

## Travail de Bachelor 2014

# Détection d'activité à l'aide de Kinect 2.0



Etudiant : Martin Tazlari

Professeur : Yann Bocchi

Déposé le : 28 juillet 2014

Martin Tazlari

## Source de l'illustration de la page de titre

[http://i-cms.journaldunet.com/image\\_cms/original/1723090-kinect-2-un-nouvel-horizon-s-ouvre-pour-les-usages-pros.jpg](http://i-cms.journaldunet.com/image_cms/original/1723090-kinect-2-un-nouvel-horizon-s-ouvre-pour-les-usages-pros.jpg)

## Avertissement

Ce travail nécessite des connaissances en programmation et en mathématiques afin de pouvoir être compris dans son ensemble.

La technologie présentée dans ce rapport est également en phase bêta aussi bien au niveau hardware que software. Cette technologie évoluera dans le futur, ce qui aura des influences sur ce travail.

La partie « Etat de l'art » de ce travail a été réalisée de façon commune avec M. Joël Voiselle, étudiant à la Hevs avec l'accord du M. Yann Bocchi, superviseur du projet.

## Remerciements

Je tiens à remercier toutes les personnes m'ayant soutenu durant la durée de ce travail de Bachelor.

Je remercie tout particulièrement :

M. Yann Bocchi, pour l'encadrement du travail et le suivi qu'il a effectué durant ces 3 mois.

Messieurs, Gianluca Rizzo et Stefano Bromuri pour leurs conseils précieux et leur disponibilité lors de la rédaction de ce travail.

Merci également aux personnes ayant relu ce rapport

Merci enfin à ma famille et amis qui m'ont soutenu durant toute ma formation.

## Résumé

Peut-on détecter des activités de façon efficace ?

Cette question fut la base de mon travail de Bachelor, son but est donc de rechercher différentes techniques de classification de données afin de détecter une action concrète.

Pour cela, j'ai pu utiliser la dernière caméra Kinect 2.0, ce choix de technologie a été motivé par plusieurs critères. Tout d'abord le fait que cette technologie soit très récente, ce qui permettrait donc au final de découvrir les premières possibilités de l'appareil et ainsi pouvoir déterminer des possibilités d'améliorations une fois la technologie arrivée à maturité.

Le but de ce travail est donc d'étudier et de classer différentes activités filmées à l'aide de Kinect 2.0 et d'en déterminer quelles applications pratiques on peut en faire.

Par exemple :

- Pourrons-nous détecter un incendie en se basant sur la détection de plusieurs personnes en train de courir ?
- Pourrons-nous détecter si le mouvement d'une partie du corps se fait correctement en se basant sur des images filmées ?

Pour cela il m'a fallu effectuer tout un processus de recherche :

- Définir le nombre et le choix des activités à étudier
- Définir les premiers critères importants de cette analyse
- Rechercher des algorithmes de classification de données
- Comparer les résultats afin de déterminer le modèle de données le plus pertinent

Plus spécifiquement, je me suis concentré sur deux types d'analyses :

- Les analyses de séries temporelles
- Les analyses de séries fréquentielles

J'ai ainsi pu obtenir des résultats montrant un faible taux d'erreurs de classification, ce qui démontre qu'en se basant sur des analyses simples on obtient déjà des premiers résultats pertinents.

Afin d'aller plus loin dans mon travail j'ai également cherché à corriger et adapter mes données de départ pour augmenter la précision des modèles de données. J'ai ainsi essayé plusieurs techniques de transformation des données qui ont permis d'améliorer le taux d'erreurs de classification en dessous des 15% d'erreurs.

De plus, cela m'a permis de détecter dans quelles situations un algorithme de classification est mieux adapté qu'un autre.

Toutes ces analyses m'ont alors permis de déterminer des premières limites de la détection d'activité en utilisant Kinect 2.0. Elles ont également démontré qu'en utilisant des techniques de classification simple on pouvait obtenir des résultats déjà utilisables pour de futures applications de détection d'activité.

En conclusion, la technologie encore très jeune qu'est Kinect 2.0 combinée à de simples algorithmes de classification, offre déjà d'excellents résultats. De plus, il existe encore de nombreuses possibilités d'analyses sur les données que j'ai pu récolter, ce qui signifie qu'on peut facilement imaginer qu'à l'avenir on pourra définir des modèles et ainsi pousser encore plus loin la détection d'activité.

## Table des matières

Source de l'illustration de la page de titre .....	ii
Liste de figures.....	x
Liste de tableaux.....	xii
Abréviations et Glossaire.....	xiv
1. Présentation du travail .....	2
1.1 Problématique .....	3
1.1 Synthèse de présentation du Travail de Bachelor .....	4
1.2 Outils utilisés .....	5
1.2.1 Kinect Studio .....	5
1.2.2 Visual Studio 2013 .....	5
1.2.3 Knime.....	6
2. Etat de l'art .....	7
2.1 Kinect 1.0 vs Kinect 2.0.....	7
2.3 Fonctionnalités de Kinect pour Xbox One .....	8
2.4 Les fonctionnalités actuelles de Kinect 2.0 .....	9
2.5 Les fonctionnalités futures de Kinect 2.0 .....	10
2.6 Description des fonctionnalités actuelles .....	10
2.6.1 Caméra couleur .....	10
2.6.2 Caméra de profondeur .....	11
2.6.3 Reconnaissance basique du corps.....	11
2.6.4 Caméra infrarouge.....	13
2.6.5 Mapping de coordonnées .....	14
2.7 Cas d'utilisation de Kinect .....	14

2.7.1 Robotique : .....	14
2.7.2 Réalité augmentée : .....	14
2.7.3 Application interactive .....	15
2.7.4 Ecologie et Domotique .....	15
2.7.5 Domaine médical.....	15
3.0 Protocoles et planifications .....	16
3.1 Création d'activités .....	16
3.2 Ecriture des protocoles.....	17
3.2.1 Représentation des différentes activités.....	18
3.4 Choix des sujets tests .....	20
3.5 Environnement de test .....	21
4.0 Prises de données.....	23
4.1 Prise de vues des sujets avec Kinect 2.0. ....	23
5.0 Utilisation des données .....	25
5.1 Récupération des données .....	25
5.2 Exportation des données.....	26
5.2.1 Le barycentre.....	28
5.3 Analyse préalable des données.....	28
6.0 Analyse .....	31
6.1 Techniques d'analyse.....	31
6.2 Analyse en séries temporelles.....	34
6.2.1 Qu'est-ce qu'une série temporelle ? .....	34
6.2.2 Transformation des données.....	34
6.3 Workflow d'analyse de données .....	36

6.3.1 Méthodes d'analyses utilisées.....	36
6.4 Workflow d'analyses temporelles .....	38
6.4.1 Résultats des algorithmes .....	39
6.4.2 Validation croisée .....	40
6.5 Workflow d'analyses temporelles avec validation croisée.....	40
6.5.1 Résultats des algorithmes .....	41
6.6 Workflow d'analyses temporelles (réduction du nombre de points).....	43
6.6.1 Résultat des algorithmes sans validation croisée.....	44
6.6.2 Résultat des algorithmes avec validation croisée .....	45
7. Analyse de données en fréquence .....	46
7.0.1 Qu'est-ce qu'une fréquence ? .....	46
7.1 Workflow d'analyse fréquentielle complet .....	47
7.1.1 Résultats des algorithmes .....	47
7.1.2 Résultats des algorithmes avec validation croisée .....	48
7.3 Workflow d'analyse fréquentielle se basant sur le point minimum et maximum .....	50
7.3.1 Résultats des algorithmes .....	51
7.3.2 Résultats des algorithmes avec Cross-validation .....	52
8. Création d'un regroupement hiérarchique .....	54
8.1 Qu'est-ce que le regroupement hiérarchique .....	54
8.2 Workflow de regroupement hiérarchique .....	55
9.0 Analyse se basant sur des histogrammes normalisés .....	56
9.1 Transformation des données.....	56
9.2 K-means.....	56
9.3 Centroïdes .....	57

9.4 Histogrammes.....	57
9.5 Output de données utilisées .....	58
10. Analyse .....	59
10.1 Résultats des analyses sans validation croisée.....	59
10.2 Résultats des analyses avec validation croisée .....	60
11. Analyse sur différents nombres de centroïdes.....	62
11.1 Résumé des résultats sans validation croisée .....	62
11.2 Résumé des résultats avec validation croisée .....	63
12. Comparaison des différentes techniques d'analyses .....	64
12.1 Tableau comparatif des algorithmes .....	65
13. Hypothèses d'utilisations réelles.....	66
14. Bilan du travail et améliorations possibles.....	67
15. Annexes : .....	69
Références .....	70
Déclaration sur l'honneur.....	76

## Liste de figures

Figure 1 Fonctionnalités de Kinect for Xbox One .....	9
Figure 2: Protocole de marche .....	18
Figure 3: Protocole de course.....	18
Figure 4 : Protocole de saut.....	18
Figure 5: Protocole de position assise .....	19
Figure 6 : Cadre de prise de vue « Figure de l'auteur ».....	22
Figure 7 : Filtre permettant de gérer la latence avec Kinect 2.0 .....	24
Figure 8 : Programme de récupération de données « Figure de l'auteur » .....	25
Figure 9 : Fichier csv récupérant tous les éléments « Figure de l'auteur » .....	26
Figure 10 : Fichier csv récupérant le barycentre « Figure de l'auteur » .....	27
Figure 11 : Point de capture de Kinect 2.0 .....	27
Figure 12 : Evolution du barycentre d'un sujet pour l'activité assise « Figure de l'auteur »...	28
Figure 13 : Evolution du barycentre d'un sujet pour l'activité de saut «Figure de l'auteur » ..	29
Figure 14 : Evolution du barycentre d'un sujet pour l'activité de marche « Figure de l'auteur » .....	29
Figure 15 : Evolution du barycentre d'un sujet pour l'activité de course « Figure de l'auteur » .....	30
Figure 16 : Différence entre une série temporelle et une fréquence .....	31
Figure 17 : Différences entre deux activités de saut « Figure de l'auteur ».....	33
Figure 18 : Extrait d'un dataset d'activité « Figure de l'auteur ».....	34

Figure 19 : Extrait du fichier de data set en entrée de Knime « Figure de l’auteur » .....	35
Figure 20 : Arbre de décision.....	36
Figure 21 : K-Nearest Neighbor .....	37
Figure 22 : Logique Floue .....	37
Figure 23 : Extrait du workflow d’analyses se basant sur les séries temporelles dans Knime « Figure de l’auteur ».....	38
Figure 24 : Extrait du workflow d’analyses se basant sur les séries temporelles avec validation croisée « Figure de l’auteur » .....	40
Figure 25: Extrait du fichier d’entrée pour Knime avec les fréquences « Figure de l’auteur »	46
Figure 26 : Extrait du dataset min-max pour Knime « Figure de l’auteur » .....	50
Figure 27 : Exemple de dendrogramme .....	54
Figure 28 : Workflow de regroupement hiérarchique « Figure de l’auteur » .....	55
Figure 29 : Deux clusters K-means avec leur centroïdes .....	56
Figure 30 : A gauche : histogramme d’une activité non normalisée, à droite : histogramme d’une activité normalisée « Figure de l’auteur ».....	57
Figure 31 : Extrait du fichier d’entrée pour les analyses Knime « Figure de l’auteur ».....	58

## Liste de tableaux

Tableau 1 : Kinect 1.0 vs Kinect 2.0 .....	7
Tableau 2 : Système de coordonnées .....	14
Tableau 3 : Récapitulatif du panel de test « Données de l'auteur ».....	21
Tableau 4 : Moyenne du taux d'erreurs des algorithmes de classification sur 10 processus lancés « Tableau de l'auteur » .....	39
Tableau 5 : Moyenne du taux d'erreurs des algorithmes de classification sur 48 processus lancés avec validation croisée « Tableau de l'auteur » .....	42
Tableau 6 : Taux d'erreurs des modèles selon la durée en image (sans validation croisée) « Tableau de l'auteur » .....	44
Tableau 7 : Taux d'erreurs des modèles selon la durée en image (avec validation croisée) « Tableau de l'auteur » .....	45
Tableau 8 : Moyenne du taux d'erreurs des algorithmes de classification sur 10 processus lancés « Tableau de l'auteur » .....	47
Tableau 9 : Moyenne du taux d'erreurs des algorithmes de classification sur 48 processus lancés avec cross-validation « Tableau de l'auteur » .....	49
Tableau 10 : Moyenne du taux d'erreurs des algorithmes de classification sur 10 processus lancés « Tableau de l'auteur » .....	51
Tableau 11 : Moyenne du taux d'erreurs des algorithmes de classification sur 48 processus lancés avec cross-validation « Tableau de l'auteur » .....	53
Tableau 13: Moyenne du taux d'erreurs lancé sur 20 processus sans validation croisée « Tableau de l'auteur » .....	59
Tableau 14 : Moyenne du taux d'erreurs lancé sur 48 processus avec validation croisée	

« Tableau de l’auteur » .....	61
Tableau 15 : Moyenne du taux d’erreurs des algorithmes avec un nombre de centroïdes variable (sans validation croisée) « Tableau de l’auteur » .....	62
Tableau 16 : Moyenne du taux d’erreurs des algorithmes avec un nombre de centroïdes variable (avec validation croisée) « Tableau de l’auteur » .....	63
Tableau 17: Comparatif des techniques d'analyses « Tableau de l’auteur ».....	64
Tableau 18 : Comparatif des résultats d'algorithmes « Tableau de l’auteur » .....	65

## Abréviations et Glossaire

Activité	Action vidéo enregistrée par un sujet de test (saut, marche, course...)
Error Rate	Taux d'erreurs des modèles de classification
Workflow	Processus d'analyse de Knime
Cluster	Groupe, Segment de données
Domotique	Techniques électroniques permettant l'automatisation de bâtiments à l'aide de l'informatique et des télécommunications. <sup>[46]</sup>
Dataset	Set de données
Quaternion	Représente un nombre de type hypercomplexe
Input	Set de données utilisé en entrée pour un modèle d'analyse
Itération	Action de répéter un processus
V-C	Validation croisée
CSV	Format de fichier de type texte

## 1. Présentation du travail

Fin 2013, Microsoft présente sur le marché la nouvelle version de Kinect pour Windows, cet appareil est une caméra dernière technologie pouvant être utilisée dans de nombreuses applications.

