



**HAL**  
open science

# Approche topologique de la métrologie du mouvement pour des applications en réalité virtuelle

Chakib Bensekka

► **To cite this version:**

Chakib Bensekka. Approche topologique de la métrologie du mouvement pour des applications en réalité virtuelle. Traitement du signal et de l'image [eess.SP]. Ecole nationale supérieure d'arts et métiers - ENSAM, 2018. Français. NNT : 2018ENAM0040 . tel-02054184

**HAL Id: tel-02054184**

**<https://pastel.hal.science/tel-02054184v1>**

Submitted on 1 Mar 2019

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

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

École doctorale n° 432 : Science des Métiers de l'ingénieur

## Doctorat ParisTech

# THÈSE

pour obtenir le grade de docteur délivré par

**l'École Nationale Supérieure d'Arts et Métiers**

**Spécialité "Informatique-traitement du signal "**

*Présentée et soutenue publiquement par*

**Chakib Arslane BENSEKKA**

Le 15/10/ 2018

## **Approche topologique de la métrologie du mouvement pour des applications en réalité virtuelle**

Directeurs de thèse : **Frédéric MERIENNE, Thierry POZZO**

Co-encadrement de la thèse : **Christophe GUILLET**

### **Jury**

**M. Charles GUTTMANN, Ass. Prof, Brigham and Women's Hospital (USA)**  
**Mme France MOUREY, Professeur, Université de Bourgogne (France)**  
**M. Naufal M. SAAD, Ass. Prof., Universiti Teknologi Petronas (Malaysia)**  
**M. Frédéric MERIENNE, Professeur, Arts et Métiers Paris Tech**  
**M. Thierry POZZO, Professeur, Université de Bourgogne**  
**M. Christophe GUILLET, Maître de Conférences, Université de Bourgogne**

Président  
Rapporteur  
Rapporteur  
Invité  
Invité  
Invité

**T  
H  
È  
S  
E**



## Remerciements

Je voudrais commencer par remercier mes directeurs de thèse Frédéric Merienne et Thierry Pozzo pour la confiance qu'ils m'ont accordée en acceptant d'encadrer ce travail doctoral, pour leurs multiples conseils et pour toutes les heures qu'ils ont consacrées à diriger cette thèse.

J'adresse tous mes remerciements et ma gratitude à mon co-directeur de thèse Monsieur Christophe Guillet qui a toujours été présent tout au long de ces années, et sans qui ce travail n'aurait pu être accompli.

Je tiens à remercier également Pauline Hilt ainsi que toute l'équipe Cognition, Action, and Sensorimotor Plasticity de l'université de Bourgogne pour leur aide lors de la réalisation des expériences.

Je remercie tous les membres de l'équipe de l'Institut Image : Géraldine, Julien, Jérémy, Sébastien, Thomas, Rémy, Damien, Jean-Rémy, Jean-Luc, Ruding et Bo pour m'avoir accueilli parmi eux ainsi que pour leur aide précieuse dans l'élaboration de ce travail.

Je remercie tout particulièrement ma famille qui m'a encouragé tout au long de mon parcours ainsi que ma femme qui m'a soutenu dans la réalisation de ce travail.





## Résumé

Dans le domaine médical, une meilleure connaissance de la fonction motrice est susceptible de permettre d'établir des thérapies adaptées à chaque lésion motrice et des outils d'études et de dépistage dans le cas de maladies neurodégénératives. Dans le domaine de la réalité virtuelle, la reconnaissance du mouvement est un point angulaire dans l'interaction de l'avatar ou de la personne en immersion avec son environnement. Plusieurs travaux ont été menés dans le but de proposer des approches de classification du mouvement humain. L'idée principale de ces méthodes est d'extraire des invariants des données enregistrées afin de les regrouper en clusters. Cependant, l'étude du mouvement humain avec des systèmes de capture de mouvement génère une quantité de données volumineuse, avec des relations non linéaires entre elles. Les méthodes présentées dans la littérature scientifique utilisent ces données soit directement comme entrée à des algorithmes de classification, soit en appliquant une méthode de réduction dimensionnelle, comme l'analyse en composantes principales avant la classification. Ces méthodes restent extrêmement sensibles au bruit blanc pendant l'enregistrement ainsi qu'aux différences morphologiques entre les sujets. Dans notre travail, nous allons présenter une méthodologie de classification et de reconnaissance du mouvement humain, qui se base sur l'analyse topologique des données cinématiques. L'analyse topologique sera réalisée via la persistance homologique, qui est une méthode d'analyse des données volumineuses qui permet de leur associer une signature topologique. On combinera cette méthode d'analyse topologique avec des algorithmes d'apprentissage afin d'augmenter la précision de la reconnaissance des mouvements en réduisant l'impact des différences morphologiques entre les sujets ainsi que l'impact du bruit blanc issu pendant l'étape d'acquisition du mouvement. Par ailleurs, on combinera la méthode d'analyse topologique avec un algorithme de réseaux de neurones temporels, afin de construire une approche qui permet de prédire la suite d'un mouvement à partir d'une partie d'un intervalle d'enregistrement.

Les résultats ont montré la capacité de l'approche proposée à obtenir une précision avec un taux élevé lors de la classification, ainsi que sa robustesse face au bruit blanc et aux différences morphologiques entre les personnes. Les résultats ont montré aussi le coût élevé en temps de calcul de notre approche. Nous avons proposé des méthodes (algorithme et parallélisme) de façon à réduire les temps de calculs.

## Abstract

In the medical field, a better knowledge of the motor functions is susceptible to allow for establish therapies adapted to each motor lesion and tools for studies and screening for neurodegenerative diseases. In the field of virtual reality, the movement recognition is an angular point in the interaction of the avatar or the person in immersion with their virtual environment.

Several studies have been carried out in order to propose approaches for human movements classification. The main idea of these methods is to extract invariants from the recorded data in order to classify them into clusters. However, the study of human movements with motion capture systems generates a huge volume of data with nonlinear relations between them.

The methods presented in the scientific literature use these data either directly as input to classification algorithms or by applying a dimensional reduction method such as principal component analysis prior to classification. These methods remain extremely sensitive to white noise during recording as well as morphological differences between subjects.

In our work, we are going to present a methodology of classification and recognition of human movement which is based on the topological analysis of kinematic data. Topological analysis will be carried out via homological persistence which is a method for big data analysis, that allows to associate them with a topological signature.

We will combine this topological analysis method with learning machine algorithms to increase the accuracy of movement classification by reducing the impact of the morphological differences between subjects as well as the impact of the white noise resulting during the step of data acquisition.

Otherwise, we will combine the topological analysis with an algorithm of temporal neural networks, in order to propose an approach that can allow to predict the continuation of a movement from a part of a recording interval.

The results showed the ability of the proposed approach to achieve high accuracy for classification, as well as its robustness against white noise and morphological differences between subjects. The results also showed the high cost of computing time of our approach which we tried to reduce by modifying its steps and by rewriting the code so that it can be executed in parallel.



# Table des matières

Chapitre I Introduction.....	15
I.1 Contexte scientifique .....	17
I.1.1 L'analyse du mouvement humain .....	17
I.1.2 La persistance homologique .....	18
I.2 Contexte applicatif .....	20
I.3 Problématique .....	21
I.4 Plan de lecture .....	22
Chapitre II Etat de l'art.....	23
II.1 Systèmes d'acquisition de mouvement .....	23
II.1.1 Systèmes d'acquisition non visuels de mouvement .....	23
II.1.2 Systèmes d'acquisition visuels de mouvement .....	31
Chapitre III Outils.....	35
III.1 Persistance Homologique.....	35
III.1.1 Les simplexes .....	35
III.1.2 Complexe Simplicial .....	36
III.1.3 Filtration d'un complexe simplicial .....	37
III.1.4 Complexes de Čech et de Vietoris-Rips .....	37
III.1.5 Echantillonnage « Landmark selection » .....	38
III.1.6 Chaines et faces de simplexes, bord d'une face .....	40
III.1.7 Groupes d'homologie .....	40
III.1.8 La persistance homologique .....	41
III.1.9 Stabilité des diagrammes de persistance .....	43
III.1.10 Travaux basés sur la persistance homologique dans un contexte applicatif .....	44
III.1.11 Machine à vecteurs de support .....	45

## Introduction

III.2	Réduction dimensionnelle .....	53
Chapitre IV	Méthodologie .....	57
IV.1	Questions de recherche .....	57
IV.2	Approche proposée .....	57
Chapitre V	Classification de mouvements .....	62
V.1	Introduction .....	62
V.2	Evaluation avec un mouvement simple acquis avec Vicon .....	63
V.2.1	Protocole expérimental .....	63
V.2.2	Préparation des données et études comparatives .....	64
V.2.3	Résultats .....	67
V.3	Evaluation avec un mouvement complexe acquis avec Vicon ....	75
V.3.1	Protocole expérimental .....	75
V.3.2	Préparation des données .....	77
V.3.3	Résultats obtenus .....	78
V.3.4	Test de robustesse .....	82
V.4	Evaluation avec un mouvement bruité acquis avec une Kinect .	84
V.4.1	Résultats .....	85
V.5	Discussion .....	86
V.6	Démarche avec réseaux de neurones .....	88
V.6.1	Introduction aux réseaux de neurones .....	90
V.6.2	Résultats et comparaison avec la classification en utilisent la machine à vecteurs de support .....	99
Chapitre VI	Parallélisme .....	105
VI.1	Introduction .....	105
VI.2	Principe du parallélisme et état de l'art .....	105
VI.2.1	Communication entre unités de calcul .....	108
VI.2.2	Communication réseau .....	109
VI.2.3	Communication avec la bibliothèque MPI .....	110
VI.2.4	Implémentation .....	113
VI.2.5	Résultats .....	116
Chapitre VII	Anticipation de mouvement .....	120

## Introduction

VII.1	Réseaux de neurones récurrents .....	120
VII.1.1	Introduction .....	120
VII.1.2	Réseaux de neurones récurrents .....	121
VII.1.3	Travaux basés sur l'utilisation des réseaux de neurones récurrents .....	124
VII.1.4	Application sur la prédiction du mouvement .....	129
Chapitre VIII	Conclusion et perspectives .....	134
Annexe A	.....	138
Annexe B	.....	142

# Liste des figures

Figure 1. Utilisation de l'analyse de mouvement dans la rééducation. ....	15
Figure 2. Utilisation de la motion capture pour l'analyse des activités sportives. ....	15
Figure 3. Applications de réalité virtuelle qui se basent sur la reconnaissance du mouvement. ....	16
Figure 4. Décomposition du mouvement d'un homme qui monte un escalier. ....	17
Figure 5. Première étape de l'analyse topologique via la persistance homologique d'une forme et échantillonnage sous forme d'un nuage de points. ....	19
Figure 6. L'utilisation de la persistance homologique afin de trouver la configuration optimale pour une couverture réseau. ....	19
Figure 7. Analyse topologique des données qui représentent la distribution des éléments d'une galaxie. ....	20
Figure 8. A gauche le système de capture de mouvement Vicon, à droite la Kinect, qui est le second système utilisé dans la capture de mouvement pour nos travaux. ....	21
Figure 9. Classification des différents type de systèmes d'acquisition du mouvement. ....	23
Figure 10. Capteur EMG (à gauche), signal electromyographique enregistré(à droite). ....	24
Figure 11. Dispositif expérimental utilisé par Abend et al (Abend , Bizzi, & Morasso, 1982), pour enregistrer la trajectoire de la main et les angles des articulations du bras lors du mouvement. ....	25
Figure 12. La partie supérieure représente les différentes trajectoires effectuées par les participants. Les trajectoires varient d'une trajectoire droite 1 à la trajectoire la plus courbée 5. La partie inférieure représente le profil de la vitesse de la main pour chacune des trajectoires. ....	25
Figure 13. Données des différents mouvements projeter en 3D à l'aide de l'ACP. ....	26
Figure 14. Configuration expérimentale pour l'acquisition des données sélectionnées. ....	27
Figure 15. (a) Protocole de l'expérience de Bizzi et al [5]. (b) Chaque point représente la position de fixation de la patte et de la stimulation, les flèches illustrent le champ de force généré au niveau de la cheville après chaque stimulation. ....	27
Figure 16. Mouvement de flexion et extension du bras sur les deux plans vertical et horizontal(à gauche). Synergies spatiotemporelles (Syn. 1 et Syn. 2) extraites à partir d'un mouvement de flexion extension du bras dans les plans vertical et horizontal (à droite), dans l'étude de Chiovetto et al (2013). Chaque muscle est représenté dans chaque synergies et les couleurs représentent le niveau d'activation de ces muscles. ....	29

Figure 17. Les schémas des mouvements effectués, B: Condition basique, K: exécuter le mouvement sous la contrainte d'extension du genou, R: effectuer le mouvement sur une petite plateforme «difficulté de garder l'équilibre», S : imposer la condition d'une trajectoire droite du doigt, C :imposer une trajectoire semi-circulaire. ....	30
Figure 18. Précision de la classification binaire entre chaque condition et une autre, les résultats obtenus sont tous supérieurs à 80%. ....	30
Figure 19. Comparaison des précisions avec et sans le prétraitement avec l'analyse en composantes principales, on remarque une nette amélioration de la précision avec l'application de PCA avant la classification. ....	32
Figure 20 (A) & (B) positionnement des angles et des articulations utilisées lors de la classification. (c) Les exercices de gym effectués par les participants. ....	33
Figure 21. Un complexe simplicial. ....	36
Figure 22.(1) $\epsilon^2$ voisinages de points (2) Rips-Vietoris complexe(3) Cechs complexe. ....	38
Figure 23. Filtration de Rips. ....	38
Figure 24. Nombres de Betti. ....	41
Figure 25. Représentation du code-barres (à gauche) et de son équivalent, le diagramme de persistance d'ordre 0 (à droite). ....	42
Figure 26.(a) des images médicales de lésions hépatiques, (b) la représentation des images selon l'intensité de leurs pixels et la réalisation d'une filtration simpliciale, (c) la signature topologique (code-barres) obtenue. ....	44
Figure 27. a) (1,2,3,4) démarche proposée par Heo et al , $a_1$ placer des marqueurs sur la mâchoire du patient, $a_2$ récupération des coordonnées 3D de ces marqueurs, après la réalisation de la filtration $a_3$ , on obtient le barres-code $a'$ représentant la signature topologique. b) travaux de Lee et al pour analyser l'activité des réseaux de neurones avec la persistance homologique. ....	45
Figure 28. Hyperplan séparateur de deux classes de données. ....	45
Figure 29. L'hyperplan séparateur optimal. ....	46
Figure 30. Les vecteurs de support. ....	47
Figure 31. Chevauchement de classes. ....	48
Figure 32. Projection des données dans un espace de dimension supérieure en utilisant une fonction noyau en cas de non séparabilité linéaire des données. ....	49
Figure 33. Classification multi-classes avec l'approche Un-contre-tous. ....	50
Figure 34.L'approche Un-contre-un. ....	51
Figure 35.Classification par graphe de décision. ....	52
Figure 36. Pour une distribution non-linaire (A), l'utilisation de la distance géodésique(B) permet de respecter les distances entre les individus de la distribution initiale (C). ....	56

Figure 37. (a) approche classique présentée dans la littérature scientifique, (b) approche proposée qui consiste à ajouter une étape d'analyse topologique avant la classification.....	58
Figure 38. Les étapes de la démarche proposée.....	58
Figure 39. Le système de capture de mouvement « Vicon » basé sur les caméras infrarouges et les marqueurs.....	59
Figure 40. Chronophotographie pour la décomposition du mouvement. ....	59
Figure 41. Le mouvement est représenté par une séquence de postures. ....	60
Figure 42. Analyse topologique de chacune des postures du mouvement. ....	60
Figure 43. Application de l'approche proposée, pour classifier deux mouvements. ....	61
Figure 44. Illustration du paradigme expérimental de l'étude 2. A. Représentation du positionnement de la surface par rapport aux sujets (distance épaule-surface : 130% de la longueur du bras). B. Les trois positions initiales, P1, P2 et P3 demandées aléatoirement aux sujets.....	63
Figure 45. Matrice C dans laquelle sont enregistrées les coordonnées de marqueurs de chaque posture du mouvement. ....	65
Figure 46. La trajectoire de l'index du même participant répétant le même mouvement séquentiel.....	70
Figure 47. Comparaison entre le temps d'exécution pour les deux approches, pour une classification binaire. ....	70
Figure 48. Comparaison de temps de calcul en secondes entre les deux approches, pour une classification multi-classes. ....	74
Figure 49. Illustration du protocole expérimental mis en place pour l'étude des synergies musculaires. Chaque cible est représentée par un chiffre allant de 1 à 9, et positionnée sur trois fils (B1, B2 et B3). Les distances aux sujets, et hauteurs des cibles sont indiquées en % de la taille du sujet.....	76
Figure 50. La trajectoire de l'index du même participant répétant le même mouvement séquentiel.....	79
Figure 51. Comparaison du temps de calcul en secondes entre les deux approches, pour la classification multi-classes. ....	80
Figure 52. Test de robustesse pour les deux approches, (a1) signatures topologiques de deux mouvements sans bruit (a2) avec bruit. ....	82
Figure 53. Les données cinématiques avant (A) et après l'ajout du bruit blanc (A'). Les distributions des signatures topologiques avant (B) et après l'ajout du bruit blanc (B). ....	83
Figure 54. Protocole de la troisième expérience : le déplacement d'un objet selon différentes trajectoires, droite et semi-circulaire, (pendant la troisième tâche un tremblement de la main est introduit). ....	85

Figure 55. L'approche proposée qui se base sur la reconnaissance des formes issues de la réduction dimensionnelle des matrices de distance entre les postures du même mouvement via les réseaux de neurones.....	89
Figure 56. Modèle du neurone biologique.....	90
Figure 57. Modélisation du neurone artificiel à deux entrées.....	91
Figure 58. Graphe de La fonction de Heaviside, utilisant la valeur 0,5 comme seuil.....	92
Figure 59. Perceptron avec seuil.....	93
Figure 60. Perceptron avec perceptron avec entrée supplémentaire (biais). ....	93
Figure 61. (a) données linéairement séparables, (b) données non-linéairement séparable qui représentent la limite des réseaux de neurones monocouche.....	94
Figure 62. Utilisation de la descente de gradient, pour déterminer la valeur du poids optimal W, qui minimise l'erreur.....	95
Figure 63. Un réseau de neurones multicouches, composé de trois couches.....	97
Figure 64. Le temps d'exécution moyen des deux approches proposées, pour une classification binaire de deux mouvements de pointage vers une seule cible.....	102
Figure 65. Comparaison de temps de calcul moyen d'une classification binaire, de deux mouvements de pointages vers neuf cibles.....	102
Figure 66. Comparaison du temps d'exécution moyen, des deux approches lors d'une classification multi-classe.....	103
Figure 67. (a) évolution de la puissance des micro-processeurs en GHz de 1990 jusqu'à 2010. (B) schéma expliquant la loi de Moore.....	106
Figure 68. (gauche ligne 1) Control Data 6600 1964,(droite ligne 1) Cray X-MP 1982.(gauche ligne 2) Connection Machine CM-1 1986. (droite ligne 2) kalray processor 2017.....	107
Figure 69. Architecture parallèle mono-machine.....	108
Figure 70. Architecture parallèle avec mémoire partagée.....	109
Figure 71. La communication entre les unités de calcul, permet une distribution optimale des tâches entre ces unités.....	111
Figure 72.Figure démonstrative de la loi de Amdahl.....	113
Figure 73. Les différentes étapes d'exécution d'un code écrit en Java.....	114
Figure 74. Comparaison entre le temps de calcul obtenu après l'exécution parallèle de la démarche (en vert) et celui des exécution séquentielle, lors de la classification binaire de la deuxième expérience.....	117
<i>Figure 75.Comparaison entre le temps de calcul obtenu après l'exécution parallèle de la démarche (en vert) et celui des exécution séquentielle, lors de la classification multi-classe de la deuxième expérience.....</i>	<i>117</i>

Figure 76. Estimation de temps de calcul par rapport au nombre d'unités de calcul utilisées, selon la loi de Amdahl. ....	118
Figure 77. Déterminer la suite d'un mouvement à partir d'un intervalle d'enregistrement. ...	120
Figure 78. Graphe détaillant une chaîne de Markov, les arêtes pondérées représentent les probabilités de passer d'un état vers un autre. ....	121
Figure 79. (a) architecture d'un réseau de neurones classique, (b) une architecture d'un réseau de neurones récurrent proposée par Elman. ....	122
Figure 80. Un réseau de neurones récurrent dont le but est d'apprendre une séquence. ...	124
Figure 81. (à gauche) L'utilisation des réseaux de neurones récurrents, pour suivre le mouvement du visage humain, sur des trajectoires en 2D. (à droite) la reconnaissance du mouvement humain sur un plan 2D. ....	125
Figure 82. Traitement des enregistrements vidéo en images qui extraient la silhouette de la personne en mouvement. ....	125
Figure 83. La quantité réelle d'énergie produite, par rapport à la quantité prédite par les réseaux de neurones récurrents. ....	126
Figure 84. Prédiction de l'offre et de la demande en utilisant les réseaux de neurones récurrents. ....	128
Figure 85. Prédiction des cours de différents index boursiers, en utilisant les réseaux de neurones récurrents. ....	128
Figure 86. Les étapes de la démarche proposée, ....	130
Figure 87. Les mouvements de marche (en haut) et de courses (en bas) ajoutés depuis la base de mouvement de l'Université de Carnegie Mellon. ....	132
Figure 88. Précision du système à déterminer la suite d'un mouvement, par rapport au nombre de neurones de la couche cachée et la couche contexte. ....	132



## Liste des tableaux

Tableau 1. Comparaison du temps de calcul en secondes entre les algorithmes de classification SVM et arbres de décision par rapport au nombre de participants. ....	32
<i>Tableau 2. Exemples de k-simplexes pour <math>k \in [0,1,2,3]</math>.....</i>	<i>36</i>
Tableau 3. Interprétation des différents résultats attendus. ....	67
Tableau 4. Résultats de comparaison de la classification binaire. ....	68
<i>Tableau 5. La complexité de calcul de chaque étape de l'approche proposée. ....</i>	<i>71</i>
Tableau 6. Les résultats de comparaison de la classification multi classes, de la première expérience.....	73
Tableau 7. Résultats de comparaison de la classification binaire. ....	78
Tableau 8. Les résultats de comparaison de la classification multi classes, de la deuxième expérience.....	81
Tableau 9. Résultats de comparaison de la classification multi classes.....	85
Tableau 10. Comparaison des résultats de la première expérience, entre les deux approches proposées, le calcul de la matrice de distance globale et la classification avec la machine à vecteurs de support, et la seconde qui consiste à calculer. ....	100
Tableau 11. Comparaison des résultats obtenus par les deux approches proposées, lors de la classification des mouvements de la deuxième expérience.....	101
Tableau 12. Les activités enregistrées dans le cadre du projet : e-health monitoring open data project. ....	127
Tableau 13. Les précisions de détermination de la suite des différents mouvements étudiés. ....	133

# Chapitre I Introduction

L'analyse des mouvements avec des systèmes de caméra est de plus en plus utilisée dans le domaine médical. L'une des applications médicales qui nécessite le plus de précision en analysant les mouvements est la surveillance de l'évolution de l'état d'un patient pendant la rééducation (Fig 1).



Figure 1. Utilisation de l'analyse de mouvement dans la rééducation.

L'analyse des mouvements est aussi utilisée dans le domaine sportif pour étudier la manière dont les sportifs réalisent l'activité afin d'améliorer leurs performances (Fig 2). Enfin, dans le domaine de la réalité virtuelle, la reconnaissance des mouvements permet de rendre efficace l'expérience d'immersion virtuelle (Fig 3).



Figure 2. Utilisation de la motion capture pour l'analyse des activités sportives.



Figure 3. Applications de réalité virtuelle qui se basent sur la reconnaissance du mouvement.

L'utilisation d'images pour l'analyse du mouvement humain génère une quantité importante de données. Plusieurs travaux ont été menés dans le but de proposer des approches qui permettent d'extraire des invariants à partir des données enregistrées pour une meilleure classification des mouvements. Dans la plupart des travaux existants dans la littérature, l'approche suivie se base sur l'utilisation des données issues de la capture du mouvement directement après leur filtration ou normalisation (comme entrée aux différents algorithmes de classification). Cette approche reste extrêmement sensible au bruit blanc ainsi qu'aux différences morphologiques entre les personnes (Kalapanidas, Nikolaos, Viorel Craciun, & Neagu, 2003), (Bregler, 1997).

Le but de notre travail est de définir une approche plus robuste face au bruit blanc et aux différences morphologiques entre les personnes. Pour cela, nous proposons d'ajouter une étape d'analyse topologique avant de réaliser la classification avec les algorithmes utilisés directement dans les autres approches.

Pour l'analyse topologique, notre travail se base sur la persistance homologique. Cet outil a été introduit par H. Edelsbrunner et al (Edelsbrunner, Letscher, & Zomorodian, 2002). Il permet d'analyser des données multidimensionnelles volumineuses qui peuvent être représentées sous forme de nuage de points. L'utilisation de cet outil pour les données du mouvement permettrait de donner une signature topologique à l'ensemble des mouvements.

L'objectif de ces travaux est de permettre une classification des mouvements indépendamment de la morphologie du sujet et également une anticipation d'un mouvement par l'analyse de ses premiers instants.

## I.1 Contexte scientifique

### I.1.1 L'analyse du mouvement humain

L'analyse du mouvement reste un enjeu majeur de la recherche théorique et appliquée en médecine et en biologie. La complexité de notre système musculo-squelettique ainsi que l'émergence des nouveaux domaines de recherche, tels que la biomécanique, la vision par ordinateur et la réalité virtuelle, rendent le développement d'outils d'analyse de mouvements une opportunité pour la médecine.

Les travaux pionniers relatifs à l'analyse du mouvement, remontent à la fin du 19<sup>ème</sup> siècle avec les travaux de Etienne-Jules Marey en 1870 et de Eadweard Muybridge en 1885 (EADWEARD MUYBRIDGE, 1885) (Fig 4), qui utilisent la photographie pour décomposer le mouvement.

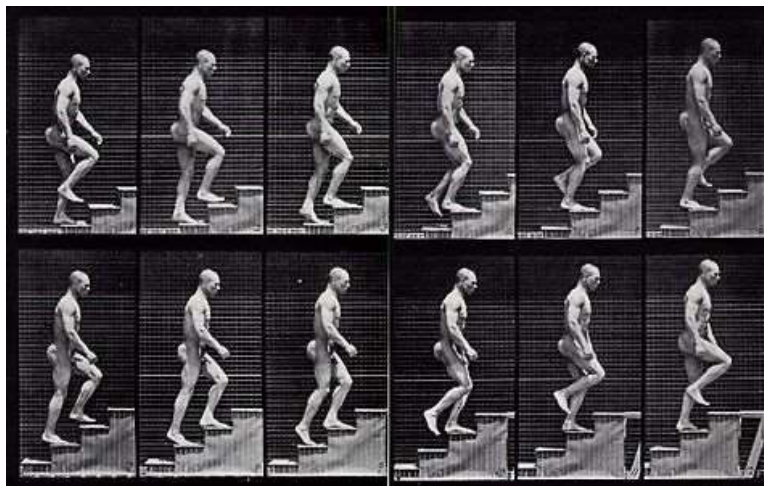


Figure 4. Décomposition du mouvement d'un homme qui monte un escalier.

Depuis, plusieurs techniques ont été utilisées afin d'analyser et développer des approches de reconnaissance et de classification de mouvements, en se basant soit sur des données cinématiques (vitesse, accélération, position) d'un ou plusieurs points du corps soit sur des données électromyographiques.

Abend et al (Abend , Bizzi, & Morasso, 1982) ont développé un dispositif qui permet de trouver une relation entre la vitesse et le mouvement de la main (Fig 11). Plusieurs travaux dont ceux de A.Mannini et A.Sabatini (Andrea Mannini & Angelo Maria Sabatini, 2010), utilisent les accéléromètres pour classifier les mouvements (Fig 14).

Le travail de Bizzi et al (Bizzi, Mussa-Ivaldi, & Giszter , 1991) sur le mouvement de la patte d'une grenouille, a permis de trouver une relation entre l'activité électromyographique et le mouvement. Cette relation est définie dans la notion de synergies.

Tolambiya et al (Tolambiya , Thomas , Chiovetto , Berret, & Pozzo, 2011) utilisent cette notion dans la classification de mouvements humains qui impliquent l'ensemble du corps (Fig 17).

Ces dernières années, les systèmes de capture de mouvement sont de plus en plus utilisés. L'acquisition de mouvement avec ce genre de dispositif se divise en deux sous-groupes :

- Les systèmes de capture avec marqueurs.
- Les systèmes sans marqueurs.

Avec les systèmes de capture de mouvements, les travaux qui existent dans la littérature scientifique consistent à utiliser les données issues de l'acquisition comme entrées aux différents algorithmes de classification, comme dans les travaux de Hachaj et al (Hachaj, R. Ogiela, & Koptyra, 2016).

Les différentes approches d'analyse de mouvement citées dans cette section, sont détaillées dans la section analyse de mouvement dans le chapitre de l'état de l'art.

### **I.1.2 La persistance homologique**

L'homologie persistante est devenue l'un des outils d'analyse topologique les plus populaires pour analyser des données volumineuses. La méthode a été introduite par Edelsbrunner et al (H. Edelsbrunner, D. Letscher, & A. Zomorodian, 2000). Les données d'entrée peuvent être représentées sous la forme d'un nuage de points multidimensionnel (Fig 5). Le principe de cette méthode d'analyse consiste à extraire les caractéristiques topologiques d'une suite de complexes simpliciaux imbriqués qui sont des approximations topologiques successives de la forme du nuage de points considéré et de restituer une signature topologique propre à cette forme.

L'homologie persistante donne une description algébrique aux caractéristiques topologiques des données. Cette description peut être représentée sous forme d'une signature topologique (code-barres ou diagramme de persistance) (Fig 25) qui permettent d'identifier cet ensemble de données parmi d'autres.

Dans un espace à trois dimensions, les caractéristiques topologiques non triviales qui sont comptabilisées pour définir la signature topologique de l'espace étudié sont les composantes connexes, les « trous » (en dimension 1) et les cavités (en dimension 2) (Figure 24. Nombres de Betti.)

## Introduction

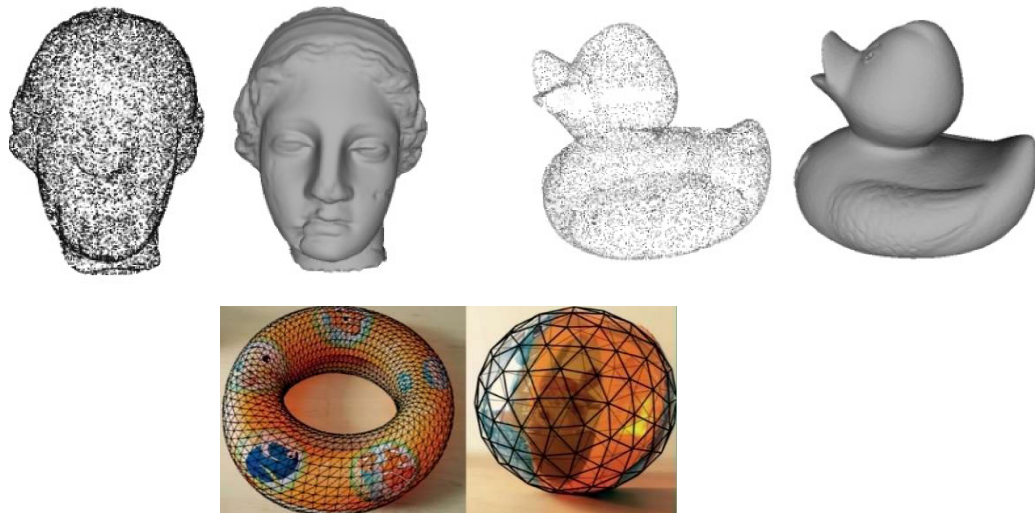


Figure 5. Première étape de l'analyse topologique via la persistance homologique d'une forme et échantillonnage sous forme d'un nuage de points.

Dans la littérature scientifique, plusieurs travaux dans différents domaines se sont basés sur la persistance homologique afin d'analyser les gros volumes de données. Parmi ces travaux, on peut citer ceux de V. de Silva et R. Ghrist qui utilisent la persistance homologique pour résoudre un problème  $N_p$  complet (VIN DE SILVA & ROBERT GHRIST, 2007). L'étude consiste à trouver la configuration optimale qui permet une couverture maximale d'un espace avec le plus petit nombre d'émetteurs possibles (Fig 6).

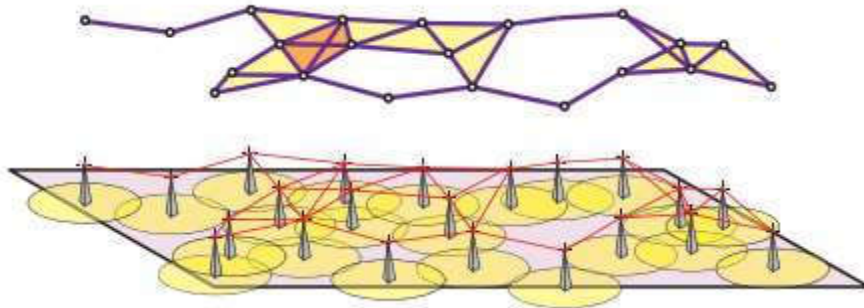


Figure 6. L'utilisation de la persistance homologique afin de trouver la configuration optimale pour une couverture réseau.

R. van de Weygaert et al (R. van de Weygaert, G. Vegter, E. Platen, B. Eldering, & N. Kruithof, 2011) analysent les données d'une compilation des positions de certaines galaxies issues de la base 2dF Galaxy Redshift Survey (Anglo-Australian Observatory, 2003) réalisée par l'observatoire anglo-australien (AAO).

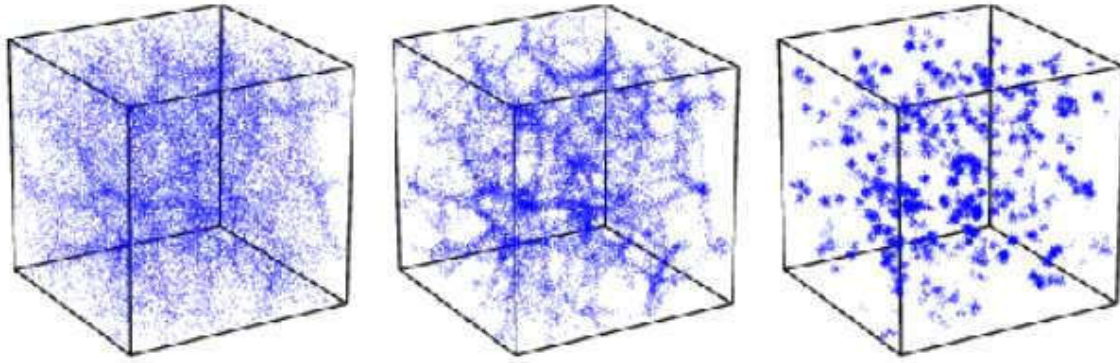


Figure 7. Analyse topologique des données qui représentent la distribution des éléments d'une galaxie.

La persistance homologique est de plus en plus utilisée dans l'analyse des données médicales. Dans les travaux de Nicolau M et al (M. Nicolau, A. J. Levine, & G. Carlsson, 2011), la méthode est utilisée pour analyser les données biomédicales afin de dépister le cancer du sein. Elle est aussi utilisée par Adcock et al (Adcock, Rubin, & Carlsson, 2014) dans le dépistage et la classification des lésions hépatiques (Fig 27).

La persistance homologique est détaillée dans la deuxième section du chapitre de l'état de l'art.

## I.2 Contexte applicatif

Ces travaux ont été réalisés au sein de l'Institut Image des Arts et Métiers, dont l'axe de recherche principal est le développement de méthodes et outils de l'immersion virtuelle au service de l'ingénieur. Ces travaux ont été réalisés en étroite collaboration avec l'Unité Cognition, Action, et Plasticité Sensorimotrice de l'Institut National de la Santé Et de la Recherche Médicale, dont une partie des travaux sont orientés vers l'étude du contrôle moteur.

Dans le domaine médical, l'analyse du mouvement humain est un point central dans le diagnostic et la proposition de traitement pour les lésions motrices. Une signature topologique du mouvement permettrait d'évaluer les performances des fonctionnalités motrices des patients et de les comparer avec les performances de sujets témoins afin d'évaluer l'évolution de la thérapie (Fig 1).

Dans le domaine de la réalité virtuelle, la reconnaissance du mouvement est essentielle pour l'interaction entre la personne en immersion et l'environnement virtuel (Fig 3). En plus de la classification des mouvements, on proposera dans notre travail une approche qui permet d'anticiper la suite d'un mouvement à partir d'une partie d'un intervalle d'enregistrement. Cette capacité de déterminer la suite d'un mouvement peut avoir plusieurs applications, comme le précalcul des interactions de l'environnement virtuel avec la personne en immersion. L'approche d'anticipation de la suite d'un mouvement peut

## Introduction

être utilisé pour l'assistance et le lancement d'alertes en cas de chute ou des mouvements dangereux dans des espaces équipés de capteurs.

Les systèmes de capture de mouvements utilisés dans nos travaux sont de deux types : un système avec marqueurs sur le sujet offrant un enregistrement du mouvement précis et avec une fréquence élevée : Vicon motion capture Systems (Fig 8), et un système de capture de mouvement sans marqueurs sur le sujet : le dispositif Kinect (Fig 8), moins sophistiqué et plus accessible à l'utilisation, mais qui enregistre le mouvement avec une fréquence plus faible.



Figure 8. A gauche le système de capture de mouvement Vicon, à droite la Kinect, qui est le second système utilisé dans la capture de mouvement pour nos travaux.

### I.3 Problématique

La problématique scientifique de nos travaux est l'analyse des mouvements humains permettant d'une part une classification de mouvements indépendamment de la morphologie du sujet et d'autre part une anticipation de la fin du mouvement. Dans cet objectif, notre ambition est de mettre en œuvre l'outil de persistance homologique pour analyser les données cinématiques issues des systèmes de capture de mouvements. Cet outil devrait permettre d'extraire des invariants moteurs pour l'établissement d'une signature topologique. Cette méthode pourra être combinée avec des algorithmes d'apprentissage tels que les machines à vecteurs de support et les réseaux de neurones. De cette façon, une approche robuste de classification et de reconnaissance des mouvements humain pourra être proposée dans un premier temps. Dans un second temps, une technique permettant d'anticiper la suite d'un mouvement sera proposée.



## **I.4 Plan de lecture**

Dans le premier chapitre, on a détaillé le contexte de nos travaux et introduit notre problématique.

Le second chapitre sera consacré à l'état de l'art des différents travaux réalisés dans le domaine de l'analyse et la classification des mouvements humains.

Le troisième chapitre décrira les outils utilisés dans notre approche de la problématique :

- Une section détaillant la persistance homologique qu'on utilisera pour l'analyse topologique des données.
- Une section détaillant la méthode utilisée pour l'étape d'apprentissage et de classification des signatures topologiques issues de l'étape précédente.
- Une section décrivant les méthodes de réduction dimensionnelle utilisées.

Dans le quatrième chapitre, on détaillera l'approche proposée.

Le cinquième chapitre sera consacré à l'évaluation de notre approche par des études expérimentales. Une étude comparative sur les résultats obtenus au niveau de la précision de classification ainsi que le temps de calcul sera réalisé entre notre approche et celle utilisée dans la littérature.

Le sixième chapitre exposera la méthodologie proposée pour l'anticipation de la suite d'un mouvement à partir d'un intervalle d'enregistrement.

Le chapitre 7 clôturera ce mémoire en exposant un bilan de nos travaux à travers une discussion générale. Des perspectives et pistes de recherches à explorer dans de futurs travaux seront proposées.

# Chapitre II Etat de l'art

## II.1 Systèmes d'acquisition de mouvement

Dans cette section, nous allons présenter les différentes approches utilisées par les systèmes de capture afin d'enregistrer le mouvement.

Pour une analyse pertinente des démarches utilisées, nous allons commencer par une vue d'ensemble sur les différents systèmes de capture, qui se divisent entre systèmes de capture visuel et non visuel (Fig 9).

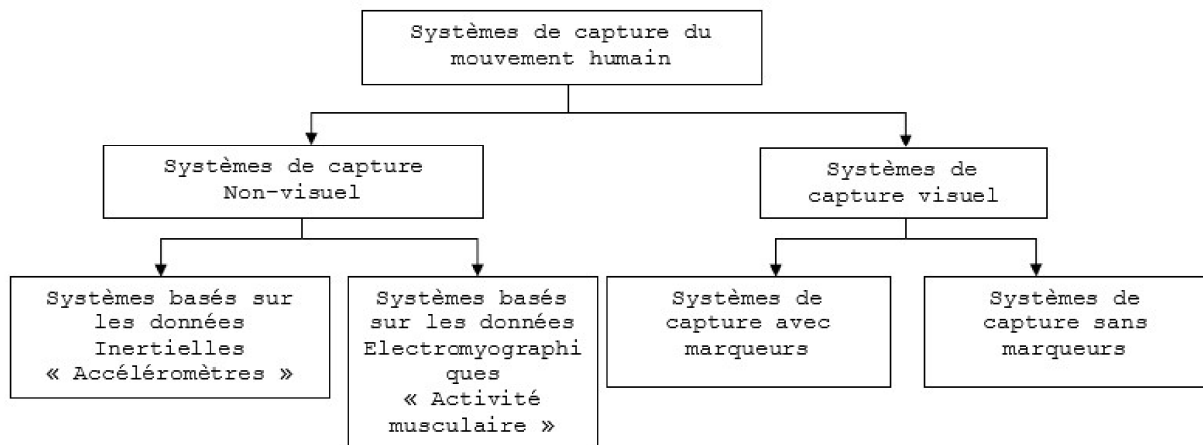


Figure 9. Classification des différents type de systèmes d'acquisition du mouvement.

### II.1.1 Systèmes d'acquisition non visuels de mouvement

Les capteurs de ce genre de système peuvent être regroupés selon les deux familles des données les plus utilisées dans la littérature scientifique : ceux qui se basent sur les données inertielles et ceux qui se basent sur les données électromyographiques.

Pour les données inertielles, on peut citer les accéléromètres comme premier type de capteurs utilisés dans l'analyse du mouvement. Le principe consiste à enregistrer les accélérations linéaires ou les vitesses angulaires.

Pour le second type de données, les captures électromyographiques sont utilisées pour enregistrer l'activité électrique des muscles (Fig 10).

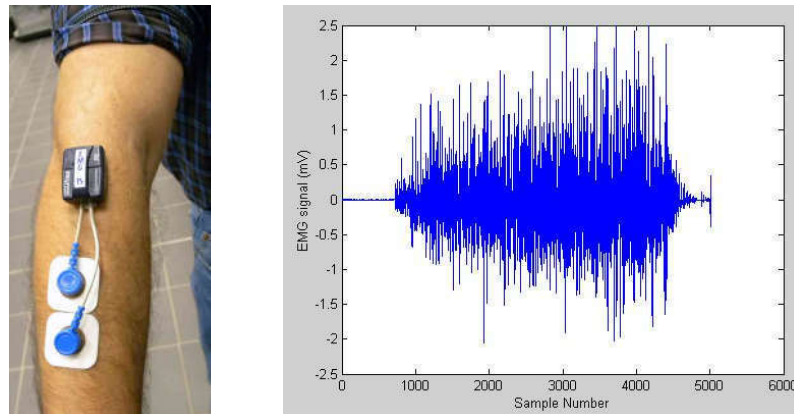


Figure 10. Capteur EMG (à gauche), signal electromyographique enregistré (à droite).

Les capteurs optiques (exemple : caméras infrarouges) améliorent nettement la précision de la capture des positions pendant le mouvement. On a deux sous-groupes de systèmes de tracking visuel : avec ou sans marqueur.

Les systèmes de tracking avec marqueurs sont un ensemble de technologies qui consistent à équiper les sujets de marqueurs, placés sur différentes parties de leurs corps. Les caméras suivent et enregistrent les trajectoires des marqueurs qui retournent les coordonnées en 3 dimensions (x,y,z).

L'un des premiers travaux réalisés sur la classification du mouvement remonte à 1981 avec Abend et al (Abend , Bizzi, & Morasso, 1982). Leur travail avait comme but de trouver une relation entre la vitesse de la main et la trajectoire lors d'un mouvement de pointage.

Le dispositif utilisé consistait en un transducteur qui détecte la position de la main (Fig 11) ainsi que les angles du coude et de l'épaule chaque 10 ms. A partir de ces données, les vitesses linéaires de la main et angulaires des deux articulations sont calculées en utilisant la dérivation des positions et des angles.

Le rayon de courbure  $\rho$  est calculé à chaque point de la trajectoire du mouvement selon la formule suivante :

$$\rho = (\dot{X}\ddot{Y} - \ddot{X}Y) / (\dot{X}^2 + \dot{Y}^2)^{3/2}$$

où  $\dot{X}$  et  $\dot{Y}$  sont la dérivation par rapport au temps des deux coordonnées X et Y,  $\ddot{X}$  et  $\ddot{Y}$  leurs accélérations respectives.

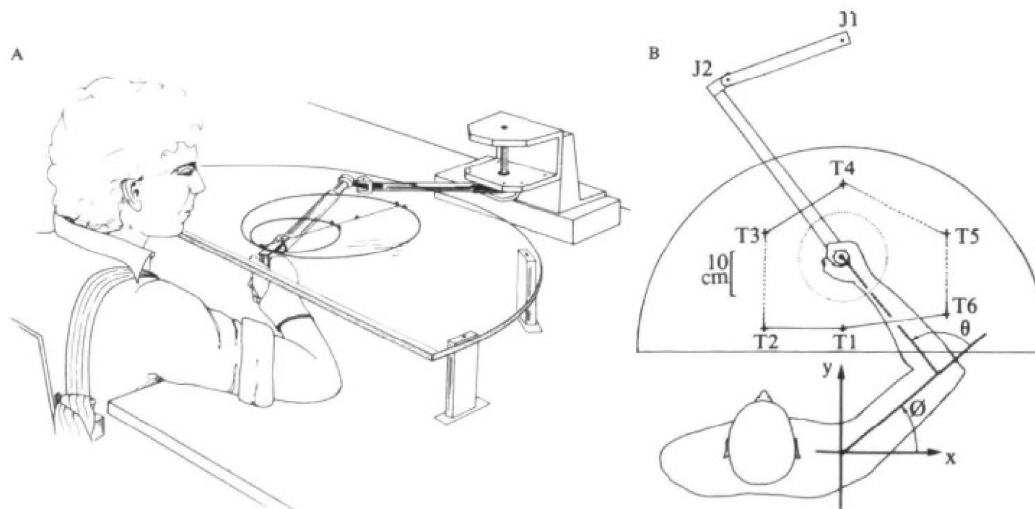


Figure 11. Dispositif expérimental utilisé par Abend et al (Abend , Bizzi, & Morasso, 1982), pour enregistrer la trajectoire de la main et les angles des articulations du bras lors du mouvement.

Les sujets ont exécuté des mouvements de pointage horizontal vers une cible, en suivant des trajectoires soit droites soit courbées. La Figure 12 illustre la série de mouvements effectués par un sujet. Pour chaque mouvement, la trajectoire de la main, la vitesse ainsi que la trajectoire sont enregistrées et représentées.