Ce travail de Bachelor a pour but d'étudier cette technologie et d'en tirer des conclusions afin de déterminer à quel point Kinect 2.0 pourrait être intéressant dans des applications à venir. Ce travail est regroupé en plusieurs étapes :

- Recherche des spécificités de Kinect 2.0
- Test d'utilisation pratique
- Rédaction d'un rapport d'expérience

Le test d'utilisation pratique est le suivant :

Kinect 2.0 permet de détecter plusieurs activités du corps humain, certaines concentrées sur les mouvements de la main et d'autres sur des mouvements du visage. Mon but est de définir et d'enregistrer d'autres activités (le saut, la marche, ...) et ainsi de déterminer un algorithme permettant à Kinect 2.0 de reconnaître ces activités à l'avenir.

*A noter : il est possible que Microsoft déploie une mise à jour permettant de détecter ces activités au cours de ce travail. Le cas échéant le travail s'adaptera afin de détecter de nouveaux mouvements.*

## 1.1 Problématique

Depuis maintenant plusieurs années, la détection de mouvements se retrouve de plus en plus au sein d'applications que nous utilisons au quotidien, par exemple : un détecteur allumant la lumière.

L'étape suivante est de pouvoir définir des activités précises pour des individus. Peut-on détecter si une personne est en train de se gratter la tête juste en la filmant ? L'intérêt est certes moindre dans cet exemple, mais plus concrètement est-il possible de détecter si une personne fuit un danger juste en regardant sa façon de courir ?

Pour ce travail, j'ai cherché à détecter des activités basiques en me basant sur une technologie dernière génération : Kinect 2.0. Cette caméra permet de recomposer le squelette d'une personne juste sur une image vidéo, ce qui fait d'elle la plus adaptée pour pouvoir récolter des données servant à la détection d'activité.

J'espère ainsi pouvoir déterminer un modèle de données pertinent permettant de détecter plusieurs activités simples avec un faible taux d'erreurs. Ce résultat permettrait la création d'applications simples de détection d'activité et pourrait aussi servir de base à des recherches plus poussées qui débouchent sur des applications complexes dans des domaines comme la médecine.

## 1.1 Synthèse de présentation du Travail de Bachelor



Récupération des activités avec la caméra Kinect 2.0 :

- Choix des activités à analyser
- Choix de l'environnement de tests
- Choix des sujets de tests
- Contraintes d'utilisations
- Exportation des fichiers vidéo

Récupération des fichiers vidéo et transformation en données.

- Choix des données à récupérer
- Récupération des données
- Exportation des données

Récupération des données de la vidéo :

- Choix des nœuds de traitements utilisés
- Choix du meilleur algorithme de décision
- Récupération des données critiques
- Choix des critères d'analyse
- Ecriture du rapport

## 1.2 Outils utilisés

### 1.2.1 Kinect Studio

Kinect Studio est un outil qui aide à l'enregistrement et à la lecture de vidéos filmées avec Kinect. Il peut récupérer plusieurs flux de stream en même temps : Le flux infrarouge et le flux de profondeur ainsi que les différents points d'articulations du corps reconnus par Kinect. Il n'est pas possible de filmer et de sauvegarder la vidéo avec la caméra couleur HD.



L'utilisation de cet outil permet la lecture et l'écriture des flux de données vidéo qui pourront aider à déboguer des fonctionnalités, créer des scénarii de tests ou analyser la performance.

[31]

Kinect Studio offre également la possibilité de visualiser en 2D et en 3D le flux de Kinect en direct. Il est possible de sélectionner plusieurs options lors de l'enregistrement ou de la lecture des vidéos :

- Couleur noir-blanc
- Effet profondeur à base de gris
- Récupération des positions X et Y dans l'espace de la personne filmée
- Zoom de certains éléments [31]

Cet outil me sera utile dans ce travail pour la capture vidéo des activités effectuées par les sujets tests. Je pourrai ainsi visualiser leurs mouvements et les traiter dans un second temps pour en récupérer les données nécessaires.

### 1.2.2 Visual Studio 2013

Visual Studio 2013 est une suite complète d'outils et de services d'aides au développement informatique. C'est plus qu'un simple IDE (Integrated development environment). Cette suite d'outils permet la création d'applications pour le Windows Store ou le développement de site internet avec différents langages : .Net, C#, HTML/JavaScript, C++ etc... [33]



La version 2013 permet une meilleure intégration web via le cloud ainsi que l'utilisation de la méthodologie Agile au cours du développement. Cet outil est le seul permettant le développement d'applications avec Kinect, il m'est donc indispensable pour la création d'une application récupérant les coordonnées d'un sujet filmé avec Kinect 2.0.

### 1.2.3 Knime

Knime est un programme d'analyses de processus et de traitements de données. Ce programme Open-Source permet la manipulation et la transformation de données afin de pouvoir en tirer une analyse et une classification. <sup>[34]</sup>

Il est alors possible de prédire le résultat d'une donnée avec un certain pourcentage de probabilité. Cet outil me sera utile lors de la partie traitement des données récupérées par mon interface développée en C#.

Je pourrai déterminer la probabilité pour un nouveau sujet d'avoir effectué une certaine action.

Cet outil permet également la création de rapports via un serveur Knime qui illustrent de façon graphique le résultat des prédictions.



## 2. Etat de l'art

### 2.1 Kinect 1.0 vs Kinect 2.0

La version précédente de Kinect proposait de nombreuses fonctionnalités, mais la nouvelle version 2.0 n'est pas qu'une simple mise à jour, elle offre de nouvelles possibilités. Les composants techniques de la version 2.0 ont largement évolué <sup>[11; 1]</sup>.

<b>Éléments</b>	<b>Kinect 1.0</b>	<b>Kinect 2.0</b>
<i>Visuel</i>		
<i>Date de sortie</i>	16 juin 2011	22 novembre 2013
<i>Angle de vue</i>	43° verticale par 57° horizontale	70° horizontale par 60° verticale
<i>Distance d'utilisation recommandée</i>	1.35 m	1.8 m
<i>Inclinaison verticale</i>	±27°	Aucun
<i>Images par seconde (caméra de profondeur et infrarouge)</i>	30 Images par seconde	30 Images par seconde
<i>Latence</i>	102ms	20-50ms
<i>Champ de profondeur</i>	0.4m – 4.0m	0.4m – 4.5m
<i>Flux en couleur</i>	640 x 480	1920 x 1080
<i>Flux de profondeur</i>	320 x 240	512 x 424
<i>Flux infrarouge</i>	Aucun	541 x 424 (active camera)
<i>Flux audio</i>	4-micros	4-micros
<i>Format audio</i>	16-kHz, 24-	4 micros fonctionnant à 48 kHz
<i>USB</i>	2.0	3.0

Tableau 1 : Kinect 1.0 vs Kinect 2.0

## 2.3 Fonctionnalités de Kinect pour Xbox One

Kinect pour Xbox One est sortie le 22 novembre 2013, cet appareil était automatiquement livré avec la nouvelle console de jeu Xbox de Microsoft. Kinect pour Xbox One possède une caméra intégrée de 30 images par seconde avec une qualité HD de 1080p.

Cette caméra est capable de gérer jusqu'à 2 gigas de données par seconde afin de lire l'environnement qui l'entoure. Elle utilise la technologie « Time in Flight »<sup>[25]</sup> qui permet une capture rapide de l'image, cette Kinect for Xbox One intègre également des microphones. Elle propose plusieurs fonctionnalités dont ne dispose pour l'instant pas la Kinect 2.0 pour Windows :

- La navigation dans le menu d'une Xbox One par le geste ou la voix
- La détection de commandes vocales pour lancer certaines actions.
- La reconnaissance biométrique des personnes qui permet de reprendre le contrôle de la console par le son de la voix. Il faut avoir été enregistré sur la console au préalable.<sup>[26]</sup>

A partir de tests préliminaires plusieurs fonctionnalités intéressantes sont ressorties :

- Kinect pour Xbox One détecte une liste complète de commandes vocales comme « Xbox démarrer », « Xbox aller à », cette détection se fait par rapport à la langue de paramétrage de la console.
- Kinect pour Xbox One peut détecter les mouvements des mains à partir d'une liste d'activités qui permet de sélectionner un élément, de naviguer à travers un menu, de revenir à l'écran d'accueil ou de maintenir une sélection.
- Kinect pour Xbox One peut également reconnaître les joueurs qui se trouvent en face de la caméra, la console adaptera alors les préférences utilisateurs au joueur présent.

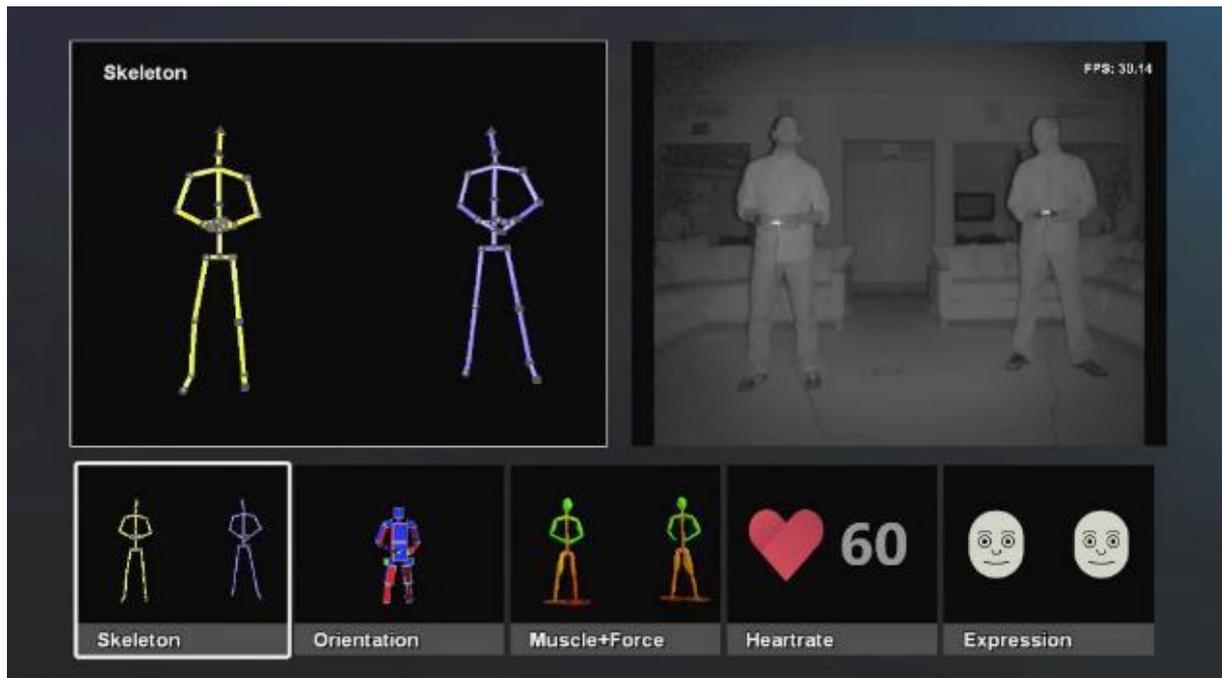


Figure 1 Fonctionnalités de Kinect for Xbox One

## 2.4 Les fonctionnalités actuelles de Kinect 2.0

Kinect version 2.0 pour Windows est actuellement en version bêta, les fonctions suivantes présentées sont susceptibles de changer et d'évoluer. Au moment de l'écriture de ces lignes, la mise à jour d'avril du SDK de Kinect pour Windows est disponible. Voici donc les listes des fonctions actuelles :

- Caméra couleur : Récupération des données de la caméra couleur
- Caméra de profondeur : Récupération des données de la caméra de profondeur
- Reconnaissance simple du corps : Récupération des données du squelette sur chaque humain que Kinect détecte.
- Caméra infrarouge : Récupération des données de la caméra infrarouge
- Mapping de coordonnées : Corrélation des données de la caméra de profondeur avec des valeurs RGB. [3; 11].

## 2.5 Les fonctionnalités futures de Kinect 2.0

Kinect 2.0 pour Windows aura plusieurs fonctionnalités implémentées dans les mois à venir. Certaines sont déjà disponibles dans la version de Kinect 2.0 pour Xbox One. Voici une liste non exhaustive de ces fonctionnalités :

- Capture de données de la part de plusieurs senseurs Kinect 2.0 en même temps.
- Reconnaissance faciale <sup>[3]</sup>
- Une API audio permettant la gestion de commandes vocales via Microsoft Speech <sup>[30]</sup>

## 2.6 Description des fonctionnalités actuelles

Cette partie illustre les différentes fonctionnalités de Kinect 2.0 en donnant quelques détails au niveau hardware et software. Certaines fonctionnalités utilisent des algorithmes propriétaires et sont basées sur des présomptions de la part de personnes sur internet.

### 2.6.1 Caméra couleur

C'est la fonctionnalité la plus basique disponible sur Kinect 2.0, elle est utilisée pour récupérer une image Full HD et envoyer les images au format YUY2 par défaut. Il est possible de convertir les données dans un autre format beaucoup plus léger et facile à traiter pour un processeur. <sup>[3]</sup>.

### 2.6.2 Caméra de profondeur

Cette caméra est utilisée pour récupérer les mouvements d'un utilisateur dans l'espace. Elle est composée d'une caméra infrarouge photosensible. La caméra de profondeur semble fonctionner en analysant un modèle de tacheture de lumière laser infrarouge envoyée par un 2<sup>ème</sup> composant. [15].



Cette caméra est également capable de traquer les index du corps au pixel correspondant, déterminé par un algorithme propriétaire de détection du squelette.

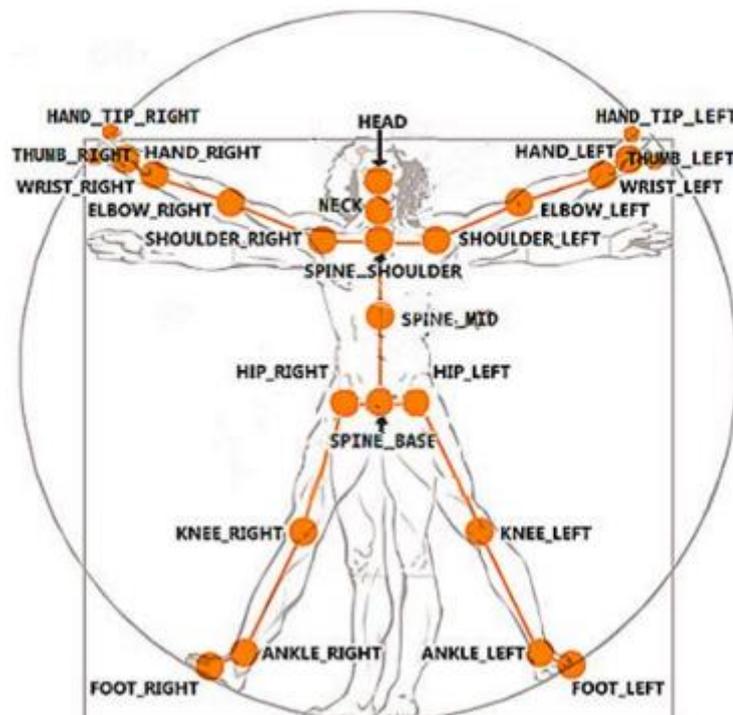
### 2.6.3 Reconnaissance basique du corps

Pour reconnaître les mouvements utilisateurs, ce qui reste l'utilisation principale de Kinect, l'appareil peut ajouter un squelette virtuel sur l'utilisateur via la caméra de profondeur.

La reconnaissance du squelette semble être basée sur une utilisation de la « decision forest » [16], récupérée sur plus d'un million d'exemples d'essais. Il y a 2 sous étages. Le premier avec les images de la caméra de profondeur de 100'000 images qui reconnaissent le squelette, (récupéré à partir de « motion capture ».) et utilisé dans la « decision forest ». Le second transforme l'image du corps humain en squelette.

Kinect 2.0 est capable de reconnaître jusqu'à 6 corps en même temps avec 25 articulations citées ci-dessous [3]:

SpineBase	WristLeft	HipLeft	AnkleRight
SpineMid	HandLeft	KneeLeft	FootRight
Neck	ShoulderRight	AnkleLeft	SpineShoulder
Head	ElbowRight	FootLeft	HandTipLeft
ShoulderLeft	WristRight	HipRight	ThumbLeft
ElbowLeft	HandRight	KneeRight	HandTipRight



*Articulations du squelette reconnus par Kinect 2.0*

Chaque articulation a une position 3D avec l'orientation indiquée par un quaternion. L'état de chaque articulation ne peut pas être reconnu ou déduit par Kinect 2.0. Par exemple, il ne reconnaît pas si un utilisateur est face ou dos à lui.

Kinect 2.0 peut reconnaître plusieurs états concernant les mains de l'utilisateur (limité à deux utilisateurs) :

- Poings fermés
- Mains ouvertes
- Doigts en lasso (le V de la victoire avec les doigts)
- Inconnu
- Non détecté

Kinect 2.0 peut détecter certaines activités du visage de l'utilisateur :

- Les yeux ouverts
- Les yeux fermés
- La bouche ouverte
- La bouche fermée
- Détection de lunettes
- Détection de l'émotion (sujet souriant ou neutre)

Toutes ces activités sont enregistrées dans le SDK et sont reconnues selon 4 critères de probabilité (inconnu, non, peut-être, oui).

#### 2.6.4 Caméra infrarouge

La caméra infrarouge utilisée par la caméra de profondeur peut également être utilisée indépendamment afin de produire des images en noir et blanc.

La caméra infrarouge est un élément actif de Kinect 2.0 qui imite une source de lumière infrarouge afin d'illuminer l'environnement. La caméra reflète la lumière de la même façon qu'une caméra classique. Cette caméra permet une vue claire dans un environnement avec très peu de lumière. Il y a deux formes d'utilisation : infrarouge et infrarouge à longue exposition. <sup>[3; 6]</sup>.

## 2.6.5 Mapping de coordonnées

Grâce à sa variété de senseurs, Kinect 2.0 possède 3 systèmes de coordonnées indépendants

<u>Nom</u>	<u>Application</u>	<u>Dimensions</u>	<u>Unité</u>	<u>Gamme</u>	<u>Origine</u>
<i>ColorSpacePoint</i>	Couleur	2	pixels	1920x1080	Coin en haut à gauche
<i>DepthSpacePoint</i>	Index de couleur, caméra de profondeur	2	pixels	512x424	Coin en haut à gauche
<i>CameraSpacePoint</i>	Détection du corps	3	Mètres	-	Caméra infrarouge et profondeur

Tableau 2 : Système de coordonnées

Le mapping de coordonnées permet par exemple la création d'un effet « fond vert » sans avoir besoin d'utiliser un fond particulier.

## 2.7 Cas d'utilisation de Kinect

### 2.7.1 Robotique :

Le laboratoire « Jet Propulsion Laboratory » de la Nasa essaie de manipuler un bras robotique en utilisant Kinect 2.0. Des ingénieurs sont capables de connecter Kinect 2.0 à un Oculus Rift afin de créer une interface des plus immersives. Grâce à l'utilisation simultanée de ces deux technologies, ils sont capables de contrôler entièrement la rotation d'un bras robotisé. <sup>[23]</sup>.

### 2.7.2 Réalité augmentée :

La version 1.0 de Kinect a donné naissance à de nombreuses applications utilisant la réalité augmentée telles que Kinect Fitnect, une application permettant de voir nos propres vêtements sur nous sans devoir les essayer. Un écran retransmet notre image avec le vêtement imprimé par-dessus. <sup>[27]</sup>.

### 2.7.3 Application interactive

Kinect offre également la possibilité de créer et gérer des murs interactifs afin de naviguer à travers des menus ou jouer à de petits jeux simples basés sur la détection de mouvements. [28].

### 2.7.4 Ecologie et Domotique

Le senseur Kinect permet également grâce à son système de microphone de détecter des ordres vocaux et ainsi automatiser une maison en allumant ou éteignant des lumières selon le nombre de personnes dans la pièce. [29].

### 2.7.5 Domaine médical

Des applications de coaching pour la réhabilitation sont développées avec Kinect, ces véritables coach virtuels encouragent les patients et leur permettent d'imiter des mouvements de rééducation. De nombreux pays essaient d'améliorer cette technologie afin de la rendre toujours plus précise. [23]

## 3.0 Protocoles et planifications

### 3.1 Création d'activités

La première étape de mon travail a été de déterminer et planifier le protocole des activités qui seront filmées puis analysées. Cette tâche initiale a pour but de définir les mouvements de base qui me serviront pour la suite de mon travail de Bachelor.

En me basant sur les 25 points d'articulations du corps disponibles avec Kinect 2.0, il m'est apparu que des activités complexes, comme se gratter la tête, ne pouvaient pas être détectées.

Je me suis donc décidé sur quatre activités basiques qui pourront être employées dans le futur comme base à des mouvements plus complexes.

Voici ces quatre activités :

- Détection de la marche
- Détection de la course
- Détection du saut
- Détection de la position assise

Ces quatre activités ont la particularité d'être très différentes de par leur sens du mouvement :

La marche et la course : Mouvement de gauche à droite

Le saut : Mouvement de bas en haut

La position assise : Mouvement de recul puis vers le bas

Ces directions différentes permettront une meilleure analyse des données par la suite. Le nombre de quatre activités a été décidé par rapport à la durée de réalisation de ce travail.

*A noter que Microsoft peut très bien ajouter la détection d'une ou plusieurs de ces activités dans les mises à jour à venir. C'est une contrainte mineure car la partie primordiale de ce travail*

*se trouve dans la manipulation des données avec Knime. Dans le cas où Microsoft proposerait la détection de ses activités, à l'avenir, je pourrai comparer la précision de leur algorithme par rapport à mon modèle de données.*

Grâce à cette première recherche, j'ai pu commencer la rédaction des protocoles d'activités qui délimiteront exactement les mouvements des sujets ainsi que leurs contraintes.

### 3.2 Ecriture des protocoles

Une fois les activités choisies, il a fallu définir un protocole strict pour chaque action. Ce protocole a pour objectif de fixer le cadre de l'activité afin que chaque sujet exécute le mouvement dans les mêmes conditions et de la même façon.

Chaque protocole définit plusieurs caractéristiques clés comme la durée du test, la distance du capteur face au sujet, l'environnement qui l'entoure ainsi que la description du mouvement à exécuter.

Par exemple, pour le protocole de marche, j'ai choisi de placer la caméra au maximum de portée du sujet afin d'avoir le plus de recul possible ce qui permet de pouvoir enregistrer quelques pas supplémentaires. La portée est assez faible en fin de compte : les sujets se trouvent à 2.5 mètres du capteur. En effet, la capture de tous les points d'articulation n'est possible qu'à faible distance.

La durée des tests est fixée à 15 secondes par activité, la caméra Kinect 2.0 peut en théorie capturer 30 points par seconde ce qui fait un total de 450 points pour chaque activité. Cette quantité de données est suffisante pour être analysée.

Chaque protocole possède aussi des caractéristiques spécifiques, par exemple le protocole de saut impose au sujet un saut sans élan tout en gardant les bras le long du corps. Ces critères sont définis afin d'éviter l'ajout de données non essentielles lors de l'analyse.

En effet, je ne cherche pas à détecter l'activité du saut avec élan mais uniquement le saut.

Vous pouvez retrouver les quatre protocoles écrits pour ce travail en annexe.