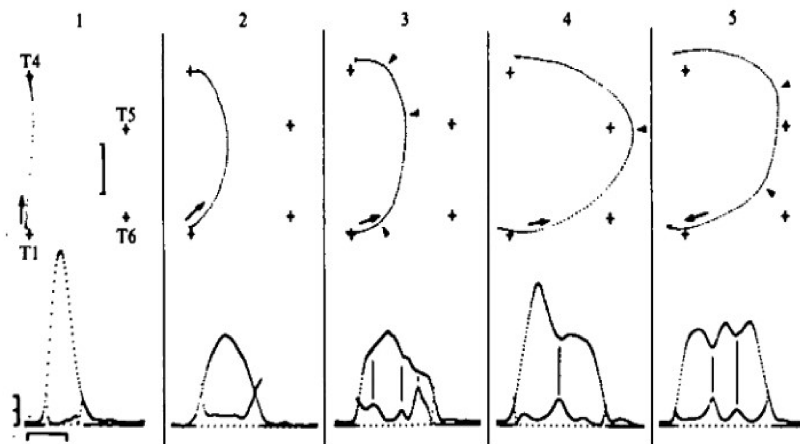


Figure 12. La partie supérieure représente les différentes trajectoires effectuées par les participants. Les trajectoires varient d'une trajectoire droite 1 à la trajectoire la plus courbée 5. La partie inférieure représente le profil de la vitesse de la main pour chacune des trajectoires.

Les résultats montrent que quand la trajectoire de la main est droite, le profil de vitesse de la main est en forme de cloche. Quand la trajectoire est incurvée le profil de vitesse est irrégulier. Cette irrégularité augmente avec l'augmentation de la courbure, comme le montre bien la Figure 12.

Comme cela a été décrit dans le premier paragraphe, l'accélération est la donnée inertielle la plus utilisée dans l'analyse du mouvement. Wang et Wu (Wang & Wu, 2014) utilisent les données issues des accéléromètres, placés sur « l'oreille, le torse, le bras, le poignet, au niveau de la taille, le genou et la cheville », pour classifier un ensemble de mouvements : « s'asseoir, se lever, marcher ». La première étape de l'expérience consistait à utiliser directement les données électromyographiques comme exemple d'apprentissage, et dans la deuxième partie la classification est effectuée après un traitement de données par une mise à l'échelle et par analyse en composantes principales (ACP). Les résultats montrent une nette amélioration dans la précision avant et après le traitement des données pour atteindre une précision de 99% en utilisant la machine à vecteurs de supports après filtration et normalisation des données. En appliquant une réduction dimensionnelle avec l'analyse en composantes principales (ACP) combinée à une machine à vecteur de support, la classification multi-classes est réalisée avec une grande précision (Fig 13).

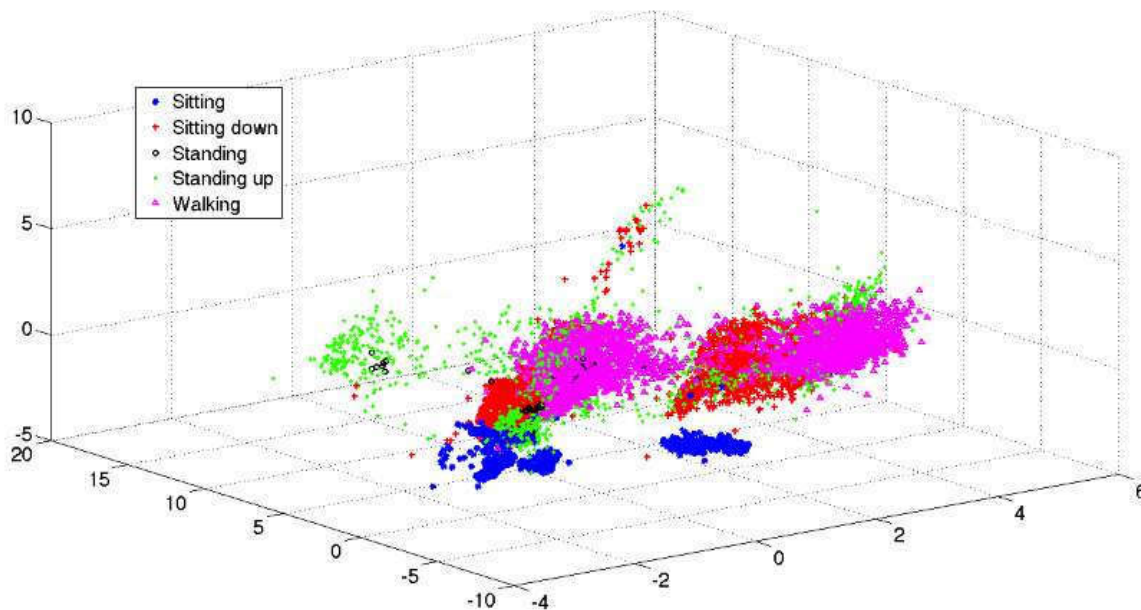


Figure 13. Données des différents mouvements projetés en 3D à l'aide de l'ACP.

A.Mannini et A.Sabatini (Andrea & Sabatini Maria, 2010) proposent une approche pour classifier les mouvements afin d'être utilisée dans les systèmes d'aide aux mouvements, destinés pour des personnes atteintes de lésion-motrice ou des personnes âgées. Leur approche consistait à enregistrer les données à partir de cinq accéléromètres, situés sur la hanche, au poignet, au bras, à la cheville et à la cuisse (Fig 14), de 20 participants à qui on demande d'effectuer des activités de la vie quotidienne.

Il s'agit notamment d'activités qui impliquent principalement l'utilisation des membres supérieurs comme « broser les dents, frotter, plier le linge, manger ou boire et travailler sur ordinateur », des activités du corps entier tel que passer l'aspirateur, et des mouvements qui impliquent principalement les membres inférieurs : « marcher et monter les escaliers ». Les résultats montrent qu'avec la limitation du bruit blanc en posant un seuil de filtration pour les données bruitées, la précision s'améliore considérablement en passant de 64% à 86%.

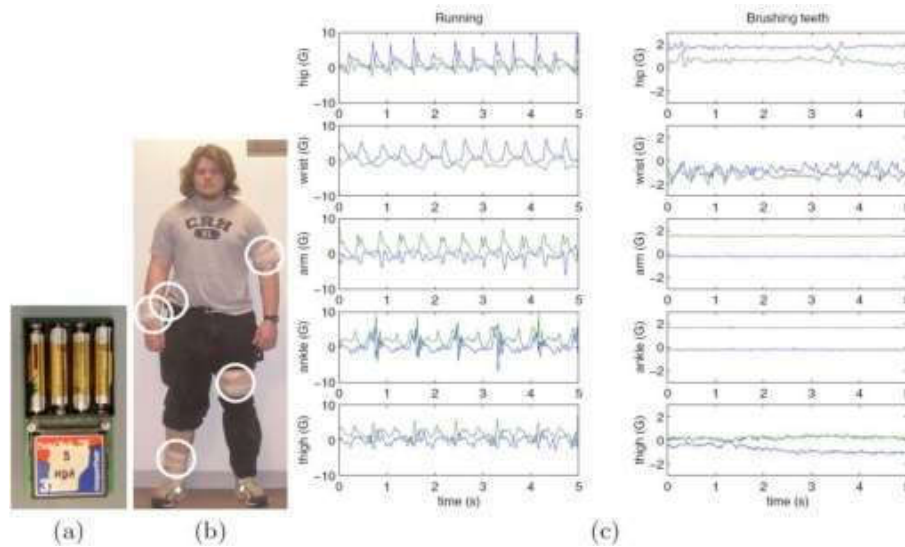


Figure 14. Configuration expérimentale pour l'acquisition des données sélectionnées.

Pour l'approche qui se base sur les données électromyographiques, des études pionnières ont été menées par Bizzi et al (Bizzi, Mussa-Ivaldi, & Giszter, 1991) sur l'activité musculaire des grenouilles. La grenouille est fixée sur une plateforme, et la moelle épinière est activée par une micro stimulation électrique en différents points au niveau de la patte. Pour chaque point stimulé, la force isométrique produite à la cheville est enregistrée (Fig 15).

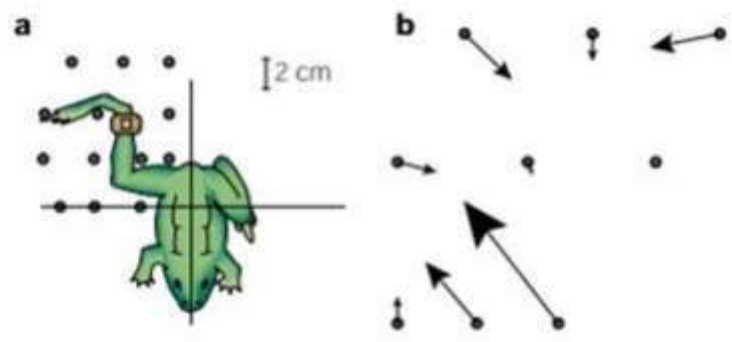


Figure 15. (a) Protocole de l'expérience de Bizzi et al [5]. (b) Chaque point représente la position de fixation de la patte et de la stimulation, les flèches illustrent le champ de force généré au niveau de la cheville après chaque stimulation.

L'expérience a montré que pour chaque site stimulé au niveau de la moelle épinière, on obtient un champ de force bien précis au niveau de la cheville. Ce résultat a permis de définir la notion de «primitive musculaire» qui contrôle l'activité d'un bloc de muscles.

La stimulation simultanée des deux sites produit un mouvement de la cheville équivalent à une combinaison linéaire de la stimulation de chacun des deux sites séparément. Ce second résultat montre que les mouvements complexes sont engendrés par une combinaison de primitives musculaires. La théorie des synergies musculaires a été proposée pour mieux formaliser cette notion.

### II.1.1.1 Les synergies musculaires

Les synergies musculaires correspondent à des régularités trouvées dans les enregistrements électromyographiques pendant l'exécution d'un mouvement. La recherche se fait en utilisant des méthodes d'analyses de données telles que l'analyse en composantes principales ou bien en utilisant des algorithmes de classification qui se basent sur de l'apprentissage non supervisé ; le but étant de déterminer l'ensemble de muscles qui s'activent régulièrement avec un certain décalage temporaire pour exécuter un mouvement précis.

L'une des formalisations des synergies musculaires est celle proposée par d'Avella et al (d'Avella, Saltiel, & Bizzi, 2003), nommée modèle de décomposition spatio-temporelle. Une synergie  $W_K$  qui décrit le profil d'activité d'un ensemble de muscles lors d'un mouvement peut être combinée à d'autres synergies afin de réaliser une tâche qui se compose d'un ensemble de mouvements, par exemple un mouvement de flexion-extension du bras. La combinaison de ces synergies se fait à l'aide de deux coefficients, un coefficient d'amplitude  $a_i$  et un autre de décalage temporel  $t_i$ .

Le modèle de combinaison de R synergies est représenté par l'équation suivante :

$$m^S(t) = \sum_{k=1}^R \alpha_k^S W_k(t - t_k^S) + \text{residu}.$$

Où  $W_k(t)$  est le vecteur de synergie décrivant l'activité sur l'intervalle de temps,  $t_k^S$  est le décalage temporel de la synergie par rapport au début du mouvement, et  $\alpha_k^S$  est le coefficient paramétrant l'amplitude de la synergie.

La figure 16 illustre deux synergies  $W_1$  et  $W_2$  qui décrivent l'activation des deux muscles (Biceps et Triceps) lors de l'étude du mouvement de flexion (synergie  $W_1$ ) et extension (synergie  $W_2$ ) du bras par Chiovetto et al (Chiovetto, Berret, Delis, Panzeri, & Pozzo, 2013). L'amplitude de l'activité musculaire est caractérisée par des nuances de couleurs variant du bleu clair au rouge qui représente le pic d'activité.

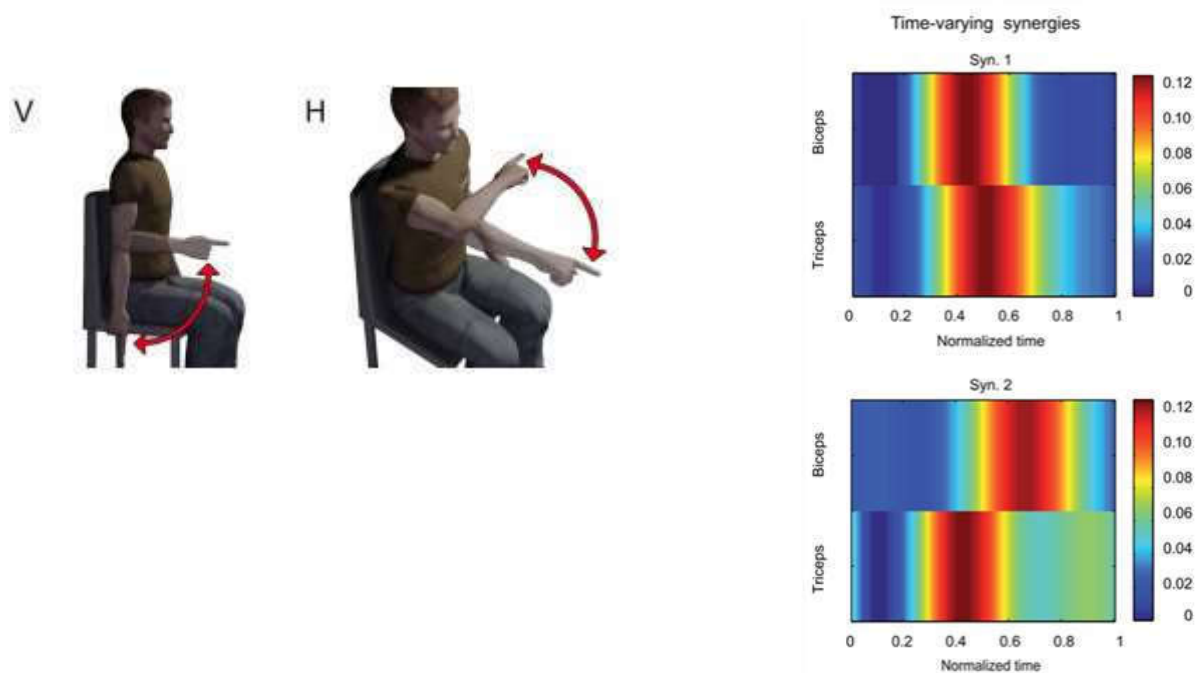


Figure 16. Mouvement de flexion et extension du bras sur les deux plans vertical et horizontal (à gauche). Synergies spatiotemporelles (Syn. 1 et Syn. 2) extraites à partir d'un mouvement de flexion extension du bras dans les plans vertical et horizontal (à droite), dans l'étude de Chiovetto et al (2013). Chaque muscle est représenté dans chaque synergies et les couleurs représentent le niveau d'activation de ces muscles.

La notion de synergie a été utilisée dans plusieurs travaux qui traitent de l'analyse du mouvement. Dans Nair et al ( Sumitra S. , Robert M., Davy , & Thomas, 2010), les données électromyographiques sont utilisées pour diagnostiquer les maladies d'arthrite (polyarthrite rhumatoïde et arthrose de la hanche). Des capteurs EMG sont posés sur les muscles (soleus, gactrocnemius medialis, gluteus medialis, tibialis anterior, vastus lateralis et biceps femoris) au niveau des membres inférieurs de trois ensemble de participants. Le premier groupe est constitué de patients atteints de polyarthrite rhumatoïde, le second groupe de patients est atteint d'arthrose de la hanche et le troisième est un groupe témoin. Les données ont été filtrées avant d'être utilisées comme entrée à des machines de classification, le but étant de diagnostiquer si la personne est atteinte de lésion ou pas, et ensuite d'arriver à identifier les différences musculaires entre les participants témoins et les patients. Dans Tolambiya et al (Tolambiya , Thomas , Chiovetto , Berret, & Pozzo, 2011) les données électromyographiques de 24 muscles sont enregistrées lors d'un mouvement de pointage impliquant tout le corps « whole body reaching », dans le but de différencier les mouvements effectués sans ou sous contraintes. Le mouvement est exécuté dans un premier temps sans aucune contrainte, et dans un second temps sous deux contraintes impliquant des muscles focaux et deux autres impliquant des muscles posturaux. Les deux contraintes



posturales consistent à ajouter une difficulté à garder l'équilibre, en utilisant une base de support réduite (RBoS) où les participants devaient effectuer le mouvement avec les pieds posés sur une base de support réduite (Fig 17 R), et en demandant aux participants d'effectuer le mouvement sans fléchir les genoux (Fig 18). Pour les contraintes focales, une trajectoire directe du doigt est imposée ainsi qu'une trajectoire semi-circulaire (Fig 17 S et C).

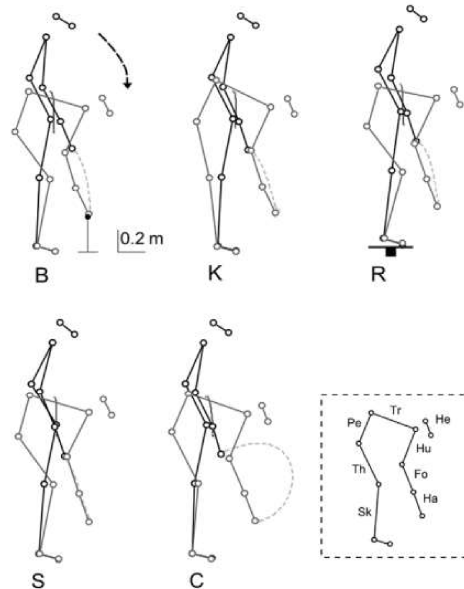


Figure 17. Les schémas des mouvements effectués, B: Condition basique, K: exécuter le mouvement sous la contrainte d'extension du genou, R: effectuer le mouvement sur une petite plateforme «difficulté de garder l'équilibre», S : imposer la condition d'une trajectoire droite du doigt, C : imposer une trajectoire semi-circulaire.

Les données électromyographiques sont enregistrées avec une fréquence de 100 Hz, et filtrées avec le filtre de Butterworth (Roger G., Liliam F., & Jurandir, 2007) à 5 Hz de fréquence de coupure. La machine à vecteurs de support est utilisée pour une classification binaire afin de reconnaître un mouvement d'un autre. Les résultats obtenus montrent une précision qui s'approche de 80%, comme le montre la figure 18.

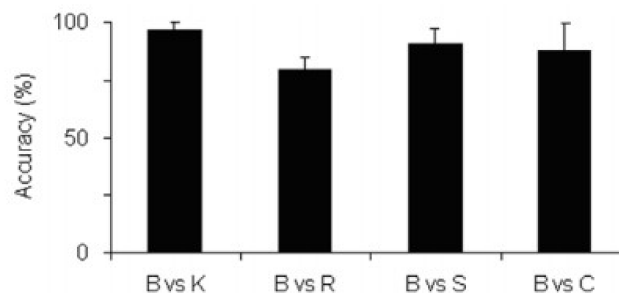


Figure 18. Précision de la classification binaire entre chaque condition et une autre, les résultats obtenus sont tous supérieurs à 80%.

### **II.1.2 Systèmes d'acquisition visuels de mouvement**

Ces dernières années, les systèmes de capture de mouvement sont de plus en plus utilisés dans l'analyse du mouvement. Von Tscharner et al (Von Tscharner, Enders, & Maurer, 2013) utilisent ces systèmes pour étudier l'impact de la semelle des chaussures sur le mouvement de la marche. Les participants étaient priés d'effectuer deux séries d'aller-retours sur une piste de 30 mètres en portant une paire de chaussures différente pour chaque série de tests. La seule différence entre les deux paires utilisées est au niveau des semelles, la première est équipée d'une semelle viscoélastique et la seconde est complètement en élastique. Les données cinématiques ont été enregistrées à l'aide de 13 marqueurs réfléchissants qui étaient placés au niveau du bassin et du membre inférieur droit, le but étant d'être en mesure de reconnaître si le mouvement est issu d'un essai effectué avec la première ou la seconde semelle. Les données enregistrées pour chaque essai sont normalisées et étiquetées « Mouvement semelle 1 » ou « Mouvement semelle 2 ». Ensuite, la partie classification est effectuée en utilisant deux analyses en composantes principales (ACP) et les machines à vecteurs de support (SVM). Les résultats montrent que la machine à vecteurs de support est mieux adaptée pour améliorer la précision que la classification obtenue avec l'analyse en composantes principales.

Le travail de Leightley et al (Leightley, Darby, Li, S. McPhee, & Yap, 2017) consiste à analyser les données cinématiques du corps entier en utilisant la Kinect comme système de capture de mouvement dans le but de classifier les mouvements effectués lors de la rééducation physique des patients. Les données sont issues de l'enregistrement des coordonnées 3D de 20 articulations, dont les pieds, chevilles, genoux, hanches, torse, épaules, coudes, poignets, mains et tête, et enregistrées sous forme de matrices. Les participants effectuaient les mouvements dans un espace bien défini en face de la Kinect. Les activités sélectionnées étaient : un saut d'environ 10 cm avec les deux jambes, une extension des bras vers l'avant, ramasser un objet par terre avec la main droite à partir d'une position debout, un exercice de squat, de marche, courir, se mettre debout d'une position assise, un mouvement de torsion de la partie supérieure du corps, extension des bras avec les deux pieds à plat au sol et en soulevant les deux bras verticalement aussi haut que possible. Une réduction dimensionnelle à l'aide de l'analyse en composantes principales est effectuée, avant de passer à la phase d'apprentissage en utilisant deux algorithmes de classification supervisée : la machine à vecteurs de support et les forêts d'arbres décisionnels (Liaw & Wiener, 2001). Une comparaison du niveau de précision lors de la classification a été réalisée, avant et après l'utilisation de la réduction dimensionnelle (Fig 19).

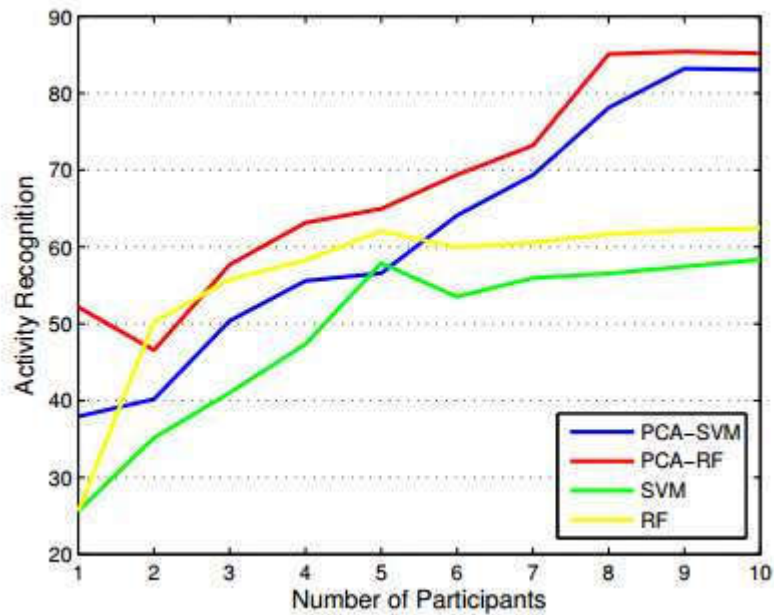


Figure 19. Comparaison des précisions avec et sans le prétraitement avec l'analyse en composantes principales, on remarque une nette amélioration de la précision avec l'application de PCA avant la classification.

Les résultats montrent que la précision augmente nettement avec l'application de la réduction dimensionnelle. La précision atteint 85% avec l'algorithme de forêts d'arbres décisionnels et 83% pour la machine à vecteurs de support. Une autre comparaison a été réalisée sur le temps d'apprentissage nécessaire pour chaque méthode (Tab 1), qui montre que l'algorithme de machine à vecteur de support est moins coûteux en temps de calcul que les arbres de décision. L'algorithme des arbres de décision rencontre des difficultés de classification quand le mouvement est continu tel que les mouvements des bras.

$\lambda$ participants/ Classifier	2	4	6	8	10
	Training time [Sec]				
<i>SVM</i>	2.381	14.69	30.39	42.24	55.96
<i>RF</i>	5.08	19.35	49.06	79.21	123.17

Tableau 1. Comparaison du temps de calcul en secondes entre les algorithmes de classification SVM et arbres de décision par rapport au nombre de participants.

Dans le travail de Hachaj et al (Hachaj, R. Ogiela, & Koptyra, 2016), la classification et la reconnaissance du mouvement humain sont réalisées sans faire de prétraitement sur les données avant la partie d'apprentissage.

Les données utilisées sont les coordonnées 3D des articulations ou les angles des articulations (Fig 20), enregistrées avec une Kinect lors d'activités humaines qui consistent à des exercices de gymnastique (Fig 20).

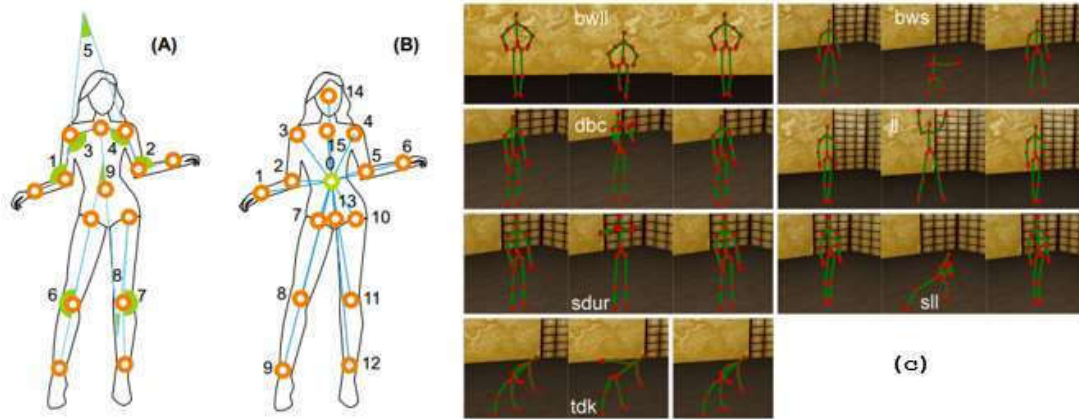


Figure 20 (A) & (B) positionnement des angles et des articulations utilisées lors de la classification. (c) Les exercices de gym effectués par les participants.

Les algorithmes utilisés sont les réseaux de neurones combinés à un algorithme parallèle de descente de gradient stochastique et les machines à vecteurs de support avec un noyau de base radiale (RBF) (J.A.K. & Vandewalle, 1999) ainsi qu'une chaîne de Markov (Brooks, Gelman, Jones, & Meng, 2011).

Les résultats obtenus se rapprochent de ceux obtenus dans les travaux précédents, mais avec l'utilisation d'un nombre important de neurones dans la couche intermédiaire équivalent à 256 neurones. L'algorithme de machine à vecteurs de support fournit toujours un bon résultat soit en utilisant les coordonnées des articulations ou les angles entre eux. Les chaînes de Markov ne sont pas capables de différencier les mouvements qui se ressemblent comme dans le cas des mouvements de fente latérale gauche et fente latérale droite (sll Fig 20), et perdent rapidement leur capacité de généralisation (Hachaj, R. Ogiela, & Koptyra, 2016).

Dans cette section, nous avons présenté un état de l'art sur les différentes approches utilisées pour l'analyse du mouvement dans le but de faire de la classification, en utilisant divers types de données (électromyographiques et cinématiques), et différentes techniques de capture du mouvement. La majorité des travaux réalisés ont montré la capacité de l'algorithme de machine à vecteurs de support à retourner d'excellents taux de précision lors de la classification.

On remarque aussi que la majorité des travaux présentés utilisaient les données issues des systèmes de capture de mouvement directement comme entrée aux algorithmes de classification. Ces données sont généralement très bruitées et leur utilisation directe montre que ces systèmes sont très sensibles à la présence de bruit blanc lors de l'enregistrement comme le montrent Jiang et al (Jiang, et al., 2011) et avec des capacités limitées face à un nombre important de mouvements à classer.

Pour ces raisons, on se propose de les analyser à partir d'invariants topologiques qui leur sont associés avant de faire leur classification. L'intérêt est que cette approche a la particularité d'être stable par perturbation. L'analyse topologique sera basée sur

Etat de l'art

la notion de persistance homologique qu'on détaillera dans les sections suivantes.

# Chapitre III Outils

## III.1 Persistance Homologique

Dans cette partie, nous allons définir une approche qui va permettre d'attribuer une signature topologique à un mouvement humain enregistré à partir de systèmes de capture de mouvement. Compte tenu du dispositif de capture de mouvement, c'est à partir d'une version discrétisée du corps humain que son mouvement est analysé. Le nuage de points correspondant peut-être perçu comme un ensemble fini de sommets de la triangulation d'une surface plongée dans l'espace euclidien tridimensionnel. En topologie algébrique, le calcul d'homologie est une procédure qui associe des invariants algébriques à un espace topologique. Cela permet de classifier de tels espaces selon ces invariants qui sont associés à certaines de leurs caractéristiques topologiques. Les espaces topologiques que nous considérons dans notre étude sont des complexes simpliciaux définis à partir du nuage de points considérés. Ainsi, via le calcul de la persistance homologique, notion introduite par Edelsbrunner et al ( Edelsbrunner, Letscher, & Zomorodian, 2002), nous allons étudier l'évolution d'invariants topologiques associés à chacun des éléments d'une suite de complexes simpliciaux imbriqués approchant la forme sous-jacente du nuage de points. Le calcul d'invariants topologiques pour des complexes simpliciaux relève de l'homologie simpliciale. On ne considèrera ainsi, dans tout ce qui suit, que l'homologie simpliciale à coefficients dans l'espace vectoriel  $\mathbb{Z}/2\mathbb{Z}$ .

### III.1.1 Les simplexes

**Définition 1** : Soit  $S = \{ s_0, s_1, s_2, \dots, s_k \} \subset R^n$  un ensemble de  $(k+1)$  points de  $R^n$  ( $n \geq k$ ). Un  $k$ -simplexe de  $R^n$  est l'enveloppe convexe de  $(k+1)$  points affinement indépendants. Dans  $R^3$ , un  $k$ -simplexe ( $k > 3$ ) est trivial. Les  $k$ -simplexes non triviaux sont : un point (0-simplexe), une arête (1-simplexe), un triangle (2-simplexe), un tétraèdre (3-simplexe). (Tab 2).





<i>0-Simplex</i>	<i>vertex</i>	
<i>1-Simplex</i>	<i>edge</i>	
<i>2-Simplex</i>	<i>triangle</i>	
<i>3-Simplex</i>	<i>tetrahedron</i>	

Tableau 2. Exemples de  $k$ -simplexes pour  $k \in [0,1,2,3]$

**Définition 2 :** Si  $\sigma$  est un  $k$ -simplexe défini par l'ensemble  $S$  de ses sommets, et  $\hat{S}$  désigne un sous-ensemble de  $S$  qui engendre un autre simplexe noté  $\epsilon$ , alors  $\epsilon$  est appelé face de  $\sigma$ .

### III.1.2 Complexe Simplicial

Un complexe simplicial  $K$  est une union d'ensembles finis de simplexes, défini comme suit (Fig 21) :

1. Si  $\theta$  désigne un des simplexes de  $K$  alors toutes les faces de  $\theta$  sont dans  $K$ .
2. Si deux simplexes appartiennent à  $K$ , leur intersection est soit vide soit une face commune à ces deux simplexes.

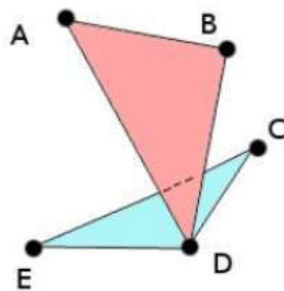


Figure 21. Un complexe simplicial.

### III.1.3 Filtration d'un complexe simplicial

**Définition :** Une filtration d'un complexe simplicial (fini)  $K$  est une suite  $(K^i)$  de sous-complexes imbriqués de  $K$  définis comme suit :

$$\begin{cases} (i) K^0 \subseteq K^1 \subseteq K^2 \dots \subseteq K \\ (ii) K^{i+1} = K^i \cup \sigma^{i+1} \text{ où un } \sigma^{i+1} \text{ est un simplexe de } K \end{cases}$$

Deux types de complexes simpliciaux sont couramment utilisés pour construire une filtration : les complexes de Čech et de Vietoris-Rips qui peuvent être définis sur des espaces métriques:

**Définition:** Un espace métrique est la donnée d'une paire  $(S, d)$  où  $S$  désigne un ensemble et  $d$  une fonction distance  $d : M \times M \rightarrow \mathbb{R}$  satisfaisant les propriétés suivantes:

1.  $d(x, y) \geq 0 \quad \forall (x, y) \in M \times M$
2.  $d(x, y) = 0$  si et seulement si  $x = y$
3.  $d(x, y) = d(y, x) \quad \forall (x, y) \in M \times M$
4.  $d(x, y) \leq d(x, z) + d(z, y) \quad \forall (x, y, z) \in M \times M \times M$

### III.1.4 Complexes de Čech et de Vietoris-Rips

**Définition :** Soit  $(S, d)$  un espace métrique où  $S$  désigne un ensemble de points. Etant donné un réel  $\varepsilon > 0$ , le complexe de Čech est le complexe simplicial noté  $Cechs(S)$  et défini par :

$$\sigma \in Cechs(S) \text{ si il existe une paire de points } p, q \text{ de } \sigma \text{ telle que } d(p, q) < \varepsilon.$$

**Définition :** Soit  $(S, d)$  un espace métrique où  $S$  désigne un ensemble de points. Etant donné un réel  $\varepsilon > 0$ , le complexe de Vietoris-Rips est le complexe simplicial noté  $Rips(S)$  et défini par :

$$\sigma \in Rips(S) \text{ si et seulement si } d(p, q) < \varepsilon$$

Pour toute paire de points  $p, q$  de  $\sigma$ .



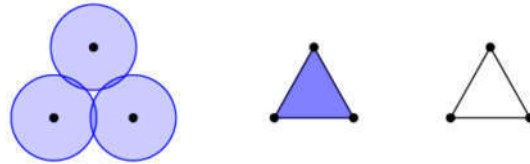


Figure 22. (1)  $\frac{\varepsilon}{2}$  voisinages de points (2) Rips-Vietoris complexe (3) Čech complexe.

*Remarque* : Le problème du complexe de Čech est son coût de calcul important (Kerber & R. Sharathkumar, 2013).

Exemple : étapes de la construction d'un complexe de Rips :

- Pour  $\varepsilon = \varepsilon_0$  on a :  $\beta_0 = 5$
- Pour  $\varepsilon = \varepsilon_1$  on a :  $\beta_0 = 2$
- Pour  $\varepsilon = \varepsilon_2$  on a :  $\beta_0 = 1$

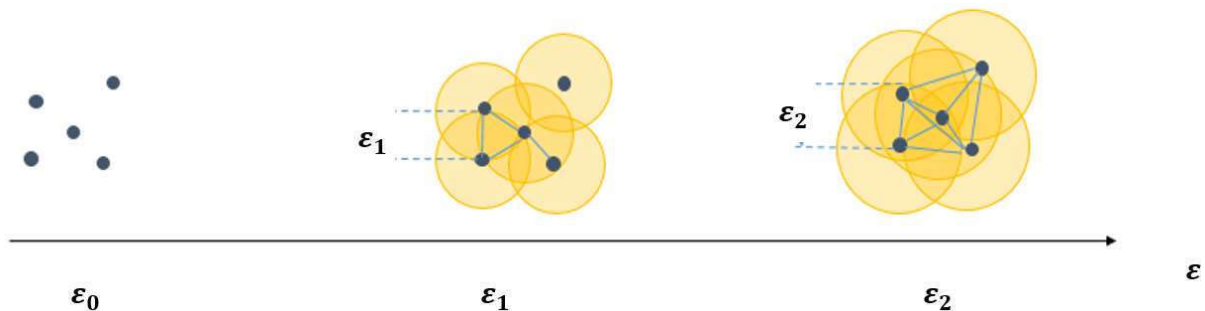


Figure 23. Filtration de Rips.

### III.1.5 Echantillonnage « Landmark selection »

Dans le cas où le nuage de points est volumineux, l'utilisation de la filtration de Rips sera coûteuse en temps de calcul et engendrera un très grand nombre de simplexes. L'alternative qui se présente est d'échantillonner le nuage en sélectionnant un sous-ensemble de sommets qui seront les seuls points sur lesquels on appliquera la filtration.

La sélection de ce sous-ensemble  $L = \{l_1, \dots, l_m\} \subset S = \{s_1, \dots, s_n\}$  de sommets peut se faire selon deux approches d'échantillonnage :

- La première est la sélection du sous-ensemble  $L$  d'une façon complètement aléatoire à partir du nuage  $S$ .
- La seconde est un processus appelé « sequential maximin » : le premier point de l'échantillon est choisi d'une manière aléatoire. Ensuite, les points suivants sont définis par une relation de récurrence, le «i-ème» point étant sélectionné en tenant compte du (i-1)ième point de manière à maximiser la distance  $d(l_i, l_{i-1})$ . Maximiser la distance entre les points de l'échantillon permet de couvrir au mieux le nuage de points initial  $S$ .

Après l'échantillonnage du nuage de points, on a souvent recours à deux méthodes pour construire le complexe simplicial : « Witness streams » et « Lazy witness streams » ( Adams, Tausz, & Vejdemo-Johansson, 2014) .

### III.1.5.1 Witness streams

Soit l'espace  $X$  défini par le nuage de point  $S$  et  $L$  le sous-ensemble obtenu après l'échantillonnage,  $M_k(s)$  est la distance entre un point  $s \in S$  et le  $(k+1)$ -ième plus proche point dans le sous-ensemble  $L$ . Le witness stream complex noté  $W(S, L, t)$  est défini comme suit ( Adams, Tausz, & Vejdemo-Johansson, 2014):

- $L = \{l_0, l_1, \dots, l_n\}$  est l'ensemble des sommets.
- $t$  est un réel  $>0$ .
- Pour  $k > 0$  et les sommets  $l_i$ , le  $k$ -simplexe  $\sigma \in W(S, L, t)$  si toutes les faces du  $k$ -simplexe sont aussi dans  $W(S, L, t)$ , et qu'il existe un point  $s \in S$ , appelé point témoin qui vérifie la condition suivante :  $\max\{d(l_0, s), d(l_1, s), \dots, d(l_k, s)\} \leq t + m_k(s)$ .

### III.1.5.2 Lazy witness streams

C'est une variante de la méthode witness streams, en ajoutant un paramètre  $v$  qui peut prendre les valeurs 0, 1 ou 2, il détermine la manière dans laquelle le complexe  $LW_v(S, L, t)$  est construit.

$S$  est le nuage de points initial et  $L$  le sous-ensemble obtenu avec l'échantillonnage. Si  $v=0$  on pose  $m(z) = 0$ . Pour les autres valeurs de  $v$ ,  $m(s)$  est la distance entre le point témoin  $s \in S$  et le  $v$ -ième point le plus proche dans  $L$ , pour tous les points  $z$  dans  $Z$  si  $v > 0$  alors  $m(z)$  est la distance entre  $z$  et le  $v$ -ième plus proche point dans le sous-ensemble d'échantillons  $L$ . Le lazy witness complexe est défini comme suit ( Adams, Tausz, & Vejdemo-Johansson, 2014):

- $L : \{l_0, l_1, \dots, l_n\}$  est l'ensemble des sommets.
- Pour les sommets  $l_\alpha$  et  $l_{\hat{\alpha}}$ , l'arête  $[l_\alpha, l_{\hat{\alpha}}]$  appartient au complexe  $LW_v(S, L, t)$ , si il existe un point témoin  $s \in S$ , tel que :  $\max\{d(l_\alpha, s), d(l_{\hat{\alpha}}, s)\} \leq t + m(s)$ .

### III.1.6 Chaines et faces de simplexes, bord d'une face

Dans tout ce qui suit,  $K$  désigne un complexe simplicial.

**Définition 1:** On note  $C_p(K)$  l'espace vectoriel à coefficients dans  $Z/2Z$  engendré par les  $p$ -simplexes dans  $K$ . On appelle  $p$ -chaîne de  $C_p(K)$ , tout élément  $c$  défini par:  $c = \sum \epsilon_i \sigma_i$ , où  $\epsilon_i \in Z/2Z$ , et  $\sigma_i$  désigne un  $p$ -simplexe dans  $K$ .

**Définition 2:** On définit le bord du  $p$ -simplexe  $\sigma$  de  $C_p(K)$ , comme la somme de ses  $(k-1)$  faces, qu'on note  $\partial(\sigma)$ .

**Définition 3:** Le sous-espace vectoriel  $Z_p(K)$  de  $C_p(K)$  défini par:

$$Z_p(K) = \{c \in C_p(K) \mid \partial_p(c) = \emptyset\}$$

dont les éléments sont appelés des  $k$ -cycles est l'ensemble des  $p$ -chaines sans bord.

**Définition 4:** Le sous-espace vectoriel  $B_p(K)$  de  $C_p(K)$  défini par:  $B_p(K) = \{c \in C_p(K) \mid \exists d \in C_{p+1}(K) : \partial_{p+1}(d) = c\}$  est l'ensemble des  $p$ -bords.

La propriété:  $\partial \circ \partial = \partial^2 = 0$  implique  $B_p(K) \subseteq Z_p(K)$ . On peut alors définir:

### III.1.7 Groupes d'homologie

**Définition 5:** Le groupe d'homologie simpliciale de dimension  $p$  de  $K$ , noté  $H_p(K)$ , est défini par:  $H_p(K) = Z_p(K)/B_p(K)$ . Le rang de  $H_p(K)$  est appelé nombre de Betti de dimension  $p$  de  $K$  et noté  $\beta_p(K)$ .

Ainsi,  $\beta_0(K)$  désigne le nombre de 0-cycles non bordants ou composantes connexes de  $K$ ,  $\beta_1(K)$  le nombre de 1-cycles non bordants ou "trous" et  $\beta_2(K)$  le nombre de 2-cycles non bordants ou "cavités" (Fig 24).

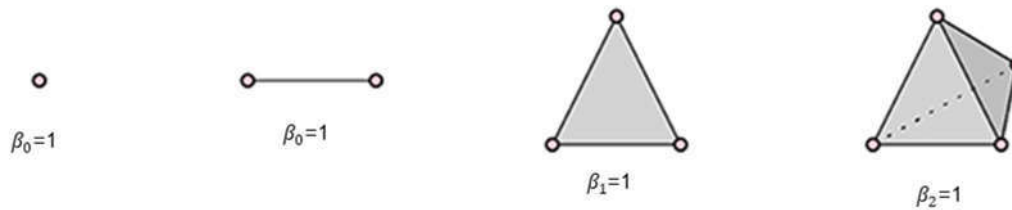


Figure 24. Nombres de Betti.

### III.1.8 La persistance homologique

#### III.1.8.1 Principe

L'homologie persistante prend un complexe simplicial filtré  $K$  et a comme particularité de calculer comment les caractéristiques topologiques (nombres de Betti) persistent au cours de la filtration.

Plus précisément, l'inclusion  $f_i^j: K^i \rightarrow K^j$ , pour  $i \leq j$ , induit une application linéaire pour chaque dimension  $p$ :  $h_i^j: H_p(K^i) \rightarrow H_p(K^j)$ .

**Définition:** On dit alors qu'une classe d'homologie  $\alpha \in H_p(K^i)$  est née en  $\varepsilon_i$  si  $\alpha \notin \text{Im}(h_{i-1}^i)$ . On dit qu'une classe  $\alpha \in H_p(K^i)$  née en  $\varepsilon_i$  meurt en  $\varepsilon_j$  si  $h_i^{j-1}(\alpha) \notin \text{Im}(h_{i-1}^{j-1})$  mais  $h_i^j(\alpha) \in \text{Im}(h_{i-1}^j)$ . La différence  $\varepsilon_i - \varepsilon_j$  est appelée persistance de  $\alpha$ .

Elle mesure combien de "temps" une caractéristique homologique survit durant la filtration. Les classes nées en  $\varepsilon_i$  qui ne meurent pas sont associées à  $[\varepsilon_i; +\infty[$  et sont appelées essentielles, les autres classes étant appelées inessentielles.

**Définition:** Pour  $i \leq j$ , le groupe d'homologie persistante d'ordre  $p$  est défini par:  $H_p^{i,j} = Z_p(K^i) \setminus (B_p(K^j) \cap Z_p(K^i))$ . Les éléments non triviaux de  $H_p^{i,j}$  correspondent aux classes d'équivalence des  $p$ -cycles nés avant ou en  $\varepsilon_i$  et qui persistent, i.e. qui ne meurent pas durant la filtration pour  $\varepsilon \in [\varepsilon_i; \varepsilon_j[$ .

Remarque: Pour  $i = j$ , cette notion recouvre celle d'homologie usuelle:

$$H_p^{i,i} = H_p(K^i) = Z_p(K^i) \setminus B_p(K^i).$$

La "durée de vie" des caractéristiques topologiques, pour chaque dimension  $p$ , peuvent être visualisées soit en utilisant le diagramme persistant d'ordre  $p$  ou le barcode  $\beta_p$  (Fig 25). Dans le barcode, l'axe

vertical représente le nombre de générateurs homologiques de dimension  $p$  (nombre de Betti de dimension  $p$ ) et l'axe horizontal correspond à la valeur de filtration. Une autre représentation graphique, le diagramme de persistance d'ordre  $p$  associe  $(\varepsilon_i; \varepsilon_j)$  aux classes "nées" en  $\varepsilon_i$  et "mortes" en  $\varepsilon_j$  et l'intervalle  $[\varepsilon_j; +\infty[$  aux classes essentielles nées en  $\varepsilon_i$ . Le nombre de points dans la région du plan  $]-\infty; \varepsilon_i] \times [\varepsilon_j; +\infty[$  correspond à  $\dim(H_p^{i,j})$ .

Exemple : Le diagramme obtenu ci-dessous décrit la persistance des composantes du groupe d'homologie d'ordre 0 tout au long de la filtration :

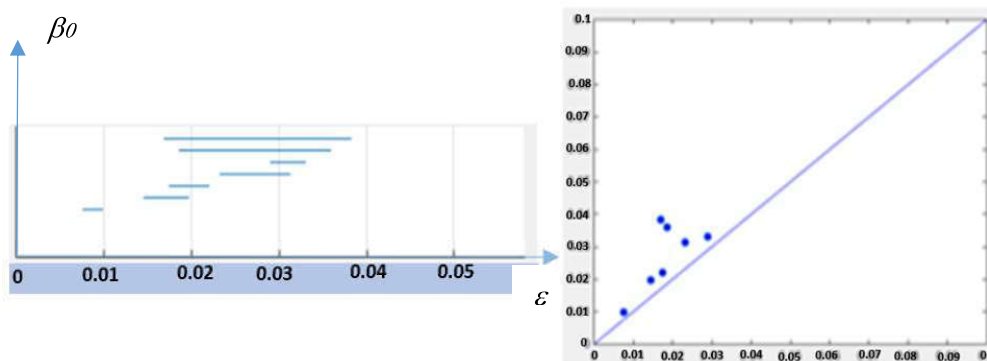


Figure 25. Représentation du code-barres (à gauche) et de son équivalent, le diagramme de persistance d'ordre 0 (à droite).

Afin de comparer les mouvements, nous devons trouver un moyen d'estimer la similarité ou dissimilarité entre leurs signatures topologiques. Pour cela, nous pouvons définir une mesure de distance entre les diagrammes de persistance d'ordre  $p$  en utilisant un algorithme de "matching" dit algorithme bipartite hongrois (EDELSBRUNNER & MOROZOV, 2012) (Mileyko, Mukherjee, & Harer, 2011). Nous pouvons ainsi définir des métriques qui sont appelées distance de Wasserstein distance et de Bottleneck.