3.2.1 Représentation des différentes activités

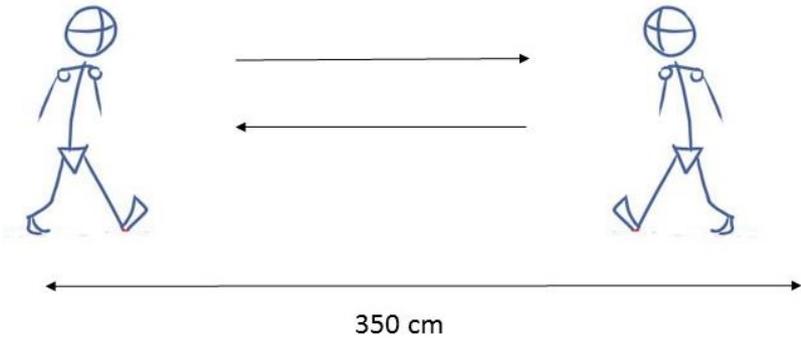


Figure 2: Protocole de marche

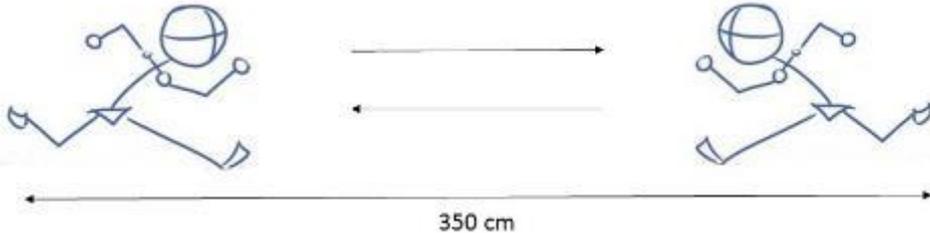


Figure 3: Protocole de course

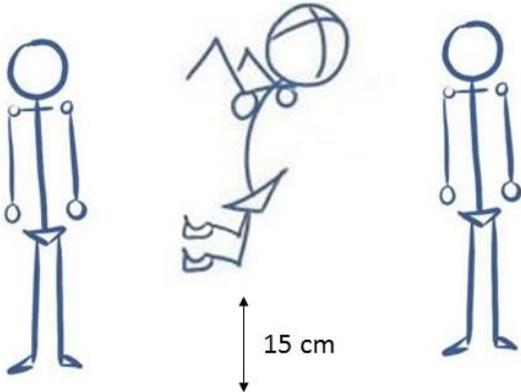


Figure 4 : Protocole de saut

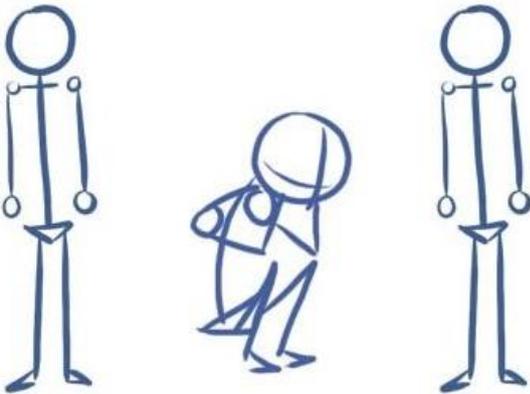


Figure 5: Protocole de position assise

### 3.4 Choix des sujets tests

Le choix du panel de test est également important, il a tout d'abord fallu déterminer le nombre de personnes qui participeraient au test. Après discussion avec les superviseurs du projet et vu la faible durée disponible, il ne serait pas possible de tester les protocoles sur un grand nombre de personnes.

Il a été décidé de sélectionner 10 personnes, cinq personnes serviront pour la partie « Training » de l'analyse et définiront le modèle ; les cinq autres testeront le modèle afin de valider leur activité. Le nombre de personnes aura une influence sur le résultat final de la classification des activités. En effet, avec plus de personnes le résultat serait certainement plus précis ; cependant, faute de temps et de moyens à disposition, il a été décidé que ce panel de départ serait suffisant pour mener à bien ce projet.

Dans un premier temps, j'envisageais de sélectionner 10 personnes avec une morphologie similaire afin d'améliorer l'homogénéité des données. Cependant cette hypothèse était erronée : la morphologie de la personne ne doit pas avoir d'influence sur la détection du type de mouvement. De plus Kinect 2.0 est sensée pouvoir détecter les activités sur une population normale et non tronquée avec des personnes similaires.

J'ai récupéré certains critères concernant chaque sujet du panel test :

- Age
- Taille
- Sexe
- Poids

*Ces facteurs n'auront en théorie aucune incidence sur mes protocoles de tests, cependant il se peut qu'ils servent dans le cas d'études complémentaires se basant sur ce travail.*

Sujet			Taille	Poids
No	Age	Sexe	(cm)	(kg)
1	26	Homme	179	89
2	23	Homme	175	95
3	28	Homme	183	86
4	24	Homme	183	90
5	27	Homme	180	120
6	23	Homme	177	79
7	24	Homme	183	78
8	33	Homme	178	71
9	30	Homme	172	60
10	26	Femme	173	54
11	20	Homme	190	95
12	22	Femme	170	60

Tableau 3 : Récapitulatif du panel de test « Données de l'auteur »

### 3.5 Environnement de test

L'environnement de test est important pour plusieurs raisons, c'est lui qui déterminera les constantes lors des tests : distance à la Kinect 2.0, position de départ du sujet, exposition à la lumière.

La fonctionnalité de détection des points d'articulation de Kinect est ce qui importe le plus, l'arrière-plan de la caméra Kinect n'a que peu d'importance.

Les tests se réaliseront dans une pièce fermée avec une lumière artificielle afin d'éviter une exposition trop forte au soleil. En effet, Kinect se base principalement sur une caméra infrarouge pour récupérer les points d'articulations, une trop forte exposition à la lumière rend cette caméra inutilisable. Une alternative en extérieur est donc inenvisageable.

Certains critères demeurent cependant importants, la taille de la pièce doit être suffisante pour que le sujet puisse effectuer le mouvement correctement. Le recul de la caméra doit également être adapté afin de capter entièrement les mouvements du sujet.

La caméra Kinect 2.0 se situe légèrement en hauteur d'une petite table haute, et est face au sujet. Une zone au sol est délimitée pour permettre au sujet de connaître l'espace disponible pour la marche ou la course.

J'ai donc sélectionné une pièce fermée avec un espace de test contre un mur, le sujet se déplace sur 3.5 m de large sur un maximum de 1.10 m de long. Au centre se trouve un point de repère pour le positionnement du point de départ des mouvements suivants :

- position assise
- saut

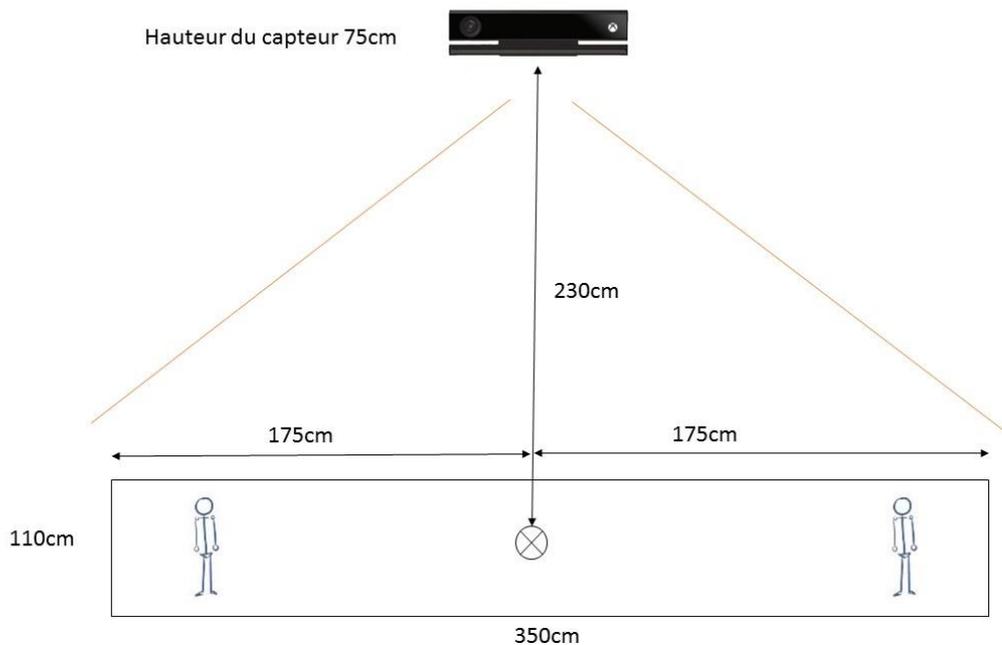


Figure 6 : Cadre de prise de vue « Figure de l'auteur »

## 4.0 Prises de données

### 4.1 Prise de vues des sujets avec Kinect 2.0.

La prise de vue des sujets est l'étape cruciale de tout le processus de réalisation de ce projet. Cette étape consiste à filmer tous les protocoles d'activités sur les sujets de tests dans un environnement déterminé.

Le challenge de cette tâche consiste à réussir à maintenir une égalité dans l'enregistrement de chaque vidéo. Il ne faut pas qu'une vidéo soit plus courte qu'une autre par exemple, ou que l'environnement ne soit plus adapté d'une prise de vue à une autre.

La durée des vidéos est fixée à 15 secondes. Chaque séquence sera enregistrée grâce au programme Kinect Studio et sauvegardée dans le répertoire Kinect au format .xef<sup>[35]</sup>

Chaque protocole de test est exécuté de la même façon, les sujets effectuent le même mouvement à la même position de départ. Par exemple, pour le protocole de marche le sujet démarre sur la gauche pour se diriger vers la droite.

En théorie, je devrais récupérer 450 images par activité (30 images par seconde pendant 15 secondes) ce qui reviendrait à avoir 450 points à comparer afin de détecter l'activité.

En pratique, la caméra Kinect 2.0 ne capte pas réellement 30 images par seconde mais entre 27-28 images par seconde. De plus Kinect 2.0 bénéficie d'un système de filtre qui permet d'améliorer la stabilité de l'image et de corriger, prédire certains points du squelette sur les images manquantes.

Ce système va arrondir les points récupérés et va aussi en supprimer certains qu'il n'arrive pas à analyser. <sup>[37,37]</sup>

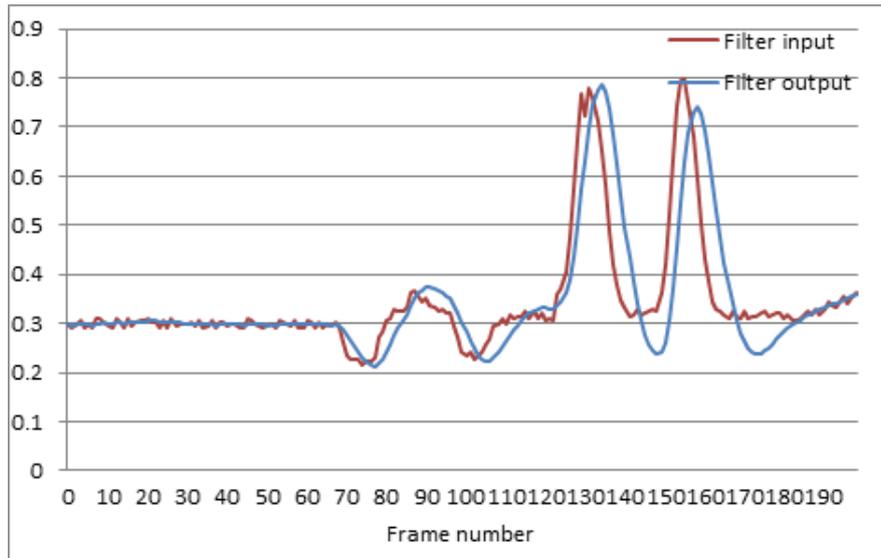


Figure 7 : Filtre permettant de gérer la latence avec Kinect 2.0

Ces deux éléments ont pour conséquence de diminuer le nombre de points capturés pour chaque vidéo. Deux personnes effectuant donc la même action avec la même durée ne généreront pas le même nombre de points lors de l'exportation de mes données.

## 5.0 Utilisation des données

### 5.1 Récupération des données

Cette tâche est la suite logique des prises de vue avec Kinect Studio. Tous les fichiers vidéo sont maintenant enregistrés et prêts à être exportés sous un format permettant une analyse des données. La récupération des données se fait à l'aide d'un programme développé spécialement pour ce travail.

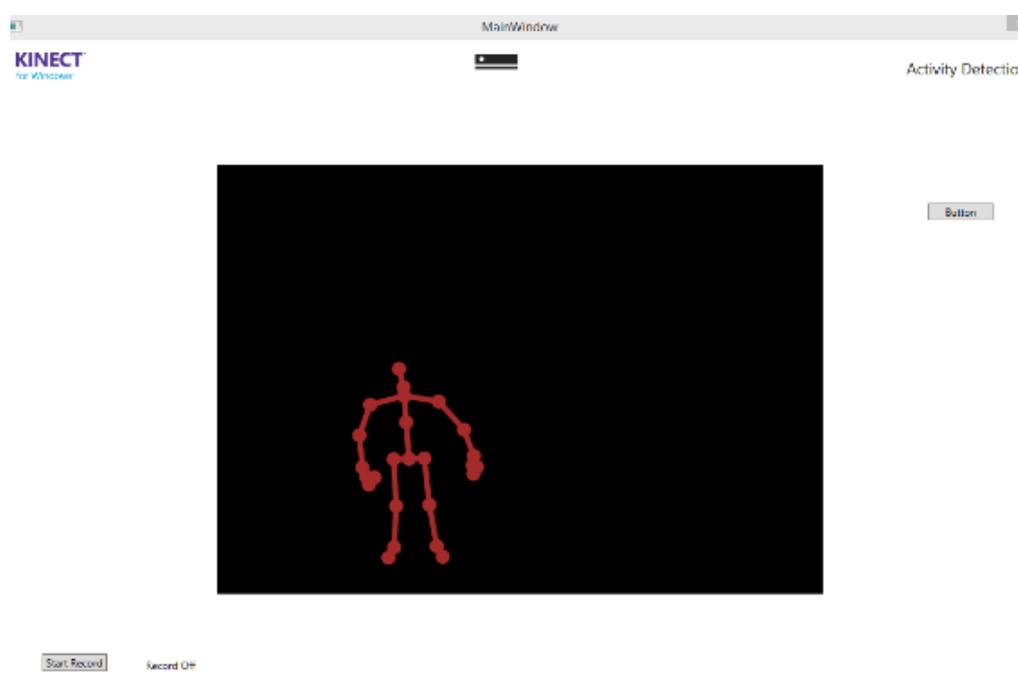


Figure 8 : Programme de récupération de données « Figure de l'auteur »

Ce programme a pour fonction de récupérer les points d'articulations du squelette formé par la caméra de Kinect 2.0. La caméra Kinect 2.0 permet une capture à environ 30 images par seconde, ce qui en pratique est légèrement inférieur.