### III.1.8.2 La distance de Wasserstein

**Definition:** Soient  $X$  et  $Y$  des espaces topologiques définis à partir de deux nuages de points. On note  $Dgm_p(X)$  et  $Dgm_p(Y)$  leurs diagrammes de persistance d'ordre  $p$ . Soit  $\theta: Dgm_p(X) \rightarrow Dgm_p(Y)$  un matching entre les diagrammes de persistance d'ordre  $p$ , alors la distance de Wasserstein de degré  $l$  est définie par:

$$W_l(X, Y) = \left[ \sum_p \inf_{\theta_p: Dgm_p(X) \rightarrow Dgm_p(Y)} \sum_{x \in Dgm_p(X)} \|x - \theta(x)\|_\infty^l \right]^{1/l}$$

### III.1.8.3 La distance de Bottleneck

**Definition:** Soient  $X$  et  $Y$  des espaces topologiques définis à partir de deux nuages de points. On note  $Dgm_p(X)$  et  $Dgm_p(Y)$  leurs diagrammes de persistance d'ordre  $p$ . Soit  $\theta: Dgm_p(X) \rightarrow Dgm_p(Y)$  un matching entre les diagrammes de persistance d'ordre  $p$ , alors la distance de Bottleneck est définie par:

$$d_p^B(X, Y) = \inf_{\theta_p: Dgm_p(X) \rightarrow Dgm_p(Y)} \left[ \sup_{x \in Dgm_p(X)} \|x - \theta(x)\|_\infty \right]$$

### III.1.9 Stabilité des diagrammes de persistance

Si  $(X, d_X)$  et  $(Y, d_Y)$  sont des espaces métriques, la distortion d'une correspondance  $C: X \rightrightarrows Y$  est définie comme suit:

$$dis(C) = \sup\{|d_X(x, x') - d_Y(y, y')| : (x, y), (x', y') \in C\}$$

La distance de Gromov-Hausdorff entre  $(X, d_X)$  et  $(Y, d_Y)$ , est alors définie par :

$$d_{GH}(X, Y) = \frac{1}{2} \inf\{dis(C) : C \text{ est une correspondance } X \rightrightarrows Y\}$$

Le théorème de stabilité est un des résultats majeurs de l'analyse topologique de données, donnant des garanties théoriques pour calculer approximativement la persistance homologique.

Une première version de ce théorème due à Cohen-Steiner et al. (2005) concerne la persistance homologique des fonctions à valeurs réelles.

Il a été revisité depuis dans un cadre plus général. En particulier, Chazal et al. (2009) en donnent une version suivante :

**Théorème** : Pour tous espaces métriques  $(X, d_X)$  et  $(Y, d_Y)$ , pour tout  $k \in \mathbb{N}$ :

$$d_{\infty}^B(D_k(R(X)); D_k(R(Y))) \leq d_{GH}(X, Y)$$

où  $D_k(R(X))$  et  $D_k(R(Y))$  désignent les diagrammes de persistance d'ordre  $k$  des complexes de Vietoris-Rips de  $X$  et  $Y$ ,  $d_{\infty}^B$  la distance de Bottleneck et  $d_{GH}$  la distance de Gromov-Hausdorff.

### III.1.10 Travaux basés sur la persistance homologique dans un contexte applicatif

Dans Collins et al (Collins, Zomorodian, & Carlsson, 2004), l'homologie persistante est utilisée dans le but de classer des formes représentées par des nuages de points. Cette méthode d'analyse est de plus en plus utilisée dans le domaine médical. Adcock et al (Adcock, Rubin, & Carlsson, 2014) analysent les images médicales en utilisant la persistance homologique pour dépister et classer les lésions hépatiques. Les images sont représentées sous forme de nuages de points, selon l'intensité de leurs pixels (Fig 26), sur lesquels l'analyse topologique est appliquée.

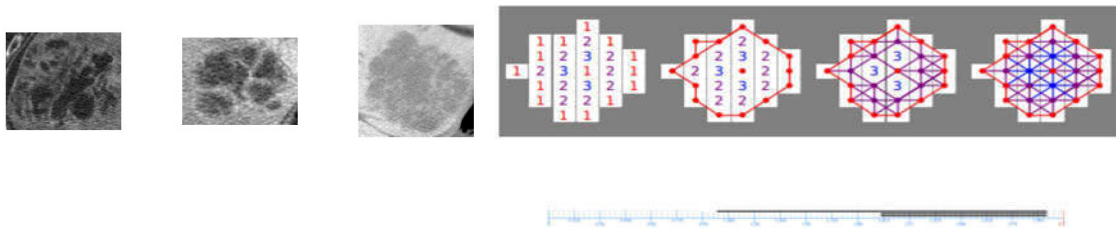


Figure 26. (a) des images médicales de lésions hépatiques, (b) la représentation des images selon l'intensité de leurs pixels et la réalisation d'une filtration simpliciale, (c) la signature topologique (code-barres) obtenue.

Les articles de Heo et al (Heo & Gamble, Exploring uses of persistent homology for statistical analysis of landmark-based shape data, 2010) et et (Heo, Gamble, & T. Kim, Topological Analysis of Variance and the Maxillary Complex, 2012), présentent une approche basée sur l'homologie persistante, dans l'objectif de surveiller les effets cliniques des traitements orthodontiques. Les marqueurs sont placés à différents points dans la mâchoire des patients (Fig 28), les coordonnées de ces marqueurs sont utilisées comme entrées pour l'homologie persistante.

La méthode d'analyse topologique est aussi utilisée par Lee et al (Lee, K. Chung, Kang, Bung-Nyun, & Dong Soo, 2011), pour classer l'activité des réseaux de neurones naturels (Fig 27).

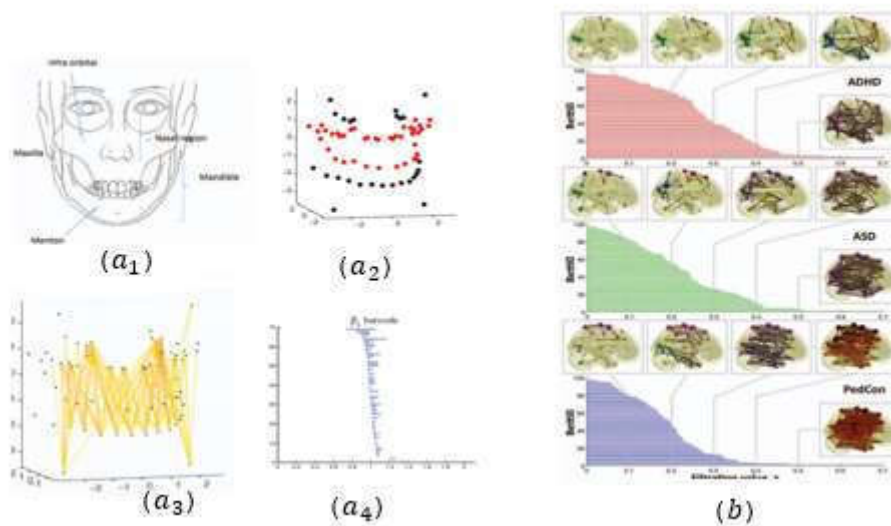


Figure 27. a) (1,2,3,4) démarche proposée par Heo et al ,  $a_1$  placer des marqueurs sur la mâchoire du patient,  $a_2$  récupération des coordonnées 3D de ces marqueurs, après la réalisation de la filtration  $a_3$  , on obtient le barres-code  $a$ , représentant la signature topologique. b) travaux de Lee et al pour analyser l'activité des réseaux de neurones avec la persistance homologique.

### III.1.11 Machine à vecteurs de support

La Machine à vecteurs de supports ou SVM (pour Support Vector Machine), est une méthode de classification supervisée introduite par V. Vapnik et A. Lerner en 1963 (Pattern Recognition using Generalized Portrait Method, 1963) dont le but est de résoudre des problèmes de discrimination, autrement dit de décider à quelle classe appartient un échantillon donné. Si on se place dans un espace muni d'un produit scalaire comme  $R^n$ , le problème de discrimination linéaire consiste à trouver une fonction qui définit l'hyperplan optimal séparant deux classes de données (Fig 28).

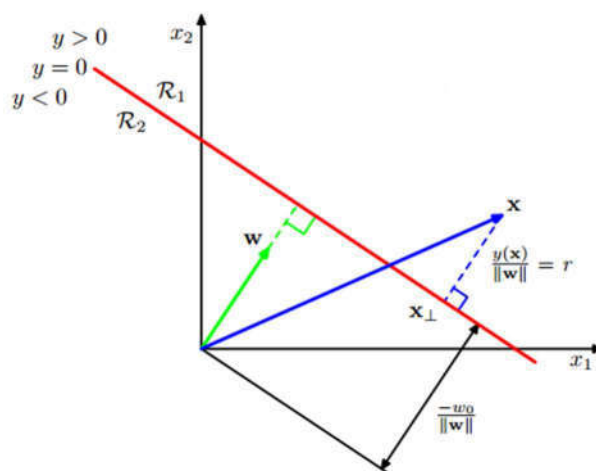


Figure 28. Hyperplan séparateur de deux classes de données.



Un hyperplan  $H$  est défini par la fonction suivante :  $y(x) = Wx + b$  où  $x$  désigne une entrée,  $W$  est un vecteur normal à l'hyperplan  $H$  qui va déterminer son orientation et  $b$  sa position. La fonction  $y$  retourne une valeur supérieure à 0 si le point appartient à la classe  $C_1$  et une valeur inférieure à 0 si le point appartient à la classe  $C_2$  :

- $y(x) \geq 0$  si  $x \in C_1$
- $y(x) \leq 0$  si  $x \in C_2$

La distance entre une entrée  $x$  et l'hyperplan séparateur  $H$  est définie par (Fig 29) :

$$d(x, H) = \frac{y(x)}{\|W\|} = \frac{Wx + b}{\|W\|}$$

On appelle marge la distance  $d$  entre les deux classes de données. Pour séparer les deux ensembles, il existe plusieurs hyperplans possibles. L'hyperplan optimal qu'on cherche est celui qui maximise la marge. La marge correspond à la plus petite distance entre l'hyperplan optimal et le plus proche point de chacune des classes (Fig 29).

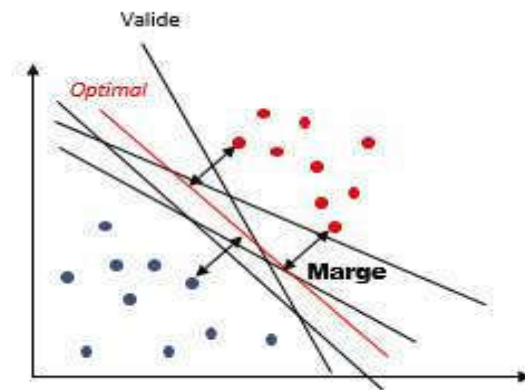


Figure 29. L'hyperplan séparateur optimal.

Les points qui se retrouvent sur la marge (Fig 31) sont nommés les supports de vecteurs. Pour ces points, on suppose que la fonction de classification  $y$  retourne exactement 1 ou -1 :

$$y(x_i) = W \cdot x_i + b = \pm 1$$

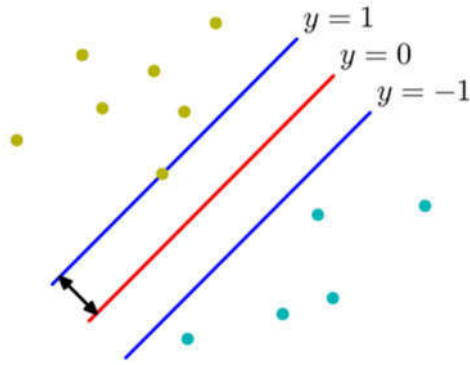


Figure 30. Les vecteurs de support.

La marge  $d$  est alors donnée par :

$$d = \frac{2}{\|W\|}$$

Maximiser la valeur de la marge revient à minimiser la valeur de  $W$ . Le problème revient alors à trouver  $W$  et  $b$  tels que  $d$  soit maximale pour tout  $(x_i; y_i)$  sous les contraintes:

$$\begin{cases} W \cdot x_i + b \geq 1 \text{ si } y_i = 1 \\ W \cdot x_i + b < 1 \text{ si } y_i = -1 \end{cases}$$

Le problème peut s'écrire de manière équivalente sous la forme :

$$\begin{cases} \min_{w,b} \frac{1}{2} \|W\|^2 \\ \forall i = 1..N \quad y_i(w \cdot x_i + b) \geq 1 \end{cases}$$

Minimiser la valeur de  $W$  sous la contrainte représente un problème d'optimisation quadratique (H.K. Xu, 2003) qu'on peut simplifier en un problème dual de minimisation en utilisant les multiplicateurs de Lagrange (Bierlaire, 2006):

$$\begin{cases} \sum_{i=1}^N \alpha_i - \frac{1}{2} \sum_{i,j=1}^N \alpha_i \alpha_j y_i y_j x_i \cdot x_j \\ \sum_{i=1}^N y_i \alpha_i = 0, \quad \alpha_i \geq 0 \end{cases}$$

où  $\alpha_i$  sont les multiplicateurs de Lagrange.

### III.1.11.1 Classification avec une marge d'erreur

Dans les cas pratiques, on se retrouve souvent confronté à des problèmes avec des classes qui se chevauchent (Fig 31). Dans ces cas, on doit faire un compromis entre la complexité du classifieur et sa précision.

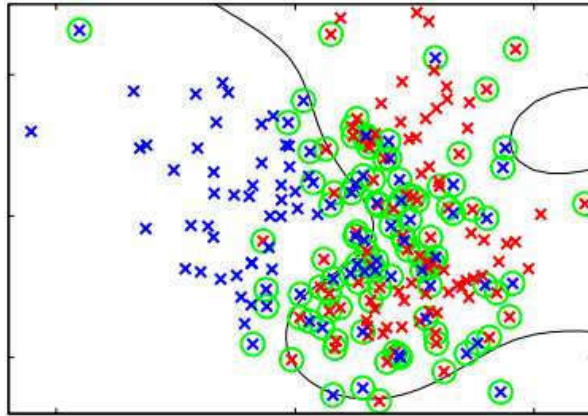


Figure 31. Chevauchement de classes.

L'idée est d'ajouter une variable d'ajustement  $\varepsilon_i$  appelée variable de ressort dans la formulation du problème de minimisation pour prendre en compte les erreurs de classification ou le bruit ( V. Vapnik, 1995) :

$$\left\{ \begin{array}{l} \min_{w,b} \frac{1}{2} \|W\|^2 + C \sum_{i=1}^N \varepsilon_i \\ \forall i = 1..N \quad y_i(w \cdot x_i + b) \geq 1 - \varepsilon_i \end{array} \right.$$

Si la valeur de  $\varepsilon_i$  est comprise entre 0 et 1, l'individu ne respecte pas la marge mais reste dans le bon côté du classifieur. Par contre, si elle est supérieure à 1, l'individu est du mauvais côté.

On ajoute aussi une constante C qui ajuste la précision du classifieur. Plus la valeur de C est grande plus le classifieur est précis et moins il est indulgent par rapport aux individus mal classés.

La formulation sous forme d'un problème dual d'optimisation est similaire au cas précédent, à l'exception que les multiplicateurs de Lagrange sont bornés par la variable C ( V. Vapnik, 1995) :

$$\left\{ \begin{array}{l} \sum_{i=1}^N \alpha_i - \frac{1}{2} \sum_{i,j=1}^N \alpha_i \alpha_j y_i y_j x_i \cdot x_j \\ \sum_{i=1}^N y_i \alpha_i = 0, \quad 0 \leq \alpha_i \leq C \end{array} \right.$$

### III.1.11.2 Machines à vecteurs de supports non-linéaires

La limite de l'utilisation de la variable de ressort  $\varepsilon_i$  est atteinte quand les données sont complètement non-linéairement séparables. Dans ce cas, au lieu de chercher l'hyperplan de séparation optimal dans l'espace initial des données, on projette les données dans un espace de dimension supérieure, en utilisant une fonction noyau (Fig 32).

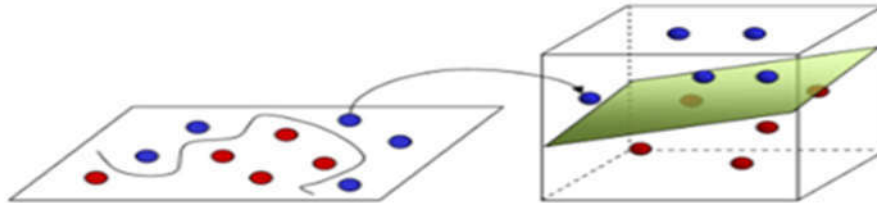


Figure 32. Projection des données dans un espace de dimension supérieure en utilisant une fonction noyau en cas de non séparabilité linéaire des données.

Le changement de dimension s'effectue via une transformation  $\theta: x \rightarrow \theta(x)$  telle que, pour deux entrées  $x_i$  et  $x_j$ , la fonction de transformation  $K$ , appelée fonction noyau, soit définie par :  $K(x_i, x_j) = \langle \theta(x_i), \theta(x_j) \rangle$ .

Remarque : Les fonctions noyaux les plus utilisées sont (Nello Cristianini & John Shawe-Taylor, 2000) :

- Le noyau Gaussien :  $K(x_i, x_j) = \exp\left(-\frac{\|x_i - x_j\|^2}{2\sigma^2}\right)$
- Le noyau polynomial de degré  $p$  :  $K(x_i, x_j) = (1 + x_i \times x_j)^p$

### III.1.11.3 Machine à vecteur de support multi classes

La machine à vecteurs de support est à l'origine une méthode de classification binaire. Cependant, la majorité des problèmes de classification rencontrés dans le monde réel sont multi-classes. Pour cela, plusieurs extensions de la méthode de machine à vecteur de support ont été proposées. L'idée principale de ces extensions est de décomposer le problème multi-classes en une multitude de sous-ensembles représentant chacun un problème de classification binaire.

### III.1.11.4 Un-contre-tous (one against all)

C'est l'extension la plus ancienne et la plus basique. Elle a été introduite par Vapnik (Vapnik, V, 1995). Elle consiste à définir un hyperplan  $H_k$  pour chaque classe  $k$  pour la séparer des autres classes du problème (Ryan Rifkin & Aldebaro Klautau, 2004). Cet hyperplan  $H_k$  retourne (+1) si l'individu appartient à la classe  $k$  et (0) s'il appartient aux autres classes.

Pour un problème à  $N$  classes on aura  $N$  hyperplan  $H_k$  ( $k=1, \dots, N$ ) défini par la fonction suivante :

$$H_k(x) = \text{signe}(Wx + b) = \begin{cases} +1 & \text{si } x \in k \\ -1 & \text{sinon} \end{cases}$$

Le résultat retourné par l'hyperplan  $H_k$  permet de savoir si l'entrée  $x$  appartient à la classe  $k$  ou pas. Ainsi, pour classer une entrée, il faut la présenter à tous les hyperplans. On peut résumer le choix de la classe par la fonction de décision:

$$k^* = \text{Arg}_{1 \leq k \leq N} \text{Max}(H_k(x))$$

Si un seul hyperplan retourne 1, on peut conclure que l'entrée  $x$  appartient à la classe de cet hyperplan. L'inconvénient de cette méthode est que dans certains cas plusieurs hyperplans retournent 1 simultanément ce qui engendre des régions d'ambiguïté (Fig 33).

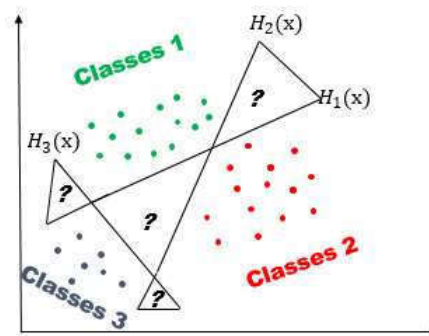


Figure 33. Classification multi-classes avec l'approche Un-contre-tous.

### III.1.11.5 Un-contre-un (one-against-one)

C'est une extension proposée par Kner et al (S. Knerr, L. Personnaz, & G. Dreyfus, 1990). Elle consiste à utiliser un hyperplan pour chaque paire de classes au lieu d'un hyperplan pour chaque classe. De cette manière, pour  $N$  classes, cette approche nécessite  $N(N-1)/2$  fonctions de décision au lieu de  $N$  pour l'approche d'un-contre-reste.

Pour chaque paire de classes  $(k_1, k_2)$ , on définit un hyperplan binaire  $H_{k_1 k_2}$  qui retourne (+1) si l'entrée appartient à la classe  $k_1$  et (-1) si elle appartient à  $k_2$ .

$$H_{k_1 k_2}(x) = \text{sign}(Wx + b) = \begin{cases} +1 & \text{si } x \in k_1 \\ -1 & \text{si } x \in k_2 \end{cases}$$

La méthode d'affectation reste la même et se fait par vote majoritaire. On présente l'entrée à tous les hyperplans et on l'affecte à la classe la plus votée.

Cette approche réduit le nombre de zones d'ambiguïté, mais elle ne les supprime pas complètement (Fig 34).

Cette approche réduit aussi le temps d'apprentissage parce que chaque hyperplan va apprendre à différencier entre deux classes au lieu d'une seule.

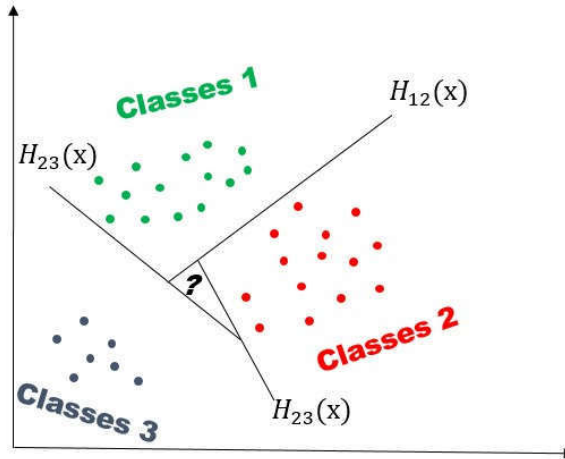


Figure 34. L'approche Un-contre-un.

### III.1.11.6 Graphe de décision

Afin d'éliminer l'inconvénient des zones d'ambiguïté dans l'approche une-contre-une et J.C. Platt et al (John C. Platt & Nello Cristianini, 1990) proposent pour la dernière étape de l'approche une-contre-une d'utiliser un graphe de décision au lieu du vote majoritaire.

La méthode d'apprentissage reste la même en définissant  $N(N-1)/2$  hyperplans. Après, on construit un graphe de décision qui va déterminer à quelle classe on affecte l'individu "x".

La construction du graphe s'appuie sur une mesure  $E_{k_1k_2}$  qui représente la capacité de généralisation de chaque hyperplan, elle est définie comme le rapport entre le nombre de vecteurs supports figure 12 et les nombres d'entrées des deux classes de l'hyperplan :

$$E_{k_1k_2} = \frac{\text{Nombre}_{\text{vecteur supports}}}{\text{Nombre}_{\text{entrées des deux classes}}}$$

La classification selon le graphe se fait selon les étapes suivantes :

1. Initialiser une liste  $L$  avec toutes les classes du problème.
2. Si  $L$  contient une seule classe : on crée un nœud avec l'étiquette de cette classe.
3. Calculer la mesure  $E_{k_1k_2}$  pour chaque hyperplan  $H_{k_1k_2}$ .

## Outils

4. Chercher les deux classes  $k_1$  et  $k_2$  séparées par l'hyperplan dont la mesure  $E_{k_1 k_2}$  est maximale.
5. Créer un nœud  $N$  avec l'étiquette  $E_{k_1 k_2}$ .
6.  $L = L - \{k_1\}$  retourner à l'étape 1 pour construire un graphe de décision à partir de la nouvelle liste  $L$  en l'attachant au fils gauche de  $N$
7.  $L = L - \{k_2\}$  retourner à l'étape 1 pour construire un graphe de décision à partir de la nouvelle liste  $L$  en l'attachant au fils droit de  $N$

On obtient à la fin un graphe de décision avec des classes comme feuilles et dont les nœuds internes sont les hyperplans (Fig 35).

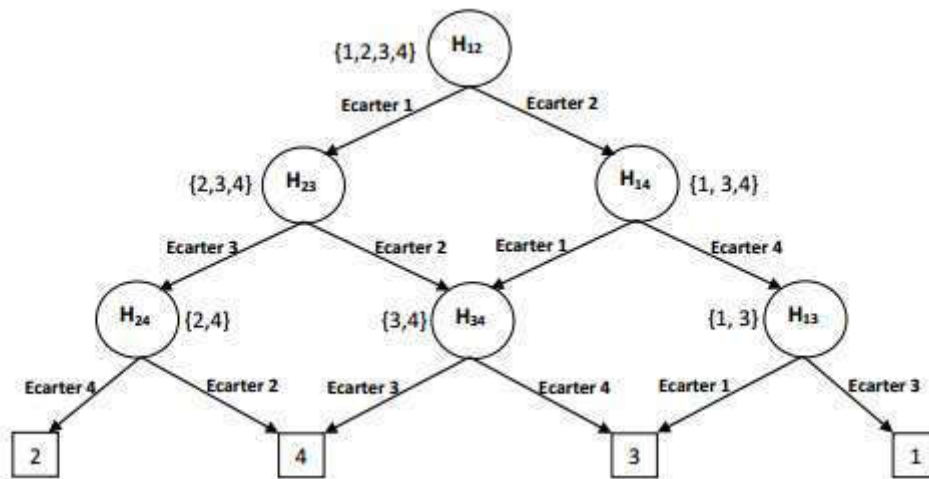


Figure 35. Classification par graphe de décision.

Pour une entrée  $x$  qu'on veut classer, on commence par la tester avec l'hyperplan qui se trouve à la racine du graphe. Si le résultat est positif, on continue avec le nœud fils gauche sinon le fils droit, et on répète les tests jusqu'à ce qu'on arrive à une des feuilles qui représentent les classes.

## III.2 Réduction dimensionnelle

Les données qu'on souhaite analyser sont souvent représentées dans un tableau de mesures qu'on peut considérer comme un nuage de points multidimensionnel. Le nuage de points repose sur une structure géométrique sous-jacente qui est une variété. Il convient alors de déterminer la dimension intrinsèque de la variété en question. Les méthodes classiques pour estimer la dimension intrinsèque d'un nuage de points multidimensionnel sont des méthodes de représentation des données dans des sous-espaces de plus petite dimension (2 ou 3) appelées méthodes de réduction dimensionnelles. L'objectif est de visualiser la distribution des données dans des espaces de dimension faible tout en conservant la même quantité d'information présente dans les données. Il existe de nombreuses méthodes qui permettent de déterminer la structure des données : des méthodes qui relèvent de processus linéaires comme l'ACP (Analyse en Composantes Principales), MDS (Multi-Dimensional Scaling) ou permettant de mettre en évidence les structures non-linéaires des données comme Isomap. Les deux dernières méthodes ont la particularité de préserver au mieux les distances entre points du nuage lors de la projection dans des sous-espaces de dimension plus faible. Dans le cas de la MDS, il s'agit de la distance euclidienne. Dans celui de l'Isomap, c'est la distance géodésique. Dans ce chapitre, nous présenterons ces méthodes utilisées dans notre approche.

### III.2.1.1 Analyse en composantes principales

La plus ancienne des méthodes de réduction dimensionnelle est l'analyse en composantes principales noté ACP dans la littérature. Les premiers travaux sur la méthode remontent à 1901 avec la publication de Karl Pearson (Pearson, 1901). A l'époque, la méthode est connue sous le nom de transformation de Karhunen-Loève, dont le but était d'utiliser la corrélation entre les variables, non pas pour expliquer la liaison entre une variable et une ou un ensemble de variables, mais pour résumer et structurer les informations contenues dans ces variables. La méthode a été développée en 1933 par l'économiste Harold Hotelling (Hotelling, H, 1933), qui a ajouté l'analyse factorielle à la méthode. Depuis, le nom d'analyse en composantes principales est utilisé.

Soit un ensemble de  $p$  individus, où chaque individu est défini par un ensemble de  $n$  variables quantitatives qui le décrivent. Ces informations sont représentées dans un tableau de données à  $p$  lignes et  $n$  colonnes.

Le but de l'ACP est d'étudier le tableau correspondant. L'étude du tableau peut se faire soit en considérant le tableau comme un ensemble de lignes ou bien en le considérant comme un ensemble de colonnes. Dans une étude sur l'ensemble des lignes, on va s'intéresser à l'étude



des  $p$  individus, en essayant de les regrouper en classes homogènes selon les variables  $n$  qui les caractérisent.

De la même manière, une étude au niveau des colonnes (variables) a comme but d'analyser les informations apportées par ces dernières, en recherchant les variables qui apportent une information presque identique sur les individus et celles qui apportent une information différente. Pour mesurer la ressemblance entre les informations apportées par les différentes variables et leurs impacts sur la description et la classification des individus, on peut se référer au coefficient de corrélation entre ces variables dans le but de visualiser les liaisons entre les variables sans passer en revue les liaisons entre couples de variables séparément. Cette visualisation peut se faire en cherchant des indicateurs synthétiques qui résument les informations fournies par l'ensemble des variables.

L'étude des individus (sur les lignes) et les variables (sur les colonnes) permet de :

- Utiliser les variables pour caractériser les différences entre les classes d'individus.
- Illustrer la liaison (corrélation) entre variables par des exemples d'individus extrêmement différents.

L'objectif final de l'analyse par composantes principales est de synthétiser l'ensemble de variables qui décrivent un ensemble d'individus en cherchant une représentation (projection des individus) visuelle dans une dimension réduite (en 2 ou 3 dimensions) qui respecte le plus fidèlement possible les informations fournies dans le tableau descriptif initial.

On peut résumer les étapes de l'ACP en quatre points principaux :

1. Centrer et réduire la matrice des individus.
2. Calculer les valeurs et vecteurs propres.
3. Déterminer la dimension du sous-espace dans lequel les individus seront projetés.
4. Projeter les individus dans le sous-espace sélectionné.

### **III.2.1.2 Positionnement multidimensionnel**

Le positionnement multidimensionnel est une approche qui permet d'estimer les coordonnées d'un ensemble d'individus dans un espace de dimension réduite généralement en deux ou en trois dimensions, et cela en partant des valeurs de similarité ou de dissimilarité de ces individus.

La méthode suppose que la (di)similarité entre chaque paire d'individus est définie par leur distance euclidienne. Les valeurs sont issues d'une matrice carrée symétrique ce qui nous donne  $n \times (n-1)/2$  valeurs de (di)similarité où  $n$  est la dimension de la matrice.

Les étapes du positionnement multidimensionnel sont les suivantes :

1. Calculer le carré de la matrice de (di)similarité  $P^{(2)} = [p^2]$  .
2. Calculer la matrice de double centrage  $B = -\frac{1}{2}JP^{(2)}J$  où la matrice  $J$  est calculée comme suit :  $J = I - n^{-1}\mathbf{1}^n$  avec  $I$  qui est la matrice identité  $n$  la dimension de la matrice de départ,  $\mathbf{1}^n$  une matrice de dimension  $n$  qui contient des uns.
3. Extraire les  $n$  plus grands valeurs propres positives  $\lambda_1, \dots, \lambda_n$  de  $B$  et les  $e_1, \dots, e_n$  vecteurs propres qui les correspondent. Le nombre  $n$  correspond à la dimension dans laquelle les individus vont être représentés. Ce nombre peut être choisi soit directement, par exemple si on veut que la représentation soit dans deux ou trois dimensions, ou bien en sélectionnant les vecteurs propres qui correspondent aux valeurs propres les plus significatives par rapport aux autres.
4. Une fois qu'on a déterminé les  $n$  valeurs propres, on calcule la matrice de coordonnées :

$$X = \lambda \times e$$

### III.2.1.3 Isomap (isométric feature mapping)

La méthode du positionnement multidimensionnel est une méthode de projection partant du principe que les éléments dans leur espace d'origine sont répartis sur une variété linéaire. Elle s'appuie sur la distance euclidienne pour calculer la distance entre eux.

Dans le cas où les données sont représentées dans un espace non-linéaire (Fig 37), l'utilisation du positionnement multidimensionnel qui se base sur la distance euclidienne biaise la représentation des données dans une dimension réduite lors de la projection.

La solution proposée par Joshua B. Tenenbaum et al (B. Tenenbaum, de Silva, L. Schwartz, & C. Langford, 2002) est de transformer la distribution non-linéaire en distribution linéaire (Fig 36), en utilisant la distance géodésique (Michel, 1994) au lieu de la distance euclidienne.

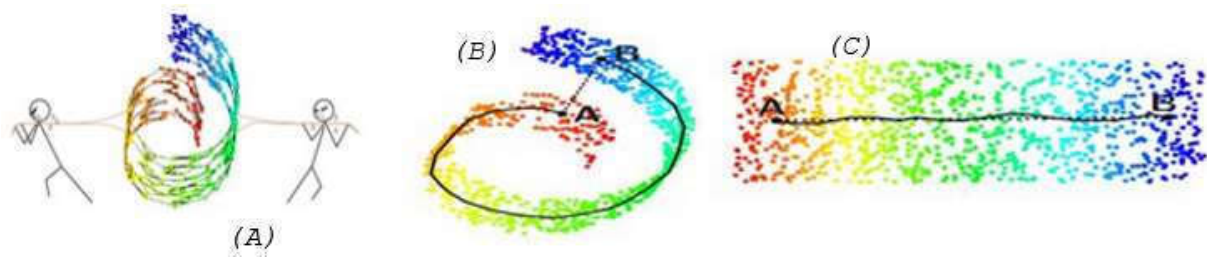


Figure 36. Pour une distribution non-linéaire (A), l'utilisation de la distance géodésique (B) permet de respecter les distances entre les individus de la distribution initiale (C).

Les étapes de la méthode d'Isomap sont les suivantes :

- Étape 1 : déterminer l'ensemble de voisinage pour chaque point dans l'espace de plongement, et cela soit en connectant chaque point avec tous les autres points qui sont à une distance inférieure à une constante  $\epsilon$  ou bien en fixant une valeur à  $K$  et connecter chaque point avec ses  $K$ -plus proches voisins.
- Étape 2 : une fois que chaque point est connecté avec ses voisins, on obtient un graphe  $G$  qu'on va pondérer par ses arêtes. Le poids de chacune d'elles est égale à la distance euclidienne  $d_e(i, j)$ . Pour les petites distances, on estime que la distance euclidienne est égale à la distance géodésique entre les deux sommets de l'arête.
- Étape 3 : On calcule la distance géodésique entre tous les points du graphe  $G$ . Cela revient à calculer le plus petit chemin « le chemin qui contient le moins d'arêtes possibles » entre deux points. La distance géodésique est la somme des poids de ces arêtes. Pour trouver le chemin le plus court entre deux points, on utilise l'algorithme de Dijkstra (Michel Minoux, 1975). Toutes ces distances sont enregistrées dans une matrice.
- Étape 4 : Une fois qu'on obtient la matrice de distance carrée, on applique le positionnement dimensionnel détaillé de la partie précédente.

Comme le montre la figure (Fig 43), on réalise tout d'abord une réduction dimensionnelle et la labélisation des coordonnées obtenues qui représentent la distribution des signatures topologiques. Ensuite, on effectue une classification pour être en mesure de reconnaître si une signature topologique d'une posture donnée appartient à tel ou tel mouvement.

Plusieurs travaux comme T. Hachaj (Hachaj, R. Ogiela, & Koptyra, 2016), S. Nair et al (S. Nair, M. French, Laroche, & Thomas, 2010) et Asha et al (Asha, Sudalaimani, Devanand, Thomas, & Sudhamony, 2013) illustrent les capacités des machines à vecteurs de supports (SVM) dans la classification des données biomécaniques.

# Chapitre IV Méthodologie

## IV.1 Questions de recherche

Notre problématique scientifique est relative à la classification de mouvements humains par une analyse topologique des données de capture de mouvement.

Dans ce contexte, nos travaux se sont focalisés sur trois questions de recherche.

La première question de recherche est relative à l'étape d'analyse topologique : est-ce que l'ajout d'une étape d'analyse topologique des données provenant de la capture de mouvement permet d'obtenir une classification plus précise et plus robuste des mouvements ?

La deuxième question de recherche traite de la mesure de différence entre deux mouvements : est-ce qu'une signature topologique du mouvement humain permet de définir une valeur quantifiant la différence entre deux mouvements ?

La troisième question de recherche concerne l'anticipation de mouvement : est-ce qu'une combinaison entre une analyse topologique et des réseaux de neurones permet d'anticiper un mouvement à partir de ces premiers instants ?

## IV.2 Approche proposée

L'état de l'art a montré que les systèmes de capture de mouvement sont de plus en plus utilisés pour des études thérapeutiques ou des applications médicales autres.

Les différentes méthodes de reconnaissance de mouvements qui utilisent les données enregistrées comme entrées directes aux algorithmes de classification font face à deux problèmes majeurs :

- La robustesse de la méthode face au bruit blanc lors de l'enregistrement.
- Les différences morphologiques entre les sujets, qui peuvent induire les algorithmes de classification en erreur.

Le but de notre travail est de proposer une approche qui permet d'améliorer la classification et la reconnaissance des mouvements humains en ajoutant une étape d'analyse topologique des données via la persistance homologique avant la classification (Fig 37).

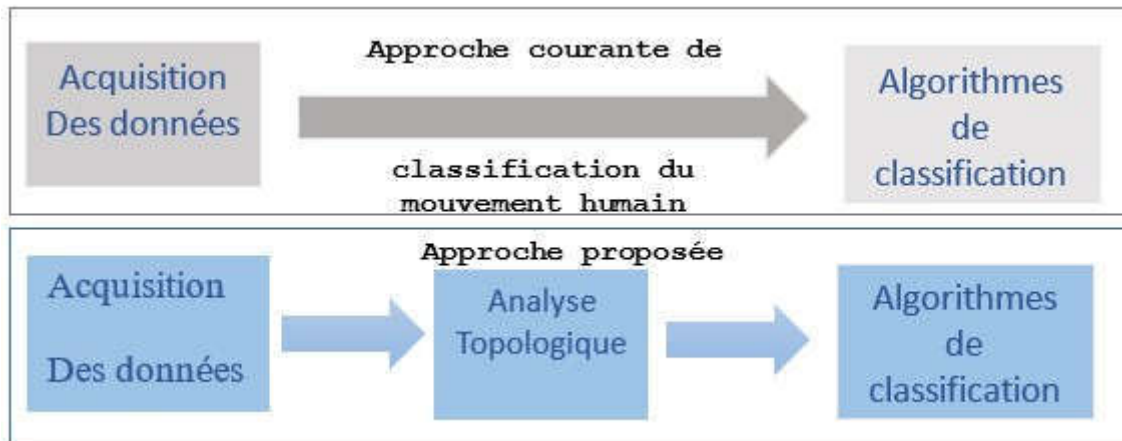


Figure 37. (a) approche classique présentée dans la littérature scientifique, (b) approche proposée qui consiste à ajouter une étape d'analyse topologique avant la classification.

Dans cette partie, nous allons détailler les étapes de notre démarche d'analyse. La Figure (Fig 38) donne une vue générale sur les étapes de l'analyse.

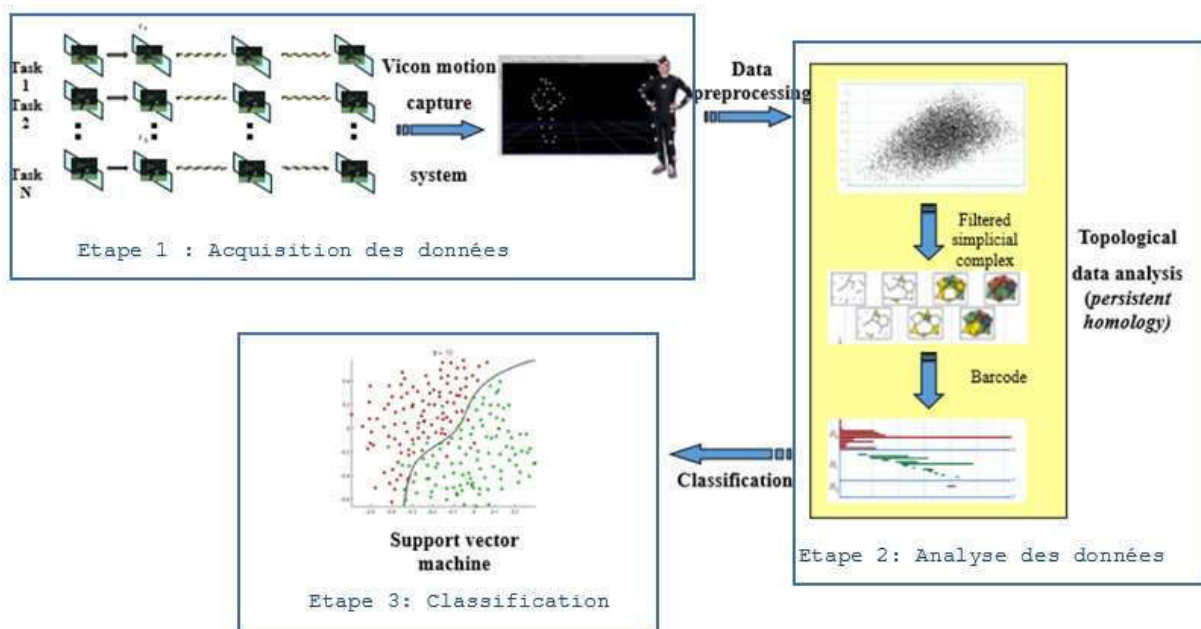


Figure 38. Les étapes de la démarche proposée.

La première étape consiste à définir l'ensemble des mouvements à étudier (ensemble des mouvements que le système doit être en mesure de reconnaître à la fin de l'analyse). L'acquisition des mouvements

## Méthodologie

est effectuée dans la seconde étape. Pour l'enregistrement des données, nous utilisons principalement un système de motion capture, avec des caméras tel que le Vicon motion capture (Fig 39). Enfin, la troisième étape consiste à classer les mouvements.



Figure 39. Le système de capture de mouvement « Vicon » basé sur les caméras infrarouges et les marqueurs.

En 1882 Etienne-Jules Marey ( Marey, 1894) a développé une technique de photographie nommée « chronophotographie » qui permet de décomposer le mouvement en poses successives (Fig 40).

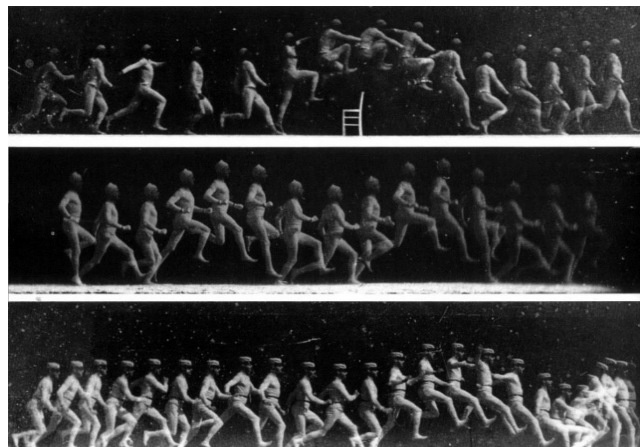


Figure 40. Chronophotographie pour la décomposition du mouvement.

Les systèmes de capture de mouvements actuels se basent sur le même principe pour décomposer les mouvements. Les coordonnées des marqueurs sont enregistrées à une fréquence  $\alpha$ , ainsi le mouvement est défini comme étant une suite  $N$  de postures  $P_{i=1\dots N}$ , (Fig 41).

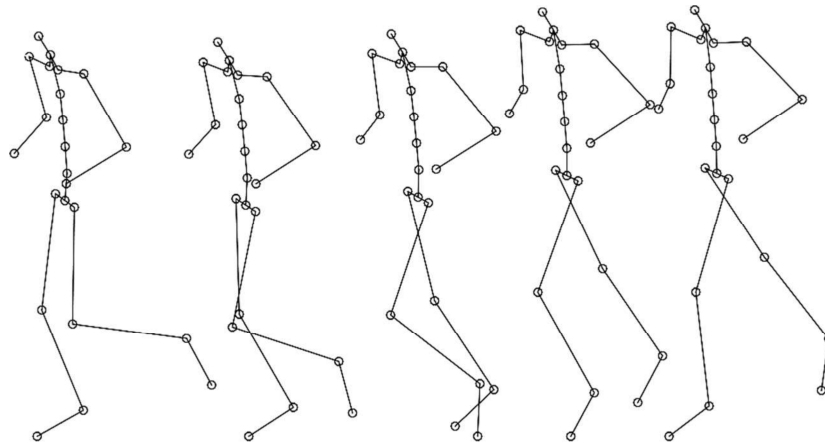


Figure 41. Le mouvement est représenté par une séquence de postures.

La deuxième étape représente le cœur de notre démarche : l'analyse topologique du mouvement. Comme indiqué dans l'étape précédente, le mouvement est enregistré sous forme d'une suite de postures, ce qui fait que le mouvement va être analysé posture par posture. Pour chaque posture du mouvement, on utilise les positions 3D des marqueurs qui forment un nuage de points, comme entrées à l'analyse topologique via la persistance homologique (détaillée dans le chapitre 2) (Fig 42), de telle façon que le mouvement soit représenté par une suite de signatures topologiques au lieu d'une suite de postures.

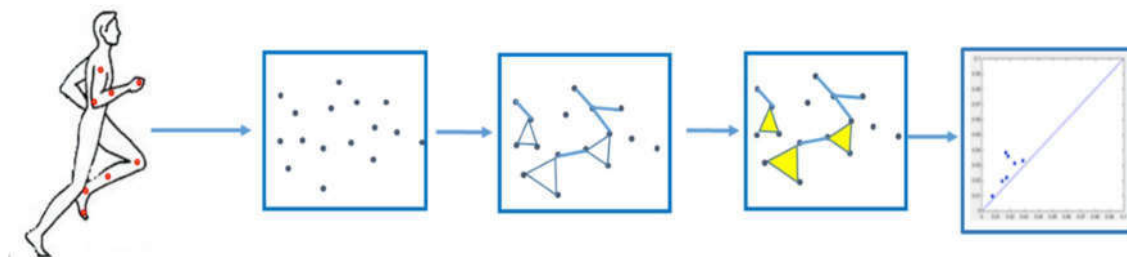


Figure 42. Analyse topologique de chacune des postures du mouvement.

Une fois que tous les mouvements qu'on veut étudier sont codifiés sous forme de suites de signatures topologiques, pour faire de la classification entre deux ou plusieurs mouvements, il suffit de construire une matrice de distances (carrée et symétrique) (Fig 44) entre les signatures topologiques de chaque mouvement en utilisant l'une des métriques choisies (distance de Wasserstein ou de Bottleneck) détaillées dans le chapitre 3. Une fois la matrice de distances calculée, on applique une méthode de réduction dimensionnelle, soit par une approche linéaire avec le positionnement multidimensionnel (L. Davison, 1991) (MDS pour Multidimensional scaling) soit par une approche non linéaire avec Isomap (B. Tenenbaum, de Silva, L. Schwartz, & C. Langford, 2002) afin de pouvoir visualiser en 2D ou 3D la distribution des signatures topologiques.

## Méthodologie

Si par exemple on veut classer deux mouvements constitués respectivement de  $N$  et  $N'$  postures, on doit calculer la matrice de distances entre toutes les signatures topologiques des postures des deux mouvements. Ensuite on applique une réduction dimensionnelle. Il suffit alors de récupérer les coordonnées des traces des signatures topologiques dans le sous-espace de dimension réduite, et de faire de la classification supervisée avec la machine à vecteurs de support ou les réseaux de neurones (Fig 43).