Le programme affiche la forme du squelette en mouvement effectuant l'action. Cette animation permet d'identifier que le mouvement capturé est bien le bon.

Le processus de capture programme fonctionne de deux façons différentes :

- Récupération des mouvements en direct depuis Kinect 2.0
- Récupération des mouvements depuis un fichier vidéo préenregistré et lancé sur Kinect Studio.

## 5.2 Exportation des données

Une fois la vidéo lancée ou le mouvement effectué le programme écrit deux fichiers csv permettant deux récupérations différentes. Le premier fichier récupère chaque point d'articulations avec sa position dans l'espace.

	A	B	C	D	E
1	JointType	Time	X	Y	Z
2	SpineBase	7,54938E+11	0,01401972	-0,02875001	2,560978
3	SpineMid	7,54938E+11	0,00219281	0,2360281	2,454176
4	Neck	7,54938E+11	-0,00938281	0,4935887	2,336053
5	Head	7,54938E+11	-0,01882295	0,6459597	2,293004
6	ShoulderLef	7,54938E+11	-0,1867901	0,3743278	2,354627
7	ElbowLeft	7,54938E+11	-0,2827421	0,1442158	2,469727
8	WristLeft	7,54938E+11	-0,2902799	-0,09876218	2,487249
9	HandLeft	7,54938E+11	-0,2800181	-0,1754198	2,481454
10	ShoulderRig	7,54938E+11	0,1584944	0,3810345	2,362464
11	ElbowRight	7,54938E+11	0,2789916	0,1548304	2,497073
12	WristRight	7,54938E+11	0,3119958	-0,08679289	2,505745
13	HandRight	7,54938E+11	0,2977687	-0,1596014	2,493456
14	HipLeft	7,54938E+11	-0,06477124	-0,02990812	2,526674
15	KneeLeft	7,54938E+11	-0,08757722	-0,3741766	2,370862
16	AnkleLeft	7,54938E+11	-0,07566537	-0,7002756	2,46679
17	FootLeft	7,54938E+11	-0,0986663	-0,7829521	2,401217
18	HipRight	7,54938E+11	0,09244284	-0,02684338	2,528601
19	KneeRight	7,54938E+11	0,1198848	-0,3726434	2,38452
20	AnkleRight	7,54938E+11	0,1010737	-0,6958324	2,464839
21	FootRight	7,54938E+11	0,1392044	-0,7786575	2,420141
22	SpineShould	7,54938E+11	-0,00652579	0,4303049	2,367647
23	HandTipLeft	7,54938E+11	-0,2667128	-0,2460272	2,486932
24	ThumbLeft	7,54938E+11	-0,3043571	-0,1856962	2,469471
25	HandTipRigh	7,54938E+11	0,2921799	-0,231449	2,492765
26	ThumbRight	7,54938E+11	0,3203977	-0,1692536	2,472

Figure 9 : Fichier csv récupérant tous les éléments « Figure de l'auteur »

A noter que ce fichier csv ne sera pas utilisé dans la suite de mon travail, je le présente afin de montrer les autres possibilités qu'offre Kinect 2.0 pour la récupération des données. Ce fichier pourrait servir dans d'autres travaux à l'avenir ou comme amélioration de celui-ci.

Le deuxième fichier récupère le barycentre du squelette étudié et donne sa position dans l'espace. Ce fichier servira pour toute l'analyse et la détermination de l'algorithme de tri que j'utiliserai dans la phase suivante de mon travail.

A	B	C	D	E
JointType	Frame	X	Y	Z
BaryCenter	1	0,00625339	-0,09131006	2,445939
BaryCenter	2	0,00572521	-0,1060254	2,447037
BaryCenter	3	0,00447238	-0,1049806	2,438574
BaryCenter	4	0,00332473	-0,09662695	2,431519
BaryCenter	5	0,00092329	-0,05409301	2,427447
BaryCenter	6	-0,00503641	0,01605535	2,41965
BaryCenter	7	-0,00618555	0,09190736	2,415219
BaryCenter	8	-0,01320315	0,1709388	2,418983
BaryCenter	9	-0,01511995	0,2073572	2,430392

Figure 10 : Fichier csv récupérant le barycentre « Figure de l'auteur »

Dans chaque fichier on récupère un point sélectionné ainsi que l'image exacte où elle a été sauvegardée. On retrouve également la position X, Y, Z par rapport au point central de Kinect 2.0

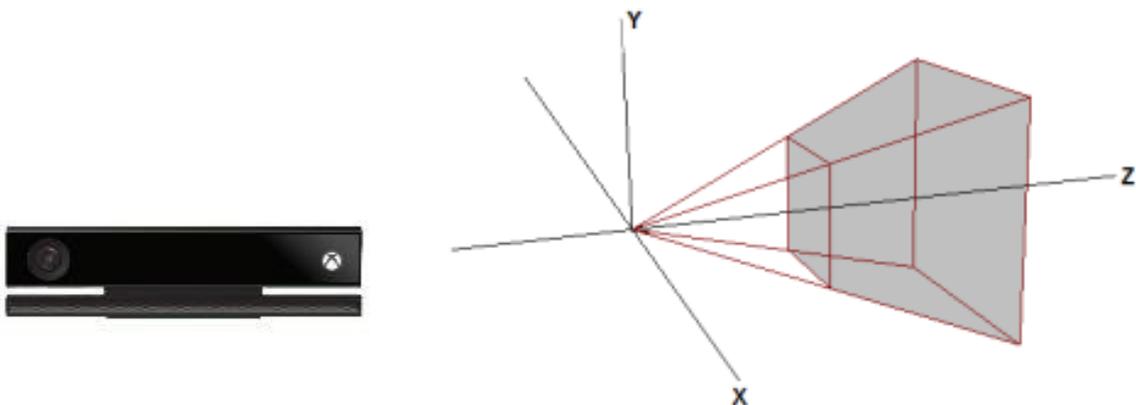


Figure 11 : Point de capture de Kinect 2.0

Kinect possède un point de base de capture, qui correspond au point 0 ; 0 ; 0. Les données des fichiers correspondent à la différence en mètre des points x, y, z par rapport à ce point de base.

### 5.2.1 Le barycentre

A ne pas confondre avec le centre de gravité, le barycentre ou centre de masse permet de déterminer le point central d'une forme géométrique. Je l'utilise dans ce travail afin d'estimer le centre du squelette en mouvement. En effet, il n'est pas possible de calculer un centre de gravité exacte sans la masse, c'est pourquoi le barycentre est utilisé afin de nous donner une estimation de celui-ci. Le calcul que j'utilise est simple, je fais la moyenne de chacun des 25 points d'articulation à leur position X, Y, Z et j'utilise ce point comme point centre <sup>[35]</sup>.

### 5.3 Analyse préalable des données

A parti des fichiers csv du premier sujet j'ai pu créer les premières courbes représentant les mouvements, en voici les représentations.

*A noter : sur les graphiques suivants le point de base correspond au point 0.0.0 dans l'espace soit le point central de la sortie optique de Kinect 2.0*

#### a) La position assise

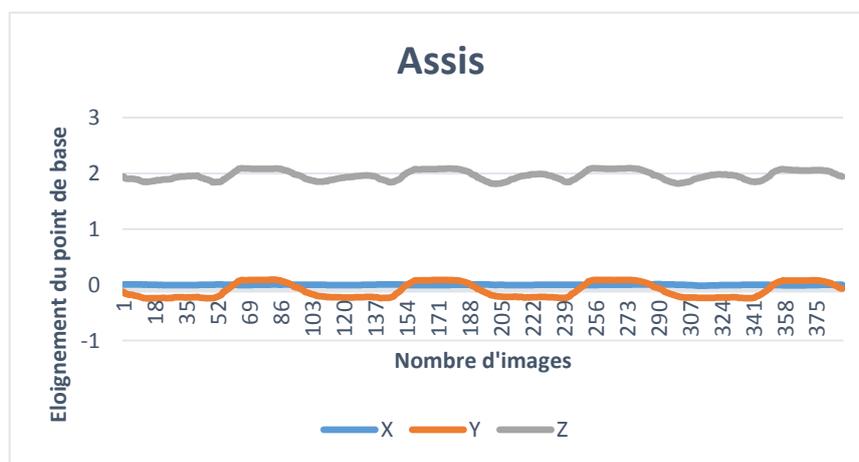


Figure 12 : Evolution du barycentre d'un sujet pour l'activité assise « Figure de l'auteur »

b) Le saut

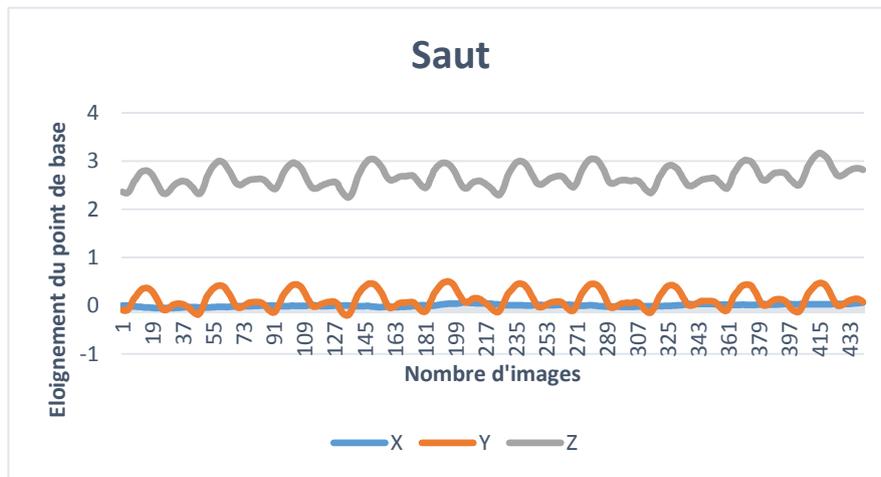


Figure 13 : Evolution du barycentre d'un sujet pour l'activité de saut «Figure de l'auteur »

c) La marche

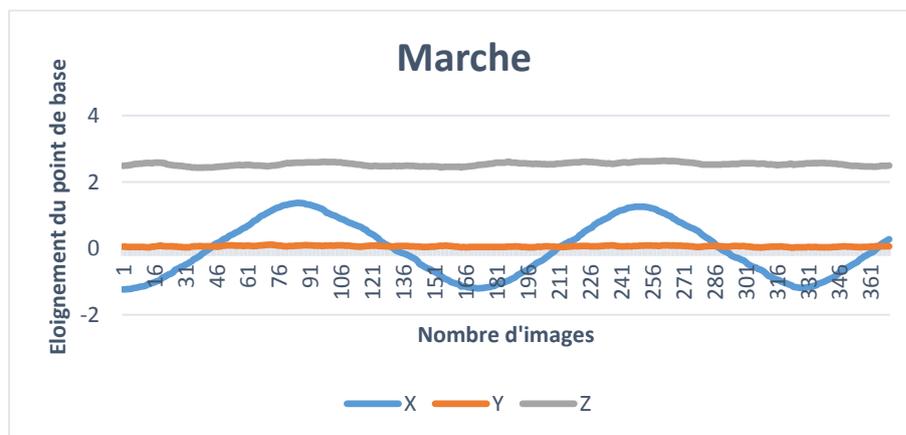


Figure 14 : Evolution du barycentre d'un sujet pour l'activité de marche « Figure de l'auteur »

d) La course



Figure 15 : Evolution du barycentre d'un sujet pour l'activité de course « Figure de l'auteur »

On peut voir des similitudes dans les courbes de la position assise et dans celles du saut. En effet quand on s'assied et qu'on se relève notre centre de gravité va baisser puis s'élever à nouveau. Lors d'un saut l'effet est le même, notre centre de gravité va monter et s'abaisser à nouveau.

On remarque aussi que la courbe de course ressemble fortement à la courbe de marche, il y a simplement un mouvement plus rapide et une plus grande oscillation dans l'axe des X.

## 6.0 Analyse

### 6.1 Techniques d'analyse

Dans un premier temps, je pensais me baser sur le fichier de l'illustration 3, ce fichier contient la position dans l'espace du sujet pendant une période de temps donnée. Ce fichier m'aurait permis de rechercher une courbe moyenne qui représenterait chaque mouvement puis ensuite pouvoir classer mes données en me basant sur cette courbe. Il m'est apparu une autre possibilité d'analyse, la transformation des données temporelles en fréquences.