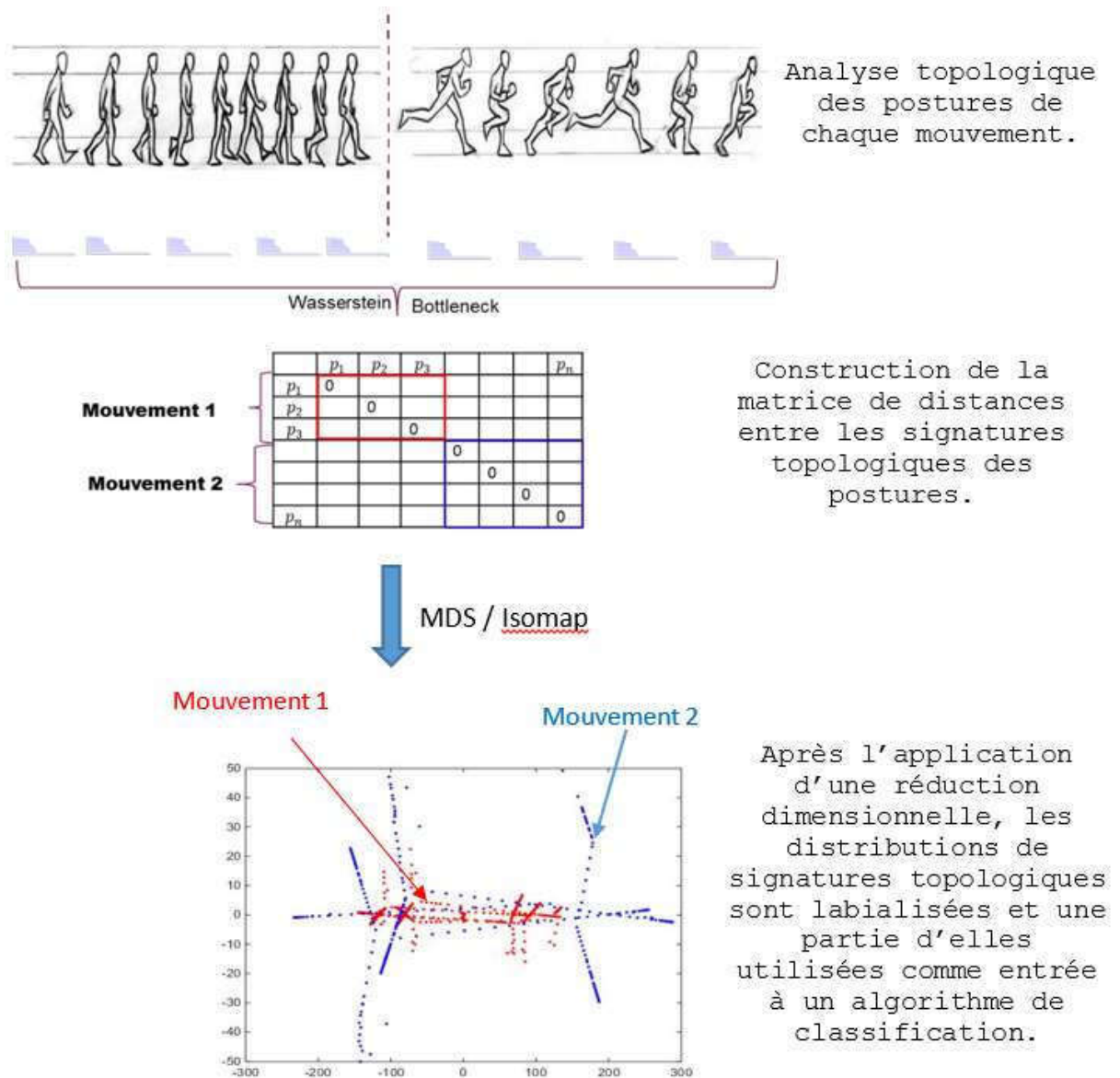


Figure 43. Application de l'approche proposée, pour classer deux mouvements.



# Chapitre V Classification de mouvements

## V.1 Introduction

Les mouvements sélectionnés sont des mouvements de pointage avec la main. Il s'agit de mouvements qui sont effectués de manière indépendante afin d'éviter la facilité de classer des mouvements cycliques. Pour la première expérience, les sujets sont priés d'exécuter un mouvement de pointage vers une surface homogène positionnée devant eux à partir de trois positions initiales différentes (Fig 44). Dans la seconde expérience, les participants exécutent un ensemble de mouvements de pointage successifs impliquant tout le corps (whole-body pointing) vers 9 cibles. Pour chaque mouvement de l'expérience, un ordre des cibles à atteindre est prédéfini (Fig 49). Les deux expériences suivent le protocole proposé par Hilt et al (Hilt P. , 2015) (Hilt, Berret, Papaxanthis, Stapley, & Pozzo, 2016). Le matériel utilisé ainsi que le protocole de chaque expérience vont être détaillés dans les sections suivantes.

Quatre participants dont l'âge est  $25 \pm 3$  ans, le poids est  $70 \pm 7$  kg et la taille de  $172 \pm 8$  cm ont été sélectionnés. Ils étaient tous droitiers et ne souffraient d'aucun trouble moteur. De plus, ils n'ont pas connaissance des objectifs et/ou des hypothèses de l'étude avant la passation.

Les mouvements sont enregistrés à l'aide de capteurs rétro-réfléchissants posés sur différents points du corps humain dont on enregistre leurs coordonnées en 3D, à l'aide d'un système de motion capture (Vicon, Oxford, UK) (Laboratoire INSERM U1093, 2011) (Figure 8. A gauche le système de capture de mouvement Vicon, à droite la Kinect, qui est le second système utilisé dans la capture de mouvement pour nos travaux.) composé de sept caméras qui enregistrent les coordonnées des capteurs avec une fréquence de 100 Hz.

## V.2 Evaluation avec un mouvement simple acquis avec Vicon

### V.2.1 Protocole expérimental

Pour la première expérience, les participants debout, avec les pieds fixés au sol, devaient toucher avec leurs deux index une surface qui se trouve en face d'eux. La surface est constituée d'un drap uniforme tendu sur un cadre en bois de dimensions 2,5m×1.5m (H×L). Le choix d'une surface souple (drap) a pour but de s'assurer que les participants ne puissent s'appuyer dessus.

Pour déterminer la distance optimale entre le participant et la surface, plusieurs pré-tests ont été réalisés. La distance optimale sélectionnée entre l'épaule du participant au repos et la surface cible est équivalente à 130% de la longueur du bras du participant. De plus, un angle d'inclinaison de 15° entre la verticale et la surface est imposé (Fig 44).

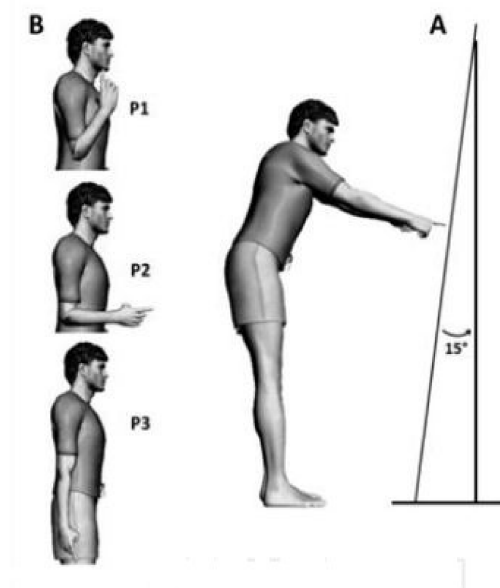


Figure 44. Illustration du paradigme expérimental de l'étude 2. A. Représentation du positionnement de la surface par rapport aux sujets (distance épaule-surface : 130% de la longueur du bras). B. Les trois positions initiales, P<sub>1</sub>, P<sub>2</sub> et P<sub>3</sub> demandées aléatoirement aux sujets.

On note P<sub>1</sub>, P<sub>2</sub> et P<sub>3</sub> les trois positions initiales à partir desquelles les participants doivent atteindre la surface avec leur index. Le premier mouvement est effectué à partir de la position P<sub>1</sub> qui consiste à se tenir debout avec un angle 0° entre le bras et l'avant-bras (Fig

44). Pour le second mouvement, la position initiale notée  $P_2$  est obtenue en formant un angle de  $90^\circ$  entre le bras et l'avant-bras. Enfin, pour le dernier mouvement, la position initiale  $P_3$  est obtenue en maintenant le participant debout mais cette fois-ci avec le bras le long du corps ( $180^\circ$  entre le bras et l'avant-bras).

Les étapes sont les suivantes : une fois que le participant se tient droit face à la surface à atteindre en respectant une distance équivalente à 130% de la longueur du bras du sujet définie ci-dessus, on lui demande de toucher la surface avec ses deux index le plus spontanément possible, la seule instruction donnée étant de garder les pieds fixés au sol.

Chaque participant répète alors 12 fois chacune des 3 tâches à partir des positions initiales successives  $P_1$ ,  $P_2$  puis  $P_3$ , ce qui représente 36 mouvements de pointage par participant.

Etant donné que le mouvement des deux bras est simultané, nous nous limitons à l'étude de l'hémicorps droit et aux déplacements réalisés dans le plan para-sagittal. L'enregistrement se fait à l'aide de 10 marqueurs positionnés au niveau de : processus acromial, condyle latéral de l'humérus, processus styloïde ulnaire, extrémité de l'index, grand trochanter, partie externe de l'articulation du genou, malléole externe, et 5e métatarse de la tête du pied. Les quatre marqueurs restants étaient positionnés sur le cantus externe de l'œil droit, le méat auditif gauche, le sommet du crâne et la vertèbre D1 (Hilt P. , 2015).

### V.2.2 Préparation des données et études comparatives

Le système de capture du mouvement a été calibré de manière à ce que les axes X, Y et Z correspondent respectivement aux axes horizontal (H), antéropostérieur (AP) et vertical (V).

Le début et la fin de chaque tâche de pointage sont désignés comme étant les instants qui correspondent respectivement aux moments où la vitesse sort de l'intervalle de confiance à 5% centrée en sa valeur maximale (Berret, Bonnetblanc, Papaxanthis, & Pozzo, 2009).

Les données cinématiques sont filtrées (filtre passe-bas, fréquence de coupure de 10Hz) par un filtre Butterworth d'ordre 5 (fonction de Matlab) (Sanjit K Mitra, 2011) (filtfilt Matlab Zero-phase digital filtering, s.d.). Ensuite, une normalisation à 200 pas est effectuée par un algorithme d'interpolation (fonction spline de Matlab) (Boor, 1978) (Matlab Inc) .

Les données enregistrées à chaque répétition des trois mouvements ( $Mouv_1$ ,  $Mouv_2$  et  $Mouv_3$ ) pour chacun des participants  $Sub_i$  sont placées dans une matrice notée  $C$  qui compte  $N$  lignes et  $S \times 3$  colonnes, où  $N$  est le nombre de postures échantillonnées du mouvement et  $S$  le nombre

## Classification de mouvements

de marqueurs. Ainsi, la ligne  $i$  de la matrice  $C$  correspond à la concaténation des coordonnées  $(x_i^j, y_i^j, z_i^j)$  des  $S$  marqueurs  $(M_j)_{j=1,\dots,S}$  enregistrés par le système de motion capture pour la posture  $P_i$  avec  $i=1\dots N$ . On obtient une matrice (3 mouvements  $\times$  12 répétitions  $\times$  4 participants).

On parcourt la matrice  $C$  ligne par ligne en appliquant une filtration (voir section filtration) sur le nuage de points défini à partir des données de chacune d'entre elles. Un mouvement sera donc caractérisé par une suite de  $N$  codes-barres ou diagrammes de persistance à un ordre donné (Etape 3 section démarche) (Fig 45).

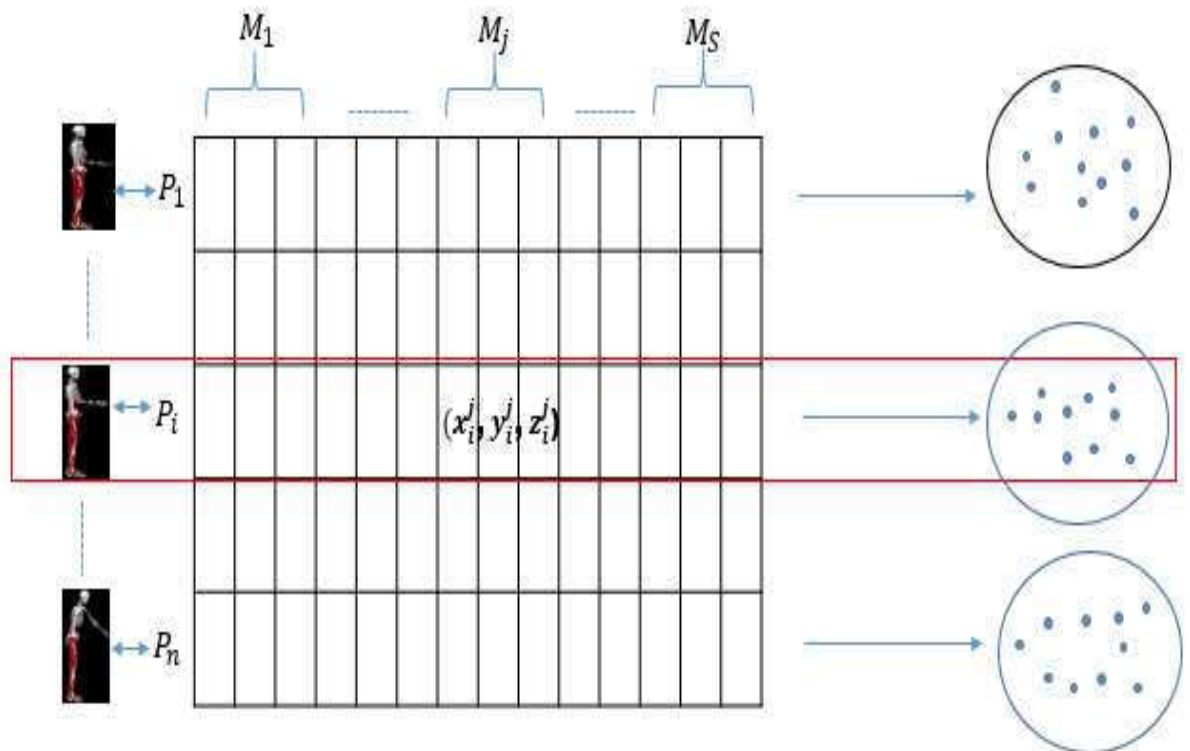


Figure 45. Matrice  $C$  dans laquelle sont enregistrées les coordonnées de marqueurs de chaque posture du mouvement.

A la fin de l'analyse topologique, chacune des répétitions des trois mouvements ( $Mouv_1$ ,  $Mouv_2$  et  $Mouv_3$ ) sera codifiée sous forme d'une suite de signatures topologiques (codes-barres) (voir chapitre Persistance homologique).

Comme mentionné dans les sections précédentes, le but de notre travail est de présenter une approche qui permet d'améliorer la robustesse de l'approche classique (qui consistait à injecter directement les données cinématiques dans un classifieur) en ajoutant une étape d'analyse topologique via la persistance homologique afin de pallier

## Classification de mouvements

aux problèmes de bruit blanc et l'absence éventuelle de distinction de différences morphologiques entre les sujets (participants, patient).

Pour confirmer la robustesse de la démarche proposée, on procède à une comparaison des précisions des résultats obtenus après classification des différents mouvements par chacune des deux approches, dans les configurations suivantes :

### A. Classification inter-mouvements avec :

A.1 Inter-participants (différents participants, exécutent des mouvements différents).

A.2 Intra-participants (même participant qui exécute des mouvements différents).

### B. Classification intra-mouvements avec :

B.1 Inter-participants (différents participants qui exécutent les mêmes mouvements).

B.2 Intra-participants (le même participant exécute les mêmes mouvements).

Comme noté ci-dessus, Mouv1, Mouv2 et Mouv3 désignent les mouvements associés respectivement aux trois positions de départ suivantes :  $P_1$ ,  $P_2$  et  $P_3$ . Nous ne présenterons que les résultats obtenus après comparaison entre les données issues des mouvements effectués par trois participants tirés aléatoirement dans l'ensemble des participants (l'ensemble des résultats qui combinent les données des trois mouvements effectués par tous les participants sont détaillés dans l'annexe numéro 1). Pour chaque participant sélectionné, chacun des trois mouvements de pointage est répété 12 fois.

Dans un premier temps, on va utiliser un algorithme de classification binaire (machine à vecteurs de support). Les paramètres de la méthode (type et valeurs de la fonction noyau, valeur de la variable ressort) ont été déterminés en utilisant la K-validation (The support vector machine under test, 2003). Pour chacune des deux approches, 80% des données ont été utilisées dans l'apprentissage et les 20% restantes pour la phase des tests qui permet de calculer la précision.

L'algorithme de classification binaire utilisé (SVM) retourne un pourcentage de précision qui décrit comment les données sont classifiées, que ce soit directement les données cinématiques pour l'approche qui utilise la classification directe ou les signatures topologiques pour l'approche basée sur l'analyse topologique. Le tableau ci-dessous (Tab 3) détaille l'interprétation des différents résultats qui sont attendus.

Précision	Interprétation
$\cong 100\%$	L'approche considère les données cinématiques ou les signatures topologique comme étant des données appartenant à deux mouvements complètement différents.
$\cong 50\%$	L'approche considère les données cinématiques ou les signatures topologique comme étant des données appartenant au même mouvement.
$\cong 0\%$	L'approche n'est pas en mesure de classier les données des deux mouvements.

Tableau 3. Interprétation des différents résultats attendus.

Comme mentionné dans les sections précédentes, le but de notre travail est de proposer une approche qui améliore la robustesse de la classification de données volumineuses issues de l'enregistrement des mouvements humains.

En première partie, on analyse les résultats de classification binaire, que ce soit inter-mouvements et inter-participants. On s'attend à avoir un résultat proche des 100%, indiquant la capacité des deux approches à distinguer les deux mouvements à classier.

En seconde partie, on compare les résultats retournés lors de la classification des données de deux mouvements différents exécutés par le même participant (inter-mouvements et intra-participants). On s'attend à avoir également une précision proche des 100%.

Pour la troisième et la quatrième partie, on effectue des classifications binaires (intra-mouvements et inter participants) et (intra-mouvements et intra-participants), comme indiqué dans le tableau 4. Les deux approches doivent être en mesure de reconnaître que les données appartient au même mouvement, en retournant un résultat proche des 50% qui représente le « chance level » pour une classification binaire (Combrisson & Jerbi, 2015). Cela démontre que les deux approches ne font pas de différences entre les données appartenant au même mouvement, qu'il soit exécuté par le même ou différents participants.

Une fois qu'on compare les résultats de classification retournés par les deux approches, on effectue une comparaison du temps d'exécution de chacune des deux approches.

### V.2.3 Résultats

Dans cette section on va présenter et analyser les résultats obtenus, comme décrit dans la partie précédente. Pour des raisons de présentation, les résultats de classification binaire d'un ensemble de données de trois participants est résumé dans le tableau suivant. Les résultats de classification binaire de l'intégralité des combinaisons possibles entre les mouvements effectués par l'ensemble des participants sont détaillés dans l'annexe A.

Le tableau des résultats sera divisé en quatre sous-parties.

## Classification de mouvements

Dans la première colonne du tableau, on affiche le/les mouvements à classer, ainsi que le/les participant(s) qui les exécutent :

Par exemple : Subj 1 (Task1) / Subj 2 (Task 3) : une classification binaire, entre le mouvement 1 exécuté par le premier sujet et le mouvement 3 exécuté par le participant numéro 2.

Dans les colonnes qui suivent, pour chaque approche, on indique le résultat retourné, ainsi qu'une colonne qui indique si le résultat est conforme aux attentes des hypothèses.

	Mouvements classifiés	Classification directe des données	satisfaction par rapport au résultat attendu	Classification après analyse topologique	satisfaction par rapport au résultat attendu
<b>Inter-Mouvements Inter-Subjects</b>	Subj 1 (Task1) / Subj 2 (Task 2)	100%	😊	90,53%	😊
	Subj 1 (Task 2) / Subj 2 (Task 1)	100%	😊	96,25%	😊
	Subj 1 (Task 1) / Subj 3 (Task 2)	100%	😊	91,72 %	😊
	Subj 2 (Task 2) / Subj 3 (Task 1)	100%	😊	93,04%	😊
<b>Inter-Mouvements Intra-Subjects</b>	Subj 1 (Task 1) / Subj 1 (Task 2)	84,82%	😊	86,02%	😊
	Subj 1 (Task 1) / Subj 1 (Task 3)	90,25%	😊	79,04%	😊
	Subj 2 (Task 1) / Subj 2 (Task 2)	81,02%	😊	83,56%	😊
	Subj 3 (Task 2) / Subj 3 (Task 3)	87,51%	😊	90.17%	😊
<b>Intra-Mouvements Inter-Subjects</b>	Subj 1 (Task 1) / Subj 2 (Task 1)	100%	😞	48,52%	😊
	Subj 1 (Task 2) / Subj 2 (Task 2)	100%	😞	52,65%	😊
	Subj 2 (Task 3) / Subj 3 (Task 3)	100%	😞	50%	😊
	Subj 1 (Task 1) / Subj 3 (Task 1)	100%	😞	51,72%	😊
<b>Intra-Mouvements Intra-Subjects</b>	Subj 1 (Task 1) / Subj 2 (Task 1)	52,4%	😊	50%	😊
	Subj 2 (Task 1) / Subj 2 (Task 1)	51,3%	😊	53,22%	😊
	Subj 3 (Task 2) / Subj 3 (Task 2)	52,78%	😊	47,95%	😊
	Subj 3 (Task 3) / Subj 3 (Task 3)	54,27%	😊	50,61%	😊

Tableau 4. Résultats de comparaison de la classification binaire.

## Classification de mouvements

En analysant le tableau des résultats ci-dessus (Tab 4), on peut remarquer que pour les deux premières parties (classification inter-mouvements), les deux approches peuvent distinguer correctement les postures des mouvements avec une grande précision. On note qu'on obtient un résultat de 100% avec l'approche directe quand deux participants exécutent deux mouvements différents (inter mouvements et inter participants). Cela s'explique par la sensibilité élevée de l'approche directe en ce qui concerne la différence morphologique entre participants (Fig 46).

Pour les deux dernières parties du tableau (intra-mouvements), dans le cas où les mouvements sont répétés par le même participant (intra-participant), les résultats retournés qui s'approchent de 50% pour les deux approches, indiquent que les deux approches sont en mesure de reconnaître que les postures appartiennent au même mouvement. Mais dans le cas où le mouvement est exécuté par deux participants différents (intra-mouvement et inter-participants), contrairement à l'approche proposée qui consiste à utiliser les signatures topologiques des postures comme entrée à l'algorithme de classification, l'approche directe n'est pas en mesure de reconnaître que les participants effectuent le même mouvement. Le résultat de 100% retourné par l'approche indique que l'approche considère les données des postures comme appartenant à des mouvements différents, à cause de la sensibilité de l'approche directe aux différences morphologiques entre participants. La figure (Fig 46) montre les trajectoires des doigts (index), de deux participants qui exécutent le même mouvement. On peut remarquer l'impact de la différence morphologique entre les participant, sur le résultat de la classification directe des données cinématiques.

L'approche proposée montre une robustesse face à cette différence morphologique. De plus, elle reste en mesure de reconnaître les postures qui composent chacun des mouvements Mouv1, Mouv2 et Mouv3, même s'ils sont effectués par différentes personnes. La figure (Fig 46) montre la distribution des signatures topologiques des postures du même mouvement, après réduction dimensionnelle via l'isomap (voir section Réduction dimensionnelle), exécuté par deux participants différents. On observe que les deux distributions se chevauchent, ce qui conduit à obtenir un résultat proche de 50% avec un classifieur binaire.



## Classification de mouvements

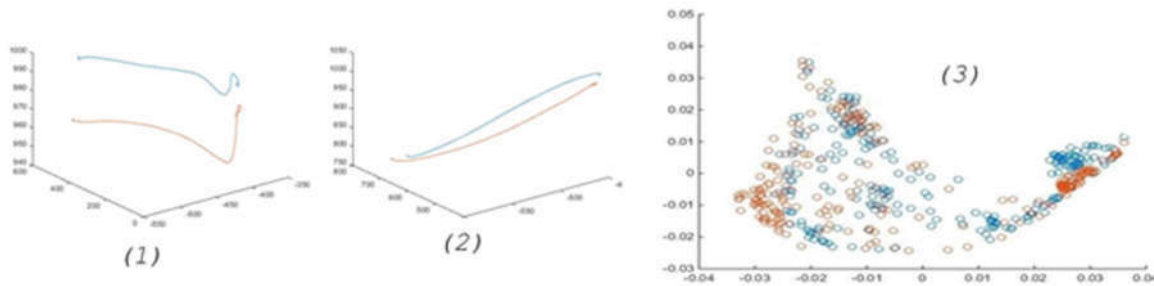


Figure 46. La trajectoire de l'index du même participant répétant le même mouvement séquentiel.

Le second point de comparaison entre les deux approches est le temps d'exécution. Ci-dessous un histogramme résumant le temps d'exécution de chacune des deux approches.

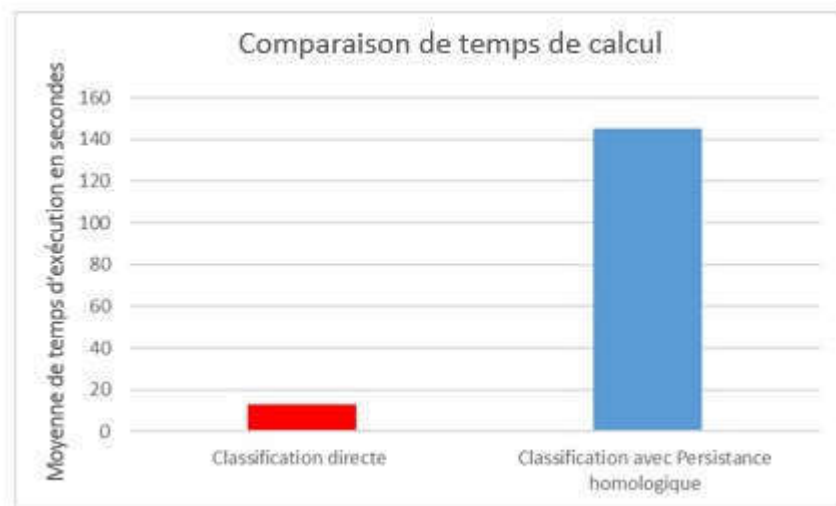


Figure 47. Comparaison entre le temps d'exécution pour les deux approches, pour une classification binaire.

D'après l'histogramme (Fig 47), on peut remarquer que l'ajout d'une étape d'analyse topologique avant la classification des données conduit à une augmentation conséquente du temps de calcul qui passe de 16 secondes à 142 secondes.

Cette grande différence en temps de calcul est due aux différentes étapes qu'implique l'approche proposée avant de procéder à la classification (Fig 38) :

- Déterminer la signature topologique (diagramme de persistance), via la filtration de Rips, de chaque posture à partir des données cinématiques de chaque mouvement (voir section III.1.3)

## Classification de mouvements

- Calculer la distance entre les diagrammes de persistance obtenus (Fig 25) en utilisant une des métriques définies dans la section III.1.8.3).
- Appliquer une réduction dimensionnelle aux données topologiques obtenues.
- Enfin, labéliser les coordonnées issues de la réduction dimensionnelle, comme entrées à l'algorithme de classification.

Afin de mieux expliquer la cause de cette différence de temps de calcul, dans le tableau ci-dessous (Tab 5), on expose la complexité de calcul de chacune de ces étapes:

Etape	Complexité de calcul
Filtration de Rips (construction de la signature topologique).	$o(n^2)$ (Afra Zomorodian, 2010)
Calculer la distance entre deux signatures topologique (persistance homologique).	$o\left(\sum_{i=1}^k \dot{n}_i \times n_M - \left(\sum_{i=1}^k \dot{n}_i \times n_M - 2\right) \cdot n^4\right)$
Appliquer une réduction dimensionnelle	$o(n^3)$ (Ming-Hsuan Yang, 2002)
Algorithme de classification (Machine à vecteur de supports)	$o(n^2)$ (Anderson Rocha & Siome Klein Goldenstein, 2014)

Tableau 5. La complexité de calcul de chaque étape de l'approche proposée.

Le tableau (Tab 5), détaillant la complexité de calcul de chaque étape, montre que les différentes étapes qui composent l'approche proposée, ont une complexité exponentielle, ce qui explique l'importance augmentation en temps de calcul. On peut remarquer aussi, que l'étape le plus couteuse en temps de calcul est celle du calcul de distance entre les signatures topologique (diagramme de persistance) des postures de mouvements.

Dans la deuxième partie de cette expérience, on va réaliser une comparaison entre les résultats retournés par les deux approches dans le cas d'une classification multi-classes. Pour une classification inter ou intra participants, le but est de tester la capacité des approches à déterminer à quel mouvement appartient un sous-ensemble de postures parmi un ensemble qui regroupe des postures de plusieurs mouvements.

La même version de la machine à vecteurs de support (un contre tous) est utilisée pour les deux approches. Pour l'approche proposée, on effectue une analyse topologique via la persistance topologique avant d'effectuer une classification multi-classes. Pour l'approche contrôle, on utilise les données de la capture de mouvement directement comme entrée à l'algorithme de classification. On essaie de confirmer

## Classification de mouvements

la capacité de l'approche à bien classifier les mouvements humains et de reconnaître la signature d'un mouvement, en effectuant des tests de classification multi-classes.

Contrairement à la première analyse, où 80% des données ont été utilisées pour la phase d'apprentissage et les 20% restant pour la partie test, les données des mouvements avec lesquelles on a calculé la précision des approches n'ont pas été utilisées dans la partie d'apprentissage.

Les données issues de la capture du mouvement (les coordonnées tridimensionnelles des marqueurs) des mouvements cités dans la première partie de la première colonne du tableau (Tab 6), ont été utilisées comme entrée pour la machine à vecteurs de supports afin de faire l'apprentissage de la première approche. Pour l'approche proposée, les données ont été analysées en utilisant la persistance homologique, une fois que les données ont été transformées en signatures topologiques, les coordonnées issues de leur réduction dimensionnelle (voir Fig 43) sont utilisés comme entrée à l'algorithme de classification (machine à vecteur de support) pour la partie d'apprentissage.

Les mêmes étapes de préparation de données ont été appliquées sur les données (partie gauche de la première colonnes) des mouvements sélectionnés pour la partie de test (calcul de la précision).

Cette étape d'analyse de classification multi-classes se divise en deux sous-parties : une classification inter et intra participants. On n'a pas effectué d'analyse enter et intra mouvements étant donné que la classification est multi classes (l'ensemble d'apprentissage contient plusieurs mouvements parmi lesquels les mêmes mouvements que celui utilisé pour le test). Le but est de mesurer la précision avec laquelle les deux approches reconnaissent un mouvement parmi un ensemble de mouvements effectués par la même ou par différentes personnes.

Pour les deux approches. Les résultats obtenus sont détaillés dans la tableau (Tab 6) ci-dessous. Dans la première colonne du tableau, on affiche le/les mouvements à classifier, ainsi que le/les participant(s) qui les exécutent, par exemple :

- Subj 1 (Task1, Task2, Task3) / Subj 1 (Task 2) : une classification multi-classes, intra-participants, où l'ensemble d'apprentissage est composé, des mouvement 1,2 et 3 exécutés par le participant numéro 1, et pour le test on essaye de classifier le mouvements numéro 2 exécuté par le même participant numéro 1.
- (Subj 1, Subj 3) (Task1, Task3) / Subj 2 (Task 1): une classification multi-classes et inter-participants, dont le groupe d'apprentissage se compose des données des trois mouvements (Mouv1,Mouv2,Mouv3) exécutés par deux participants différents ( Subj1 et Subj3), et dans l'ensemble de test qu'on

## Classification de mouvements

veut classifier a les données du mouvement numéro 1 effectué par le participant 2.

Dans les colonnes qui suivent, pour chaque approche, on indique le résultat retourné par chaque approche, ainsi qu'une colonne qui indique si le résultat est conforme aux attentes des hypothèses.

	Mouvements classifiés	Classification directe des données	satisfaction par rapport au résultat attendu	Classification après analyse topologique	satisfaction par rapport au résultat attendu
Intra-Participants	Subj 1 (Task1, Task2, Task3) / Subj 1 (Task 1)	91.14%	😊	92.45%	😊
	Subj 1 (Task1, Task2, Task3) / Subj 1 (Task 2)	89.38%	😊	90.65%	😊
	Subj 1 (Task1, Task2, Task3) / Subj 1 (Task 3)	92.02%	😊	85.22%	😊
	Subj 2 (Task1, Task2, Task3) / Subj 2 (Task 1)	91.56%	😊	93.67%	😊
	Subj 2 (Task1, Task2, Task3) / Subj 2 (Task 2)	85.73%	😊	91.05%	😊
	Subj 2 (Task1, Task2, Task3) / Subj 2 (Task 3)	90.61%	😊	89.72%	😊
	Subj 3 (Task1, Task2, Task3) / Subj 3 (Task 1)	84.05%	😊	90.57%	😊
	Subj 3 (Task1, Task2, Task3) / Subj 3 (Task 2)	92.56%	😊	91.45	😊
	Subj 3 (Task1, Task2, Task3) / Subj 3 (Task 3)	86.71%	😊	93.52%	😊
Inter-Participants	(Subj 1, Subj 2, Subj 3 ) (Task1, Task2, Task3) / Subj 1 (Task 1)	82.45%	😊	87.19%	😊
	(Subj 1, Subj 2, Subj 3 ) (Task1, Task2, Task3) / Subj 1 (Task 2)	71.54%	😊	86.47%	😊
	(Subj 1, Subj 2, Subj 3 ) (Task1, Task2, Task3) / Subj 1 (Task 3)	80.49%	😊	94.98%	😊
	(Subj 1, Subj 2, Subj 3 ) (Task1, Task2, Task3) / Subj 2 (Task 1)	79.25%	😊	95.68%	😊
	(Subj 1, Subj 2, Subj 3 ) (Task1, Task2, Task3) / Subj 2 (Task 2)	84.17%	😊	92.56%	😊
	(Subj 1, Subj 2, Subj 3 ) (Task1, Task2, Task3) / Subj 2 (Task 3)	83.01%	😊	94.57%	😊
	(Subj 1, Subj 2, Subj 3 ) (Task1, Task2, Task3) / Subj 3 (Task 1)	74.58%	😊	87.45%	😊
	(Subj 1, Subj 2, Subj 3 ) (Task1, Task2, Task3) / Subj 3 (Task 2)	81.45%	😊	94.75%	😊
	(Subj 1, Subj 2, Subj 3 ) (Task1, Task2, Task3) / Subj 3 (Task 3)	83.84%	😊	91.49%	😊

Tableau 6. Les résultats de comparaison de la classification multi classes, de la première expérience.

En analysant le tableau des résultats (tab 6), on peut remarquer que pour la classification intra-participant (classification d'une répétition d'un mouvement parmi les trois mouvements effectués par le même participant), la précision retournée par les deux approches pour

## Classification de mouvements

les différentes combinaisons, est très élevée en se situant entre 84% et 93%. Pour la classification inter-participant (classifier un mouvement parmi un ensemble de mouvements effectuées par d'autres participants), les résultats confirment la robustesse de l'approche proposée (ajout d'une analyse topologique) par rapport à la classification directe des données des mouvements. On peut voir une différence de précision qui s'approche de  $\mp 12\%$  entre les deux approches.

De la même manière, que l'analyse de la classification binaire, on compare le temps d'exécution des deux approches lors d'une classification multi-classes. Ci-dessous un histogramme (Fig 48) qui montre le temps de calcul enregistré pour chaque approche. Le temps de calcul de l'approche avec analyse topologique reste toujours plus conséquent que le temps d'exécution de l'approche de classification directe des données avec l'algorithme de la machine à vecteurs de support.

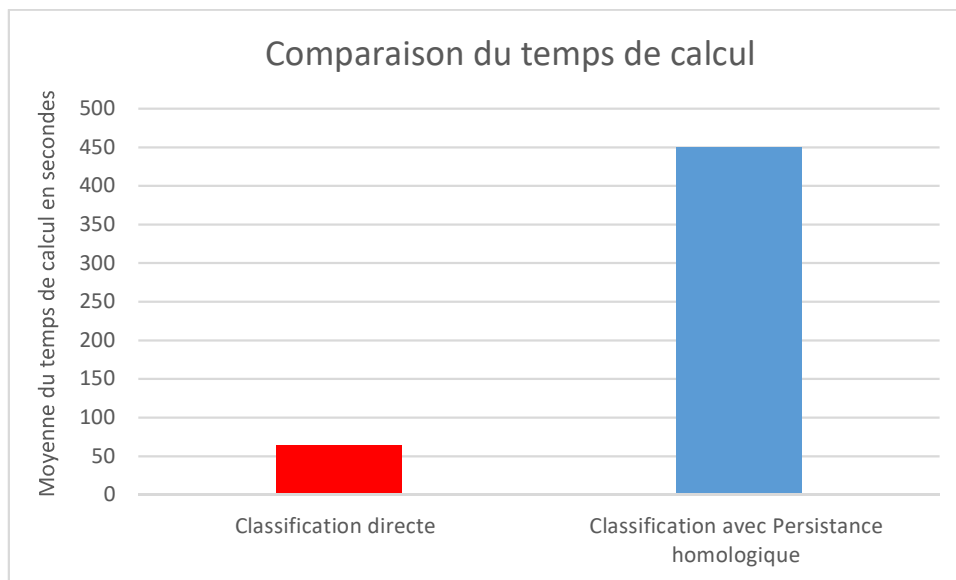


Figure 48. Comparaison de temps de calcul en secondes entre les deux approches, pour une classification multi-classes.

### **V.3 Evaluation avec un mouvement complexe acquis avec Vicon**

Le but de cette étape est de tester et de comparer les capacités des approches, lors de l'analyse d'un mouvement plus complexe (un mouvement de ciblage séquentiel vers neuf cibles, au lieu d'une seule cible dans la première expérience) et avec un nombre de données plus important (enregistrement des coordonnées de 20 marqueurs au lieu de 3 marqueurs).

Quatre participants volontaires étaient âgés entre  $25 \pm 3$  ans, avec une taille  $175 \pm 8$  cm et un poids  $75 \pm 7$  kg. Tous les participants étaient droitiers et ne souffraient d'aucun trouble moteur. Et comme pour le premier protocole, les sujets n'étaient pas en connaissance des objectifs et/ou des hypothèses de l'étude avant la passation.

#### **V.3.1 Protocole expérimental**

Lors de la première expérience, le mouvement de pointage était vers une seule cible et n'impliquait que la partie supérieure du corps. Dans cette seconde expérience, les participants devaient atteindre plusieurs cibles avec un mouvement impliquant tout leur corps (whole-body-reaching).

Le dispositif se composait de trois fils situés à la droite, en face et à la gauche du participant à une distance respective de 55%, 50% et 45% de la taille du participant. Sur chaque fil on avait attaché trois cibles situées à trois hauteurs différentes (20%, 60% et 90% de la taille du participant) (Fig 49).

Comme dans la première expérience, le protocole est détaillé dans les travaux de Hilt et al (Hilt P. , 2015) (Hilt, Berret, Papaxanthis, Stapley, & Pozzo, 2016). Le matériel utilisé ainsi que le protocole de chaque expérience vont être détaillés dans les sections suivantes.

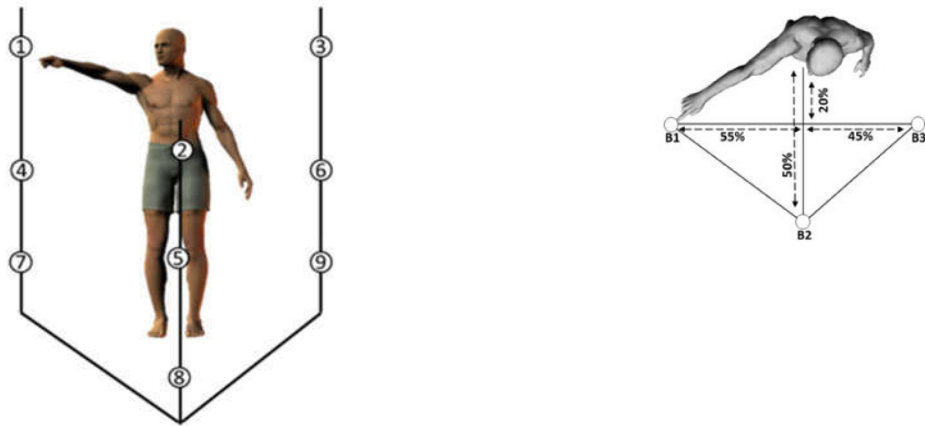


Figure 49. Illustration du protocole expérimental mis en place pour l'étude des synergies musculaires. Chaque cible est représentée par un chiffre allant de 1 à 9, et positionnée sur trois fils (B1, B2 et B3). Les distances aux sujets, et hauteurs des cibles sont indiquées en % de la taille du sujet.

Au signal sonore de départ, le participant réalise un mouvement de pointage successif d'une cible à une autre, à partir d'une cible initiale présélectionnée vers les autres huit cibles restantes dans un ordre croisant. Les mouvements de pointage réalisés sont notés comme suit :

- Task1 :  $cible_1 \rightarrow cible_2 \rightarrow cible_3 \rightarrow cible_4 \rightarrow cible_5 \rightarrow cible_6 \rightarrow cible_7 \rightarrow cible_8 \rightarrow cible_9$
- Task2 :  $cible_2 \rightarrow cible_3 \rightarrow cible_4 \rightarrow cible_5 \rightarrow cible_6 \rightarrow cible_7 \rightarrow cible_8 \rightarrow cible_9 \rightarrow cible_1$
- Task3 :  $cible_3 \rightarrow cible_4 \rightarrow cible_5 \rightarrow cible_6 \rightarrow cible_7 \rightarrow cible_8 \rightarrow cible_9 \rightarrow cible_2 \rightarrow cible_1$
- Task4 :  $cible_4 \rightarrow cible_5 \rightarrow cible_6 \rightarrow cible_7 \rightarrow cible_8 \rightarrow cible_9 \rightarrow cible_1 \rightarrow cible_2 \rightarrow cible_3$
- Task5 :  $cible_5 \rightarrow cible_6 \rightarrow cible_7 \rightarrow cible_8 \rightarrow cible_9 \rightarrow cible_1 \rightarrow cible_2 \rightarrow cible_3 \rightarrow cible_4$
- Task6 :  $cible_6 \rightarrow cible_7 \rightarrow cible_8 \rightarrow cible_9 \rightarrow cible_1 \rightarrow cible_2 \rightarrow cible_3 \rightarrow cible_4 \rightarrow cible_5$
- Task7 :  $cible_7 \rightarrow cible_8 \rightarrow cible_9 \rightarrow cible_1 \rightarrow cible_2 \rightarrow cible_3 \rightarrow cible_4 \rightarrow cible_5 \rightarrow cible_6$
- Task8 :  $cible_8 \rightarrow cible_9 \rightarrow cible_1 \rightarrow cible_2 \rightarrow cible_3 \rightarrow cible_4 \rightarrow cible_5 \rightarrow cible_6 \rightarrow cible_7$
- Task9 :  $cible_9 \rightarrow cible_1 \rightarrow cible_2 \rightarrow cible_3 \rightarrow cible_4 \rightarrow cible_5 \rightarrow cible_6 \rightarrow cible_7 \rightarrow cible_8$

Les participants réalisent 9 tâches de pointages (9 séquences de pointages) différentes. Chaque tâche est répétée 15 fois, ce qui fait que chaque participant réalise 144 mouvements de pointage.

Vu le nombre important de tâches à réaliser pour chaque participant et pour éviter l'impact de la fatigue, l'expérience est réalisée sur deux sessions de 2 heures chacune. Les deux sessions étaient réalisées avec un intervalle de 24h entre elles.

### V.3.2 Préparation des données

Les étapes de préparation de données sont identiques à celles de la première expérience ; la seule différence est le nombre de marqueurs utilisés. Pour cette seconde expérience, 20 marqueurs vont être utilisés, 16 marqueurs sont positionnés symétriquement sur les deux hémicorps gauche et droit, au niveau des points anatomiques suivants : processus acromial, condyle latéral de l'humérus, processus styloïde ulnaire, extrémité de l'index, grand trochanter, partie externe de l'articulation du genou, malléole externe, et 5e métatarse de la tête du pied. Les quatre marqueurs restants étaient positionnés sur le cantus externe de l'œil droit, le méat auditif gauche, le sommet du crâne et la vertèbre D1.

Un filtre passe-bas avec une fréquence de coupure à 10Hz (filtre passe-bas, fréquence de coupure de 10Hz) par un filtre Butterworth d'ordre 5 (fonction de Matlab) (Sanjit K Mitra, 2011) (filtfilt Matlab Zero-phase digital filtering, s.d.) est appliqué sur les données enregistrées avec le système de motion capture. Une normalisation est ensuite effectuée par un algorithme d'interpolation (fonction spline de Matlab) (Boor, 1978) (Matlab Inc) .

Comme indiqué dans la partie préparation des données de la première expérience, on calcule la vitesse et l'accélération des marqueurs en utilisant les dérivées des positions des marqueurs.

En se basant sur (Berret, Bonnetblanc, Papaxanthis, & Pozzo, 2009), les instants de début et de fin ( $T_{debut}$  ,  $T_{fin}$ ) de chaque tâche sont définis respectivement comme étant les instants où la vitesse du marqueur posé sur l'index droit dépassait ou revenait à 5% de sa vitesse maximale.

Dans un premier temps, les mêmes étapes de classification que la première expérience, ont été réalisées :

- A. Classification inter-mouvements avec :
  - A.1 Inter-participants
  - A.2 Intra-participants
- B. Classification intra-mouvements avec :
  - B.1 Inter-participants
  - B.2 Intra-participants

Les résultats attendus sont similaires à ceux de la première expérience, on attend à avoir un résultat proche des 50% pour une classification intra-mouvement, et un pourcentage proche des 100% pour la classification inter-mouvements.



### V.3.3 Résultats obtenus

Pour faciliter la représentation, le tableau suivant résume les résultats des différentes étapes de classification citées ci-dessous, en combinant les classifications entre les données des quatre premières tâches (Task1, Task2, Task3 et Task4). Les résultats détaillés de l'ensemble des tâches, sont présentés dans l'annexe B.

	Mouvements classifiés	Classification directe des données	satisfaction par rapport au résultat attendu	Classification après analyse topologique	satisfaction par rapport au résultat attendu
<b>Inter-Mouvements Inter-Subjects</b>	Subj 1 (Task1) / Subj 2 (Task 2)	83.75%	😊	93.56%	😊
	Subj 2 (Task1) / Subj 3 (Task 2)	81.40%	😊	91.30%	😊
	Subj 1 (Task3) / Subj 2 (Task 2)	91.25%	😊	87.19%	😊
	Subj 2 (Task1) / Subj 3 (Task 2)	82.98%	😊	88.94%	😊
	Subj 2 (Task3) / Subj 3 (Task 4)	90.23%	😊	85.67%	😊
	Subj 3 (Task3) / Subj 4 (Task 2)	89.09%	😊	92.34%	😊
	Subj 4 (Task1) / Subj 1 (Task 2)	86.17%	😊	84.12%	😊
	Subj 4 (Task 2) / Subj 2 (Task 1)	94.32%	😊	92.49%	😊
<b>Inter-Mouvements Intra-Subjects</b>	Subj 1 (Task 1) / Subj 1 (Task 2)	72.45%	😊	90,24%	😊
	Subj 2 (Task 1) / Subj 2 (Task 2)	69.48%	😊	80,59%	😊
	Subj 1 (Task 1) / Subj 1 (Task 3)	73.78%	😊	85,67%	😊
	Subj 3 (Task 1) / Subj 3 (Task 2)	71.85%	😊	92.45%	😊
	Subj 3 (Task 2) / Subj 3 (Task 3)	67.34%	😊	86.98%	😊
	Subj 4 (Task 1) / Subj 4 (Task 3)	74.36%	😊	91.56%	😊
	Subj 2 (Task 1) / Subj 2 (Task 3)	63.02%	😊	83.57%	😊
	Subj 2 (Task 1) / Subj 2 (Task 3)	74.53%	😊	81.45%	😊
<b>Intra-Mouvements Inter-Subjects</b>	Subj 1 (Task 1) / Subj 2 (Task 1)	84.56%	😐	52,02%	😊
	Subj 1 (Task 2) / Subj 2 (Task 2)	91.03%	😐	48,27%	😊
	Subj 1 (Task 3) / Subj 2 (Task 3)	84.32%	😐	51,16%	😊
<b>Intra-Mouvements Intra-Subjects</b>	Subj 1 (Task 1) / Subj 1 (Task 1-repetition)	95%	😐	52,17%	😊
	Subj 1 (Task 2) / Subj 1 (Task 2-repetition)	89,65%	😐	50,81%	😊

Tableau 7. Résultats de comparaison de la classification binaire.