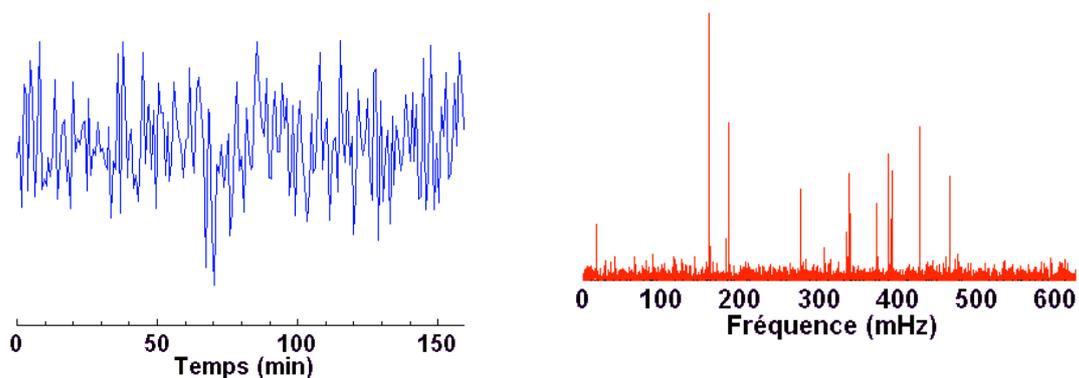
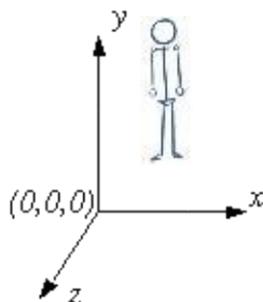


Figure 16 : Différence entre une série temporelle et une fréquence

*(A noter, les graphiques précédents n'ont aucun rapport avec les données récupérées pour ce travail, il ne sont là que pour illustrer la différence entre série temporelle et fréquence)*

L'analyse sous forme de fréquence a pour but d'obtenir une meilleure robustesse de mon modèle de données. En effet, les séries temporelles ne fonctionnent que parce que chaque sujet a été filmé de la même façon avec le même angle de vue et avec la même durée. Il y a donc de nombreux cas où mes analyses ne pourraient plus être valides. En voici un exemple simple :



Une personne se mettant à courir de profil à la caméra verra ce point bouger sur l'axe de X. Si maintenant, cette personne se met à courir face à la camera d'avant en arrière, le mouvement sera le même mais l'angle différent.

La transformation de mes données en fréquence permettrait ainsi de s'affranchir de ce problème et analyserait alors la fréquence du mouvement. Cependant, ce procédé comporte aussi certains problèmes : il est très facile de faire passer un mouvement pour un autre en bougeant à une fréquence similaire.

J'ai donc décidé, dans un premier temps, d'analyser mes données selon les séries temporelles. Ce choix est motivé par l'environnement où ont été prises les données. En effet, avec un environnement cadré comme celui qui a été mis en place, le sujet n'a jamais été amené à bouger ni à effectuer les mouvements d'une autre manière.

Il faut cependant tenir compte de plusieurs éléments qui influenceront mes analyses :

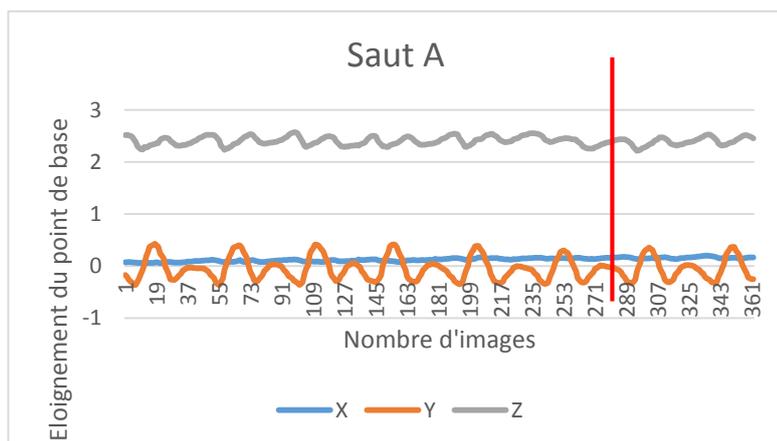
Le nombre de points de coordonnées pour chaque activité :

En exportant les données de mes fichiers vidéo il s'est avéré qu'ils n'avaient pas tous la même quantité de points à la sortie. Ce phénomène est dû à deux éléments : la capacité de la caméra Kinect 2.0, qui n'est pas réellement de 30 images seconde mais de 27 à 28 images seconde, ainsi qu'au filtre que Kinect applique elle-même supprimant les images dans lesquelles elle n'arrive pas à lire les données. (Voir point 4. Prise de vue)

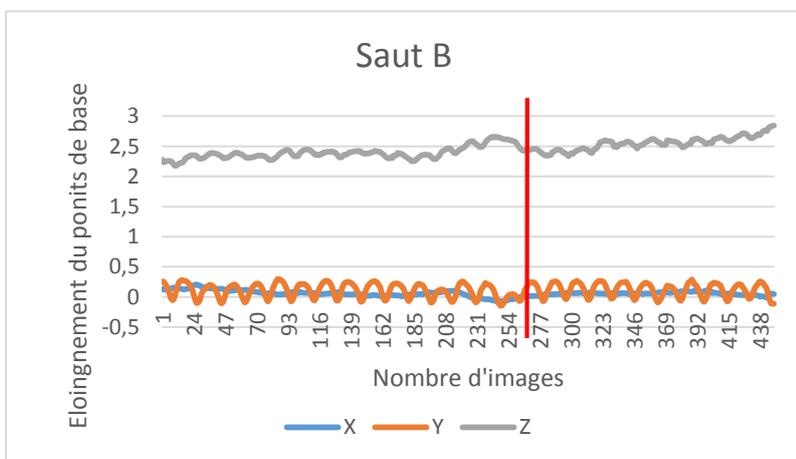
Le nombre de points récupérés pour chaque activité étant différent j'ai décidé de me baser sur le nombre de points minimum qui a pu être récupéré sur une activité soit 275 points.

En ne prenant que les 275 premiers points de chaque fichier j'ai donc pu construire un modèle pour me baser sur une analyse en série temporelle.

A noter : les données ne seront pas correctement uniformes d'une activité à l'autre. En effet, si une première activité de saut de 15 secondes a généré 350 points et une deuxième 300 points, les données seront décalées lorsque l'on récupèrera les 275 premiers points. Ce problème peut être réglé en se basant sur un histogramme des données, ce qui les nettoiera.



Prenons l'exemple d'un saut : les deux sujets avaient les mêmes consignes et ont effectué le même mouvement durant la même durée.



Pourtant les résultats sont très différents d'une exportation des données à une autre. De plus, je ne me limite qu'aux 275 premiers points de chaque vidéo. On remarque qu'à gauche du trait, pour le saut B, on effectue beaucoup plus d'oscillations que dans le saut A.

Figure 17 : Différences entre deux activités de saut « Figure de l'auteur »

Le fait de me limiter à 275 points faussera donc obligatoirement une partie de mon modèle.

## 6.2 Analyse en séries temporelles

### 6.2.1 Qu'est-ce qu'une série temporelle ?

Une série temporelle ou chronologique est une suite de valeurs numériques représentant l'évolution d'une variable à travers le temps. On peut alors étudier les courbes pour mieux analyser son comportement et ainsi pouvoir prévoir le comportement futur. On utilise alors des concepts de probabilité et de statistiques. <sup>[42]</sup>

### 6.2.2 Transformation des données

	A	B	C
1	X1	-1,235284	
2	Y1	0,05242087	
3	Z1	2,487758	
4	X2	-1,232472	
5	Y2	0,05098482	
6	Z2	2,493935	
7	X3	-1,227903	
8	Y3	0,04525353	
9	Z3	2,49838	
10	X4	-1,219447	
11	Y4	0,04410569	
12	Z4	2,505262	
13	X5	-1,211846	
14	Y5	0,04338188	
15	Z5	2,514747	

Dans un premier temps, il m'a fallu générer chaque fichier de données de chaque sujet pour chaque activité et de les comparer. Les fichiers n'ayant pas exactement le même nombre de points récupérés, je me suis basé sur le fichier en comportant le moins pour l'analyse. Je n'ai donc sélectionné que les 275 premières images de mes vidéos.

Chaque fichier représente un dataset de l'activité pour un sujet donné. Le fichier ci-après est un extrait de l'activité « Marche » pour le sujet 1.

Figure 18 : Extrait d'un dataset d'activité « Figure de l'auteur »

Chaque ligne représente un point récupéré à une image donnée de ma vidéo.

Afin de pouvoir analyser mes données de façon temporelle il m'a fallu définir un dataset d'entrée pour mon workflow de classification Knime. Ce fichier est simplement une fusion de mes 48 fichiers précédents. J'ai également transposé ma matrice afin que chaque ligne devienne une colonne. En effet, Knime se basera sur les informations de chaque colonne pour classer mes activités.

Activity	Sujet	X1	Y1	Z1	X2	Y2	Z2	X3	Y3	Z3
Jump	11	0.06029253	0.3995487	2.432635	0.06182191	0.4043244	2.444791	0.06278204	0.4063708	2.452801
Jump	12	0.06158914	0.1109236	2.497308	0.06157552	0.05496272	2.499336	0.03751836	-0.001215589	2.47547
Run	1	-1.139886	-0.1166579	2.514222	-1.166987	-0.1165815	2.559083	-1.19093	-0.1147015	2.55395
Run	2	-0.9022377	-0.1724564	2.570395	-0.8560084	-0.1492067	2.571559	-0.7982121	-0.1169319	2.56543
Run	3	-0.9662678	-0.05189186	2.530016	-0.9899199	-0.07138752	2.582716	-1.00272	-0.08608744	2.58851
Walk	1	-1.235284	0.05242087	2.487758	-1.232472	0.05098482	2.493935	-1.227903	0.04525353	2.49838
Walk	2	-1.080746	-0.03306674	2.608711	-1.119488	-0.04894867	2.623532	-1.134319	-0.05047619	2.62643
Sit	10	0.0701941	-0.1598616	2.060841	0.07116226	-0.1680125	2.058776	0.0704885	-0.184725	2.05555
Sit	11	0.06692135	-0.05042862	1.905805	0.07084329	-0.06096666	1.914624	0.06952227	-0.07442135	1.91062
Sit	12	0.1004253	-0.321822	1.971004	0.1003249	-0.3277941	1.975003	0.1040997	-0.3163614	1.96512

Figure 19 : Extrait du fichier de data set en entrée de Knime « Figure de l'auteur »

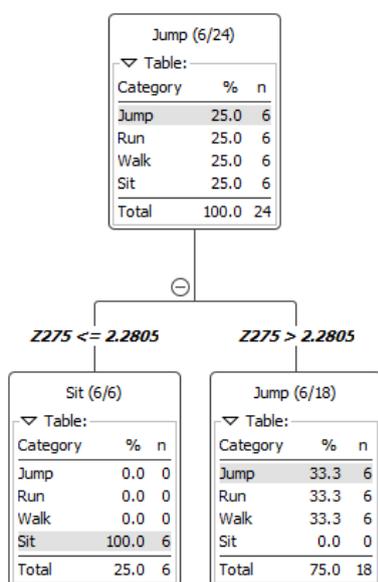
## 6.3 Workflow d'analyse de données

Le programme Knime va me permettre de créer un workflow d'analyse. Celui-ci s'effectue en plusieurs étapes :

- 1) Importation des données dans Knime
- 2) Partitionnement des données en 2 parties distinctes, la première regroupe 24 activités (6 de chaque type) et servira pour l'analyse des données. La deuxième partie (qui contient la 2<sup>ème</sup> moitié des données) servira pour le test de l'analyse.
- 3) Analyse des données selon plusieurs méthodes
- 4) Récupération des résultats afin de déterminer le meilleur modèle possible

### 6.3.1 Méthodes d'analyses utilisées

#### Decision Tree



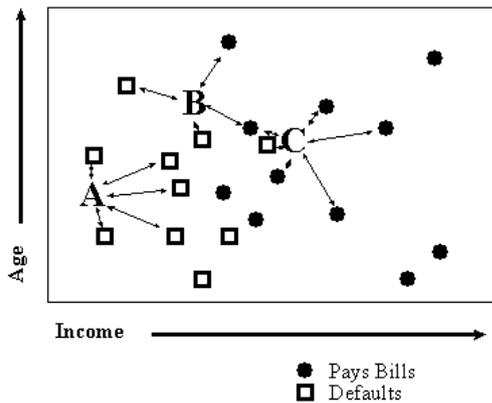
L'algorithme va se baser sur la valeur de certains points afin de séparer les données en plusieurs catégories. Par exemple, si le point Z à l'image 275 est supérieur à une certaine valeur, il va associer toutes les valeurs supérieures ensemble. Puis il va reprendre un autre point de comparaison et séparer à nouveau les données.