## Classification de mouvements

D'après le tableau (Tab 7), on peut constater que les résultats obtenus pour la classification inter-mouvements confirment ceux obtenus dans la première expérience. Dans le cas d'une classification inter-mouvements et inter-participants, les deux approches arrivent à classer les deux mouvements avec une grande précision entre 81% et 94%. Mais dans le cas d'une classification inter-mouvements et intra-participants, on remarque que la démarche de classification directe perd en précisions qui se situent entre 61% et 74% contre une précision entre 81% et 94% lors de la classification inter-participants. Cela s'explique par l'impact des différences morphologiques sur la classification directe des données des mouvements. Cette différence morphologique facilite la classification inter-participants. Dans le cas d'une classification intra-participant, cette différence morphologique s'annule ; ce qui induit une perte en précision pour la démarche de classification directe des données.

Pour la partie intra-mouvement, on remarque que même dans le cas d'une classification (intra-mouvement et intra-participant), l'approche de la classification directe n'est pas en mesure de reconnaître le même mouvement répété par le même participant à deux intervalles de temps différents.

Cela est dû à la nature du mouvement étudié dans cette expérience. Le pointage séquentiel implique en effet tout le corps (whole body reaching) vers neuf cibles. C'est un mouvement complexe, discontinu (constitué de plusieurs petits mouvements), ce qui fait que le participant ne l'exécute jamais de la même manière (même trajectoire de l'index, même vitesse) empêchant le classifieur (la machine à vecteur de support dans notre cas) à reconnaître la répétition si on utilise directement les données comme entrées (Fig 50).

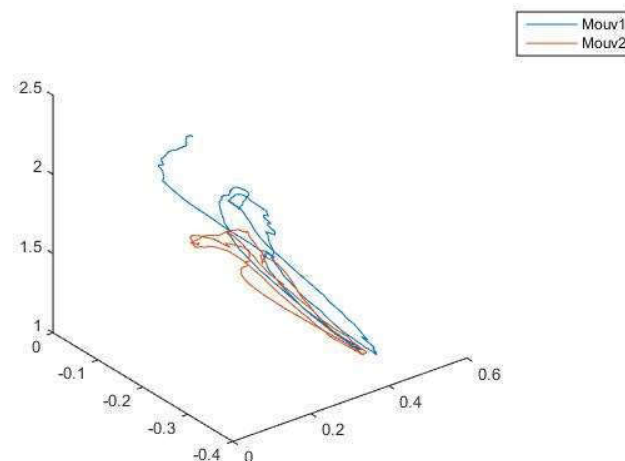


Figure 50. La trajectoire de l'index du même participant répétant le même mouvement séquentiel.

## Classification de mouvements

Avec l'augmentation du nombre des données (20 marqueurs) et la complexité du mouvement (9 cibles au lieu d'une seule), on remarque une baisse significative de la précision lors de la classification multi-classes par l'approche directe (Tab 8), contrairement à l'approche proposée. cela s'explique par la nature de la méthode d'analyse utilisée (la persistance homologique), qui est plus adaptée à l'analyse des données volumineuses (Václav Snášel, Jana Nowaková, Fatos Xhafa, & Leonard Barolli, 2017).

On observe aussi une forte corrélation entre le nombre de marqueurs (données étudiées) et le temps de calcul nécessaire pour leur analyse et la classification (Fig 51).

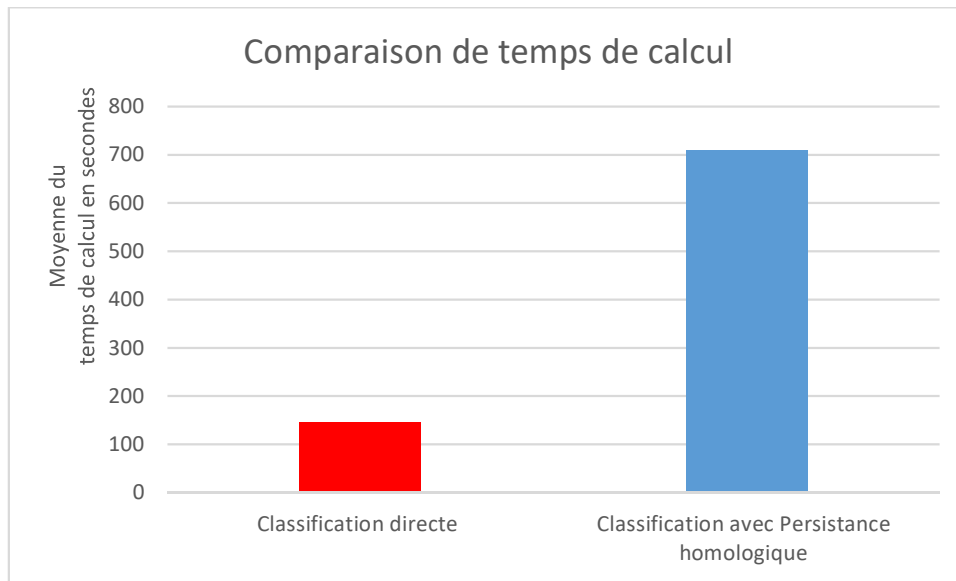


Figure 51. Comparaison du temps de calcul en secondes entre les deux approches, pour la classification multi-classes.

Classification de mouvements

	Mouvements classifiés	Classification directe des données	satisfaction par rapport au résultat attendu	Classification après analyse topologique	satisfaction par rapport au résultat attendu
Intra-Subjects	Subj 1 (Task1, Task2, Task3) / Subj 1 (Task 1)	72.15%	😊	87.45%	😊
	Subj 1 (Task1, Task2, Task3) / Subj 1 (Task 2)	64.39%	😊	92.56%	😊
	Subj 1 (Task1, Task2, Task3) / Subj 1 (Task 3)	77.44%	😊	96.06%	😊
	Subj 2 (Task1, Task2, Task3) / Subj 2 (Task 1)	83.23%	😊	89.74%	😊
	Subj 2 (Task1, Task2, Task3) / Subj 2 (Task 2)	73.56%	😊	92.85%	😊
	Subj 2 (Task1, Task2, Task3) / Subj 2 (Task 3)	62.56%	😊	83.74%	😊
	Subj 3 (Task1, Task2, Task3) / Subj 3 (Task 1)	71.89%	😊	95.63%	😊
	Subj 3 (Task1, Task2, Task3) / Subj 3 (Task 2)	58.84%	😊	93.05%	😊
	Subj 3 (Task1, Task2, Task3) / Subj 3 (Task 3)	64.98%	😊	92.75%	😊
Inter-Subjects	(Subj 1, Subj 2) (Task1, Task2, Task3) / Subj 3 (Task 1)	72.67%	😊	86.69%	😊
	(Subj 1, Subj 2) (Task1, Task2, Task3) / Subj 3 (Task 2)	64.83%	😐	91.57%	😊
	(Subj 1, Subj 2) (Task1, Task2, Task3) / Subj 3 (Task 3)	65.05%	😐	88.94%	😊
	(Subj 1, Subj 3) (Task1, Task2, Task3) / Subj 2 (Task 1)	61.46%	😐	84.71%	😊
	(Subj 1, Subj 3) (Task1, Task2, Task3) / Subj 2 (Task 2)	54.63%	😐	89.63%	😊
	(Subj 1, Subj 3) (Task1, Task2, Task3) / Subj 2 (Task 3)	49.07%	😐	94.56%	😊
	(Subj 2, Subj 3) (Task1, Task2, Task3) / Subj 1 (Task 1)	57.83%	😐	86.86%	😊
	(Subj 2, Subj 3) (Task1, Task2, Task3) / Subj 1 (Task 2)	62.85%	😐	92.26%	😊
	(Subj 2, Subj 3) (Task1, Task2, Task3) / Subj 1 (Task 3)	63.47%	😐	89.38%	😊

Tableau 8. Les résultats de comparaison de la classification multi classes, de la deuxième expérience.

### V.3.4 Test de robustesse

Dans cette section, on va tester la capacité de l'approche proposée par rapport au deuxième verrou, qui est la robustesse face au bruit blanc lors de l'enregistrement, qu'on essaie de résoudre avec l'ajout de l'étape d'analyse topologique avec la persistance homologique avant d'appliquer la classification.

Pour confirmer la robustesse de l'approche topologique face au bruit blanc lors de l'enregistrement, l'expérience consiste à ajouter des intervalles de bruit jusqu'à 100mm et de calculer pour chaque intervalle la précision retournée par l'approche classique ainsi que l'approche topologique (Fig 52).

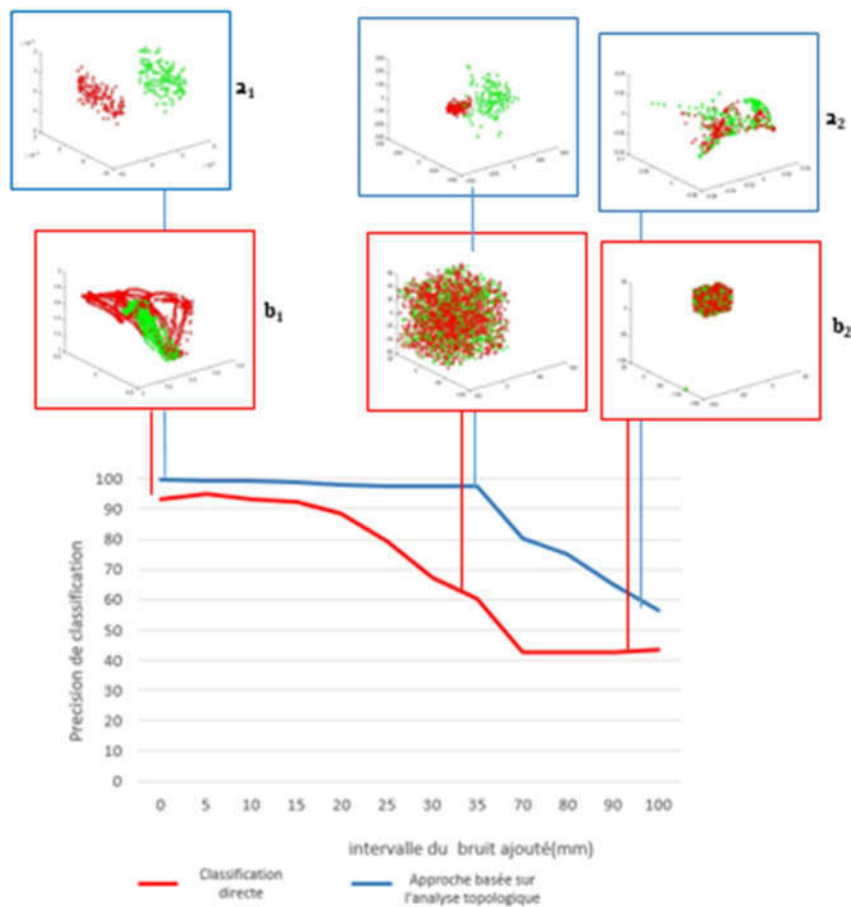


Figure 52. Test de robustesse pour les deux approches, ( $a_1$ ) signatures topologiques de deux mouvements sans bruit ( $a_2$ ) avec bruit. Représentation géométrique des données cinématiques sans bruit ( $b_1$ ) avec bruit ( $b_2$ ).

## Classification de mouvements

Quand on utilise directement le classifieur sur les données cinématiques, on peut observer une perte rapide de la précision qui chute de 95% à 40% lorsqu'on augmente l'amplitude du bruit introduit. La précision commence à chuter à partir de 20mm de bruit. Quant à l'approche basée sur la persistance homologique, la précision reste stable jusqu'à 40mm de bruit blanc ajouté puis elle chute lentement.

La figure (Fig 53) montre respectivement l'impact de l'ajout du bruit sur les données cinématiques et la signature topologique.

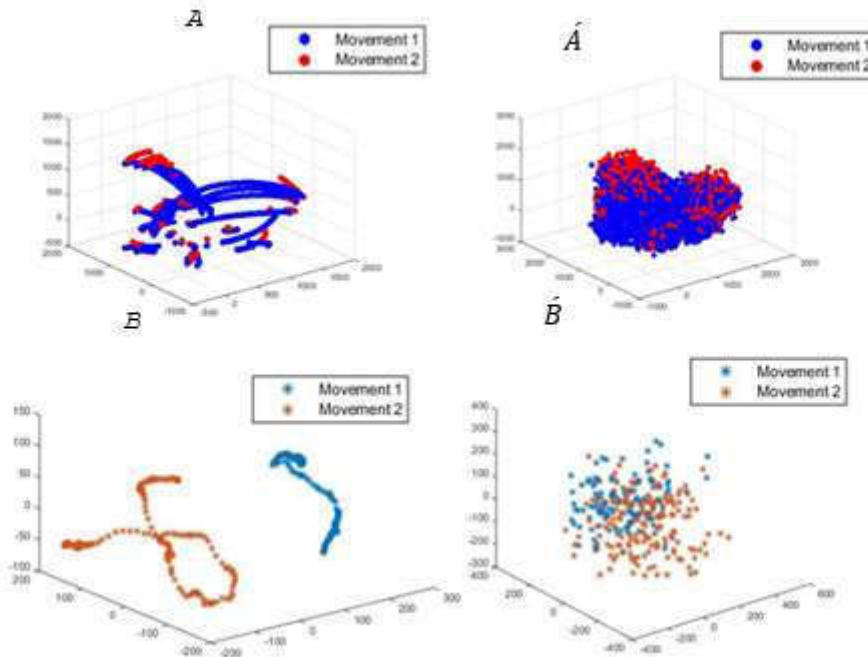


Figure 53. Les données cinématiques avant (A) et après l'ajout du bruit blanc (A'). Les distributions des signatures topologiques avant (B) et après l'ajout du bruit blanc (B').

Pour les données cinématiques avant (A) et après l'ajout du bruit (A'), on peut remarquer un chevauchement complet entre les données des deux mouvements, contrairement à la distribution des signatures topologiques (B) qui se chevauchent partiellement après l'ajout du bruit (B').

## **V.4 Evaluation avec un mouvement bruité acquis avec une Kinect**

Les deux premières expériences consistaient à analyser deux mouvements de pointage vers une ou plusieurs cibles, dont les données sont enregistrées avec un système de motion capture Vicon à 7 cameras, qui enregistrent les mouvements avec une fréquence de 100 Hz.

Dans cette troisième expérimentation, on souhaite tester la possibilité d'appliquer la démarche proposée (analyse topologique) avec des systèmes de motion capture moins onéreux et plus portatifs tel que la Kinect tout en retournant une bonne précision de classification.

Le protocole se résume à déplacer un objet de deux points initiaux différents A et B vers le même point final C selon deux trajectoires, la première étant un segment reliant les points A et C et la seconde une trajectoire semi-circulaire de B vers C (Fig 54).

Quatre participants sont priés de réaliser les mouvements. Pour les deux trajectoires définies ci-dessus, on a demandé aux participants de répéter la trajectoire semi-circulaire avec la main qui tremble afin d'ajouter du bruit au mouvement (Fig 54).

Le but est reconnaître le mouvement, même s'il n'est pas exécuté exactement de la même manière (à cause de lésion motrice, par exemple). L'enregistrement s'est fait à l'aide d'une Kinect avec une fréquence variante entre 20Hz et 30Hz (la fréquence d'échantillonnage de la Kinect n'est pas paramétrable). Les articulations enregistrées sont : l'épaule, le coude, le poignet, la main et l'index.

On note :

- Tâche 1 : le mouvement avec la trajectoire droite (Fig 54)
- Tâche 2 : le mouvement avec la trajectoire semi-circulaire chez un sujet sain
- Tâche 3 : le mouvement de la trajectoire semi-circulaire avec un sujet dont la main tremble (dans le cas où l'approche est utilisée dans le domaine médical afin de suivre l'évolution d'une thérapie de rééducation par exemple).

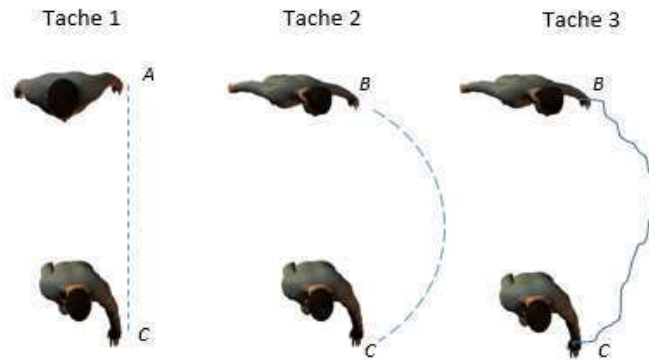


Figure 54. Protocole de la troisième expérience : le déplacement d'un objet selon différentes trajectoires, droite et semi-circulaire, (pendant la troisième tâche un tremblement de la main est introduit).

### V.4.1 Résultats

Dans le tableau ci-dessous (Tab 9) on résume les résultats obtenus des comparaisons effectuées :

- Une classification inter-mouvements dont on doit avoir un résultat qui s'approche de 100%.
- Une classification intra-mouvements à laquelle on doit avoir un résultat qui s'approche des 50% (Tab 5).

	Mouvements classifiés	Classification directe des données	satisfaction par rapport au résultat attendu	Classification après analyse topologique	satisfaction par rapport au résultat attendu
<b>Inter-Mouvements</b>	Subj 1 (Task1) / Subj 1 (Task 2)	100%	😊	97,53%	😊
	Subj 1 (Task 2) / Subj 2 (Task 2)	100%	😊	95,25%	😊
	Subj 1 (Task1) / Subj 2 (Task 3)	100%	😊	98%	😊
<b>Intra-Mouvements</b>	Subj 1 (Task 1) / Subj 2 (Task 1)	93,34%	😐	52,82%	😊
	Subj 2 (Task 2) / Subj 2 (Task 2)	86,57%	😐	58,42%	😊
	Subj 2 (Task 3) / Subj 3 (Task 3)	90,55%	😐	54,12%	😊

Tableau 9. Résultats de comparaison de la classification multi classes.



Le tableau confirme les résultats obtenus dans les deux premières expériences. Pour la classification inter-mouvement (les lignes 1,2 et 3), les deux approches arrivent à classer les postures des deux mouvements avec une très grande précision > 95%.

Pour la classification intra-mouvement (les lignes 4 et 5) contrairement à l'approche qui se base sur l'analyse topologique, on confirme l'incapacité de l'approche classique à reconnaître le même mouvement exécuté par deux sujets différents à cause de la différence morphologique ou bien par le même sujet dans le cas de l'ajout d'un bruit ou la présence d'une liaison motrice.

## V.5 Discussion

Dans ce chapitre, nous avons réalisé une étude comparative entre l'approche classique qui consiste à utiliser les données issues de la capture du mouvement directement comme entrées aux différentes méthodes de classification et l'approche proposée qui consiste à effectuer un pré-traitement en analysant topologiquement (à l'aide de la persistance homologique) les positions des marqueurs pour chaque posture du mouvement et utiliser les signatures topologiques obtenues comme entrées pour la classification.

Les résultats montrent que pour la classification binaire, dans le cas d'une classification inter-mouvements, les deux approches retournent une bonne précision de classification.

Dans le cas de la classification intra-mouvement, l'approche proposée montre sa capacité à détecter les différences morphologiques entre les participants lors de la classification. Avec la classification directe des données, les résultats montrent qu'on n'est pas en mesure de reconnaître le même mouvement réalisé par deux participants différents. Les résultats obtenus pour la classification multi-classes (Tab 8) et (Tab 10), confirment ceux obtenus pour la classification binaire (Tab 7) et (Tab 9). On remarque une baisse de précision avec l'approche de classification directe dans la cas d'inter-mouvement. Pour l'intra-mouvement, la classification directe montre toujours son incapacité à reconnaître le même mouvement exécuté par différents participants.

Dans la deuxième expérience, en analysant un mouvement plus complexe (mouvement de ciblage séquentiel vers neuf cibles), même dans le cas où un participant répète le même mouvement, on n'obtient pas les mêmes trajectoires de marqueurs, parce que la personne ne va pas répéter le mouvement avec la même exactitude (pas de trajectoire identique de l'index utilisé pour le mouvement de ciblage) (Fig 50), ce qui pousse

## Classification de mouvements

l'algorithme de classification à considérer les répétitions comme étant des mouvements différents si les données sont utilisées directement comme entrées à la méthode de classification. Cela pourrait constituer une limitation pour les applications biomédicales telles que la rééducation physique. L'ajout d'une étape d'analyse topologique via la persistance homologique a permis de s'affranchir de cette limitation. (voir les deux dernières lignes du tableau 9 ).

Les résultats retournés après l'ajout du bruit blanc (Fig 52 et 53) montrent la robustesse de l'approche proposée face à ce type de problème répandu lors de l'acquisition des données.

La dernière expérience a montré que même en utilisant une Kinect qui est un outil moins précis que les systèmes de motion capture sophistiqués de type Vicon et qui enregistre le mouvement avec une fréquence qui ne dépasse pas 30 Hz, l'application de l'étape d'analyse topologique avant la classification permet d'améliorer la précision de la reconnaissance du mouvement.

Malgré les bons résultats retournés par l'approche proposée, ainsi que sa robustesse face au bruit blanc des enregistrements, on a noté que son inconvénient est son temps de calcul important par rapport à l'approche de classification directe.

Le tableau (Tab 5) montre la complexité de calcul des différentes étapes de l'approche proposé. On remarque que l'étape de calcul de distances entre les signatures topologiques de postures est la fonction la plus coûteuse en temps de calcul et on note aussi que c'est la fonction la plus utilisé :

- Dans la partie d'apprentissage, on calcule la distance entre la signature topologique de chaque posture d'un mouvement et toutes les autres postures du même mouvement et celles des autres mouvements qui construisent l'ensemble d'apprentissage afin de construire la matrice carrée de distance.
- Lorsqu'on veut classer un mouvement donné, après le calcul des signatures topologiques pour chacune des postures de ce dernier, on doit créer une matrice de distance dans laquelle sont consignées les distances entre chaque posture du mouvement à classer et toutes les autres postures du groupe de mouvements qui constituent l'ensemble d'apprentissage.

Dans les chapitres suivants, on va essayer de proposer des approches qui permettent de réduire le temps de calcul, soit en réduisant le nombre d'itérations des étapes les plus coûteuses en temps de calcul, soit en optimisant notre code pour tirer avantage de l'exécution parallèle.

## V.6 Démarche avec réseaux de neurones

Comme détaillé dans la partie discussion précédente, l'approche basée sur la persistance homologique offre plusieurs avantages par rapport à l'approche de classification directe des données cinématiques mais elle est coûteuse en temps de calcul.

Le tableau (Tab 5) montre que le calcul de distances avec les métriques (La distance de Wasserstein et La distance de Bottleneck) est la partie la plus coûteuse en temps de calcul.

Pour chaque mouvement  $M$  à classer, on doit calculer la distance entre la signature topologique de chaque posture du mouvement  $M$  et toutes les signatures topologiques des postures des mouvements qui forment l'ensemble d'apprentissage utilisé pour l'algorithme de classification, pour s'assurer que la distribution (les coordonnées des différentes signatures) obtenue après l'application de la réduction multidimensionnelle soit dans le même repère (Fig 43).

Le nombre d'itérations nécessaires pour calculer les valeurs du triangle supérieur de la matrice symétrique de distances ainsi obtenue est :

$$Nb_{iteration} = \frac{([\sum_{i=1}^k p_i] + n + 1) + ([\sum_{i=1}^k p_i] + n)}{2}$$

Où  $k$  est le nombre de mouvements dans l'ensemble d'apprentissage,  $p_i$  est le nombre de postures de chaque mouvement  $M_i (i = 1, \dots, k)$  de l'ensemble d'apprentissage,  $n$  est le nombre de postures du mouvement à classer.

La complexité exponentielle de la méthode de calcul de distances ainsi que le nombre important d'itérations nécessaires pour construire la matrice carrée de distance font que cette étape représente la partie la plus coûteuse en temps de calcul dans la démarche proposée.

Pour pallier à ce problème, on propose de considérer les traces des signatures topologiques retournées après réduction dimensionnelle comme des formes à reconnaître au lieu d'un ensemble de nuages de points à regrouper en clusters à l'aide de la machine à vecteurs de support.

Pour la reconnaissance des formes, on utilise les réseaux de neurones qui reste l'outil le plus adapté pour ce type de tâches (CHRISTOPHER M. BISHOP , 1995) .

Pour chaque mouvement de l'ensemble d'apprentissage, on crée une matrice de distance intra-mouvement (entre les signatures topologiques des postures du même mouvement) sur laquelle on applique une réduction dimensionnelle. Une fois leurs coordonnées mises à l'échelle entre -1

## Classification de mouvements

et 1, les traces des signatures topologiques obtenues dans un espace de dimension 2 vont représenter une forme qui représente la distribution des signatures topologiques propre à ce mouvement. Ainsi, dans notre base d'apprentissage, chaque mouvement sera représenté par une forme (distribution des signatures topologiques des postures, après une réduction dimensionnelle (Fig 55).

L'apprentissage va être effectué sur les formes (Fig 55) obtenues après la réduction dimensionnelle de chacune des matrices de distance intra-mouvement. Pour renforcer l'apprentissage, comme pour la première démarche, pour chaque mouvement on aura  $n$  répétitions ( $n$  formes).

Ainsi, une fois qu'on veut classifier un nouveau mouvement, il suffit de créer sa propre matrice de distance, au lieu d'une matrice de distance entre les signatures topologiques des postures de ce dernier et toutes les postures de l'ensemble d'apprentissage. Après la réduction dimensionnelle et la mise à l'échelle, le rôle du réseau de neurone sera donc de reconnaître la forme en se basant sur les exemples (formes) de la base d'apprentissage (Fig 55).

Dans la section suivante on va détailler la méthode des réseaux de neurones. Ensuite, une comparaison sera effectuée entre l'approche qui se base sur le clustering à l'aide des machines à vecteurs de support et celle sur la reconnaissance des formes avec les réseaux de neurones.

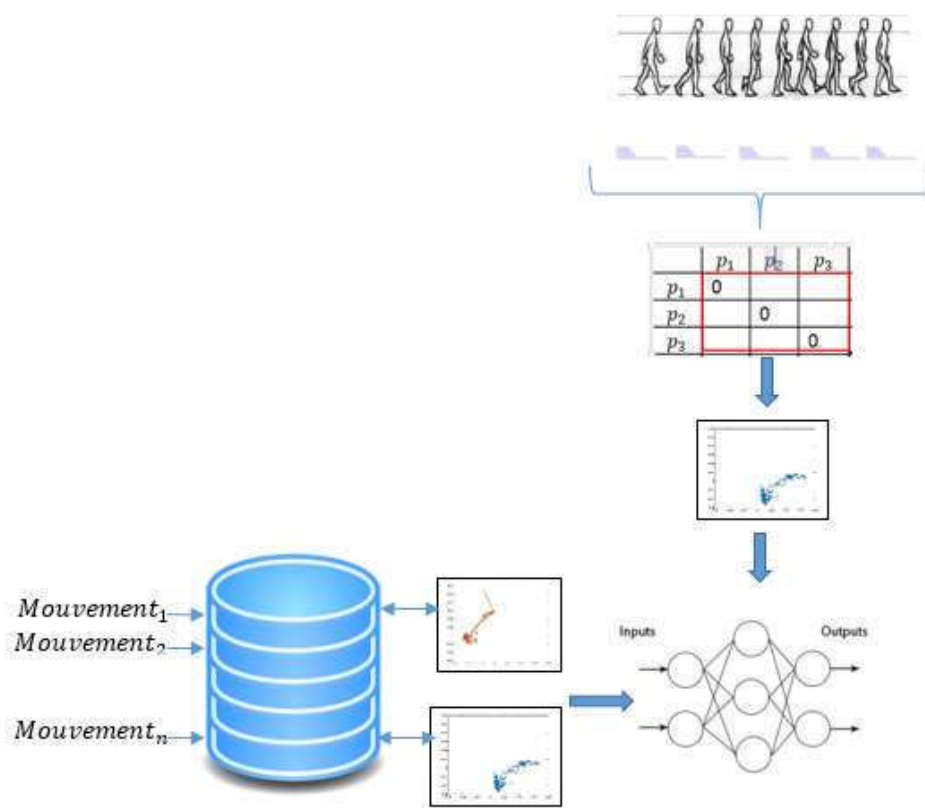


Figure 55. L'approche proposée qui se base sur la reconnaissance des formes issues de la réduction dimensionnelle des matrices de distance entre les postures du même mouvement via les réseaux de neurones.

### V.6.1 Introduction aux réseaux de neurones

Un réseau neuronal artificiel est un classificateur inspiré du fonctionnement du cerveau humain. Chacun de ces neurones est connecté à de nombreux autres neurones similaires. Chaque cellule reçoit des signaux (impulsions électriques) via les dendrites des autres cellules auxquelles elle est attachée. Si la somme de ces signaux atteint un certain seuil, le neurone transmet le signal aux autres neurones par les axones. Les contacts entre deux neurones (entre axone et dendrite) se font par l'intermédiaire des synapses (Fig 56).

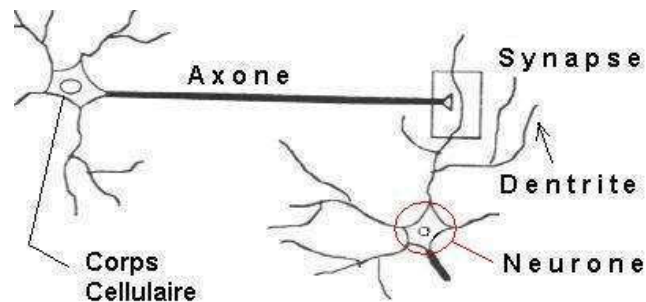


Figure 56. Modèle du neurone biologique.

- Neurone formel

En 1943 Mac Culloch et Pitts introduisent la notion de neurone formel (Warren S. McCulloch & Walter Pitts, 1943), qui est une formalisation mathématique simplifiée du neurone biologique. C'est un automate qui agit comme une fonction de transfert. Un neurone reçoit  $N$  informations (valeurs) en provenance de la couche des neurones précédents. Chacune des entrées est pondérée pour lui donner plus ou moins d'influence (la force de la connexion) vis-à-vis du neurone. Cette pondération est effectuée à l'aide d'une valeur nommée poids ou coefficient synaptique qu'on note  $w_i$  pour  $i = 1 \dots N$  (Fig 57).

Le neurone calcule la valeur de sa sortie en effectuant une somme pondérée des valeurs d'entrée :

$$\text{Somme} = \sum_{i=1}^N x_i w_i$$

La somme sera transmise à la couche des neurones suivants si la valeur de la somme est égale ou supérieure au seuil  $\theta$  (Fig 58).

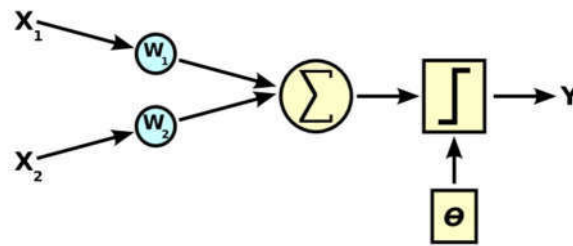


Figure 57. Modélisation du neurone artificiel à deux entrées.

Les neurones formels sont regroupés en réseaux avec différentes topologies possibles (réseaux proactifs, récurrents (Andrew Tchircoff, 2017)).

À l'instar des réseaux de neurones biologiques dont la fonction est d'apprendre tout en améliorant les capacités intellectuelles de l'individu, le rôle des réseaux de neurones artificiels est d'effectuer des classifications tout en apprenant à améliorer les résultats. L'apprentissage des réseaux de neurones consiste à définir la valeur des poids synaptiques qui permettent d'obtenir le meilleur taux de précision lors de classification.

- Perceptron et loi de Hebb

Les travaux de McCulloch et Pitts (Warren S. McCulloch & Walter Pitts, 1943) ont introduit la notion de neurone formel et des coefficients synaptiques, mais sans donner d'indication sur la méthode à suivre pour déterminer les valeurs optimales des poids synaptiques  $W_i$ .

Le physiologiste Donald Hebb (DO Hebb, 1988) a présenté une méthode pour ajuster les valeurs des poids synaptiques selon le principe que si deux neurones connectés sont activés au même temps alors la force de connexion (poids synaptiques) augmente sensiblement.

- Algorithme de Hebb

Pour décrire l'algorithme de Hebb, on pose  $\mathbf{S}$  la base d'apprentissage composée d'un ensemble de couple  $(\mathbf{E}, \mathbf{R})$  où  $\mathbf{E}$  est le vecteur des entrées  $(e_1, e_2, \dots, e_n)$  et  $\mathbf{R}$  le sortie souhaitée.

La fonction du transfert du neurone ( $Signe(R)$ ), est la fonction de Heaviside (Osgood, 2000), qui résume comme suit (Fig 58):

- Si la sortie  $R > 0$  :  $Signe(R) = +1$
- Si la sortie  $R \leq 0$  :  $Signe(R) = -1$

## Classification de mouvements

On note  $a_i$  la valeur d'activation du neurone  $n_i$ .

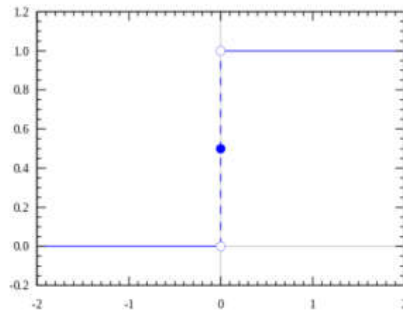


Figure 58. Graphe de La fonction de Heaviside, utilisant la valeur 0,5 comme seuil.

Les étapes de l'algorithme sont les suivantes :

- 1 On pose  $\mu$  une constante positive.
- 2 Les poids synaptiques  $W_i$  sont initiés aléatoirement.
  - Répéter
1. Sélectionner un couple d'entrée sortie  $(e, c)$  de l'ensemble  $S$ .
2. Calculer la sortie  $O$  du réseau de neurones pour l'entrée  $e$ .
  - Si  $C \neq O$
3. Modifier les poids  $W_i$  du réseau
$$w_{ij} = w_{ij} + \mu * (a_i * a_j)$$
  - Fin si
  - Fin répéter

- Perceptron

Le perceptron formalisé par Frank Rosenblatt (Rosenblatt, 1958), est la version la plus basique (monocouche) d'un réseau de neurone utilisant le principe de l'algorithme de Hebb pour ajuster les valeurs des poids synaptiques.

Un perceptron linéaire à seuil (Fig 60) à  $N$  entrées  $(e_1, e_2, \dots, e_n)$ , est défini par  $N+1$  constantes : les  $N$  poids synaptiques  $(w_1, w_2, \dots, w_n)$  ainsi que le seuil (ou le biais) noté  $\theta$ , la sortie  $O$  est calculé comme suit :

- $O = 1$  si  $\sum_{i=1}^n e_i w_i > \theta$ .
- $O = 0$  sinon.

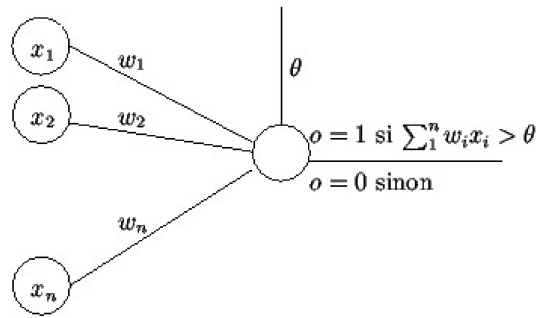


Figure 59. Perceptron avec seuil.

Afin de faciliter la notation et certaines preuves mathématiques, une variante est proposée en remplaçant le seuil  $\theta$  par une entrée supplémentaire  $e_0$  dont la valeur est constante (Fig 60).

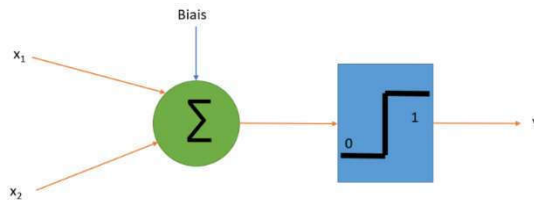


Figure 60. Perceptron avec perceptron avec entrée supplémentaire (biais).

La sortie  $O$  est calculée comme étant la somme pondérée des entrées  $(e_1, e_2, \dots, e_n)$  et le biais  $b$ :

$$O = \sum_{i=1}^n e_i w_i + b$$

L'équation de sortie correspond à l'équation d'un hyper-plan. Le perceptron (réseau de neurones monocouche) effectue une classification binaire en divisant l'espace  $R^n$  des  $n$  entrées en deux sous-ensembles. Le perceptron est donc en mesure d'apprendre à ne classer que les données linéairement séparables (Fig 61).



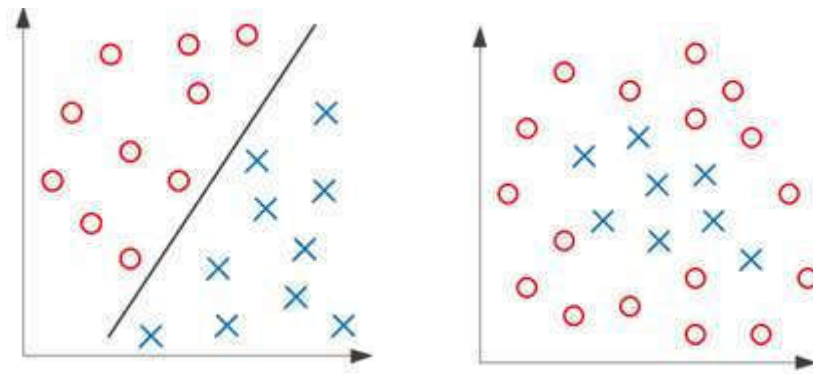


Figure 61. (a) données linéairement séparables, (b) données non-linéairement séparables qui représentent la limite des réseaux de neurones monocouche.

Après l'algorithme de Hebb (DO Hebb, 1988), plusieurs algorithmes d'apprentissage (ajustement des poids synaptiques) ont été développés. L'une des méthodes d'apprentissages supervisés les plus utilisées est la descente de gradient qui est un algorithme d'optimisation qui se base sur la notion de gradient (Bottou, 2010).

Dans notre cas, pour un perceptron, la fonction qu'on veut minimiser est la moyenne d'erreur (la différence entre la valeur obtenue et celle souhaitée) pour chaque vecteur  $x_i$  d'entrée :

$$\frac{1}{|S|} \sum_{(x_i, c_i) \in S} \text{Erreur}(C_i, P_w(x_i))$$

Où  $S$  est l'ensemble d'échantillons d'apprentissage,  $c_i$  la sortie souhaitée (la cible) pour le vecteur d'entrée  $x_i$  et  $P_w(x_i)$  le résultat réel retourné par le perceptron pour des poids  $w$ .

L'apprentissage du perceptron peut être formalisé comme étant un problème d'optimisation dont le but est de minimiser l'erreur ( $\text{Difference}(C_i, P_w(x_i))$ ). Pour chaque exemple d'apprentissage, la fonction *Erreur* :

$$\text{Erreur}(C_i, P_w(x_i)) = -(C_i - P_w(x_i)) \times w \cdot x_i$$

$w \cdot x_i$  est le produit scalaire entre les poids  $w_i$  les éléments  $e_i$  du vecteur  $\sum_{i=1}^n e_i w_i$

- Dans le cas où le résultat retourné par le perceptron est équivalent au résultat attendu  $P_w(x_i) = C_i$ , l'erreur est nulle.

## Classification de mouvements

- Si le résultat retourné est différent du résultat souhaité, l'erreur correspondra à la différence entre l'entrée  $w \cdot x_i$  est

Le but étant de trouver le minimum optimal (minimiser la moyenne d'erreur de tous les exemples d'apprentissage) par rapport au vecteur des poids  $w$  (Fig 62).

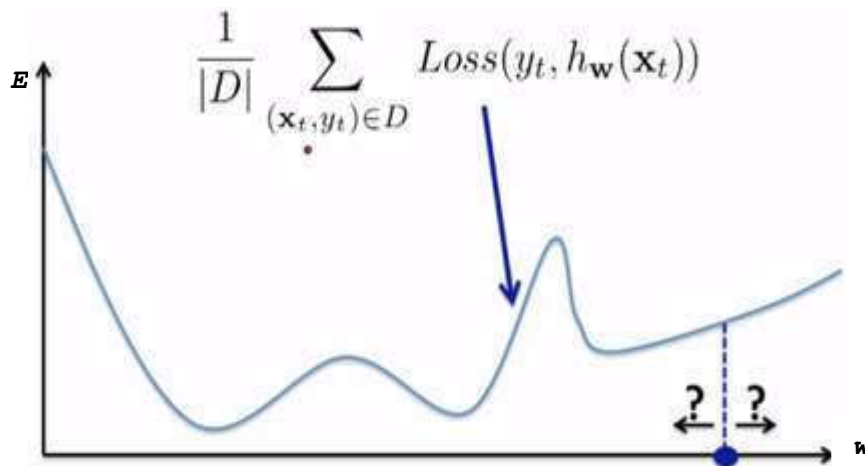


Figure 62. Utilisation de la descente de gradient, pour déterminer la valeur du poids optimal  $w$ , qui minimise l'erreur.

L'algorithme de la méthode de la descente du gradient est défini comme suit :

1. Initialiser aléatoirement les poids  $w_i$  .
2. Pour chaque vecteur  $x_i$  de l'ensemble de l'apprentissage en mis à jour les ajustements des poids :

$$\Delta w_i = \Delta w_i + t * \frac{\partial}{\partial w_i} Erreur(C_i, P_w(x_i))$$

3. Une fois la modification des poids  $\Delta w_i$  est effectuée pour chaque exemple d'apprentissage, on effectue l'ajustement final des poids :

$$w_i = w_i + \Delta w_i$$

## Classification de mouvements

La valeur introduite du taux d'apprentissage noté  $t$  correspond à la taille du pas à faire (Fig 62). La valeur de cette variable est généralement comprise entre 0 et 1. Une forte corrélation existe entre la valeur de  $t$  et l'importance des ajustement apportées aux poids  $w$ . Une grande valeur de  $t$  va conduire à d'importantes corrections de valeur pour  $w$  (Fig 62) ; ce qui peut empêcher de trouver la solution optimale, et une petite valeur de  $t$  va ralentir la convergence vers la solution optimale.

La meilleure stratégie pour le choix de la valeur du taux d'apprentissage est de l'initialiser à une grande valeur qui se rapproche de 1 au début de l'exécution, et de la diminuer progressivement avec les itérations.

- Algorithme de Widrow-Hoff

La méthode de la descente du gradient retourne de bons résultats, mais elle reste couteuse en temps de calcul, Widrow et Hoff en 1960 (Widrow & Hoff, 1960) ont proposé un algorithme qui reprenait le même principe que la descente du gradient, à la différence qu'au lieu de faire la modification des poids à la fin, après l'itérations de tout l'ensemble d'apprentissage, l'algorithme Widrow-Hoff applique le même type de modification après chaque vecteur  $x_i$  d'entrées, ce qui fait que l'algorithme converge plus rapidement vers la solution optimale .

Ci-dessous les étapes de l'algorithme :

1. Tant que condition d'arrêt est non-atteinte
2. Pour chaque vecteur d'entrée  $x_i \in S$  :
  - On calcule la sortie  $o = P_w(x_i)$
  - Pour chaque poids  $w_i$ 
$$w_i = w_i + t(c_i - o) * x_i$$
Fin PourFin Pour
3. Si besoin modification de la valeur de  $t$  .  
Fin Tant que.

- Les réseaux multicouches

Comme détaillé dans la section précédente, les réseaux de neurones monocouches ne sont pas en mesure de résoudre les problèmes de classification non linéaire ou multi-classes. Ce genre de problème représente la quasi majorité des problèmes réels.

Pour surmonter ce problème, la solution proposée était de combiner les neurones formels entre eux en s'inspirant des réseaux de neurones biologiques.

## Classification de mouvements

Un réseau de neurones multicouches est caractérisé comme suit (Fig 63) :

- Il est composé de plusieurs couches
- Chaque couche peut se constituer d'un ou de plusieurs neurones.
- Les entrées ( $e_1, e_2, \dots, e_n$ ) sont injectées dans tous les neurones de la couche d'entrée.
- Chaque neurone est connecté à tous les neurones de la couche qui le suit.

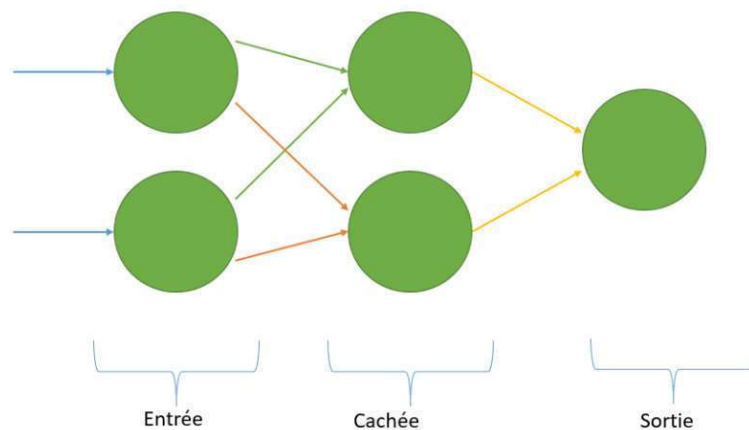


Figure 63. Un réseau de neurones multicouches, composé de trois couches.

La figure (Fig 63) présente un réseau de neurones multicouches qui se compose de trois couches : une pour l'entrée, une couche intermédiaire nommée couche cachée et une couche de sortie. Le nombre de neurones à la couche de sortie correspond au problème à résoudre (le nombre de classes). Pour le reste des paramètres (nombre de couches cachées, nombre de neurone par couche), il n'existe aucune méthode pour déterminer l'architecture optimale. Des essais de plusieurs combinaisons reste la seule solution.

Les informations d'entrées ( $e_1, e_2, \dots, e_n$ ) sont injectées à tous les neurones de la première couche. Ensuite, ces derniers sont connectés à ceux de la couche suivante et ainsi de suite jusqu'à la couche de sortie.

La méthode d'apprentissage d'un réseau de neurones multicouches, s'inspire de la méthode Widrow-Hoff (Widrow & Hoff, 1960) dont le fait de corriger les poids pour chaque exemple  $x_i$  de l'ensemble d'apprentissage  $S$ .

L'algorithme consiste à corriger en premier les poids des neurones de sortie et ensuite propager l'erreur entre la couche de sortie et celle qui le précède. L'erreur est successivement propagée en arrière d'où

## Classification de mouvements

son nom (algorithme de rétropropagation du gradient) jusqu'à la couche d'entrée.

Afin de détailler le pseudo algorithme de la rétropropagation du gradient, on considère toutes les couches à part la dernière comme étant une seule couche cachée et la dernière couche comme étant la couche de sortie. On note  $s_k$  le résultat retourné par le  $k$ ème neurone de la couche de sortie,  $c_k$  est la sortie attendue et  $o_i$  est la sortie retournée par le  $i$ ème neurone des couches cachées.

Ci-dessous les étapes de l'algorithme

- Initialiser les poids  $w$  aletoirement.
- Tant que : taux d'apprentissage < seuil
- Prendre un exemple de l'ensemble d'apprentissage  $(x_i, c_i)$ 
  - Pour chaque neurone  $k$  de la couche de sortie on calcule l'ajustement  $\Delta_k$  :
$$\Delta_k = s_k(1 - s_k)(c_k - s_k)$$
  - Fin pour.
- Pour chaque couche cachée
  - Pour chaque neurone  $i$  de la couche cachée courante
    - Initialiser Sum=0
    - Pour chaque Poids vers un neurone de la couche  $i+1$  :
$$Sum = Sum + \Delta_{i+1} \times w_{i \rightarrow i+1}$$
    - Fin pour
    - $$\Delta_i = o_i(1 - o_i) \times Sum$$
    - Fin pour
    - Pour chaque poids du réseau
      - Si lien vers neurone de sortie
$$w_k = w_k + \text{taux} \times \Delta_k \times s_k$$
      - Sinon
$$w_i = w_i + \text{taux} \times \Delta_i \times o_i$$
    - Fin si
    - Fin Pour
    - Si besoin ajuster le taux d'apprentissage.
  - Fin tant que

## **V.6.2 Résultats et comparaison avec la classification en utilisant la machine à vecteurs de support**

La modification de l'approche proposée en utilisant les réseaux de neurones a été proposée dans le but de réduire le nombre d'itérations de l'étape la plus coûteuse en temps de calcul (calcul de distances entre les signatures topologiques) afin de diminuer le temps de calcul global de l'approche proposée.

Dans cette section, on va effectuer une comparaison entre le temps de calcul de l'approche de classification de mouvement avec l'analyse de persistance homologique avant et après la modification apportée par l'utilisation des réseaux de neurones. La précision de classification sera le second point de cette comparaison. Notre objectif est de réduire le temps de calcul tout en gardant une bonne précision de classification.

Cette partie sera consacrée à la comparaison des résultats retournés par l'approche basée sur la classification des données avec les machines à vecteurs de support et celle basée sur l'apprentissage par réseaux de neurones. La comparaison se portera sur le temps de calcul ainsi que la précision de la classification.

Après plusieurs tests, l'architecture du réseau de neurones sélectionnée est une architecture multicouche (Fig 63), avec une couche d'entrée à deux neurones est une couche cachée à 4 neurones et une couche de sortie à 3 neurones.

Afin de valider les modifications apportées dans ce chapitre, on sélectionne les mouvements de pointage classifiés dans les chapitres précédents (Fig 44) et (Fig 49), afin de réaliser une comparaison entre le temps de calcul et la précision obtenue avec la démarche proposée dans le chapitre précédent (Fig 38) basée sur l'analyse topologique et la classification avec l'algorithme de la machine à vecteurs de support, et ceux obtenus après la modification de cette dernière démarche en utilisant les réseaux de neurones pour faire la classification (Fig 55).

Les résultats retournés (Tab 10) montrent que les modifications apportées à l'approche, causent une perte entre 5% et 10% en précision de classification.

## Classification de mouvements

	Mouvements à classier	Persistence homologique + SVM	satisfaction par rapport au résultat attendu	Persistence homologique + Réseaux de neurones	satisfaction par rapport au résultat attendu
<b>Inter-Mouements</b>	Subj 1 (Task 2) / Subj 2 (Task 1)	96.25%	😊	87.32%	😊
	Subj 1 (Task 1) / Subj 3 (Task 2)	91.72 %	😊	82.05%	😊
	Subj 2 (Task 2) / Subj 3 (Task 1)	93,04%	😊	89.46%	😊
<b>Intra-Mouements</b>	Subj 1 (Task 1) / Subj 2 (Task 1)	48,52%	😊	43.34%	😊
	Subj 1 (Task 2) / Subj 2 (Task 2)	52,65%	😊	56.03%	😊
	Subj 2 (Task 1) / Subj 2 (Task 1)	53,22%	😊	41.56%	😊
<b>Multi-classes</b>	Subj 1 (Task1, Task2, Task3) / Subj 1 (Task 1)	92.45%	😊	84.63%	😊
	Subj 3 (Task1, Task2, Task3) / Subj 3 (Task 1)	90.57%	😊	83.06%	😊
	(Subj 1, Subj 2, Subj 3) (Task1, Task2, Task3) / Subj 1 (Task 1)	86.47%	😊	75.81%	😊

Tableau 10. Comparaison des résultats de la première expérience, entre les deux approches proposées, le calcul de la matrice de distance globale et la classification avec la machine à vecteurs de support, et la seconde qui consiste à calculer.

Les résultats de comparaison pour le premier mouvement de pointage (Fig 44) sont détaillés dans le tableau 10 et ceux du deuxième mouvement de pointage (Fig 49) dans le tableau 11.

Classification de mouvements

	Mouvements à classier	Persistence homologique + SVM	satisfaction par rapport au résultat attendu	Persistence homologique + Réseaux de neurones	satisfaction par rapport au résultat attendu
<b>Inter-Mouvements</b>	Subj 1 (Task1) / Subj 2 (Task 2)	93.56%	😊	89.50%	😊
	Subj 2 (Task1) / Subj 3 (Task 2)	91.30%	😊	87.72%	😊
	Subj 2 (Task3) / Subj 3 (Task 4)	85.67%	😊	83.29%	😊
<b>Intra-Mouvements</b>	Subj 1 (Task 1) / Subj 2 (Task 1)	52.02%	😊	47.56%	😊
	Subj 1 (Task 2) / Subj 2 (Task 2)	48,27%	😊	42.46%	😊
	Subj 1 (Task 3) / Subj 2 (Task 3)	51,16%	😊	43.58%	😊
<b>Multi-classes</b>	Subj 1 (Task1, Task2, Task3) / Subj 1 (Task 1)	87.45%	😊	83.44%	😊
	Subj 1 (Task1, Task2, Task3) / Subj 1 (Task 2)	92.56%	😊	87.83%	😊
	Subj 3 (Task1, Task2, Task3) / Subj 3 (Task 3)	92.75%	😊	90.59%	😊

Tableau 11. Comparaison des résultats obtenus par les deux approches proposées, lors de la classification des mouvements de la deuxième expérience.



## Classification de mouvements

Le deuxième point de comparaison est le temps de calcul. On commence par la classification binaire du mouvement de pointage de la première expérience (Fig 44).

Le temps d'exécution correspond uniquement au temps de calcul de la classification une fois que la phase d'apprentissage des réseaux de neurones ou des machines à vecteurs de support sont déjà effectuée. Les histogrammes (Fig 64) et (Fig 65) montrent la moyenne du temps d'exécution pour la classification binaire, du mouvement de pointage de la première expérience (Fig 44 ) et de la second expérience (Fig 49) :

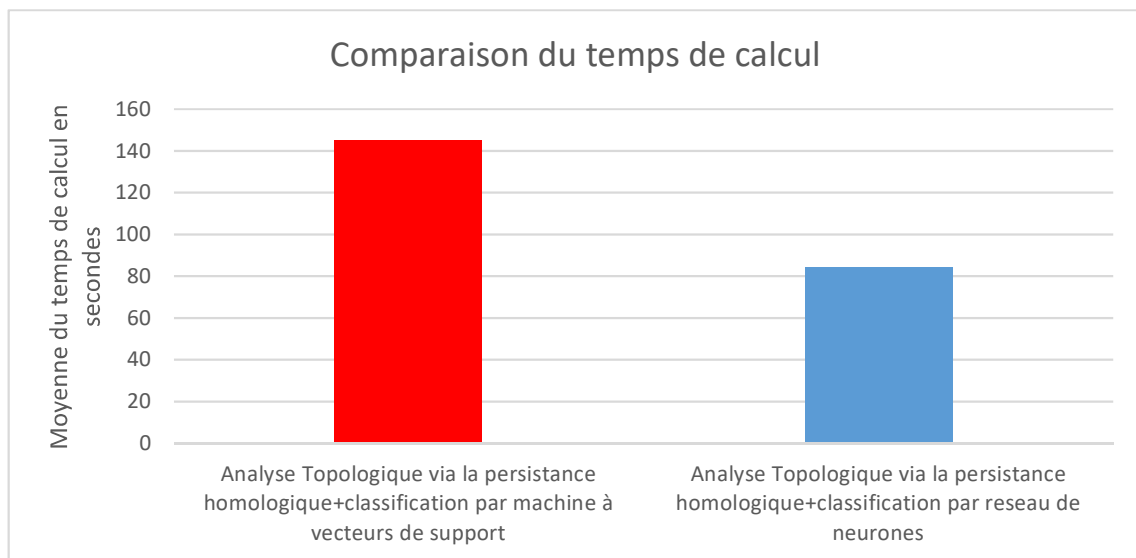


Figure 64. Le temps d'exécution moyen des deux approches proposées, pour une classification binaire de deux mouvements de pointage vers une seule cible.

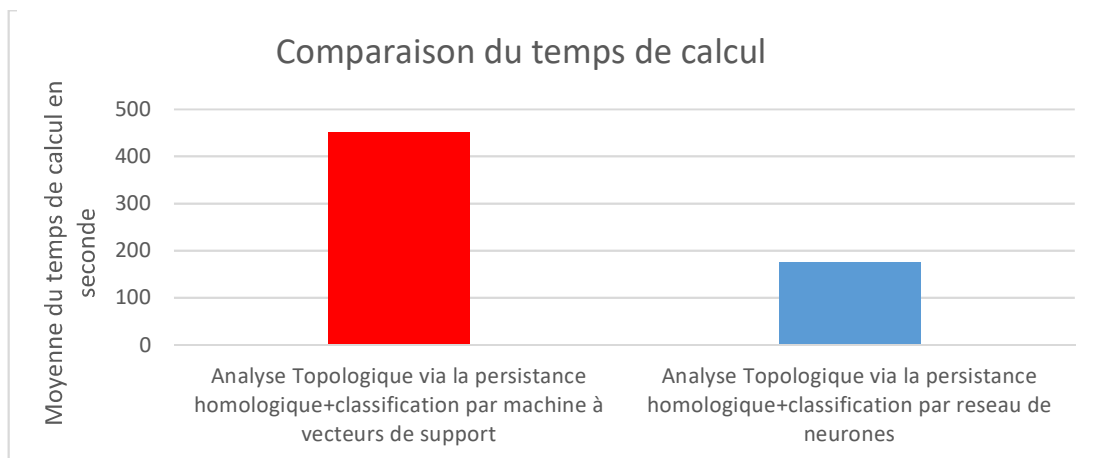


Figure 65. Comparaison de temps de calcul moyen d'une classification binaire, de deux mouvements de pointages vers neuf cibles.

## Classification de mouvements

Pour les deux expériences, on note une nette diminution du temps de calcul, qui est passé de 145 secondes à 82 secondes pour la première expérience, et de 450 secondes à 174 secondes pour la seconde expérience.

Dans le cas d'une classification multi-classes, les résultats (Fig 66) confirment encore plus l'amélioration apportée par l'utilisation des réseaux de neurones, par rapport au temps de calcul, qui est passé de 700 secondes à 450 secondes ce qui fait un gain de presque de 45% .

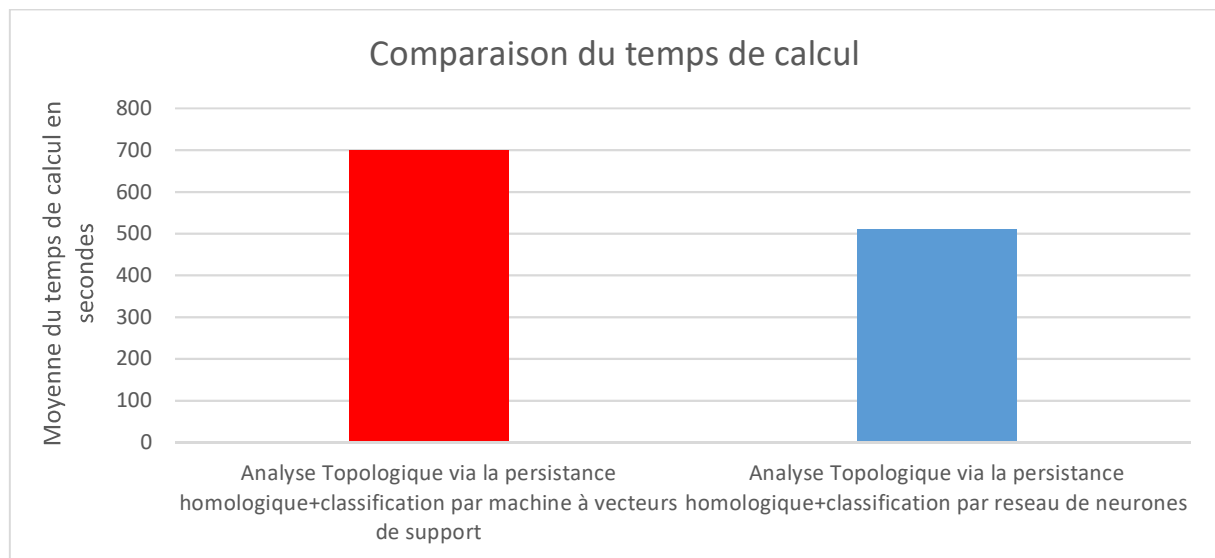


Figure 66. Comparaison du temps d'exécution moyen, des deux approches lors d'une classification multi-classe.

Partant du constat que malgré les bons résultats retournés par l'approche de classification des mouvements humains, qui se base sur l'analyse topologique par la persistance homologique, elle reste très couteuse en temps de calcul.

En comparant la complexité de calcul, de chaque étape de l'approche (Tab 5), on a déterminé que la partie de calcul de distance entre les signatures topologiques des postures des mouvements est la plus onéreuse en temps de calcul, à cause de sa complexité de calcul exponentielle, et le nombre d'itérations important afin de construire une matrice de distance entre les signatures topologiques des postures du mouvement à classifier et toutes les autres signatures des postures des mouvements de la base d'apprentissage. L'utilisation des réseaux de neurones permet de pallier à cette contrainte en supprimant cette étape. Etant donné que les matrices de distance sont calculées exclusivement entre les postures du même mouvement, cela réduit drastiquement le nombre d'itérations.

L'autre facteur qui influence l'augmentation du temps de calcul est la nature même des machines à vecteurs de support qui est une méthode destinée initialement pour résoudre des problèmes de classification binaire. Pour la classification multi-classes, la variante utilisée une-contre-une nécessite un traitement hyperplan par hyperplan, ce qui

## Classification de mouvements

augmente le temps d'exécution. Avec les réseaux de neurones, l'apprentissage prend plus de temps mais une fois réalisé, la classification des individus nécessite moins de temps de calcul.

Une autre méthode qui permet de réduire le temps de calcul est de profiter des processeurs multicœurs actuels et d'exécuter les algorithmes en parallèle. Cette approche sera détaillée dans la section suivante.

# Chapitre VI Parallélisme

## VI.1 Introduction

Comme le montre les résultats obtenus dans les chapitres précédents, l'ajout de l'étape d'analyse topologique des données apporte une nette amélioration à la précision de classification des mouvements mais augmente considérablement le temps de calcul.

L'analyse de la complexité des étapes (Tab 5), a permis de déterminer la partie de la démarche qui est la plus couteuse en temps de calcul et en nombre d'itérations. Afin de réduire le temps de calcul, une modification de l'approche initiale (Fig 38) avec l'utilisation des réseaux de neurones (Fig 55) a été détaillée dans le chapitre précédent. Avec cette modification, le temps de calcul a été réduit jusqu'à 45% dans le cas d'une multi-classes, mais cela a causé une perte de précision entre 5% et 10%.

Dans cette section on va essayer de réduire le temps de calcul, mais sans perdre en précision de classification, en s'appuyant sur l'exécution parallèle, qu'on détaillera dans ce chapitre.

## VI.2 Principe du parallélisme et état de l'art

L'une des solutions pour résoudre un problème couteux en temps de calcul est d'utiliser plusieurs unités de calcul qui vont chacune d'elles résoudre une partie du problème simultanément.

Dans la pratique jusqu'à 2005 l'année de la commercialisation des premiers processeurs-multicoques pour le marché des serveurs basés sur l'architecture x86 (Opterons d'AMD), la performance de calcul d'un ordinateur était complètement liée à la fréquence en GHz (nombre maximum d'opérations réalisées par seconde). Cette augmentation de fréquences d'une génération de processeurs à une autre en se basant sur la loi de Moore (Gordon E., 1965), qui prédit que la concentration des transistors dans un microprocesseur double tous les 18 à 24 mois (Fig 67), génère un problème de consommation d'énergie qui a atteint les 100 W pour le Intel pentium 4, ce qui engendre des difficultés pour refroidir les circuits considérant la relation directe entre fréquence du processeur en GHz et sa température ce qui a induit des problèmes de surchauffe.

Mais bien avant la commercialisation du premier micro-processeur multicœurs en 2005, plusieurs travaux ont été réalisés dans le but de

## Parallélisme

réaliser une exécution sur plusieurs unités de calcul. Dans ce qui suit, on va citer quelques dates clés :

- En 1962 les travaux de Carl Adam Petri, qui a inventé les réseaux de Petri (Petri, 1966), qui sont une modélisation graphique et mathématique d'un système dynamique, Carl Adam Petri s'est intéressé à l'accès parallèle à la mémoire
- En 1964 le premier superordinateur doté d'un processeur double-cœurs, nommé Control Data (Fig 68) fut fabriqué par la société la société Control Data Corporation.
- En 1969 le lancement du premier système d'exploitation, capable de gérer jusqu'à 8 processeurs, le système est nommé Multics qui est l'acronyme de MULTiplexed Information and Computing Service.
- En 1977 Tony Hoare introduit le communicating sequential processes (CSP) (Communicating Sequential Processes, 1985), un langage formel qui a pour but de modéliser les interactions entre les processus d'un système.
- En 1982 le superordinateur Cray X-MP (Fig 68) est lancé, basé sur un autre superordinateur le Cray 1 développé en 1976, le Cray X-MP était doté de deux processeurs.
- Le premier superordinateur massivement parallèle est produit en 1986 par la société Thinking Machines Corporation, nommé CM-1 pour Connection Machine, la machine a été cadencée avec 16 000 processeurs.

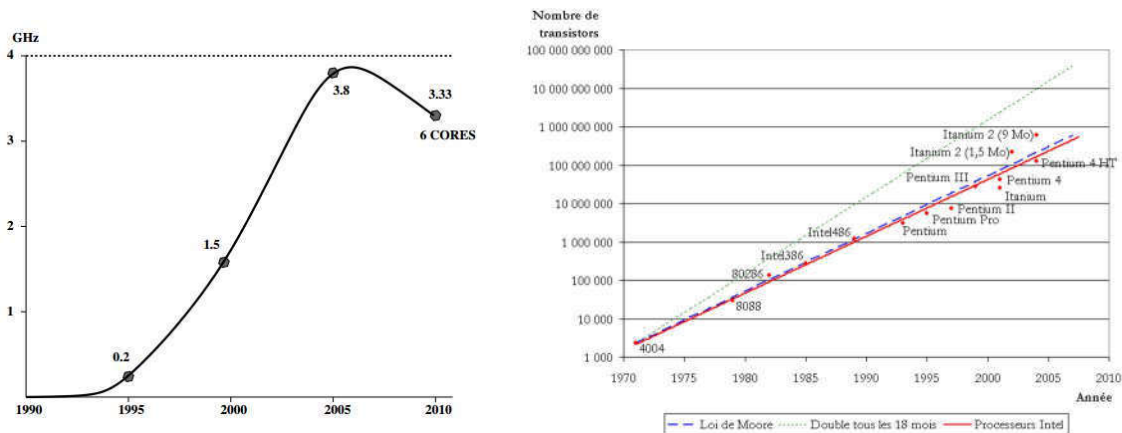


Figure 67. (a) évolution de la puissance des micro-processeurs en GHz de 1990 jusqu'à 2010. (B) schéma expliquant la loi de Moore.

Pour pallier ce problème, les constructions ont stabilisé la fréquence des micro-processeurs entre 3 et 4 GHz et se sont orientés vers une architecture parallèle multicœurs (unités de calcul) au lieu d'une

## Parallélisme

augmentation constante de la fréquence. Tout en suivant la loi de Moore le nombre de cœurs dans un processeur a augmenté en passant de 2 jusqu'à 8 pour les ordinateurs grands public tel que i7-6900K (Datasheet, 2016).

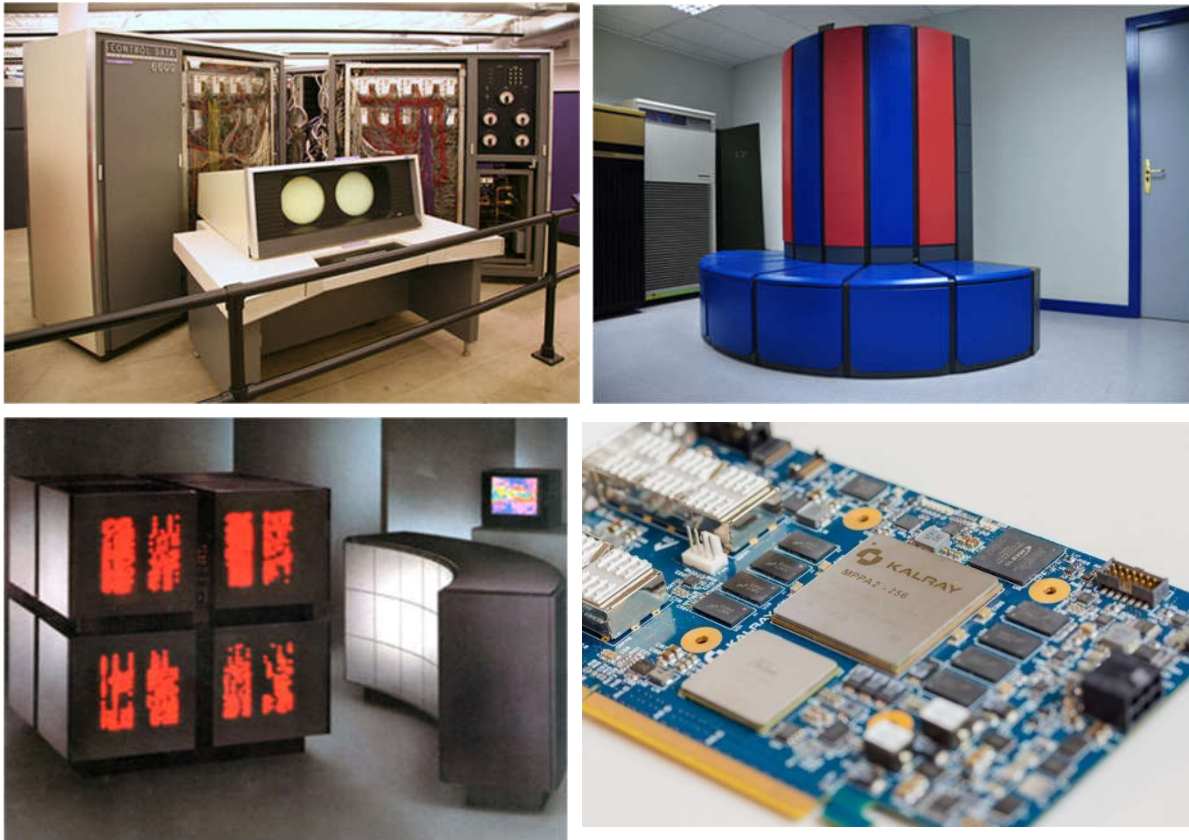


Figure 68. (gauche ligne 1) Control Data 6600 1964, (droite ligne 1) Cray X-MP 1982. (gauche ligne 2) Connection Machine CM-1 1986. (droite ligne 2) kalray processor 2017.

De nos jours, les fabricants proposent des micro-processeurs destinés au marché professionnel pour des applications nécessitant un calcul intensif tel que l'apprentissage profond. Le calcul en temps réel pour des systèmes embarqués (voiture autonome, pilotage automatique, applications médicales), kalray (Processors, 2017) développe une architecture nommée MPPA (Dupont de Dinechin, van Amstel, Poulhi, & Lager, 2014) pour (Massively parallel processor array), qui offre un calcul parallèle avec 256 cœurs (Kalray Announces Most Advanced Solution For Network Virtualization At Linley Data Center Conference, 2015).

Historiquement le terme générique utilisé dans la littérature scientifique et la documentation technique pour désigner une unité de calcul qui est chargée de résoudre un sous-problème est « worker » généralement ça correspond à un cœur d'un processeur. L'unité de calcul qui a aussi la charge de coordonner entre les différentes unités de

calcul est nommée « master », dans le cas général il y'a un seul « master ».

### VI.2.1 Communication entre unités de calcul

Afin que les unités de calcul puissent travailler ensemble, elles ont besoin de communiquer entre elles, le type de communication entre unités de calcul dépend de l'architecture parallèle exploitée, dans le cas d'une architecture mono-machine (Fig 69), la communication peut se faire, soit par :

- Variable partagée : dans ce cas une unité de calcul écrit sur une variable qui sera lue par la ou les unités destinataires.
- Echange par messages : Le système d'exploitation est chargé d'assurer la communication entre les différentes unités de calcul via des tubes de communication. Les tubes de communication sont des fichiers spéciaux gérés par le système d'exploitation.

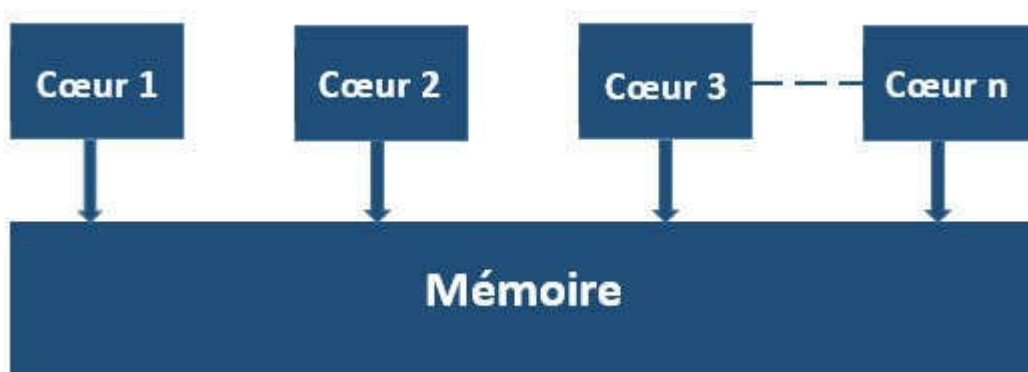


Figure 69. Architecture parallèle mono-machine.

Dans le cas où l'architecture est composée de plusieurs nœuds avec une mémoire partagée notée SMP pour Symmetric shared memory multiprocessor (Fig 70), (chaque nœud est une machine, qu'elle-même peut contenir plusieurs unités de calcul), la communication entre unités de calcul de la même machine se fait selon les méthodes décrites ci-dessus pour la communication intra-machine. Pour la communication des unités entre les différentes machines, il existe deux méthodes qui supportent le partage de mémoires entre unités de calcul des différentes machines du réseau :

- Une communication basée sur un protocole réseau TCP/IP ou UDP (User Datagram Protocol).

- La bibliothèque MPI (Message Passing Interface) ou l'une de ses variantes.

## VI.2.2 Communication réseau

Les protocoles peuvent être utilisés pour les machines qui forment des réseaux (clusters), qu'ils soient locaux LAN (Local Area Network), ou bien des réseaux étendus sur une zone géographique (MAN) :

- MAN (Metropolitan Area Network)
- WAN (Wide Area Network).

Les protocoles TCP/IP qui regroupent :

- Le protocole TPC pour (Transmission Control Protocol) qui garantit le transfert des paquets d'information dans leurs ordres d'envoi à leurs destinataires.
- Le protocole IP a le rôle d'identifier une machine du réseau à laquelle le message est destiné.

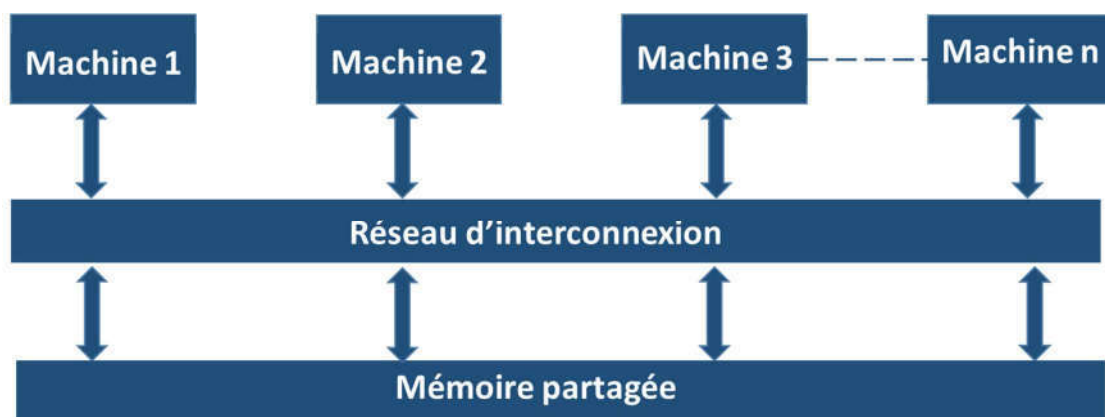


Figure 70. Architecture parallèle avec mémoire partagée.

Afin d'améliorer les capacités du protocole TPC, Miller (Barton P. Miller, Cathryn Macrander, & Stuart Sechrest, 1986) introduit la notion de sockets dans la distribution Unix de Berkeley. Les sockets sont une couche logicielle qui permet d'améliorer la communication entre les processeurs intra et inter machines.



### **VI.2.3 Communication avec la bibliothèque MPI**

La bibliothèque MPI (Message Passing Interface) permet d'assurer une communication entre différentes machines multiprocesseurs par passage de messages. La bibliothèque est devenue un standard pour exploiter un cluster de machine avec une mémoire partagée pour faire du calcul parallèle.

Les premières versions étaient implémentées dans les langages de programmation C/C++ et FORTRAN. De nouvelles versions existent pour java, Python et Perl. La bibliothèque est le fruit du travail d'un groupe de chercheur : Jack Dongarra et al ( Dongarra, Laforenza, & Orlando, 2003) qui ont publié les caractéristiques de la première version MPI1 en 1992. Le projet est porté par un consortium d'universités et de sociétés.

Le principe de ce standard est que le programme est dupliqué d'une manière à ce que chaque processeur va exécuter sa propre copie du code. Les variables d'exécution pour chaque processeur seront des variables locales sans la possibilité qu'un processeur ait la possibilité d'accéder à l'espace mémoire réservé aux variables d'un autre processeur. L'échange d'information entre processeur se fera uniquement par messages dont la transmission est assurée par les fonctions de la bibliothèque. Ce type d'exécution est nommé SPMD pour (Single Program Multiple Data) (Atallah, 1998).

La communication entre processeurs a pour but d'éviter les problèmes de synchronisation. Les deux problèmes majeurs de synchronisation sont les suivants :

1. Interblocage (ou deadlock) : cela implique deux unités (processeurs) ou plus qui s'attendent mutuellement. Ce phénomène est lié à l'accès aux ressources. Chacune des unités attend que l'autre libère les ressources.
2. Famine (ou livelock) : cela se produit quand le système ne garantit pas à une unité de calcul un accès à une ressource demandée dans un temps fini à cause du programme non équitable vis-à-vis de ses unités de calcul.

Pour assurer la cohérence des données lors de l'exécution parallèle du programme et éviter les problèmes de synchronisation, on peut appliquer un mécanisme de verrous et d'exclusions mutuelles. Cela consiste à donner l'exclusivité d'accès à certaines ressources pour une unité de calcul donnée pendant un moment. Si une autre unité demande l'accès à ces ressources, elle sera bloquée dans une file le temps que les ressources soient libérées.

Un autre paramètre important qui a un grand impact sur la performance de l'exécution est la répartition des tâches entre les unités de calcul. Comme décrit dans les sections précédentes, le principe du

parallélisme est de décomposer le problème principal en sous-problèmes et de les affecter aux différentes unités de calcul. Le temps de calcul global d'un programme exécuté en parallèle correspond à l'instant de la fin d'exécution de la dernière unité de calcul (la plus lente).

Ce qui fait qu'une bonne répartition équitable des tâches va permettre aux unités de calcul de terminer leurs exécutions avec un petit intervalle de différence. Cela évite alors d'avoir des problèmes de famine ou bien l'attente des unités de calcul qui terminent leurs tâches bien avant les autres unités de calcul (Fig 71).

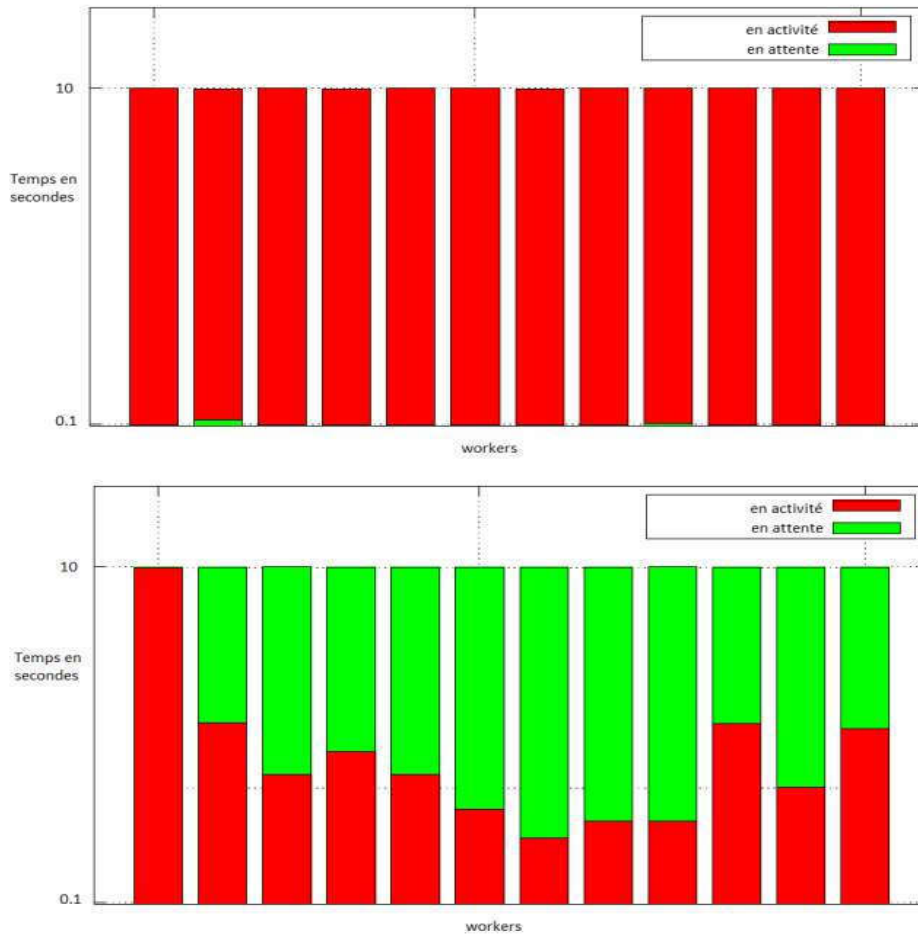


Figure 71. La communication entre les unités de calcul, permet une distribution optimale des tâches entre ces unités.

Il existe deux approches pour répartir les tâches entre les unités de calcul :

- Une répartition statique des tâches entre unités de calcul. Cela consiste à prédéfinir en amont les tâches à réaliser par chaque unité de calcul sans possibilité de changer l'attribution des tâches pendant l'exécution. Cette approche est avantageuse dans les cas où le temps d'exécution de chaque tâche (sous-problème) est connue d'avance.

- Une répartition dynamique, dont le but est d'affecter dynamiquement les tâches selon le taux d'occupation des unités de calcul afin d'obtenir un temps d'exécution global optimal. L'inconvénient de cette approche est l'augmentation des communications entre les unités de calcul.

La répartition des charges de travail sur les unités de calcul est un problème de la satisfaction des contraintes distribuées. Dans la littérature scientifique, ce type de problème est noté CSP pour Constraint Satisfaction Problem. Pour résoudre ce problème, plusieurs outils (frameworks) existent se basant sur la programmation des agents (Shoham, 1990) en associant un agent à chaque unité. Les agents vont essayer de travailler ensemble pour résoudre le problème. La première implémentation remonte à 1990 avec les travaux de Yokoo et al (Ishida, Yokoo, & Gasser, 1990). Depuis, différentes versions ont été réalisées par Hirayama et Yokoo (Distributed Partial Constraint Satisfaction Problem, 1997), Chong et Hamdi (Chong & Hamadi, 2006), Léauté et al (Léauté, Ottens, Szymanek, & Faltings, 2009) et Wahbi et al (Wahbi, Ezzahir, Bessiere, & Bouyakhf, 2011).

### VI.2.3.1 Performance du parallélisme

Théoriquement dans un cas optimal, la performance en temps de calcul d'un programme exécuté en parallèle doit être proportionnel avec le nombre d'unités de calcul exploitées. Ainsi avec un processeur quad-core, on est censé diviser le temps de calcul global par quatre. Cependant dans la pratique, il est extrêmement rare d'atteindre ce taux de performance en parallélisant l'intégralité du programme.

Dans la pratique, la quasi majorité des algorithmes contient des parties parallélisables et d'autres non- parallélisables (qui ne s'exécutent que séquentiellement). Le taux de gain de performance possible à atteindre avec la parallélisation correspond au ratio du temps d'exécution parallèle par rapport au temps de calcul séquentiel.

Ce gain de performance est défini par la loi de Amdahl (Amdahl, 1967) qui stipule que la réduction du temps d'exécution global d'un programme est limitée par le taux des parties non parallélisables. Par exemple si un algorithme peut être parallélisé à 50%, le taux de gain théorique qu'on peut obtenir est 2, si l'algorithme peut être parallélisé à 90% le taux sera de 10.

La loi est définie par l'équation suivante :

$$\text{Accélération} = \frac{1}{f + \frac{1-f}{n}}$$

- Où  $f \in [0,1]$  est la fraction du programme non parallélisable.

- N le nombre d'unité de calcul allouées.

Le graphe ci-dessous (Fig 72) montre le taux de gain obtenu par rapport au taux de parallélisation de l'algorithme et le nombre d'unités de calcul utilisées.

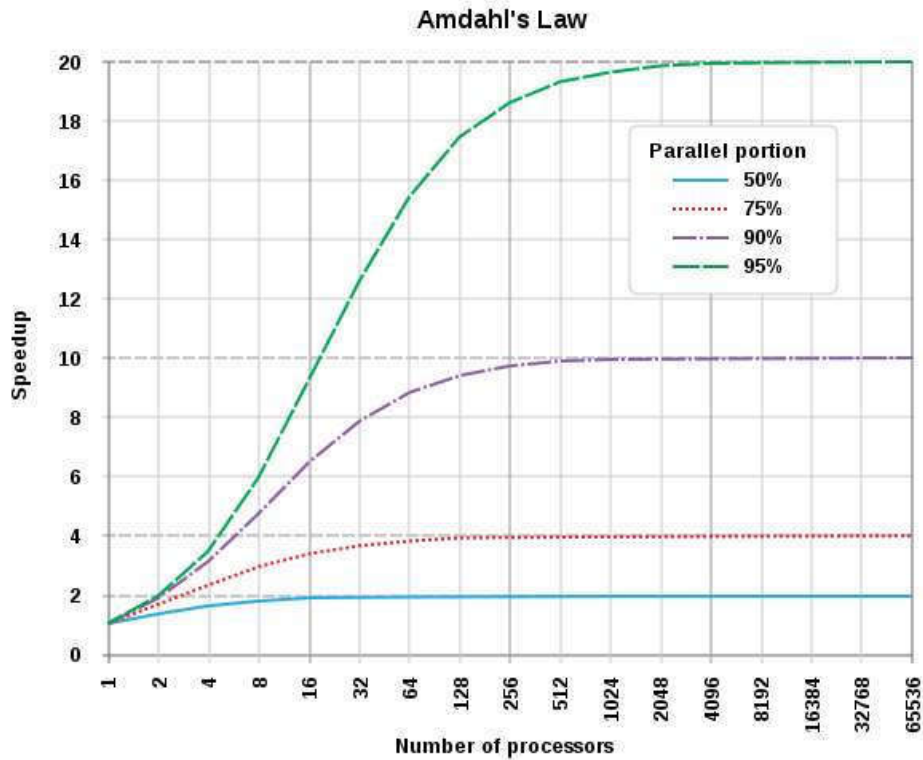


Figure 72. Figure démonstrative de la loi de Amdahl.

### VI.2.4 Implémentation

Dans un premier temps, on a implémenté notre approche avec Matlab, pour la partie d'analyse topologique. Afin de calculer la persistance homologique, on a utilisé la bibliothèque javaplex (Tausz, Vejdemo-Johansson, & Adams, 2014), écrite en java.

Le temps d'exécution du code écrit avec Matlab et java reste assez lent, surtout avec l'exécution du code de la bibliothèque en langage java, qui nécessite différentes étapes pour exécuter le code, comme le montre le schéma de la figure 73.

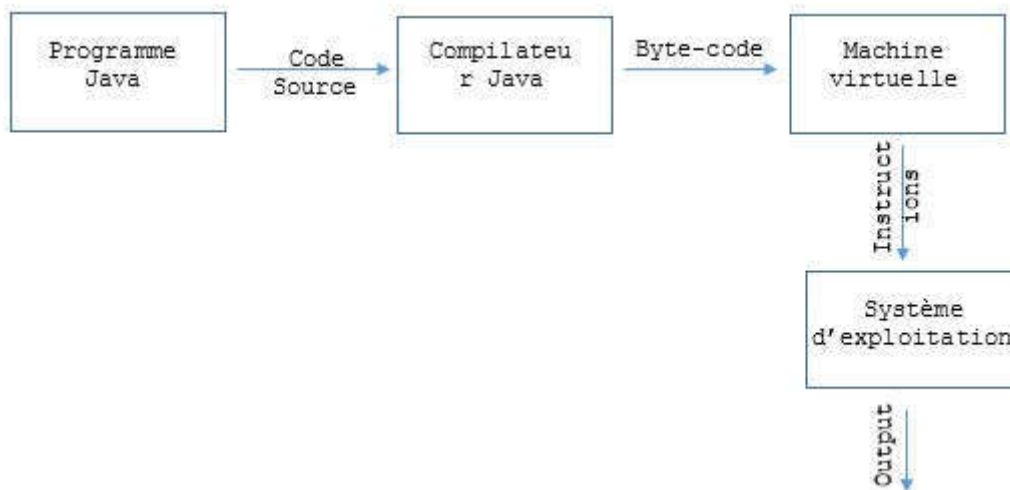


Figure 73. Les différentes étapes d'exécution d'un code écrit en Java.

Afin de réduire le temps d'exécution, on a choisi de réécrire notre approche en utilisant le package TDA (Topological Data Analysis) (Brittany Terese Fasy, Jisu Kim, Fabrizio Lecci, & Clément Maria, 2015), qui contient la bibliothèque d'analyse topologique Dionysus (MOROZOV, 2007), écrite en C++. Le choix de combiner du langage R avec du C++ permet de profiter de la rapidité d'exécution du C++ et des fonctionnalités natives du R pour gérer le parallélisme ainsi que sa capacité d'analyse des données avec différents packages.

#### VI.2.4.1 Calcul parallèle avec R sur machine multicœurs

Dans notre cas, on a utilisé un processeur i7 960 à quatre cœurs cadencés à 3.20GHz. Les étapes de l'exécution parallèle en R avec le package parallèle (Package 'parallel', 2017) sont définies comme suit (Miele & Louvet, 2016):

1. Un processus père est lancé. Cela sera le processus master qui synchronisera les trois autres unités de calcul filles qui vont être lancées.
2. L'unité de calcul « master » a la responsabilité de répartir les tâches entre toutes les unités de calcul.
3. La même unité principale « master » a la charge de collecter les résultats retournés par les différentes unités de calcul.
4. Retourner le résultat global et arrêter les unités de calcul.

Le package parallel gère automatiquement la mobilisation des unités de calcul et la communication entre elles. Plusieurs fonctions dans

## Parallélisme

ce package permettent d'accélérer l'exécution sur les tableaux et les matrices de données, en utilisant les fonctions de type apply (NSAUNDERS, 2010) (parApply, parLapply, parSapply, parVapply), ce qui permet d'appliquer la fonction de calcul sur toutes les cellules de la ligne de manière simultanée. Ces dernières fonctions sont combinées avec la boucle foreach (Steve Weston & Rich Calaway, 2017), qui est une version améliorée de la boucle «for», étant donné que la réalisation de boucles prend beaucoup de temps lors de l'exécution. La fonction «foreach» permet l'exécution de la boucle parallèlement.

Après, on fait appel à la bibliothèque parallel. On doit déterminer le nombre d'unité de calcul qu'on souhaite mobiliser pour l'exécution avec la fonction makeCluster. Leur nombre doit respecter les capacités matérielles de la machine :

```
library(parallel)
cl <- makeCluster(<taille du pool>)
parLapply(cl, <même arguments que lapply>)
stopCluster(cl)
```

On utilise aussi le package Rcpp (Dirk Eddelbuettel, et al., 2018), qui joue le rôle d'interface entre R et C++. Cela offre la possibilité de faire appel à des fonctions ou du code compilé ou écrit en C++, qui est plus rapide lors de l'exécution. Pour la partie d'analyse topologique, on a utilisé le package TDA (Topological Data Analysis) (Brittany Terese Fasy, Jisu Kim, Fabrizio Lecci, & Clément Maria, 2015) qui contient la bibliothèque Dionysus (MOROZOV, 2007) écrite en C++ pour calculer la signature topologique. La fonction de calcul de la bibliothèque a été combinée avec la boucle de type Foreach (Steve Weston & Rich Calaway, 2017) afin de déterminer les signatures topologiques des postures du mouvement, d'une manière simultanée au lieu d'une exécution séquentielle.

Après le calcul des signatures topologiques, on calcule les distances entre ces dernières avec les métriques détaillées dans la section (La distance de Wasserstein et La distance de Bottleneck), pour former la matrice de distances sur laquelle on applique une réduction dimensionnelle (voir section Réduction dimensionnelle).

Pour la partie classification, les résultats obtenus confirment la capacité de l'algorithme de machine à vecteurs de supports à obtenir de meilleurs résultats par rapport aux réseaux de neurones. Le package parallelSVM (Wannes Rosiers, Package 'parallelSVM', 2015) introduit en 2015, est une implémentation de l'algorithme de classification machine à vecteurs de support qui permet son exécution en parallèle.

Les opérations de calcul de persistance homologique (signature topologique) pour toutes les postures de chaque mouvement et le calcul de distance entre les postures des mouvements à classer, sont les

parties les plus gourmandes en temps de calcul, à cause des différentes étapes de filtration (voir Filtration d'un complexe simplicial), et le calcul de distance.

Un échantillonnage à une fréquence de 100 **Hz** est réalisé, générant un grand nombre de postures pour chaque mouvement enregistré et induisant un nombre exponentiel d'itérations. Dans le but de réduire ce temps d'exécution, on a optimisé le code afin d'exécuter les boucles en parallèle.