A chaque segment de l'arbre, on va chercher à séparer les éléments de façon à maximiser la pureté d'une des branches de l'arbre. On reproduit la procédure jusqu'à ce que l'arbre soit entièrement construit.

Figure 20 : Arbre de décision

### K-Nearest Neighbor

L'algorithme va comparer les valeurs de certains points et associer ceux qui se ressemblent le plus, il va donc rechercher le voisin le plus proche de chaque activité et les associer jusqu'à avoir classé toutes nos données dans les quatre activités possibles.

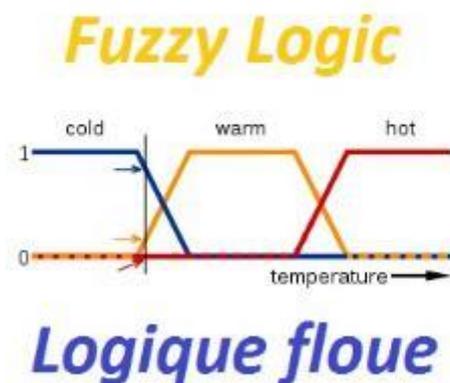


Sur l'image ci-contre on peut remarquer que les points A, B et C vont être associés à la famille qui leur est le plus proche. L'algorithme va donc regarder mes quatre activités et rassembler celles qui se rapprochent le plus.

Figure 21 : K-Nearest Neighbor

### Fuzzy Rules

L'algorithme va déterminer un certain nombre de règles en se basant sur les données d'entrées, il va ensuite classer les données en se basant sur toutes ces règles. Chaque règle est définie par une condition par exemple : Si  $x = a$ , alors  $y = b$ . [40]



Prenons en exemple l'image ci-contre, nous avons 3 courbes de température. En logique Floue (« Fuzzy Logic »), on peut donc affirmer que la courbe bleue est entièrement vraie quand elle se rapproche du point 1 et est fausse lorsqu'elle est sur le point 0. Entre les deux on détermine le pourcentage le plus proche de chaque côté.

Figure 22 : Logique Floue

### 6.4 Workflow d'analyses temporelles

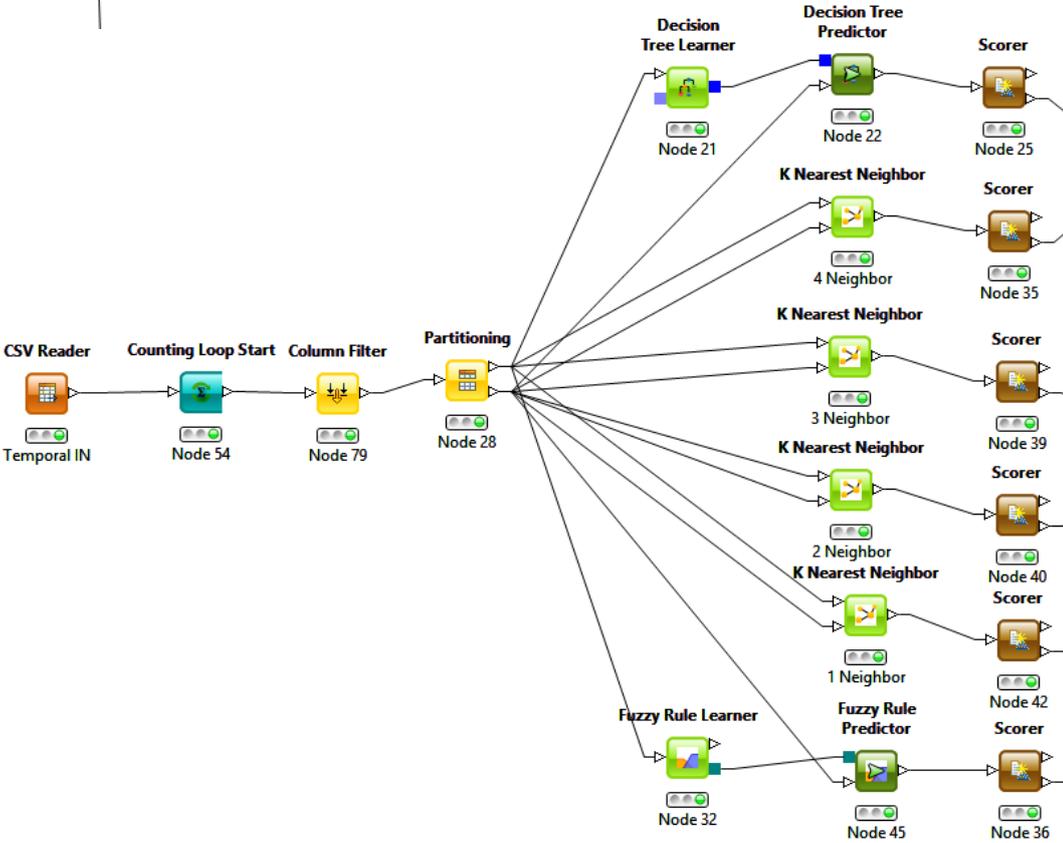


Figure 23 : Extrait du workflow d'analyses se basant sur les séries temporelles dans Kettle « Figure de l'auteur »



### 6.4.1 Résultats des algorithmes

*A noter : Le partitionnement des données s'effectue différemment à chaque fois que je lance le processus. Cette différence influencera les résultats : c'est pourquoi, j'ai exécuté 10 fois mon processus afin d'en récupérer une moyenne.*

<b>Exécution</b>	<b>Decision Tree</b>	<b>K-NN 4</b>	<b>K-NN 3</b>	<b>K-NN 2</b>	<b>K-NN 1</b>	<b>Fuzzy Rules</b>
1	41,66%	58,33%	33,33%	29,16%	25,00%	16,67%
2	25,00%	41,60%	41,60%	20,80%	25,00%	29,16%
3	37,50%	50,00%	45,83%	33,00%	25,00%	37,50%
4	45,83%	50,00%	45,83%	4,16%	37,50%	34,78%
5	50,00%	45,83%	41,60%	12,50%	33,33%	8,33%
6	41,60%	62,50%	50,00%	29,16%	37,50%	29,16%
7	37,50%	41,60%	29,16%	8,33%	25,00%	25,00%
8	29,16%	58,33%	45,83%	4,16%	16,67%	29,16%
9	50,00%	54,16%	41,60%	16,67%	33,33%	29,16%
10	41,66%	50,00%	45,83%	16,67%	25,00%	29,16%
<b>Moyenne</b>	<b>39,99%</b>	<b>51,24%</b>	<b>42,06%</b>	<b>17,46%</b>	<b>28,33%</b>	<b>26,81%</b>

Tableau 4 : Moyenne du taux d'erreurs des algorithmes de classification sur 10 processus lancés « Tableau de l'auteur »

Il ressort que la classification avec l'algorithme K-Nearest Neighbor est la plus efficace, à condition qu'on compare les données aux deux voisins les plus proches. Le manque de données totales ne permet pas de définir un résultat absolument fiable, une possibilité afin de vérifier la fiabilité serait d'utiliser la validation croisée.

### 6.4.2 Validation croisée

La validation croisée (« cross-validation ») est une méthode d'estimation de fiabilité d'un modèle fondé sur une technique d'échantillonnage. [38] Le principe est donc d'échantillonner toutes mes données afin de comparer chaque enregistrement individuellement.

### 6.5 Workflow d'analyses temporelles avec validation croisée

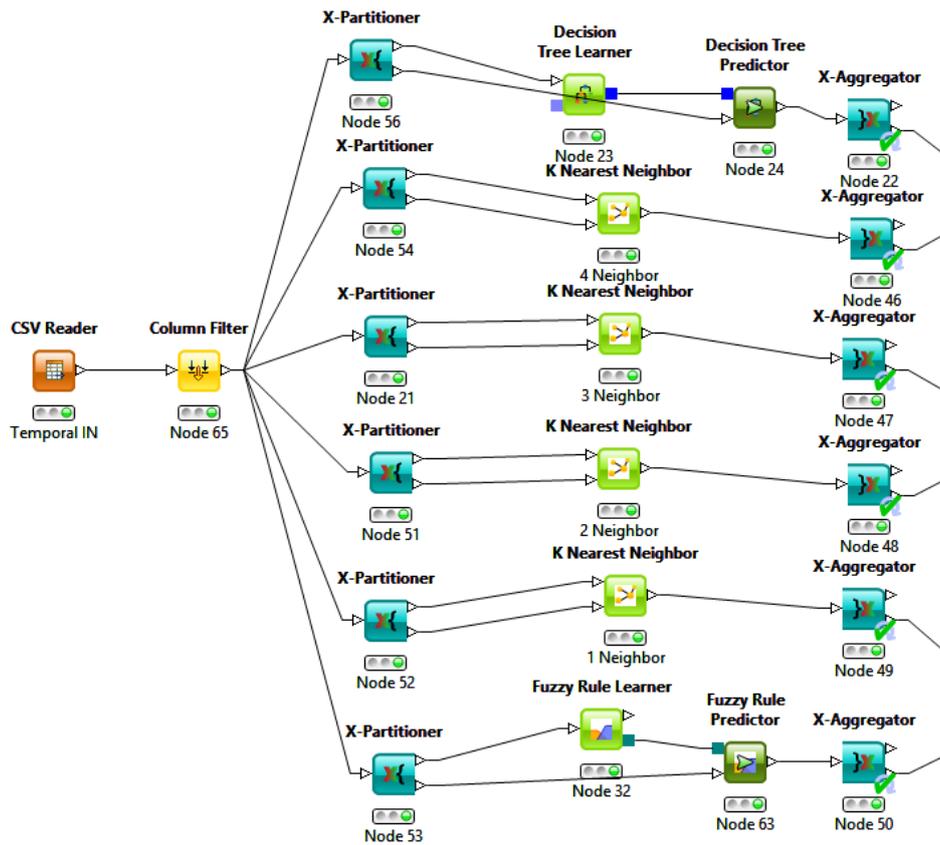


Figure 24 : Extrait du workflow d'analyses se basant sur les séries temporelles avec validation croisée « Figure de l'auteur »

### 6.5.1 Résultats des algorithmes

Le processus s'exécute 48 fois avec la validation croisée : une fois par enregistrement. J'effectue la moyenne de chaque résultat afin de déterminer l'algorithme le plus efficace.

<i>Exécution</i>	<b>Decision Tree</b>	<b>K-NN 4</b>	<b>K-NN 3</b>	<b>K-NN 2</b>	<b>K-NN 1</b>	<b>Fuzzy Rules</b>
1	0,00%	0,00%	0,00%	0,00%	0,00%	100,00%
2	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%
3	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%
4	100,00%	0,00%	0,00%	0,00%	0,00%	0,00%
5	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%
6	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%
7	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%
8	100,00%	100,00%	100,00%	100,00%	100,00%	0,00%
9	0,00%	100,00%	100,00%	100,00%	100,00%	0,00%
10	0,00%	100,00%	100,00%	100,00%	100,00%	0,00%
11	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%
12	0,00%	100,00%	100,00%	100,00%	100,00%	100,00%
13	100,00%	0,00%	0,00%	0,00%	0,00%	100,00%
14	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%
15	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%
16	100,00%	0,00%	0,00%	0,00%	0,00%	0,00%
17	100,00%	100,00%	100,00%	100,00%	100,00%	100,00%
18	100,00%	100,00%	100,00%	100,00%	0,00%	0,00%
19	0,00%	100,00%	0,00%	0,00%	0,00%	0,00%
20	100,00%	100,00%	100,00%	100,00%	100,00%	100,00%
21	0,00%	100,00%	0,00%	0,00%	0,00%	0,00%
22	0,00%	100,00%	0,00%	100,00%	0,00%	0,00%
23	0,00%	0,00%	0,00%	100,00%	100,00%	0,00%
24	100,00%	100,00%	100,00%	100,00%	100,00%	100,00%
25	0,00%	0,00%	0,00%	0,00%	0,00%	100,00%
26	100,00%	100,00%	100,00%	100,00%	100,00%	0,00%

27	0,00%	0,00%	0,00%	0,00%	0,00%	100,00%
28	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%
29	100,00%	100,00%	100,00%	100,00%	0,00%	100,00%
30	0,00%	0,00%	0,00%	0,00%	0,00%	100,00%
31	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%
32	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%
33	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%
34	100,00%	0,00%	0,00%	0,00%	0,00%	0,00%
35	100,00%	100,00%	100,00%	100,00%	0,00%	100,00%
36	0,00%	0,00%	0,00%	0,00%	0,00%	100,00%
37	100,00%	100,00%	100,00%	100,00%	100,00%	100,00%
38	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%
39	100,00%	100,00%	100,00%	100,00%	100,00%	0,00%
40	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%
41	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%
42	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%
43	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%
44	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%
45	100,00%	0,00%	0,00%	0,00%	0,00%	0,00%
46	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%
47	0,00%	100,00%	100,00%	100,00%	100,00%	0,00%
48	100,00%	100,00%	100,00%	100,00%	100,00%	100,00%
<b>Moyenne</b>	33,33%	37,50%	31,25%	35,42%	27,08%	29,17%

Tableau 5 : Moyenne du taux d'erreurs des algorithmes de classification sur 48 processus lancés avec validation croisée

« Tableau de l'auteur »

Un résultat affichant 100% de taux d'erreurs décrit simplement le cas où l'enregistrement traité en validation croisée n'a pas été classé correctement. L'analyse avec validation croisée montre que les deux algorithmes les plus efficaces sont : K-NN, avec les 2 voisins les plus proches, et la logique Floue.