### **VI.2.5 Résultats**

Les résultats de précision restent identiques à ceux de la première démarche étant donné qu'on a utilisé la même approche. Dans cette section, on va comparer les résultats obtenus en temps d'exécution, des différentes démarches exécutées dans les chapitres précédents :

1. Les temps d'exécution de la démarche proposée dans le chapitre numéro 4, (Figure 38. Les étapes de la démarche proposée.) (analyse topologique via la persistance homologique, et une classification avec la machine à vecteurs de supports). Le code de la démarche est écrit en Matlab et une bibliothèque d'analyse topologique javaplex (Tausz, Vejdemo-Johansson, & Adams, 2014) écrite en java.
2. Le temps d'exécution de l'approche qui se base sur l'utilisation des réseaux de neurones lors de la classification, démarche détaillée dans la section : Démarche avec réseaux de neurones.
3. Le temps d'exécution après la réécriture du code de la première démarche, en R et C++ afin d'utiliser le parallélisme.

On commence par comparer les temps d'exécutions de la seconde expérience détaillée dans (Fig 50), qui consiste à classer des mouvements de pointage impliquant tout le corps vers neuf cibles. On effectue une comparaison pour les deux types de classification binaire et multi-classes. Les résultats sont détaillés respectivement dans les figures (Fig 74) et (Fig 75).

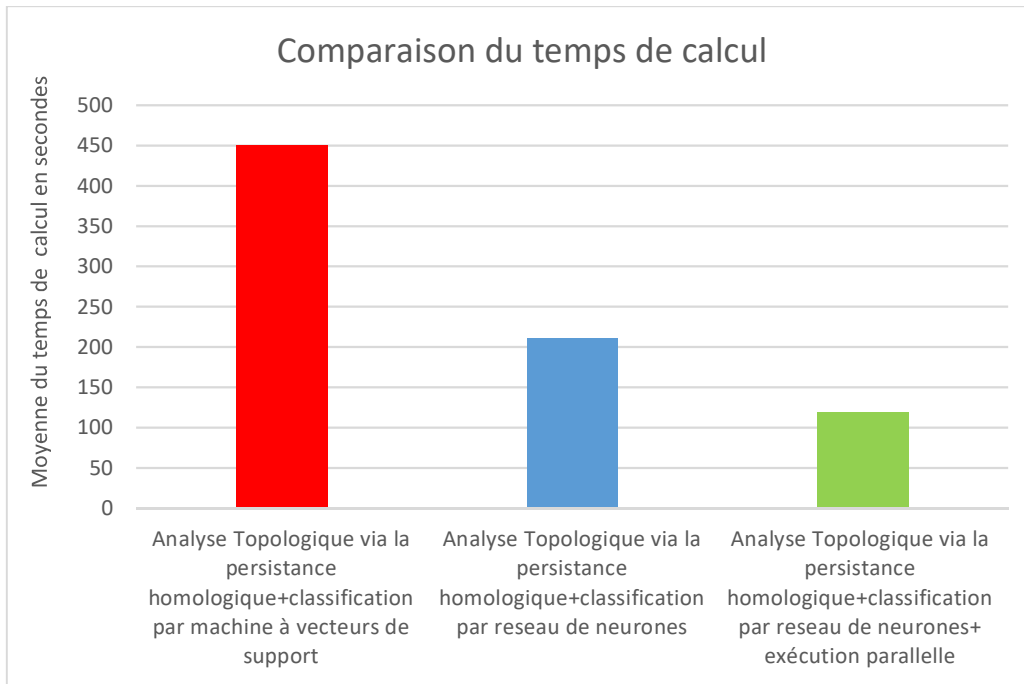


Figure 74. Comparaison entre le temps de calcul obtenu après l'exécution parallèle de la démarche (en vert) et celui des exécution séquentielle, lors de la classification binaire de la deuxième expérience.

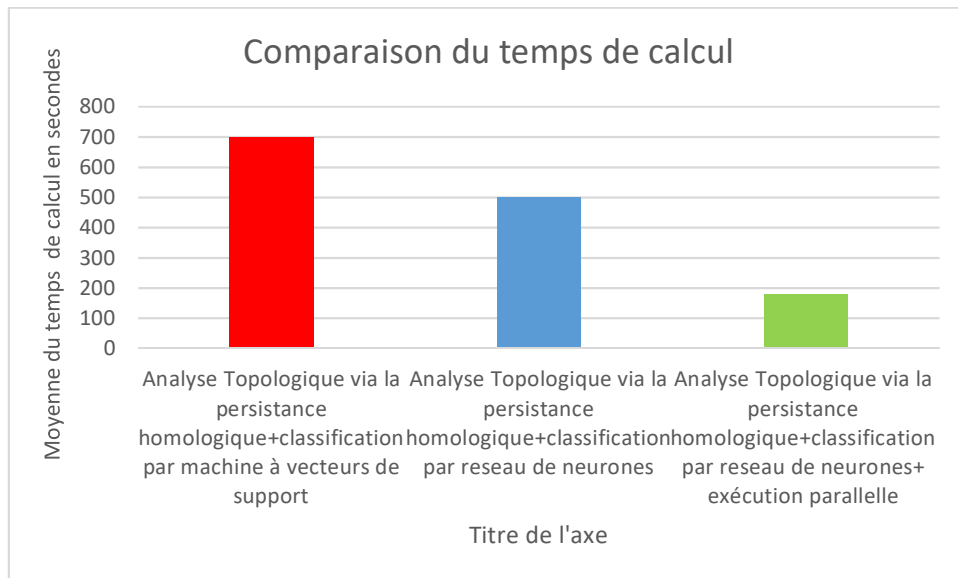


Figure 75. Comparaison entre le temps de calcul obtenu après l'exécution parallèle de la démarche (en vert) et celui des exécution séquentielle, lors de la classification multi-classe de la deuxième expérience.

On peut remarquer que l'utilisation du parallélisme a permis de diviser le temps d'exécution par rapport à la première démarche qui utilise les mêmes méthodes par quatre. Le temps d'exécution est passé de 450 secondes à 118 secondes pour la classification binaire. Pour la



## Parallélisme

classification multi-classes, le temps d'exécution est passé de 700 secondes à 180 secondes.

Les résultats obtenus (Fig 74) et (Fig 75) montrent le gain significatif en temps de calcul qu'on peut obtenir en passant d'un programme d'analyse de données écrit en Matlab et exécuté en séquentiel à un programme écrit en C++ et R et exécuté en parallèle.

L'utilisation du package `parallelSVM` (Wannes Rosiers, Package '`parallelSVM`', 2015) a permis de tirer parti des capacités de l'algorithme de classification supervisée (machine à vecteurs de support), qui a montré qu'elle est la méthode de classification la plus adaptée pour ce type de classification (Tab 13,14), tout en réduisant son temps d'exécution.

En optimisant le code, on a obtenu un programme qui est à  $\cong 90\%$  parallélisable. Le temps d'exécution obtenu avec la parallélisation est conforme à la loi de Amdahl (Amdahl, 1967). En utilisant cette même loi, on peut estimer le temps d'exécution qu'on peut obtenir si on dispose de plus d'unités de calcul (Fig 76).



Figure 76. Estimation de temps de calcul par rapport au nombre d'unités de calcul utilisées, selon la loi de Amdahl.

Dans notre travail, on a proposé une démarche qui consiste à appliquer une analyse topologique via la persistance homologique avant d'effectuer une classification avec l'algorithme de machine à vecteurs de supports. Cette démarche a permis d'améliorer considérablement les résultats de la classification surtout dans le cas d'une classification inter-mouvement (Tab 6,8,9,10,11). Mais cette approche est coûteuse en temps de calcul. On a essayé de réduire le temps de calcul en analysant la complexité de calcul des différentes étapes de

la démarche afin d'identifier les plus coûteuses en temps de calcul et celles qui nécessitent le plus d'itérations. Après la détermination des étapes les plus coûteuses en temps de calcul, on a proposé une démarche basée sur l'utilisation des réseaux de neurones. Cela nous a permis de réduire le temps de calcul mais au détriment de la précision de la classification des mouvements qui a perdu entre 5 et 10%. Pour pouvoir concilier le temps de calcul avec la précision de la classification, une réécriture du code de la première démarche a été réalisée afin qu'il soit exécutable en parallèle. Ce mode d'exécution nous a permis de conserver un bon taux de précision grâce à la démarche qui utilise la machine à vecteurs de support tout en réduisant le temps de calcul.

Après la résolution du premier verrou, la démarche proposée a permis d'avoir un très bon taux de précision lors de la classification en utilisant les réseaux de neurones et une exécution parallèle du code.

On a réussi à réduire considérablement le temps de calcul. Dans le chapitre suivant, on va s'intéresser au deuxième verrou qui consiste à proposer une approche pour apprendre à prédire la suite d'un mouvement à partir d'un intervalle d'enregistrement.

# Chapitre VII Anticipation de mouvement

## VII.1 Réseaux de neurones récurrents

### VII.1.1 Introduction

Le but de ce chapitre est de proposer une approche dont le but est de déterminer la suite d'un mouvement  $M$  parmi un ensemble de mouvements à partir d'un intervalle d'enregistrement (Fig 77).

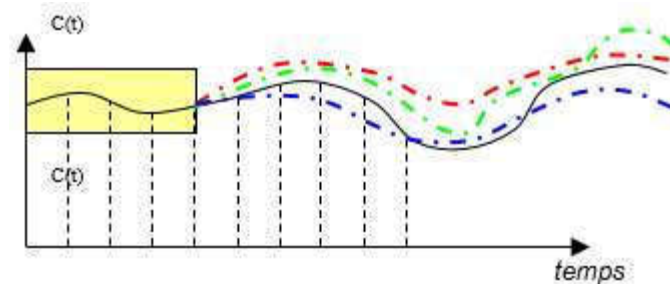


Figure 77. Déterminer la suite d'un mouvement à partir d'un intervalle d'enregistrement.

La possibilité de prédire la suite d'un mouvement peut avoir plusieurs finalités applicatives. Dans le domaine médical, l'approche peut être utilisée pour identifier les mouvements dangereux pour les patients ou les personnes âgées. Dans le domaine de la réalité virtuelle, la prédiction d'une suite de mouvement permettrait d'anticiper l'interaction entre l'avatar animé par le sujet et un environnement virtuel.

Comme défini par les travaux de Etienne-Jules Marey ( Marey, 1894), le mouvement est une suite de postures. En s'appuyant sur ce principe, nous allons utiliser un type spécifique de réseaux de neurones dont le but est d'apprendre à prédire la suite d'une séquence de postures. Il s'agit d'abord de construire une base de connaissances qui contient les données d'un ensemble de mouvements. Ensuite, les réseaux de neurones vont appliquer un apprentissage pour pouvoir les reconnaître et déterminer la suite à partir d'un intervalle d'enregistrement de mouvements (Fig 77).

On commence par présenter le principe des réseaux de neurones récurrents en exposant différents travaux de la littérature

scientifique. Ensuite, nous détaillerons les étapes de notre démarche de prédiction et terminerons par une analyse des résultats obtenus.

### VII.1.2 Réseaux de neurones récurrents

La prédiction d'une séquence est un problème de dépendance temporelle. Les chaînes de Markov se basent sur la propriété de Markov (Brooks, Gelman, Jones, & Meng, 2011) (Fig 78), qui stipule que l'information nécessaire à la prédiction d'une observation dans une séquence temporelle est contenue dans les états précédents de cette dernière.

Dans le cas où la prédiction de chaque observation dépend d'un ensemble important d'observations, cela peut vite conduire à des problèmes de complexité de calculs lors de l'apprentissage, d'espace de stockage nécessaire qui quadruple avec le nombre d'états définis pour représenter le système ainsi qu'un problème de généralisation. Pour surmonter ces limitations Jordan (JORDAN M. I, J. L. ELMAN, & D. E. RUMELHART, 1989) et Elman (ELMAN J. L, 1990) proposent une variante des réseaux de neurones classiques nommée les réseaux de neurones récurrents, afin d'offrir un outil plus puissant que les HMMs pour modéliser un mécanisme de prédiction pour les séries temporelles.

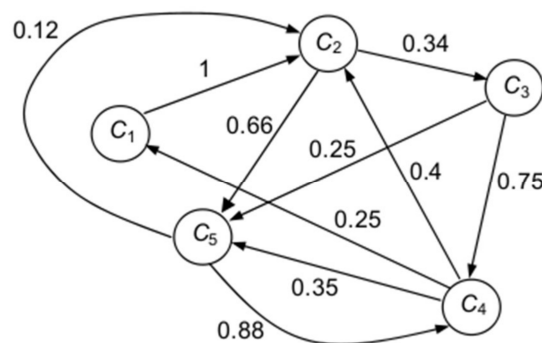


Figure 78. Graphe détaillant une chaîne de Markov, les arêtes pondérées représentent les probabilités de passer d'un état vers un autre.

Le principe des réseaux de neurones récurrents, est que le calcul de la sortie  $O(n)$  dépend de l'entrée  $n$  et de la sortie de l'élément précédent  $O(n-1)$  de la séquence. Pour cela, l'architecture de ce type de réseau de neurones est caractérisée par la présence d'une couche contexte qui sauvegarde le résultat de calcul de l'entrée précédente ( $n-1$ ) à partir de la couche cachée et l'injecte dans la couche d'entrée en formant une boucle (Fig 79). Cette connexion permet de prendre en compte à l'étape courante une information contextuelle prédite dans une étape précédente.

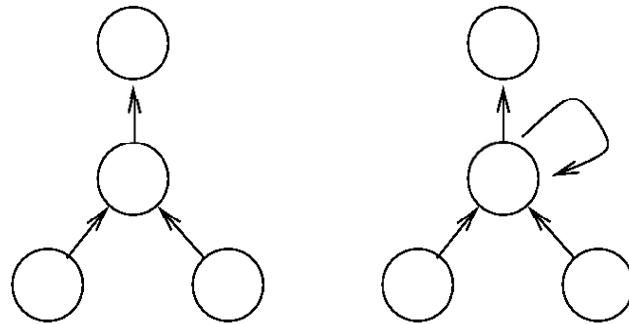


Figure 79. (a) architecture d'un réseau de neurones classique, (b) une architecture d'un réseau de neurones récurrent proposé par Elman.

L'architecture d'Elman combine un flux d'information forward pour une information qui va de la couche d'entrée à la couche de sortie, et un flux backward avec l'information de la couche cachée de l'étape précédant  $T-1$  qui est retro-propagée dans la couche d'entrée à l'étape  $T$ . De cette façon, la couche de sortie est activée par la couche cachée qui est elle-même activée par la couche d'entrée et la couche contexte. La sortie obtenue est comparée avec la cible souhaitée (apprentissage supervisé). En cas d'erreur, les poids des connexions seront ajustés avec l'algorithme du rétro-propagation du gradient (voir section Introduction aux réseaux de neurones). Les poids des connexions de la couche contexte sont fixés à 1 et ne sont pas ajustés afin qu'à l'étape  $T+1$ , la couche de contexte soit identique à la couche cachée de l'étape  $T$ , ce qui donne aux réseaux de neurones cette caractéristique de mémoire.

On peut détailler le pseudo algorithme d'apprentissage des réseaux de neurones récurrent, avec la séquences des étapes suivantes :

- Etape 1 : initialiser l'activation de la couche contexte à 0.5.
- Etape 2 : répéter les étapes 3 à 7 jusqu'à l'arrivée au symbole d'arrêt
- Etape 3 : sélectionner l'élément  $T_i$  de l'ensemble d'apprentissage.
- Etape 4 : présenter l'élément  $T_{i+1}$  de l'ensemble d'apprentissage comme étant la cible souhaitée.
- Etape 5 : calculer la sortie de l'élément  $T_i$ .
- Etape 6 : déterminer l'erreur entre la sortie obtenue et la sortie souhaitée, ajuster les poids synaptiques en utilisant l'algorithme de rétro-propagation du gradient.
- Etape 7 : copier les valeurs de la couche cachée dans la couche contexte.
- Etape 8 : vérifier si l'élément final de la séquence est atteint. Si oui : arrêter l'algorithme. Sinon : copier la couche cachée dans le couche contexte et passer à l'étape 3.

La figure (Fig 80) représente l'architecture d'un réseau de neurones récurrent avec un potentiel d'apprendre un nombre illimité de séquences de longueurs variables. Le but initial de la conception d'Elman est que le réseau soit en mesure de prendre des lettres en entrée et d'apprendre à prédire la prochaine lettre dans un mot ou une phrase.

Classic experiment on language acquisition and processing (Elman, 1990)

- **Task**
  - Elman net to predict successive words in sentences.
  - **Data**
  - Suite of sentences, e.g.
    - "The boy catches the ball."
    - "The girl eats an apple."
  - Words are input one at a time
    - **Representation**
  - Binary representation for each word, e.g.
    - 0-1-0-0-0 for "girl"
    - **Training method**
  - Backpropagation

Pour dérouler l'algorithme avec un exemple, on pose ( $Y_0, Y_1, Y_2, \dots$ ) un ensemble de caractères parmi lesquels on forme des séquences que le réseau va apprendre, comme la séquence : 0,3,5,2,0, où 0 correspond

## Anticipation de mouvement

au caractère  $Y_0$ , 3 au  $Y_3$  ...etc. Chaque séquence se termine par un symbole d'arrêt.

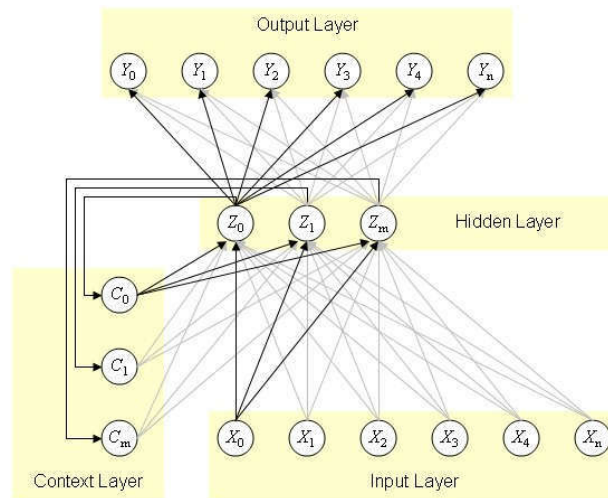


Figure 80. Un réseau de neurones récurrent dont le but est d'apprendre une séquence.

On a présenté le principe des réseaux de neurones récurrents, dont le but initial était les applications de traitement automatique du langage naturel. Mais la méthode est de plus en plus utilisée pour la prédiction dans plusieurs domaines tels que la prédiction dans les marchés financiers ou la modélisation des signaux issus de capteurs physiques.

### VII.1.3 Travaux basés sur l'utilisation des réseaux de neurones récurrents

Dans cette section, on va présenter quelques travaux de la littérature scientifique qui traitent de l'utilisation des réseaux de neurones récurrents dans différents domaines.

Psarrou et al (Alexandra Psarrou, Shaogang Gong, & Hilary Buxton, 1995) utilisent les réseaux de neurones récurrents pour suivre les trajectoires en 2D, l'approche est utilisée pour développer une application de vision par ordinateur dont le but est de suivre les trajectoires faciales (Fig 81).

## Anticipation de mouvement



Figure 81. (à gauche) L'utilisation des réseaux de neurones récurrents, pour suivre le mouvement du visage humain, sur des trajectoires en 2D. (à droite) la reconnaissance du mouvement humain sur un plan 2D.

Dans le travail de Henry Tan et al (Henry C. C. Tan, Kui Jia, & Liyanage C. De Silva, 2003), les réseaux de neurones sont utilisés pour apprendre à reconnaître dix activités humaines dans un environnement de travail au bureau (Fig 82). L'acquisition des mouvements était réalisée avec une caméra, les enregistrements étaient convertis en images en deux dimensions avant de les utiliser comme entrée d'apprentissage aux réseaux de neurones récurrents (Fig 80).



Figure 82. Traitement des enregistrements vidéo en images qui extraient la silhouette de la personne en mouvement.

Avec l'utilisation des énergies renouvelables qui est en pleine expansion, Hossein et al (Hossein Lotfi, Mohammad baghernaghibi, Toktam Lotfi, & Sadegh Hesar, 2014) se sont intéressés à l'utilisation de réseaux de neurones selon l'architecture d'Elman (Fig 80) pour prédire la quantité d'énergie éolienne (Fig 83), en s'appuyant sur des enregistrements météorologiques.



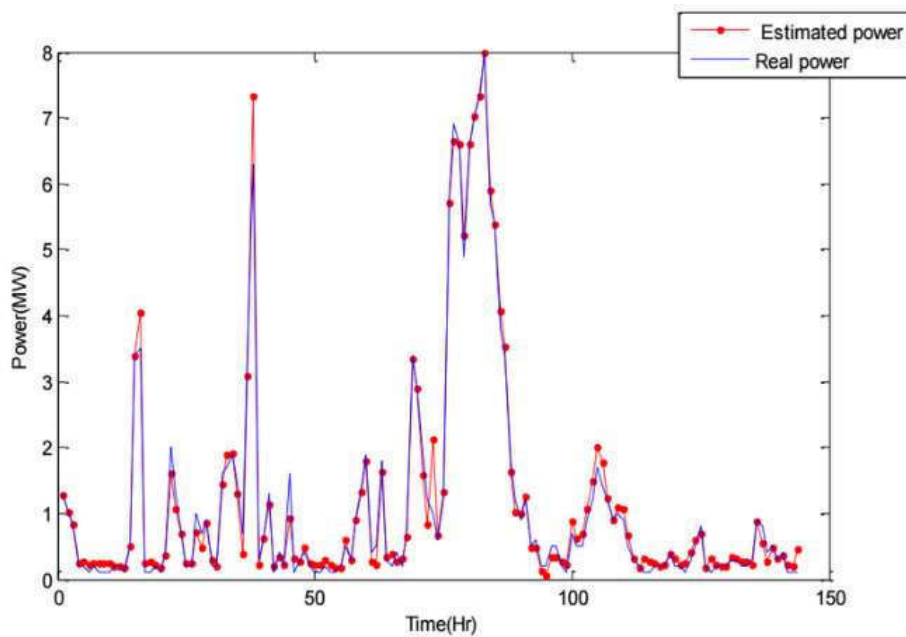


Figure 83. La quantité réelle d'énergie produite, par rapport à la quantité prédite par les réseaux de neurones récurrents.

Pour améliorer les systèmes d'aide aux personnes âgées dans un environnement doté de capteurs (maison intelligente), Liouane et al (Zaineb Liouane, Tayeb Lemlouma, Philippe Roose, Frédéric Weis, & Hassani Messaoud, 2016) proposent d'utiliser les réseaux de neurones récurrents d'Elman (ELMAN J. L, 1990) afin de reconnaître les activités des personnes et prédire les situations anormales qui peuvent présenter un danger. Pour l'apprentissage, ils ont utilisé la base d'apprentissage issue du projet « e-health monitoring open data project (Haider Mshali, Tayeb Lemlouma, & Damien Magoni, 2015) », où les données décrivent des scénarios d'activités journalières dans une maison intelligente pendant une année. Chaque activité utilisée comme entrée d'apprentissage est codifiée avec trois paramètres : l'instant du début, l'instant de fin ainsi qu'un code correspondant à l'activité. La table (Tab 12) détaille les activités étudiées :

<b>Codes</b>	<b>Activities</b>
1	Eating
2	Dressing
3	washing
{6 -4-5-34 }	grooming
7	Unary-function
8	Bowel-function
9	Toileting
10	House-keeping
11	Landry
{13-14-15-16}	Meal preparation
17	Telephone
18	Medication Use
19	Watching TV
24	Sleeping
33	Go out
35	Reading

Tableau 12. Les activités enregistrées dans le cadre du projet : e-health monitoring open data project.

Un autre domaine où les réseaux de neurones récurrents sont de plus en plus utilisés est la finance. Kendi et al (Salima KENDI, Fodil LAIB, & Mohammed Said RADJEF, 2013) proposent un système de négociation des prix sur un marché, basé sur les réseaux de neurones récurrents, qui permettent au système d'apprendre à émettre des ordres d'achat et de vente en réagissant aux flux de prévisions de l'offre et de la demande (Fig 84).

## Anticipation de mouvement

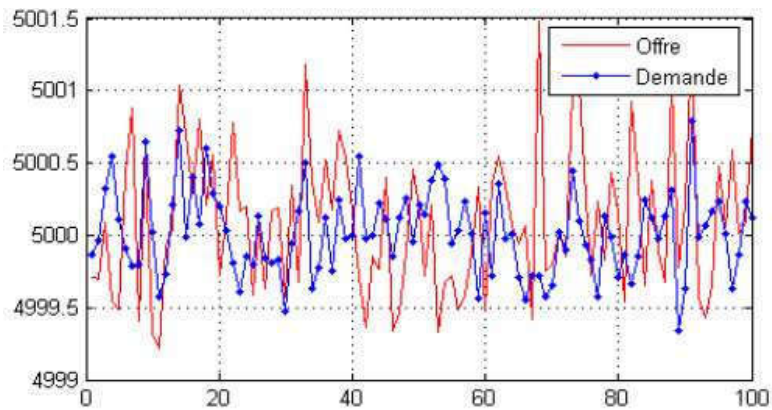


Figure 84. Prédiction de l'offre et de la demande en utilisant les réseaux de neurones récurrents.

Dans les travaux récents de Wang et al (Wei Huang, Shouyang Wang, Lean Yu, Yukun Bao, & Lin Wang, 2006), les réseaux de neurones récurrents apprennent à prédire les cours des différents indices boursiers (Nikkei225, KOSPI, SSE Composite Index, TWSE) en utilisant leur historique comme base d'apprentissage (Fig 85).

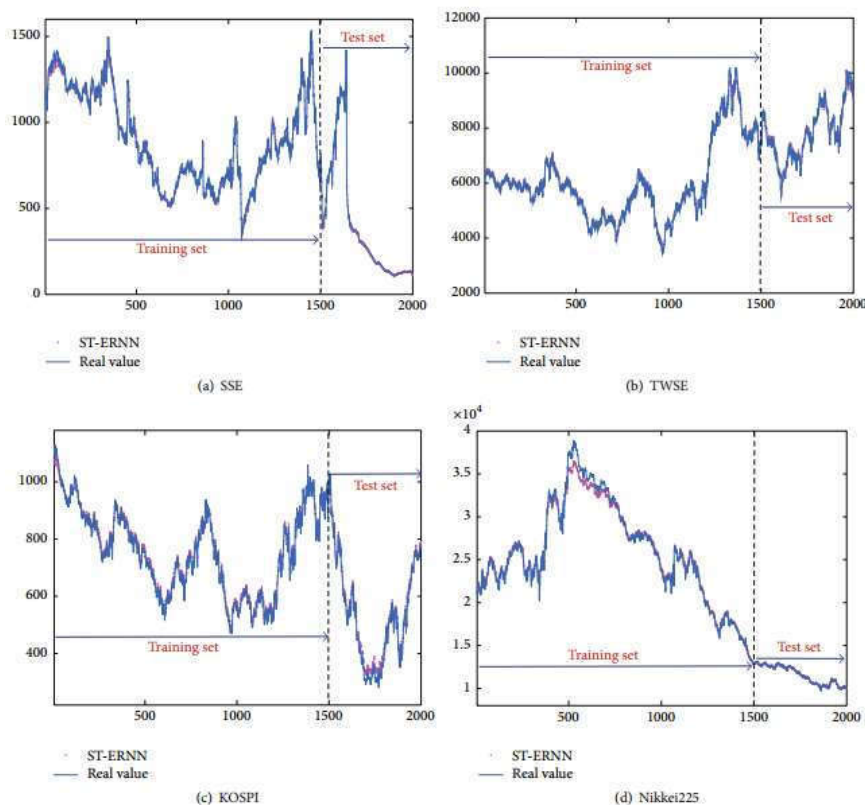


Figure 85. Prédiction des cours de différents index boursiers, en utilisant les réseaux de neurones récurrents.

#### **VII.1.4 Application sur la prédiction du mouvement**

Le chapitre III détaille la démarche proposée. La figure (Fig 55) montre que la classification se fait sur les nuages des points retournés par la réduction dimensionnelle de la matrice de distances entre les signatures topologiques des mouvements. Chaque nuage de points représente la distribution des signatures topologiques, dont chacune d'elles correspond à une posture du mouvement.

Etant donné que dans notre démarche, le mouvement est une suite temporelle de postures codifiées par des signatures topologiques, et que ces dernières sont projetées dans un espace réduit (2D ou 3D), l'idée est d'utiliser les réseaux de neurones récurrents pour apprendre l'ordre dans lequel les coordonnées qui correspondent aux postures se succèdent. Une fois l'apprentissage effectué, l'objectif est de pouvoir déterminer la suite d'un mouvement  $M$  qui appartient à l'ensemble des mouvements sur lesquels on a appliqué l'apprentissage, à partir d'un intervalle d'enregistrement (Fig 77).

Un réseau de neurones récurrents sera mis en œuvre pour l'apprentissage d'une série temporelle, dans le cas où la matrice de distance à partir de laquelle on effectue la réduction dimensionnelle est inter-mouvements (regroupement différents mouvements). Nous proposons une solution qui combine deux types de réseaux de neurones : le premier est un réseau de neurones multicouche avec un algorithme de rétropropagation du gradient (détaillé dans la section V.6) dans le but de reconnaître à quel mouvement le sous-ensemble de postures appartient. Une fois qu'on détermine le mouvement auquel appartient l'intervalle, on sélectionne le réseau de neurones récurrent (le deuxième type des réseaux de neurones utilisé) correspondant au mouvement afin de déterminer la suite de ce dernier. Pour chaque mouvement, on connaît les coordonnées qui correspondent à la dernière posture du mouvement, afin de les utiliser comme condition d'arrêt à l'algorithme de prédiction, la figure (Fig 86) détaille le fonctionnement de l'approche.

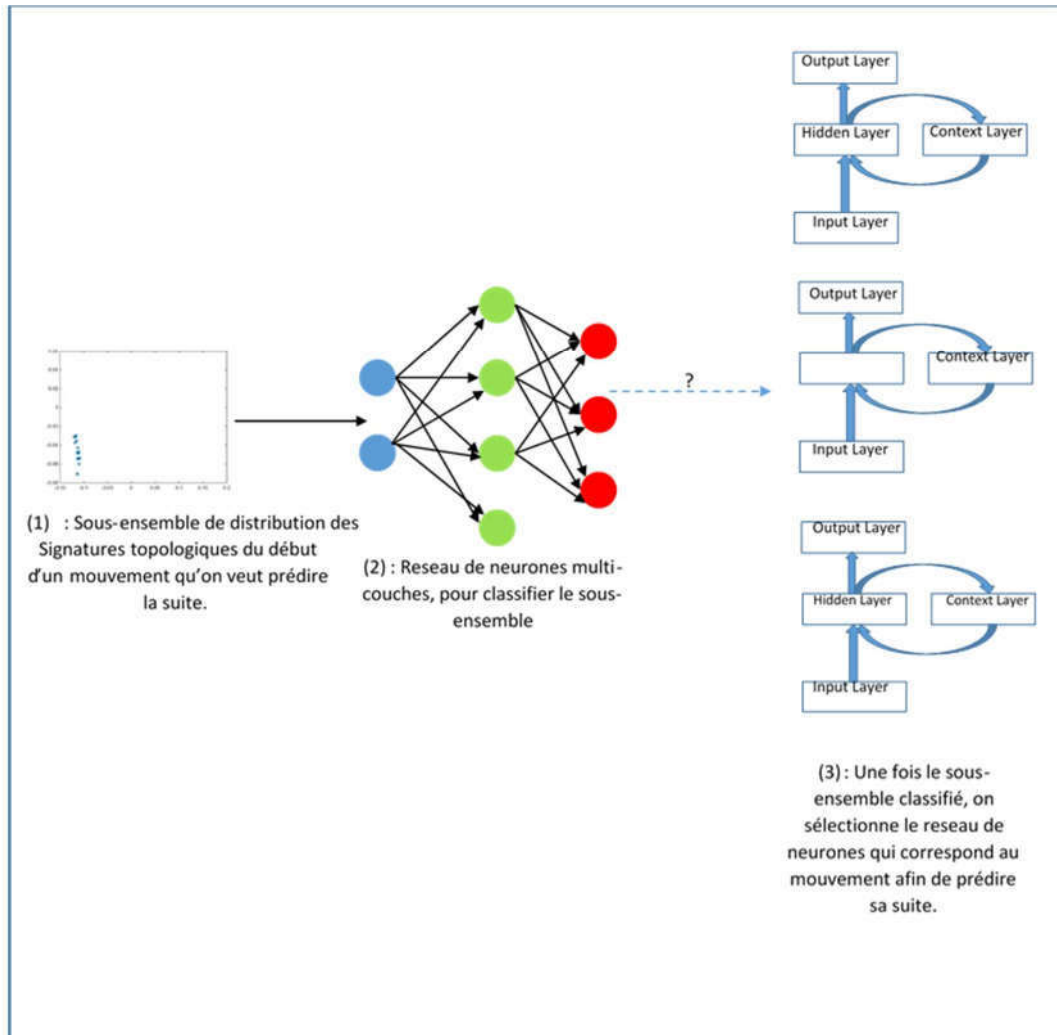


Figure 86. Les étapes de la démarche proposée, pour déterminer la suite d'un mouvement à partir d'une partie d'enregistrement.

#### **VII.1.4.1 Résultats**

Nous souhaitons tester la capacité des réseaux de neurones récurrents à prédire la suite d'un mouvement (suite des postures) à partir d'un intervalle d'enregistrement (10% de l'enregistrement total du mouvement). Comme détaillé précédemment, l'algorithme des réseaux de neurones récurrents suit une exécution itérative. Ainsi, dans la première itération, l'objectif est de prédire la posture qui vient après l'ensemble des 10% d'enregistrement initial et continuer l'itération jusqu'à la condition d'arrêt.

Pour cette expérimentation, nous avons utilisé les données issues de l'acquisition des mouvements détaillée dans le chapitre *Partie expérimentale* pour analyser la démarche.

Les mouvements de pointage vers une seule cible, impliquant la partie haute du corps (Fig 44) à partir de trois positions de départ différentes, sont notés :

- Mouvement-Bras-P1
- Mouvement-Bras-P2
- Mouvement-Bras-P3

Deux mouvements de pointage impliquant tout le corps (whole-body pointing), vers plusieurs cibles (Fig 49) qu'on note :

- M-Corp-Seq1 : pour la première séquence.
- M-Corp-Seq2 : pour la seconde séquence.
- M-Corp-Seq3 : pour la seconde séquence.
- M-Corp-Seq4 : pour la seconde séquence.

Afin d'enrichir notre base d'apprentissage (nombre de mouvements étudiés) et d'approfondir notre analyse, on a ajouté des données issues de la base de mouvements du laboratoire graphique de l'université de Carnegie Mellon (CMU Graphics Lab Motion Capture Database , <http://mocap.cs.cmu.edu/search.php>). Les mouvements sont enregistrés par un système identique à celui que l'on a utilisé : le système de capture de mouvements (Vicon motion capture) (Vicon Motion Systems, 2015).

Les mouvements ajoutés sont : un mouvement de marche, un mouvement de course et un mouvement de danse. Les mouvements ajoutés sont de nature cyclique contrairement aux deux mouvements de pointage qui sont des mouvements discontinus (Fig 87).

Les résultats montrent une classification sans erreurs, cela est dû à la nature cyclique des mouvements ce qui facilite l'apprentissage et la capacité de généralisation.

## Anticipation de mouvement

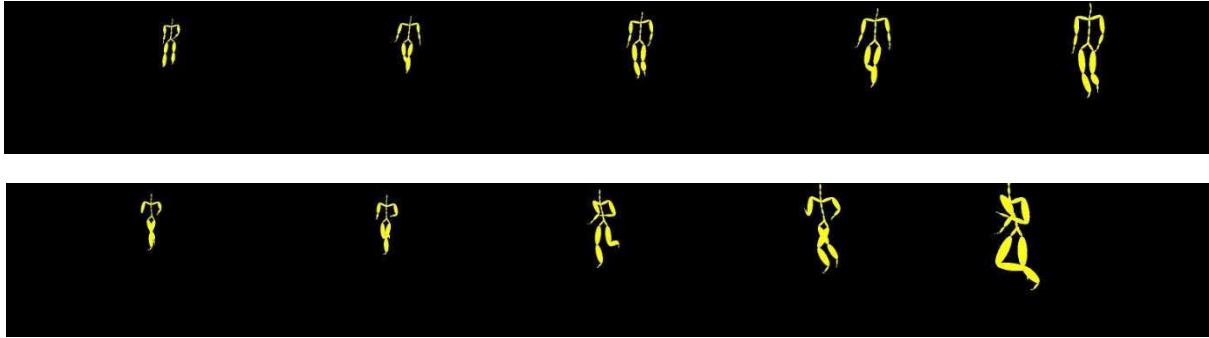


Figure 87. Les mouvements de marche (en haut) et de courses (en bas) ajoutés depuis la base de mouvement de l'Université de Carnegie Mellon.

Etant donné qu'on travaille dans la prédiction des coordonnées, le nombre des neurones de la couche d'entrée et ceux de la couche de sortie sera égale à la dimension choisie (2D ou 3D dans notre cas). Pour la couche cachée et connexe, on exécute un test de validation en utilisant les mouvements de pointages (Fig 44) et (Fig 49). Cela permet d'étudier l'impact du nombre des neurones dans la couche cachée sur la précision retournée (Fig 88).

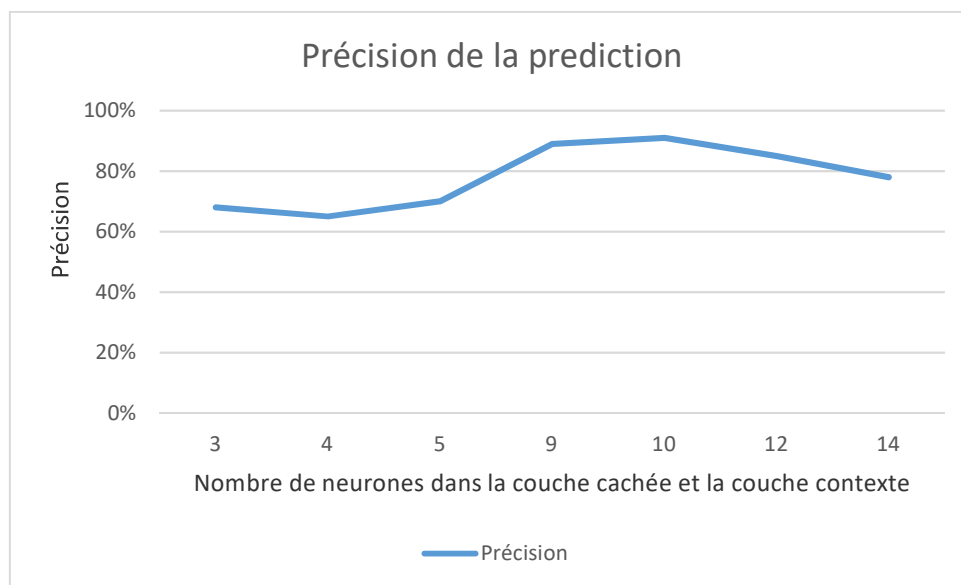


Figure 88. Précision du système à déterminer la suite d'un mouvement, par rapport au nombre de neurones de la couche cachée et la couche contexte.

Le tableau ci-dessus résume les résultats obtenus pour chacun des mouvements étudiés (Tab 13).

## Anticipation de mouvement

Mouvement analysé	Nombre optimal de neurones dans la couche cachée	Précision de la prédiction
Mouvement-Bras-P1	8	85%
Mouvement -Bras-P2	8	87,4%
Mouvement -Bras-P3	8	90%
Mouvement -Corp-Seq1	10	78%
Mouvement -Corp-Seq2	10	82%
Mouvement -courir	5	96.8%
Mouvement -marcher	6	95%
Mouvement-dance	9	81,56%

Tableau 13. Les précisions de détermination de la suite des différents mouvements étudiés.

Les résultats obtenus, dont la plus faible précision est 78%, confirment la capacité de la démarche basée sur les réseaux de neurones récurrents à apprendre à déterminer la suite d'un mouvement donné à partir d'un intervalle d'enregistrement égal à 10% de l'enregistrement total du mouvement.

A partir du tableau des résultats (Tab 13), on peut remarquer que les taux de précision les plus élevés sont ceux obtenus pour les mouvements cycliques (les mouvements de marche et de course). La nature cyclique de ces mouvements facilite grandement la prédiction de leur suite.



# Chapitre VIII Conclusion et perspectives

L'objectif de cette thèse était de proposer une approche qui permet d'améliorer la classification et la reconnaissance des mouvements humains, en ajoutant une étape d'analyse topologique des données enregistrées par un système de capture de mouvements. Pour cette étape d'analyse topologique, nous avons utilisé la persistance homologique introduite par Edelsbrunner et al ( Edelsbrunner, Letscher, & Zomorodian, 2002), qui est une méthode d'analyse des espaces topologiques représentés sous forme de nuages de points. Son but est de donner une signature topologique qui permet de distinguer un ensemble de données d'un autre.

Cette méthode d'analyse est de plus en plus utilisée dans le domaine d'analyse de données volumineuses (Collins, Zomorodian, & Carlsson, 2004) (Ben Cassidy, Caroline Rae, & Victor Solo, 2015) (Computing persistent features in big data: A distributed dimension reduction approach, 2014). Plusieurs travaux dont l'étude comparative réalisée par Andrea Mannini et Angelo Maria Sabatini (Andrea Mannini & Angelo Maria Sabatini, 2010), montrent que l'algorithme de la machine à vecteur de support, détaillé dans le chapitre III.1.11 est l'algorithme de classification le plus adapté à la classification des mouvements humains, de ce fait on a choisi cet algorithme pour la partie de classification de notre démarche, détaillée dans (Fig 38).

Les mouvements étudiés, sont des mouvements de pointage vers une ou plusieurs cibles (Fig 44) et (Fig 49). Le choix de ce type de mouvement a été proposé dans les travaux de Hilt & Pozzo (Hilt P. , 2015). Les mouvements de pointage sont des mouvements discrets non cycliques, ce qui rend leur classification plus difficile par rapport aux mouvements cycliques tels que les mouvements de marche ou de course.

Afin de valider notre démarche, nous avons réalisé une analyse comparative avec la démarche présentée dans la plupart des travaux,

## Conclusion et perspectives

qui consiste à utiliser les données enregistrées directement comme entrée aux algorithmes et méthodes de classification.

Nous avons effectué des comparaisons entre les deux types de classifications : binaire et multi-classes. Et pour chacune de ces classifications, nous avons réalisé différentes combinaisons entre inter et intra mouvements, et inter et intra participant, comme détaillé dans les tableaux (Tab 6,8,9,10,11).

En comparant les résultats obtenus entre l'approche de la classification directe, et celle proposée qui se base sur l'ajout d'une étape d'analyse topologique avant la classification, on note que pour la classification inter-mouvements, dans le cas d'une classification binaire d'un mouvement simple (pointage vers une seule cible) et enregistré avec quatre marqueurs (Fig 44), les deux approches obtiennent une bonne précision de classification (Tab 8). En revanche, dans le cas d'une classification multi-classes (plus que deux mouvements) dont l'enregistrement implique un important volume de données (20 marqueurs dans la deuxième expérience) (Fig 49), on remarque une nette baisse en précision de classification pour la démarche de classification directe, contrairement à la démarche proposée qui permet d'obtenir une bonne précision. Cela s'explique par la nature de l'analyse topologique appliquée, étant donné qu'il existe une corrélation entre la robustesse des signatures topologiques retournées par la persistance homologique et le nombre d'individus qui définissent le nuage de points.

Pour la classification intra-mouvement, dans le cas d'une classification binaire, étant donné que les deux classes appartiennent au même mouvement, la classification ne doit pas donner de différences entre les données : on doit obtenir un résultat proche de 50%. Avec l'approche classique (utilisation des données issues de la capture des mouvements directement comme entrée à l'algorithme de classification), les différences morphologiques entre les participants sont le principal facteur décisionnel lors de la classification (Fig 46 et Fig 50). Ce qui fait que, même si deux participants effectuent le même mouvement, à cause de leur différence morphologique, l'algorithme de classification n'est pas en mesure de reconnaître que les deux participants exécutent le même mouvement. Avec la démarche proposée les tableaux (Tab 6 et Tab 9) confirment la robustesse de l'approche face aux différences morphologiques.

Afin de tester la robustesse de l'approche face au bruit blanc, on a ajouté un intervalle de bruit blanc, dont on augmente l'amplitude à chaque étape jusqu'à atteindre un intervalle de 100mm (Fig 52). La précision de l'approche de classification directe commence à diminuer à partir de 10 mm de bruit blanc ajouté. Cette perte en précision continue rapidement avec l'augmentation de l'intervalle de bruit, jusqu'à atteindre 40% de précision avec un intervalle de 70mm. L'approche basée sur l'analyse topologique a apporté plus de robustesse. Sa précision de classification reste stable jusqu'à l'ajout d'un intervalle de 35mm de bruit.

Malgré la capacité de l'approche proposée à améliorer les résultats de la classification surtout pour celles qui impliquent un nombre important de données (mouvement complexe enregistré avec plusieurs

## Conclusion et perspectives

marqueurs), et sa robustesse face aux différences morphologiques entre les participants, ainsi que face aux bruits blancs, notre approche reste coûteuse en temps de calculs par rapport à l'approche de classification directe.

Afin de réduire le temps de calcul de notre démarche, on a analysé la complexité calculatoire des différentes étapes. On a effectué des modifications à notre approche dans le but d'éviter la partie la plus coûteuse en temps de calcul qui est la partie de calcul de la matrice de distances. Ces distances permettent d'évaluer les différences entre la signature topologique de chaque posture du mouvement à classifier et les autres signatures topologiques des postures du même mouvement ainsi que les signatures topologiques des postures de l'ensemble des mouvements utilisés pour l'apprentissage. Les modifications apportées consistaient à calculer les matrices des distances de chaque mouvement de l'ensemble d'apprentissage en post-classification (Fig 55) et apprendre au réseau de neurones à les classifier. Ainsi, lorsque l'on souhaite classer un mouvement, il suffit de calculer sa matrice de distance intra-mouvement et l'utiliser directement après une réduction dimensionnelle comme entrée au réseau de neurones. Ces modifications ont permis de réduire considérablement le temps de calcul, mais la comparaison entre les précisions de classification avant et après la modification de l'approche initiale ont montré une perte moyenne entre 5% et 10%.

Pour éviter de faire un compromis entre le temps de calcul et la précision obtenue, dans le Chapitre VI, on a proposé l'exécution de la première démarche proposée en parallèle, après la réécriture de l'algorithme correspondant à la première démarche avec un langage plus adapté à l'exécution parallèle. On a ainsi réussi à réduire le temps de calcul sans perdre en précision de classification.

Le second verrou scientifique posé en introduction est la possibilité de déterminer la suite d'un mouvement à partir des premières postures du mouvement. Pour cela, nous avons combiné notre approche d'analyse topologique via la persistance homologique avec un type spécifique de réseaux de neurones nommé les réseaux de neurones récurrents. Le but de ce type de réseaux de neurones est de déterminer la valeur d'une suite temporelle à l'instant  $T_{i+1}$  à partir des valeurs de la suite sur l'intervalle  $[0, T_i]$ . Avec cette approche, on était en mesure de déterminer la suite d'un mouvement à partir d'un intervalle d'enregistrement qui représente 15% de l'intégralité du mouvement.

Développer une approche qui à partir d'une base de connaissance qui englobe un ensemble de mouvements, permet de reconnaître un mouvement avec une grande précision, ainsi que la possibilité de prédire la suite d'un mouvement à partir d'un intervalle d'enregistrement, offre plusieurs possibilités applicatives. Dans le domaine médical, la classification et la reconnaissance des mouvements est très importante pour la rééducation. Par exemple, le praticien sera en mesure de suivre l'évolution du traitement en comparant les mouvements effectués par le patient par rapport aux mouvements exécutés par un participant contrôle. De même, la possibilité de calculer la différence entre un mouvement et un autre, en calculant la distance entre les signatures topologiques de leurs postures respectives, est une caractéristique

## Conclusion et perspectives

très importante apportée par notre démarche. Pour le domaine de la réalité virtuelle ou la réalité augmentée, la capacité de reconnaissance de mouvements est une partie incontournable pour le bon fonctionnement de l'expérience de l'immersion.

La combinaison de l'analyse topologique via la persistance homologique avec les réseaux de neurones récurrents nous a permis d'être en mesure de déterminer la suite d'un mouvement à partir d'une partie enregistrée qui avoisine les 10% de l'intégralité du mouvement. Cette démarche peut avoir plusieurs applications. Dans le domaine médical, la possibilité de prédire la suite d'un mouvement peut être utilisé pour surveiller et avertir en cas de mouvements dangereux pour les patients (personne âgées). Dans le domaine de réalité virtuelle, on peut utiliser la prédiction de la suite pour précalculer les interactions des éléments de l'environnement virtuel avec la personne en immersion.

Malgré les bons résultats obtenus pour la classification des mouvements humains et pour l'anticipation des mouvements, ainsi que les différentes applications possibles que ce soit dans le domaine médical ou la réalité virtuelle, l'ajout d'une étape d'analyse topologique en se basant sur la persistance homologique avant la classification est très couteuse en temps de calcul. Les différentes modifications apportées à notre approche initiale avec l'utilisation des réseaux de neurones et l'exécution parallèle ont permis de réduire considérablement le temps de calcul. Mais des travaux futurs peuvent être menés dans le but de réduire ce temps de calcul (voire même permettre une réalisation en temps réel de la méthode). Plusieurs pistes peuvent être envisagées telles que l'utilisation des processeurs massivement parallèles avec la réécriture du code pour qu'il soit exécuté par des processeurs graphiques « GPU » plus adaptés au calcul matriciel.

Dans les perspectives à explorer dans les futurs travaux, le choix de la distance entre les signatures topologiques pourrait être investigué plus en détail. Dans notre travail, on a présenté deux métriques : les distances de Bottleneck et de Wasserstein.

Actuellement le choix de la valeur du pas  $\varepsilon$  de la filtration, est fait manuellement selon les estimations de l'utilisateur. La valeur du pas ajouté à chaque étape de la filtration a un impact très important sur la précision des signatures topologiques obtenues. Le développement d'une méthode heuristique pourrait être réalisée dans le but de trouver la valeur optimale du pas  $\varepsilon$ .

# Annexe A

Les résultats détaillés de la première expérience :

	Mouvements classifiés	Classification directe des données	satisfaction par rapport au résultat attendu	Classification après analyse topologique	satisfaction par rapport au résultat attendu
Inter-Mouvements Inter-Subjects	Subj 1 (Task1) / Subj 2 (Task 2)	100%	😊	90,53%	😊
	Subj 1 (Task1) / Subj 3 (Task 2)	100%	😊	94.31%	😊
	Subj 1 (Task1) / Subj 4 (Task 2)	100%	😊	92.16%	😊
	Subj 2 (Task1) / Subj 3 (Task 2)	100%	😊	96.08%	😊
	Subj 2 (Task1) / Subj 4 (Task 2)	100%	😊	93.89%	😊
	Subj 3 (Task1) / Subj 4 (Task 2)	100%	😊	95.28%	😊
	Subj 1 (Task1) / Subj 2 (Task 3)	100%	😊	96.21%	😊
	Subj 1 (Task1) / Subj 3 (Task 3)	100%	😊	91.25%	😊
	Subj 1 (Task1) / Subj 4 (Task 3)	100%	😊	94.52%	😊
	Subj 2 (Task1) / Subj 3 (Task 3)	100%	😊	93.28%	😊
	Subj 2 (Task1) / Subj 4 (Task 3)	100%	😊	89.28%	😊
	Subj 3 (Task1) / Subj 4 (Task 3)	100%	😊	91.27%	😊
	Subj 1 (Task1) / Subj 2 (Task 4)	100%	😊	96.28%	😊
	Subj 1 (Task1) / Subj 3 (Task 4)	100%	😊	94.22%	😊
	Subj 1 (Task1) / Subj 4 (Task 4)	100%	😊	90.63%	😊
	Subj 2 (Task1) / Subj 3 (Task 4)	100%	😊	97.28%	😊
	Subj 2 (Task1) / Subj 4 (Task 4)	100%	😊	94.89%	😊
	Subj 3 (Task1) / Subj 4 (Task 4)	100%	😊	95.23%	😊
	Subj 1 (Task2) / Subj 2 (Task 3)	100%	😊	97.05%	😊
	Subj 1 (Task2) / Subj 3 (Task 3)	100%	😊	93.94%	😊
	Subj 1 (Task2) / Subj 4 (Task 3)	100%	😊	88.74%	😊
	Subj 2 (Task2) / Subj 3 (Task 3)	100%	😊	96.28%	😊
	Subj 2 (Task2) / Subj 4 (Task 3)	100%	😊	97.28%	😊
	Subj 3 (Task2) / Subj 4 (Task 3)	100%	😊	90.87%	😊
	Subj 1 (Task2) / Subj 2 (Task 4)	100%	😊	94.03%	😊
	Subj 1 (Task2) / Subj 3 (Task 4)	100%	😊	94.88%	😊
	Subj 1 (Task2) / Subj 4 (Task 4)	100%	😊	97.52%	😊
	Subj 2 (Task2) / Subj 3 (Task 4)	100%	😊	93.28%	😊
	Subj 2 (Task2) / Subj 4 (Task 4)	100%	😊	94.29%	😊
	Subj 3 (Task2) / Subj 4 (Task 4)	100%	😊	95.07%	😊
	Subj 1 (Task3) / Subj 2 (Task 4)	100%	😊	91.06%	😊
	Subj 1 (Task3) / Subj 3 (Task 4)	100%	😊	97.29%	😊
	Subj 1 (Task3) / Subj 4 (Task 4)	100%	😊	95.96%	😊
	Subj 2 (Task3) / Subj 3 (Task 4)	100%	😊	98.50%	😊
Subj 2 (Task3) / Subj 4 (Task 4)	100%	😊	91.29%	😊	

	Subj 3 (Task3) / Subj 4 (Task 4)	100%	☺	94.02%	☺
<b>Inter-Mouvements Intra-Subjects</b>					
	Subj 1 (Task 1) / Subj 1 (Task 2)	84,82%	☺	86,02%	☺
	Subj 1 (Task 1) / Subj 1 (Task 3)	90,25%	☺	79,04%	☺
	Subj 1 (Task 2) / Subj 1 (Task 3)	91.28%	☺	86.28%	☺
	Subj 2 (Task 1) / Subj 2 (Task 2)	81,02%	☺	83,56%	☺
	Subj 2 (Task 1) / Subj 2 (Task 3)	85.20%	☺	91.09%	☺
	Subj 2 (Task 2) / Subj 2 (Task 3)	89.08%	☺	92.33%	☺
	Subj 3 (Task 1) / Subj 3 (Task 2)	90.87%	☺	89.06%	☺
	Subj 3 (Task 1) / Subj 3 (Task 3)	94.08%	☺	85.25%	☺
	Subj 3 (Task 2) / Subj 3 (Task 3)	87,51%	☺	90.17%	☺
	Subj 4 (Task 1) / Subj 4 (Task 2)	92.83%	☺	91.06%	☺
	Subj 4 (Task 1) / Subj 4 (Task 3)	81.85%	☺	91.77%	☺
	Subj 4 (Task 2) / Subj 4 (Task 3)	87.09%	☺	90.25%	☺
<b>Intra-Mouvements Inter-Subjects</b>					
	Subj 1 (Task 1) / Subj 2 (Task 1)	100%	☺	48,52%	☺
	Subj 1 (Task 1) / Subj 3 (Task 1)	100%	☺	51.72%	☺
	Subj 1 (Task 1) / Subj 4 (Task 1)	100%	☺	51,49%	☺
	Subj 1 (Task 2) / Subj 2 (Task 2)	100%	☺	52,65%	☺
	Subj 1 (Task 2) / Subj 3 (Task 2)	100%	☺	49.81%	☺
	Subj 1 (Task 2) / Subj 4 (Task 2)	100%	☺	53.04%	☺
	Subj 1 (Task 3) / Subj 2 (Task 3)	100%	☺	42.89%	☺
	Subj 1 (Task 3) / Subj 3 (Task 3)	100%	☺	53.95%	☺
	Subj 1 (Task 3) / Subj 4 (Task 3)	100%	☺	55.05%	☺
	Subj 2 (Task 1) / Subj 3 (Task 1)	100%	☺	45.89%	☺
	Subj 2 (Task 1) / Subj 4 (Task 1)	100%	☺	53.07%	☺
	Subj 1 (Task 2) / Subj 3 (Task 2)	100%	☺	48.67%	☺
	Subj 1 (Task 2) / Subj 4 (Task 2)	100%	☺	51.05%	☺
	Subj 2 (Task 3) / Subj 3 (Task 3)	100%	☺	50.00%	☺
	Subj 2 (Task 3) / Subj 4 (Task 3)	100%	☺	51.37%	☺
	Subj 3 (Task 1) / Subj 4 (Task 1)	100%	☺	53.78%	☺
	Subj 3 (Task 2) / Subj 4 (Task 2)	100%	☺	56.23%	☺
	Subj 3 (Task 3) / Subj 4 (Task 3)	100%	☺	52.37%	☺
<b>Intra-Mouvements Intra-Subjects</b>					
	Subj 1 (Task 1) / Subj 1 (Task 1)	52,4%	☺	50%	☺
	Subj 1 (Task 2) / Subj 1 (Task 2)	54,39%	☺	52,39%	☺
	Subj 1 (Task 3) / Subj 1 (Task 3)	56,37%	☺	51,29%	☺
	Subj 2 (Task 1) / Subj 2 (Task 1)	51,3%	☺	53,22%	☺
	Subj 2 (Task 2) / Subj 2 (Task 2)	52,44%	☺	50,29%	☺
	Subj 2 (Task 3) / Subj 2 (Task 3)	53,84%	☺	51,29%	☺
	Subj 3 (Task 1) / Subj 3 (Task 1)	54,94%	☺	54,27%	☺
	Subj 3 (Task 2) / Subj 3 (Task 2)	52,78%	☺	47,95%	☺
	Subj 3 (Task 3) / Subj 3 (Task 3)	54,91%	☺	51,25%	☺
	Subj 3 (Task 1) / Subj 3 (Task 1)	52,34%	☺	49,67%	☺

---

Subj 3 (Task 2) / Subj 3 (Task 2)	51,37%	☺	50,94%	☺
Subj 3 (Task 3) / Subj 3 (Task 3)	54,27%	☺	50,61%	☺





## Annexe B.

Les résultats détaillés de la deuxième expérience :

	Mouvements classifiés	Classification directe des données	satisfaction par rapport au résultat attendu	Classification après analyse topologique	satisfaction par rapport au résultat attendu
Inter-Mouvements Inter-Subjects	Subj 1 (Task1) / Subj 2 (Task 2)	83.75%	☺	93.56%	☺
	Subj 1 (Task1) / Subj 3 (Task 2)	77.27%	☺	91.34%	☺
	Subj 1 (Task1) / Subj 4 (Task 2)	71.86%	☺	85.69%	☺
	Subj 1 (Task1) / Subj 2 (Task 3)	75.66%	☺	87.04%	☺
	Subj 1 (Task1) / Subj 3 (Task 3)	81.27%	☺	95.67%	☺
	Subj 1 (Task1) / Subj 4 (Task 3)	85.39%	☺	94.78%	☺
	Subj 1 (Task1) / Subj 2 (Task 4)	76.59%	☺	95.67%	☺
	Subj 1 (Task1) / Subj 3 (Task 4)	82.65%	☺	92.84%	☺
	Subj 1 (Task1) / Subj 4 (Task 4)	84.29%	☺	90.78%	☺
	Subj 1 (Task1) / Subj 2 (Task 5)	77.94%	☺	94.37%	☺
	Subj 1 (Task1) / Subj 3 (Task 5)	82.55%	☺	92.37%	☺
	Subj 1 (Task1) / Subj 4 (Task 5)	75.69%	☺	92.28%	☺
	Subj 1 (Task1) / Subj 2 (Task 6)	74,28%	☺	91,28%	☺
	Subj 1 (Task1) / Subj 3 (Task 6)	79,87%	☺	94,87%	☺
	Subj 1 (Task1) / Subj 4 (Task 6)	82,59%	☺	91,67%	☺
	Subj 1 (Task1) / Subj 2 (Task 7)	75,98%	☺	96,35%	☺
	Subj 1 (Task1) / Subj 3 (Task 7)	76,25%	☺	94,88%	☺
	Subj 1 (Task1) / Subj 4 (Task 7)	77,49%	☺	96,85%	☺
	Subj 1 (Task1) / Subj 2 (Task 8)	64,59%	☺	94,37%	☺
	Subj 1 (Task1) / Subj 3 (Task 8)	71,29%	☺	89,37%	☺
	Subj 1 (Task1) / Subj 4 (Task 8)	67,29%	☺	95,67%	☺
	Subj 1 (Task1) / Subj 2 (Task 9)	71,94%	☺	87,94%	☺
	Subj 1 (Task1) / Subj 3 (Task 9)	65,78%	☺	86,57%	☺
	Subj 1 (Task1) / Subj 4 (Task 9)	71,94%	☺	82,94%	☺
	Subj 2 (Task1) / Subj 3 (Task 2)	81.40%	☺	91.30%	☺
	Subj 2 (Task1) / Subj 4 (Task 2)	68,94%	☺	86,57%	☺
	Subj 2 (Task1) / Subj 3 (Task 3)	85,67%	☺	91,85%	☺
	Subj 2 (Task1) / Subj 4 (Task 3)	91,57%	☺	95,67%	☺
	Subj 2 (Task1) / Subj 3 (Task 4)	88,27%	☺	93,48%	☺
	Subj 2 (Task1) / Subj 4 (Task 4)	93,78%	☺	96,29%	☺
	Subj 2 (Task1) / Subj 3 (Task 5)	87,67%	☺	91,58%	☺
	Subj 2 (Task1) / Subj 4 (Task 5)	85,67%	☺	96,37%	☺
Subj 2 (Task1) / Subj 3 (Task 6)	85,67%	☺	92,67%	☺	
Subj 2 (Task1) / Subj 4 (Task 6)	69,67%	☺	81,58%	☺	

	Subj 2 (Task1) / Subj 3 (Task 7)	78,33%	☹	96,37%	☹
	Subj 2 (Task1) / Subj 4 (Task 7)	77,58%	☹	86,38%	☹
	Subj 2 (Task1) / Subj 3 (Task 8)	84,38%	☹	96,58%	☹
	Subj 2 (Task1) / Subj 4 (Task 8)	73,95%	☹	92,85%	☹
	Subj 1 (Task1) / Subj 3 (Task 9)	85,29%	☹	92,89%	☹
	Subj 1 (Task1) / Subj 4 (Task 9)	71,29%	☹	94,36%	☹
	Subj 3 (Task1) / Subj 4 (Task 2)	81,29%	☹	90,24%	☹
	Subj 3 (Task1) / Subj 4 (Task 3)	85,94%	☹	94,39%	☹
	Subj 3 (Task1) / Subj 4 (Task 4)	81,67%	☹	94,37%	☹
	Subj 3 (Task1) / Subj 4 (Task 5)	74,39%	☹	92,84%	☹
	Subj 3 (Task1) / Subj 4 (Task 6)	85,94%	☹	89,37%	☹
	Subj 3 (Task1) / Subj 4 (Task 7)	82,36%	☹	94,67%	☹
	Subj 3 (Task1) / Subj 3 (Task 8)	86,37%	☹	93,84%	☹
	Subj 3 (Task1) / Subj 4 (Task 8)	82,29%	☹	83,94%	☹
	Subj 3 (Task1) / Subj 4 (Task 9)	72,54%	☹	89,36%	☹
<b>Inter-Mouvements Intra-Subjects</b>	Subj 1 (Task 1) / Subj 1 (Task 2)	72.45%	☹	90,24%	☹
	Subj 1 (Task 1) / Subj 1 (Task 3)	73,78%	☹	85,67%	☹
	Subj 1 (Task 1) / Subj 1 (Task 4)	71.27%	☹	82.49%	☹
	Subj 1 (Task 1) / Subj 1 (Task 5)	67.29%	☹	84.02%	☹
	Subj 1 (Task 1) / Subj 1 (Task 6)	75.29%	☹	88.37%	☹
	Subj 1 (Task 1) / Subj 1 (Task 7)	81.08%	☹	87.55%	☹
	Subj 1 (Task 1) / Subj 1 (Task 8)	69.87%	☹	94.85%	☹
	Subj 1 (Task 2) / Subj 1 (Task 9)	73.78%	☹	85,67%	☹
	Subj 1 (Task 2) / Subj 1 (Task 3)	65.94%	☹	90.09%	☹
	Subj 1 (Task 2) / Subj 1 (Task 4)	75.86%	☹	85.66%	☹
	Subj 1 (Task 2) / Subj 1 (Task 5)	69.58%	☹	80.78%	☹
	Subj 1 (Task 2) / Subj 1 (Task 6)	73.68%	☹	91.55%	☹
	Subj 1 (Task 2) / Subj 1 (Task 7)	75.87%	☹	88.56%	☹
	Subj 1 (Task 2) / Subj 1 (Task 8)	75.89%	☹	82.87%	☹
	Subj 1 (Task 2) / Subj 1 (Task 9)	70.77%	☹	86.05%	☹
	Subj 1 (Task 3) / Subj 1 (Task 4)	69.58%	☹	82.07%	☹
	Subj 1 (Task 3) / Subj 1 (Task 5)	74.96%	☹	93.07%	☹
	Subj 1 (Task 3) / Subj 1 (Task 6)	66.98%	☹	83.66%	☹
	Subj 1 (Task 3) / Subj 1 (Task 7)	70.69%	☹	86.51%	☹
	Subj 1 (Task 3) / Subj 1 (Task 8)	74.22%	☹	92.06%	☹
	Subj 1 (Task 3) / Subj 1 (Task 9)	73.05%	☹	86.33%	☹
	Subj 1 (Task 4) / Subj 1 (Task 5)	63.66%	☹	82.71%	☹
	Subj 1 (Task 4) / Subj 1 (Task 6)	64.95%	☹	82.34%	☹
	Subj 1 (Task 4) / Subj 1 (Task 7)	72.05%	☹	86.26%	☹
	Subj 1 (Task 4) / Subj 1 (Task 8)	71.42%	☹	91.33%	☹
	Subj 1 (Task 4) / Subj 1 (Task 9)	73.55%	☹	86.69%	☹
Subj 1 (Task 5) / Subj 1 (Task 6)	69.20%	☹	89.67%	☹	

Subj 1 (Task 5) / Subj 1 (Task 7)	71.32%	☹️	82.96%	☹️
Subj 1 (Task 5) / Subj 1 (Task 8)	68.23%	☹️	85.77%	☹️
Subj 1 (Task 5) / Subj 1 (Task 9)	72.09%	☹️	96.85%	☹️
Subj 1 (Task 6) / Subj 1 (Task 7)	64.21%	☹️	87.09%	☹️
Subj 1 (Task 6) / Subj 1 (Task 8)	72.51%	☹️	93.59%	☹️
Subj 1 (Task 6) / Subj 1 (Task 9)	67.38%	☹️	88.47%	☹️
Subj 1 (Task 7) / Subj 1 (Task 8)	70.33%	☹️	91.87%	☹️
Subj 1 (Task 7) / Subj 1 (Task 9)	74.28%	☹️	89.27%	☹️
Subj 1 (Task 8) / Subj 1 (Task 9)	72.23%	☹️	94.95%	☹️
Subj 2(Task 1) / Subj 2(Task 2)	69,48%	☹️	80,59%	☹️
Subj 2(Task 1) / Subj 2(Task 3)	63,02%	☹️	83,57%	☹️
Subj 2(Task 1) / Subj 2(Task 4)	74,53%	☹️	81,45%	☹️
Subj 2(Task 1) / Subj 2(Task 5)	74,97%	☹️	85,69%	☹️
Subj 2(Task 1) / Subj 2(Task 6)	76,29%	☹️	95,63%	☹️
Subj 2(Task 1) / Subj 2(Task 7)	64,97%	☹️	87,26%	☹️
Subj 2(Task 1) / Subj 2(Task 8)	77,38%	☹️	96,85%	☹️
Subj 2(Task 2) / Subj 2(Task 9)	72,08%	☹️	95,75%	☹️
Subj 2(Task 2) / Subj 2(Task 3)	69,03%	☹️	93,87%	☹️
Subj 2(Task 2) / Subj 2(Task 4)	70,97%	☹️	92,41%	☹️
Subj 2(Task 2) / Subj 2(Task 5)	66,78%	☹️	92,59%	☹️
Subj 2(Task 2) / Subj 2(Task 6)	62,89%	☹️	93,07%	☹️
Subj 2(Task 2) / Subj 2(Task 7)	71,89%	☹️	94,09%	☹️
Subj 2(Task 2) / Subj 2(Task 8)	65,22%	☹️	93,87%	☹️
Subj 2(Task 2) / Subj 2(Task 9)	67,81%	☹️	93,45%	☹️
Subj 2(Task 3) / Subj 2(Task 4)	71,09%	☹️	94,62%	☹️
Subj 2(Task 3) / Subj 2(Task 5)	63,82%	☹️	92,74%	☹️
Subj 2(Task 3) / Subj 2(Task 6)	64,78%	☹️	90,04%	☹️
Subj 2(Task 3) / Subj 2(Task 7)	59,37%	☹️	89,64%	☹️
Subj 2(Task 3) / Subj 2(Task 8)	73,97%	☹️	86,37%	☹️
Subj 2(Task 3) / Subj 2(Task 9)	67,38%	☹️	91,87%	☹️
Subj 2(Task 4) / Subj 2(Task 5)	71,25%	☹️	93,78%	☹️
Subj 2(Task 4) / Subj 2(Task 6)	76,74%	☹️	96,72%	☹️
Subj 2(Task 4) / Subj 2(Task 7)	74,31%	☹️	86,74%	☹️
Subj 2(Task 4) / Subj 2(Task 8)	76,29%	☹️	95,29%	☹️
Subj 2(Task 4) / Subj 2(Task 9)	72,88%	☹️	92,67%	☹️

Subj 2(Task 5) / Subj 2(Task 6)	71.37%	☹	94.37%	☹
Subj 2(Task 5) / Subj 2(Task 7)	70.84%	☹	93.67%	☹
Subj 2(Task 5) / Subj 2(Task 8)	69.75%	☹	92.55%	☹
Subj 2(Task 5) / Subj 2(Task 9)	73.64%	☹	91.77%	☹
Subj 2(Task 6) / Subj 2(Task 7)	75.27%	☹	94.25%	☹
Subj 2(Task 6) / Subj 2(Task 8)	65.89%	☹	94.09%	☹
Subj 2(Task 6) / Subj 2(Task 9)	84.28%	☹	94.37%	☹
Subj 2(Task 7) / Subj 2(Task 8)	65.29%	☹	92.08%	☹
Subj 2(Task 7) / Subj 2(Task 9)	70.28%	☹	89.37%	☹
Subj 3(Task 1) / Subj 3(Task 2)	71,85%	☹	92,45%	☹
Subj 3(Task 1) / Subj 3(Task 3)	72.08%	☹	95.74%	☹
Subj 3(Task 1) / Subj 3(Task 4)	72.58%	☹	91.74%	☹
Subj 3(Task 1) / Subj 3(Task 5)	85.07%	☹	95.64%	☹
Subj 3(Task 1) / Subj 3(Task 6)	73.94%	☹	84.28%	☹
Subj 3(Task 1) / Subj 3(Task 7)	72.94%	☹	93.78%	☹
Subj 3(Task 1) / Subj 3(Task 8)	72.49%	☹	94.37%	☹
Subj 3(Task 2) / Subj 3(Task 9)	71.29%	☹	94.27%	☹
Subj 3(Task 2) / Subj 3(Task 3)	67,34%	☹	86,98%	☹
Subj 3(Task 2) / Subj 3(Task 4)	73.29%	☹	91.70%	☹
Subj 3(Task 2) / Subj 3(Task 5)	73.67%	☹	91.27%	☹
Subj 3(Task 2) / Subj 3(Task 6)	74.97%	☹	87.94%	☹
Subj 3(Task 2) / Subj 3(Task 7)	70.55%	☹	94.37%	☹
Subj 3(Task 2) / Subj 3(Task 8)	68.27%	☹	98.67%	☹
Subj 3(Task 2) / Subj 3(Task 9)	76.97%	☹	93.67%	☹
Subj 3(Task 3) / Subj 3(Task 4)	71.45%	☹	92.87%	☹
Subj 3(Task 3) / Subj 3(Task 5)	69.88%	☹	94.87%	☹
Subj 3(Task 3) / Subj 3(Task 6)	76.39%	☹	87.29%	☹
Subj 3(Task 3) / Subj 3(Task 7)	67.94%	☹	82.67%	☹
Subj 3(Task 3) / Subj 3(Task 8)	73.69%	☹	91.87%	☹
Subj 3(Task 3) / Subj 3(Task 9)	63.78%	☹	92.67%	☹
Subj 3(Task 4) / Subj 3(Task 5)	65.37%	☹	73.94%	☹
Subj 3(Task 4) / Subj 3(Task 6)	61.89%	☹	94.75%	☹
Subj 3(Task 4) / Subj 3(Task 7)	66.37%	☹	86.22%	☹

	Subj 3(Task 4) / Subj 3(Task 8)	71.09%	☹	92.33%	☹
	Subj 3(Task 4) / Subj 3(Task 9)	65.37%	☹	87.29%	☹
	Subj 3(Task 5) / Subj 3(Task 6)	62.27%	☹	92.74%	☹
	Subj 3(Task 5) / Subj 3(Task 7)	71.09%	☹	88.94%	☹
	Subj 3(Task 5) / Subj 3(Task 8)	65.99%	☹	93.74%	☹
	Subj 3(Task 5) / Subj 3(Task 9)	63.94%	☹	84.67%	☹
	Subj 3(Task 6) / Subj 3(Task 7)	72.97%	☹	93.74%	☹
	Subj 3(Task 6) / Subj 3(Task 8)	67.29%	☹	82.55%	☹
	Subj 3(Task 6) / Subj 3(Task 9)	74.94%	☹	86.74%	☹
	Subj 3(Task 7) / Subj 3(Task 8)	64.29%	☹	82.97%	☹
	Subj 3(Task 7) / Subj 3(Task 9)	71.88%	☹	86.79%	☹
	Subj 2 (Task 1) / Subj 2 (Task 2)	69.48%	☹	80,59%	☹
	Subj 2 (Task 1) / Subj 2 (Task 3)	74.53%	☹	81.45%	☹
	Subj 3 (Task 2) / Subj 3 (Task 3)	67.34%	☹	86.98%	☹
	Subj 4 (Task 1) / Subj 4 (Task 3)	74.36%	☹	91.56%	☹
	Subj 4 (Task 2) / Subj 4 (Task 3)	61.93%	☹	94.32%	☹
<b>Intra-Mouvements Inter-Subjects</b>	Subj 1 (Task 1) / Subj 2 (Task 1)	84,56%	☹	52,02%	☹
	Subj 1 (Task 1) / Subj 3 (Task 1)	76,26%	☹	51,20%	☹
	Subj 1 (Task 1) / Subj 4 (Task 1)	74,67%	☹	54,69%	☹
	Subj 2 (Task 1) / Subj 3 (Task 1)	72,84%	☹	48,29%	☹
	Subj 3 (Task 1) / Subj 4 (Task 1)	84,28%	☹	51,28%	☹
	Subj 1 (Task 2) / Subj 2 (Task 2)	91,03%	☹	48,27%	☹
	Subj 1 (Task 2) / Subj 3 (Task 2)	66,52%	☹	51,20%	☹
	Subj 1 (Task 2) / Subj 4 (Task 2)	74,20%	☹	49,61%	☹
	Subj 2 (Task 2) / Subj 3 (Task 2)	69,87%	☹	45,37%	☹
	Subj 3 (Task 2) / Subj 4 (Task 2)	75,29%	☹	50,94%	☹
	Subj 1 (Task 3) / Subj 2 (Task 3)	84,32%	☹	51,16%	☹
	Subj 1 (Task 3) / Subj 3 (Task 3)	62,29%	☹	50,29%	☹
	Subj 1 (Task 3) / Subj 4 (Task 3)	64,59%	☹	52,84%	☹
	Subj 2 (Task 3) / Subj 3 (Task 3)	70,94%	☹	56,37%	☹
	Subj 3 (Task 3) / Subj 4 (Task 3)	67,94%	☹	52,09%	☹
	Subj 1 (Task 4) / Subj 2 (Task 4)	67,97%	☹	53,97%	☹
	Subj 1 (Task 4) / Subj 3 (Task 4)	73,64%	☹	50,39%	☹
	Subj 1 (Task 4) / Subj 4 (Task 4)	67,67%	☹	45,97%	☹
	Subj 2 (Task 4) / Subj 3 (Task 4)	71,94%	☹	52,97%	☹
	Subj 3 (Task 4) / Subj 4 (Task 4)	67,97%	☹	51,37%	☹
	Subj 1 (Task 5) / Subj 2 (Task 5)	65,87%	☹	46,97%	☹
	Subj 1 (Task 5) / Subj 3 (Task 5)	70,97%	☹	53,81%	☹

Subj 1 (Task 5) / Subj 4 (Task 5)	74,66%	☹️	54,79%	☺️
Subj 2 (Task 5) / Subj 3 (Task 5)	74,97%	☹️	50,28%	☺️
Subj 3 (Task 5) / Subj 4 (Task 5)	69,78%	☹️	48,76%	☺️
Subj 1 (Task 6) / Subj 2 (Task 6)	72,94%	☹️	50,28%	☺️
Subj 1 (Task 6) / Subj 3 (Task 6)	71,28%	☹️	46,95%	☺️
Subj 1 (Task 6) / Subj 4 (Task 6)	69,48%	☹️	52,84%	☺️
Subj 2 (Task 6) / Subj 3 (Task 6)	76,89%	☹️	51,6%	☺️
Subj 3 (Task 6) / Subj 4 (Task 6)	82,07%	☹️	53,91%	☺️
Subj 1 (Task 7) / Subj 2 (Task 7)	71,94%	☹️	52,47%	☺️
Subj 1 (Task 7) / Subj 3 (Task 7)	82,63%	☹️	51,47%	☺️
Subj 1 (Task 7) / Subj 4 (Task 7)	77,29%	☹️	49,67%	☺️
Subj 2 (Task 7) / Subj 3 (Task 7)	69,37%	☹️	44,99%	☺️
Subj 3 (Task 7) / Subj 4 (Task 7)	68,29%	☹️	53,67%	☺️
Subj 1 (Task 8) / Subj 2 (Task 8)	80,55%	☹️	53,56%	☺️
Subj 1 (Task 8) / Subj 3 (Task 8)	77,84%	☹️	50,37%	☺️
Subj 1 (Task 8) / Subj 4 (Task 8)	82,28%	☹️	42,96%	☺️
Subj 2 (Task 8) / Subj 3 (Task 8)	68,67%	☹️	50,76%	☺️
Subj 3 (Task 8) / Subj 4 (Task 8)	75,94%	☹️	51,74%	☺️
Subj 1 (Task 9) / Subj 2 (Task 9)	69,86%	☹️	53,28%	☺️
Subj 1 (Task 9) / Subj 3 (Task 9)	71,09%	☹️	52,97%	☺️
Subj 1 (Task 9) / Subj 4 (Task 9)	69,28%	☹️	54,09%	☺️
Subj 2 (Task 9) / Subj 3 (Task 9)	73,94%	☹️	55,64%	☺️
Subj 3 (Task 9) / Subj 4 (Task 9)	68,28%	☹️	50,49%	☺️

# Bibliographie

- Abend, W., Bizzi, E., & Morasso, P. (1982). Human arm trajectory formation. *Brain*, 331-348.
- Adams, H., Tausz, A., & Vejdemo-Johansson, M. (2014). javaPlex: A Research Software Package for Persistent (Co)Homology. *International Congress on Mathematical Software*. Berlin.
- Adcock, A., Rubin, D., & Carlsson, G. (2014). Classification of hepatic lesions using the matching metric. *Computer Vision and Image Understanding*, 121, 36-42.
- Afra Zomorodian. (2010). Fast construction of the Vietoris-Rips complex. *Computers & Graphics*, 263-271.
- Amdahl, G. (1967). Validity of the single processor approach to achieving large scale computing capabilities. *AFIPS Spring Joint Computer Conference*. IBM sunnyvale, California.
- Anderson Rocha, & Siome Klein Goldenstein. (2014). Multiclass from Binary: Expanding One-Versus-All, One-Versus-One and ECOC-Based Approaches. *IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS*.
- Andrea Mannini, & Angelo Maria Sabatini. (2010). Machine Learning Methods for Classifying Human Physical Activity from On-Body Accelerometers. *Sensors*, 10, 1154-1175.
- Andrea, M., & Sabatini Maria, A. (2010). Machine Learning Methods for Classifying Human Physical Activity from On-Body Accelerometers. *Sensors*, 2(10), 1154-1175.
- Andrew Tchircoff. (2017, Août 4). *The mostly complete chart of Neural Networks, explained*. (Towards Data Science) Récupéré sur <https://towardsdatascience.com/the-mostly-complete-chart-of-neural-networks-explained-3fb6f2367464>
- Anglo-Australian Observatory. (2003, June 30). *The 2dF Galaxy Redshift Survey*. Récupéré sur <http://www.2dfgrs.net>
- Asha, S., Sudalaimani, C., Devanand, P., Thomas, T., & Sudhamony, S. (2013). Automated Seizure Detection from Multichannel EEG Signals using Support vector Machine and Artificial neural. *International Mutli-Conference on Automation, Computing, Communication, Control and Compressed Sensing*. Kottayam, India.

- Atallah, M. (1998). *Algorithms and Theory of Computation Handbook*. Purdue University .
- B. Tenenbaum, J., de Silva, V., L. Schwartz, E., & C. Langford, J. (2002). The Isomap Algorithm and Topological Stability. *Science*, 295.
- Barton P. Miller, Cathryn Macrander, & Stuart Sechrest. (1986). A distributed programs monitor for berkeley UNIX. *Wiley Online Library*, 16(2), 183-200.
- Berret, B., Bonnetblanc, F., Papaxanthis, C., & Pozzo, T. (2009). Modular control of pointing beyond arm's length. *Journal of Neuroscience*, 29(1), 191-205.
- Bierlaire, M. (2006). *Introduction à l'optimisation différentiable*. lausanne: Presses polytechniques et universitaires romandes.
- Bizzi, E., Mussa-Ivaldi, F., & Giszter, S. (1991). Computations underlying the execution of movement: a biological perspective. *Science*, 25(80), 287-291.
- Boor, C. (1978). *A Practical Guide to Splines*. New York: Springer-Verlag.
- Bottou, L. (2010). Large-Scale Machine Learning with Stochastic Gradient Descent. *19th International Conference on Computational Statistics*. Paris, France.
- Bregler, C. (1997). Learning and Recognizing Human Dynamics in Video Sequences. *IEEE Conf. Comp. Vision and Pattern Recognition*. San Juan, Puerto Rico.
- Brittany Terese Fasy, Jisu Kim, Fabrizio Lecci, & Clément Maria. (2015). *Introduction to the R package TDA*. Récupéré sur <https://hal.inria.fr/hal-01113028/document>
- Brooks, S., Gelman, A., Jones, G., & Meng, X.-L. (2011). *Handbook of Markov Chain Monte Carlo*. 2011.
- Chazal F., Cohen-Steiner D., Guibas L. J., Memoli F., and Oudot S.Y.. Gromov-Hausdorff stable signatures for shapes using persistence. *Computer Graphics Forum (proc. SGP 2009)*, pages 1393-1403, 2009.
- Chong, Y., & Hamadi, Y. (2006). Distributed Log-Based Reconciliation. *ECAI* , 141, 108-112.
- CHRISTOPHER M. BISHOP . (1995). *Neural Networks for Pattern Recognition*. OXFORD: CLARENDON PRESS.
- Cohen-Steiner D., Edelsbrunner H., and Harer J.. Stability of persistence diagrams. In *ACM Symp. Comp. Geometry*, 2005.
- Combrisson, E., & Jerbi, K. (2015). Exceeding chance level by chance: The caveat of theoretical chance levels in brain signal classification and statistical assessment of decoding accuracy. *Journal of Neuroscience Methods*, 250, 126-136.



- Communicating Sequential Processes*. (1985).
- Datasheet. (2016). Intel © Core™ i7 Processor Family for LGA2011-v3 Socket.
- Davison L., M. (1991). *Multidimensional Scaling*. Krueger.
- Dongarra, J., Laforenza, D., & Orlando, S. (2003). Recent Advances in Parallel Virtual Machine and Message Passing Interface. *10th European PVM/MPI Users' Group Meeting*. Venice.
- D'Avella, A., Saltiel, P., & Bizzi, E. (2003). Combinations of muscle synergies in the construction of a natural motor behavior. *Nat Neurosci*, 3(6), 300-308.
- Dirk Edelbuettel, Romain Francois, JJ Allaire, Kevin Ushey, Qiang Kou, Nathan Russel, . . . John Chambers. (2018, March). Package 'Rcpp'. Récupéré sur <https://cran.r-project.org/web/packages/Rcpp/Rcpp.pdf>
- Distributed Partial ConstraintSatisfaction Problem. (1997). *Principles and Practice of Constraint Programming; Springer*, 222-236.
- DO Hebb. (1988). *The Organization of Behavior*. Wiley: New York.
- Dupont de Dinechin, B., van Amstel, D., Poulhi, M., & Lager, G. (2014). Time-critical computing on a single-chip massively parallel processor. *DATE '14 Proceedings of the conference on Design, Automation & Test in Europe*. Dresden, Germany.
- EADWEARD MUYBRIDGE. (1885). *ANIMAL LOCOMOTION*. THE UNIVERSITY OF PENNSYLVANIA.
- Edelsbrunner, H., & Morozov, D. (2012). PERSISTENT HOMOLOGY: THEORY AND PRACTICE. *Proceedings of the European Congress of Mathematics*. Kraków.
- Edelsbrunner H., Letscher D., & Zomorodian A. (2000). Topological persistence and simplification. *Foundations of Computer Science*. Redondo Beach.
- Edelsbrunner, H., Letscher, D., & Zomorodian, A. (2002). Topological persistence and simplification. *Discrete & Computational Geometry*, 28(4), 511-533.
- Filtfilt Matlab Zero-phase digital filtering*. (s.d.). (Matlab) Consulté le 2015, sur [https://fr.mathworks.com/help/signal/ref/filtfilt.html?s\\_tid=gn\\_loc\\_drop](https://fr.mathworks.com/help/signal/ref/filtfilt.html?s_tid=gn_loc_drop)
- Gordon E., M. (1965). Cramming more components onto integrated circuits. *Electronics*, 38(8).
- Hachaj, T., R. Ogiela, M., & Koptyra, K. (2016). Human actions recognition from motion capture recordings using signal resampling and pattern recognition methods. *Annals of Operations Research*, 1-17.

- Heo, G., & Gamble, J. (2010). Exploring uses of persistent homology for statistical analysis of landmark-based shape data. *Journal of Multivariate Analysis*, 101(9), 2184-2199.
- Heo, G., Gamble, J., & T. Kim, P. (2012). Topological Analysis of Variance and the Maxillary Complex. *Journal of the American Statistical Association*, 107.
- Hilt, P. (2015). *les infos Décision motrice et contrôle modulaire d'un système hyper-redondant*. Dijon.
- Hilt, P., Berret, B., Papaxanthis, C., Stapley, P., & Pozzo, T. (2016). Evidence for subjective values guiding posture and movement coordination in a free-endpoint whole-body reaching task. *Scientific Reports*, 6.
- Hotelling, H. (1933). Analysis of a complex of statistical variables into principal components. *Journal of Educational Psychology*, 417-441.
- Ishida, T., Yokoo, M., & Gasser, L. (1990). An organizational approach to adaptive production systems. *Proceedings of the eighth National conference on Artificial intelligence*, 1, 52-58 .
- J.A.K. , S., & Vandewalle, J. (1999). Least Squares Support Vector Machine Classifiers. *Neural Processing Letters*, 9, 293-300.
- John C. Platt , & Nello Cristianini . (1990). large margin dags for multiclass classification. *Neural Information Processing Systems*.
- Kalapanidas, E., Nikolaos, M., Viorel Craciun, M., & Neagu, D. (2003). Machine Learning algorithms: a study on noise sensitivity.
- Kalray Announces Most Advanced Solution For Network Virtualization At Linley Data Center Conference*. SANTA CLARA (2015).
- Kerber, M., & R. Sharathkumar. (2013). Approximate Čech Complex in Low and High Dimensions. *International Symposium on Algorithms and Computation*. Hong Kong.
- Laboratoire INSERM U1093. (2011, 11 25). *PLATEAU TECHNIQUE, Vicon*. Récupéré sur [http://u1093.u-bourgogne.fr/fr/?option=com\\_content&view=article&id=137:vicon&catid=46](http://u1093.u-bourgogne.fr/fr/?option=com_content&view=article&id=137:vicon&catid=46)
- Léauté, T., Ottens, B., Szymanek, R., & Faltings, B. (2009). FRODO: An Open-Source Framework for Distributed Constraint Optimization.
- Leightley, D., Darby, J., Li, B., S. McPhee, J., & Yap, M. (2017). Human Activity Recognition for Physical Rehabilitation. *Biomedical and Health Informatics IEEE Journal* , 21, 939-948.
- Liaw, A., & Wiener, M. (2001). Classification and Regression by RandomForest. *R News*, 2(3), 18-22.
- Marey, E.-J. (1894). *Le mouvement*. Masson.

- Matlab Inc. (s.d.). *Cubic spline data interpolation*. Consulté le 2016, sur <https://fr.mathworks.com/help/matlab/ref/spline.html>
- Michel Minoux. (1975). Plus court chemin avec contraintes : Algorithmes et applications. *Annales Des Télécommunications*, 383-394.
- Miele, V., & Louvet, V. (2016). *Calcul parallèle avec R*. edpsciences.
- Mileyko, Y., Mukherjee, S., & Harer, J. (2011). Probability measures on the space of persistence. *INVERSE PROBLEMS*, 27.
- Ming-Hsuan Yang. (2002). Extended Isomap for Pattern Classification. *American Association for Artificial Intelligence*. Honda Fundamental Research Labs.
- Morozov, D. (2007). *Dionysus, a C++ library for computing persistent homology*.
- Nair S., French S.M., Laroche R.D., & Thomas E. (2010). The Application of Machine Learning Algorithms to the Analysis of Electromyographic Patterns From Arthritic Patients. *IEEE TRANSACTIONS ON NEURAL SYSTEMS AND REHABILITATION ENGINEERING*, 2(18), 174-184.
- Nello Cristianini , & John Shawe-Taylor . (2000). *An Introduction to Support Vector Machines and Other Kernel-based Learning Methods*. Cambridge University Press.
- Nicolau M., A. J. Levine, & G. Carlsson. (2011). Topology based data analysis identifies a subgroup of breast cancers with a unique mutational profile and excellent survival. *Proceedings of the National Academy of Sciences*, 108.
- Nsaunders. (2010, AUGUST 20). *A brief introduction to "apply" in R*. Récupéré sur <https://nsaunders.wordpress.com/2010/08/20/a-brief-introduction-to-apply-in-r/>
- Osgood, B. (2000). *The Fourier Transform and its Applications*. EE 261. New York.
- Package 'parallel'. (2017, December 7). Récupéré sur <https://stat.ethz.ch/R-manual/R-devel/library/parallel/doc/parallel.pdf>
- Pattern Recognition using Generalized Portrait Method. (1963). *Automation and Remote Control*(24), 774-780.
- Pearson, K. (1901). On Lines and Planes of Closest Fit to Systems of Points in Space . *Philosophical Magazine*, 2(6), 559-572.
- Petri, C. (1966). *Communication with automata*. New york.
- Processors . (2017). Récupéré sur Kalray: <http://www.kalrayinc.com/portfolio/processors/>

- R. van de Weygaert, G. Vegter, E. Platen, B. Eldering, & N. Kruthof. (2011). Alpha, Betti and the Megaparsec Universe: on the Topology of the Cosmic Web. *arXiv:1006.2765*.
- Rosenblatt, F. (1958). THE PERCEPTRON: A PROBABILISTIC MODEL FOR INFORMATION STORAGE AND ORGANIZATION IN THE BRAIN. *Psychological Review*, 65(6), 386-408.
- Ryan Rifkin, & Aldebaro Klautau. (2004). In Defense of One-Vs-All Classification. *Journal of Machine Learning Research*(5), 101-141.
- S. Knerr, L. Personnaz, & G. Dreyfus. (1990). Single-layer learning revisited: a stepwise procedure for building and training a neural network. *Springer-Verlag*, 68, 41-50.
- Sanjit K Mitra. (2011). *Digital Signal Processing*. New York: McGraw-Hill Higher Education.
- Shoham, Y. (1990). *Agent-Oriented Programming*. Technical Report STAN-CS-90-1335: Stanford University.
- Steve Weston, & Rich Calaway. ( 2017, Septembre). *Getting Started with doParallel and foreach*. Récupéré sur <https://cran.r-project.org/web/packages/doParallel/vignettes/gettingstartedParallel.pdf>
- Tausz, A., Vejdemo-Johansson, M., & Adams, H. (2014). Javaplex: A research software package for persistent (co)homology. *Stanford University*, 129-136.
- The support vector machine under test. (2003). *Neurocomputing*(55), 169 - 186.
- Tolambiya , A., Thomas , E., Chiovetto , E., Berret, B., & Pozzo, T. (2011). An ensemble analysis of electromyographic activity during whole body pointing with the use of support vector machines. *PLoS One*, 6(7).
- Václav Snášel, Jana Nowaková, Fatos Xhafa, & Leonard Barolli. (2017). Geometrical and topological approaches to Big Data. *Future Generation Computer Systems*, 67, 286-296.
- Vapnik, V. (1995). The Nature of Statistical Learning Theory. *Springer-Verlag*(4).
- VIN DE SILVA, & ROBERT GHRIST. (2007). Coverage in sensor networks via persistent homology. *Algebraic & Geometric Topology*, 339-358.
- Von Tscherner, V., Enders, H., & Maurer, C. (2013). *Subspace Identification and Classification of Healthy Human Gait*. (PLOS ONE) Récupéré sur <https://doi.org/10.1371/journal.pone.0065063>
- Wahbi, M., Ezzahir, R., Bessiere, C., & Bouyakhf, E.-H. (2011). DisChoco 2: A platform for distributed constraint reasoning. *Proceedings of DCR*, 11, 112-121.

- Wang, H., & Wu, J. (2014). *Classification of Human Posture and Movement Using Accelerometer Data*. Consulté le 2014, sur <http://cs229.stanford.edu/proj2014/Huafei%20Wang,%20Jennifer%20Wu,%20Classification%20of%20Human%20Posture%20and%20Movement%20Using%20Accelerometer%20Data.pdf>
- Wannes Rosiers. (2015, June 26). *Package 'parallelSVM'*. Récupéré sur <https://cran.r-project.org/web/packages/parallelSVM/parallelSVM.pdf>
- Wannes Rosiers. (2015, Juin 26). *Package 'parallelSVM'*. Récupéré sur <https://cran.r-project.org/web/packages/parallelSVM/parallelSVM.pdf>
- Warren S. McCulloch, & Walter Pitts. (1943). A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, 5(4), 115-133.
- Widrow, B., & Hoff, M. (1960). Adaptive Switching Circuits. *Technical Report*(1553-1).
- Xu. H.K.(2003). An Iterative Approach to Quadratic Optimization. *Journal of Optimization Theory and Applications*, 116, 659-678.