## 6.6 Workflow d'analyses temporelles (réduction du nombre de points)

Afin d'aller plus loin dans cette analyse temporelle, j'ai également essayé de diminuer le nombre de points de mon dataset, afin d'essayer mon modèle sur moins d'images. J'ai donc classifié mes données en utilisant les workflows précédents, en diminuant le nombre de données de départ.

J'ai choisi de tester mon modèle avec le nombre d'images suivant :

15 images =  $\frac{1}{2}$  seconde

120 images = 4 secondes

30 images = 1 seconde

240 images = 8 secondes

50 images

60 images = 2 secondes

275 images = la totalité du modèle

Le but est maintenant de comparer les données afin de déterminer à partir de combien d'images par seconde mon modèle est viable. J'effectue deux fois chaque analyse avec un nombre d'images réduit. En effet, l'analyse d'une demie seconde en partant de la première image de la vidéo est très courte. Cette analyse peut-être faussée si le sujet met du temps pour réagir au début de la prise de vue. C'est pourquoi je refais chaque analyse en démarrant à une image plus lointaine dans la prise de vue.

## 6.6.1 Résultat des algorithmes sans validation croisée

<i>Durée en image</i>	Images de départ	Decision Tree	K-NN 4	K-NN 3	K-NN 2	K-NN 1	Fuzzy Rules
15	0	21,88%	58,33%	47,08%	29,17%	38,54%	27,11%
15	100	39,17%	61,25%	59,58%	34,58%	49,79%	32,04%
30	0	30,83%	52,08%	36,46%	18,75%	22,29%	28,16%
30	100	41,88%	64,58%	47,50%	25,21%	33,33%	40,44%
50	0	31,67%	42,92%	26,67%	17,29%	17,71%	30,07%
50	100	39,79%	59,17%	45,00%	23,33%	32,50%	36,69%
60	0	35,21%	41,46%	26,25%	11,88%	16,88%	26,22%
60	100	47,50%	50,63%	40,21%	18,75%	27,71%	42,66%
120	0	32,50%	45,83%	30,83%	19,38%	21,88%	27,42%
120	100	49,79%	46,88%	35,21%	15,42%	23,13%	42,02%
240	0	47,92%	51,25%	40,21%	18,96%	27,71%	31,76%
240	35	45,21%	53,33%	39,38%	20,21%	27,29%	30,79%
275	0	43,33%	53,54%	44,58%	23,54%	33,96%	26,58%

Tableau 6 : Taux d'erreurs des modèles selon la durée en image (sans validation croisée) « Tableau de l'auteur »

Chaque ligne du tableau correspond à une moyenne des résultats des algorithmes sur 20 exécutions du programme. Vous trouverez le détail de chacune des exécutions dans les annexes.

L'algorithme ayant le meilleur résultat est le K-NN avec les deux plus proches voisins, celui ayant le pire résultat est le K-NN avec 4 voisins. Il est intéressant de noter que les différences de performances des algorithmes entre l'image de départ (zéro) et l'image 100 sont très proches les unes des autres selon certains algorithmes.

## 6.6.2 Résultat des algorithmes avec validation croisée

<b>Durée en image</b>	<b>Image de départ</b>	<b>Decision Tree</b>	<b>K-NN 4</b>	<b>K-NN 3</b>	<b>K-NN 2</b>	<b>K-NN 1</b>	<b>Fuzzy Rules</b>
15	0	12,50%	64,58%	35,42%	62,50%	33,33%	33,33%
15	100	31,25%	81,25%	45,83%	70,83%	50,00%	50,00%
30	0	27,08%	37,50%	18,75%	18,75%	16,67%	33,33%
30	100	45,83%	58,33%	35,42%	47,92%	27,08%	43,75%
50	0	22,92%	31,25%	14,58%	18,75%	12,50%	31,25%
50	100	41,67%	52,08%	33,33%	45,83%	27,08%	39,58%
60	0	27,08%	29,17%	16,67%	16,67%	12,50%	33,33%
60	100	43,75%	45,83%	29,17%	41,67%	22,92%	37,50%
120	0	16,67%	29,17%	22,92%	25,00%	16,67%	31,25%
120	100	60,42%	37,50%	27,08%	29,17%	16,67%	45,83%
240	0	29,17%	43,75%	31,25%	31,25%	25,00%	29,17%
240	35	43,75%	39,58%	31,25%	33,33%	20,83%	33,33%
275	0	33,33%	37,50%	31,25%	35,42%	27,08%	29,17%

Tableau 7 : Taux d'erreurs des modèles selon la durée en image (avec validation croisée) « Tableau de l'auteur »

J'effectue la même analyse mais cette fois-ci en utilisant la validation croisée. L'algorithme ayant le meilleur résultat est le K-NN avec 1 seul voisin le plus proche, celui ayant le pire résultat est le K-NN avec 4 voisins. Là encore, chaque ligne correspond à la moyenne des résultats sur 20 exécutions, chaque analyse détaillée se trouve en annexe.

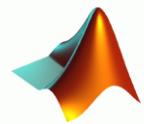
Le fait de partir à l'image de départ (zéro) ou de la 100<sup>ème</sup> image a plus d'influence avec validation croisée.

## 7. Analyse de données en fréquence

Une deuxième possibilité d'analyses, déjà évoquée dans ce travail, est l'analyse se basant sur les fréquences. Ce procédé a pour avantage de s'affranchir du nombre de points de ma série temporelle, il lui faut au minimum le double du nombre de points d'une oscillation.

Afin de pouvoir transformer mes données temporelles en fréquence il a fallu utiliser l'algorithme de Fourier. Cette transformation de Fourier est une opération qui transforme une fonction intégrable de nombre réelle en une autre fonction décrivant le spectre fréquentiel de cette dernière. <sup>[41]</sup>

J'ai utilisé Matlab, un programme permettant la gestion de fonction mathématique de façon simple. Ce programme m'a permis de transformer mes données temporelles en fréquence.



Logo :

Activity	Sujet	X1	Y1	Z1	X2	Y2	Z2	X3	Y3	Z3
Jump	1	13,201	2,934	114,86	13,118	2,9671	115,03	12,959	2,9808	115,02
Jump	2	9,2397	2,3342	5,4625	9,1966	2,3486	5,6084	9,1031	2,4044	5,6404
Jump	3	3,5162	3,0796	2,9394	3,5642	3,1601	2,8314	3,5593	3,3033	2,8412
Jump	4	6,8758	0,72799	1,7389	6,9971	0,91433	1,7706	7,055	1,0551	1,7172
Jump	5	2,889	0,15366	0,30153	3,1546	0,19952	0,42422	3,3299	0,23512	0,36441
Jump	6	3,4598	0,71639	1,4738	3,3135	0,62244	1,6156	3,2197	0,57427	1,5891
Jump	7	2,058	1,8571	1,1381	2,1422	1,5978	1,0565	2,1828	1,3426	1,0593
Jump	8	3,5999	1,2942	1,2806	3,796	1,1415	1,305	3,9789	0,94776	1,3336
Jump	9	5,5822	1,516	0,64112	5,574	1,3501	0,66582	5,4996	1,19	0,74298
Jump	10	4,3725	1,2543	0,31729	4,2855	1,0146	0,38651	4,1115	0,83592	0,37842
Jump	11	5,7089	0,80322	1,1695	5,6927	0,70829	1,1874	5,6861	0,57242	1,1883
Jump	12	5,1059	0,19189	0,80156	5,0917	0,20763	0,75929	5,0306	0,37192	0,70333
Run	1	3,8188	0,70386	0,38937	3,8517	0,94128	0,41215	3,7713	1,1884	0,44296
Run	2	4,5454	0,4742	1,3268	4,4636	0,64831	1,3401	4,3483	0,78656	1,2898
Run	3	2,5193	0,64048	1,3574	2,3646	0,36147	1,3386	2,233	0,097883	1,2713

Matlab

Figure 25: Extrait du fichier d'entrée pour Knime avec les fréquences « Figure de l'auteur »

### 7.0.1 Qu'est-ce qu'une fréquence ?

Une fréquence est le nombre de fois qu'un phénomène se produit périodiquement sur une certaine durée. Elle désigne le nombre d'occurrences de la même oscillation par unité de mesure. Elle est aussi considérée comme la variable indépendante dans l'analyse spectrale. <sup>[43]</sup>

## 7.1 Workflow d'analyse fréquentielle complet

Le workflow d'analyse fréquentielle est exactement le même que celui pour les séries temporelles. En effet, afin de pouvoir comparer au mieux les deux méthodes, il faut que les analyses soient le plus similaires possible. Les algorithmes de classification sont donc toujours les mêmes :

- Decision Tree
- K-Nearest Neighbor
- Fuzzy Rules

La première analyse fréquentielle reprend la totalité des données soit les 275 points pour chaque coordonnée.

### 7.1.1 Résultats des algorithmes

<b>Exécution</b>	<b>Decision Tree</b>	<b>K-NN 4</b>	<b>K-NN 3</b>	<b>K-NN 2</b>	<b>K-NN 1</b>	<b>Fuzzy Rules</b>
1	62,50%	58,33%	70,83%	58,33%	87,50%	83,33%
2	91,67%	95,83%	87,50%	45,83%	83,33%	95,83%
3	87,50%	75,00%	66,67%	41,67%	87,50%	91,67%
4	83,33%	70,83%	75,00%	45,83%	91,67%	83,33%
5	83,33%	79,17%	79,17%	50,00%	87,50%	87,50%
6	79,17%	79,17%	75,00%	58,33%	83,33%	82,61%
7	87,50%	87,50%	95,83%	41,67%	100,00%	86,96%
8	87,50%	70,83%	79,17%	66,67%	83,33%	91,67%
9	79,17%	79,17%	70,83%	58,33%	83,33%	87,50%
10	70,83%	75,00%	83,33%	54,17%	75,00%	86,96%
<b>Moyenne</b>	81,25%	77,08%	78,33%	52,08%	86,25%	87,74%

Tableau 8 : Moyenne du taux d'erreurs des algorithmes de classification sur 10 processus lancés « Tableau de l'auteur »

Les résultats sont largement moins précis qu'en utilisant les séries temporelles. A la 7<sup>ème</sup> exécution on trouve l'algorithme ayant la meilleure précision (K-NN avec 2 voisins) mais on trouve aussi un algorithme classant à 100% faux nos données.

### 7.1.2 Résultats des algorithmes avec validation croisée

<i>Exécution</i>	<b>Decision Tree</b>	<b>K-NN 4</b>	<b>K-NN 3</b>	<b>K-NN 2</b>	<b>K-NN 1</b>	<b>Fuzzy Rules</b>
1	0,00%	100,00%	0,00%	100,00%	100,00%	100,00%
2	100,00%	100,00%	100,00%	100,00%	100,00%	100,00%
3	100,00%	100,00%	100,00%	100,00%	100,00%	100,00%
4	100,00%	100,00%	100,00%	100,00%	100,00%	100,00%
5	100,00%	100,00%	100,00%	100,00%	100,00%	100,00%
6	100,00%	100,00%	100,00%	100,00%	100,00%	100,00%
7	100,00%	100,00%	100,00%	100,00%	100,00%	100,00%
8	100,00%	100,00%	100,00%	100,00%	100,00%	100,00%
9	100,00%	100,00%	100,00%	100,00%	100,00%	100,00%
10	100,00%	100,00%	100,00%	100,00%	100,00%	100,00%
11	100,00%	100,00%	100,00%	100,00%	100,00%	100,00%
12	100,00%	100,00%	100,00%	100,00%	100,00%	100,00%
13	100,00%	100,00%	100,00%	100,00%	100,00%	100,00%
14	100,00%	100,00%	100,00%	100,00%	100,00%	100,00%
15	100,00%	0,00%	0,00%	100,00%	100,00%	100,00%
16	100,00%	100,00%	100,00%	100,00%	100,00%	100,00%
17	100,00%	100,00%	100,00%	100,00%	100,00%	100,00%
18	100,00%	100,00%	100,00%	100,00%	100,00%	100,00%
19	100,00%	100,00%	100,00%	100,00%	100,00%	100,00%
20	100,00%	100,00%	100,00%	100,00%	100,00%	100,00%
21	100,00%	100,00%	100,00%	100,00%	100,00%	100,00%
22	100,00%	100,00%	100,00%	100,00%	100,00%	100,00%
23	100,00%	100,00%	100,00%	100,00%	100,00%	100,00%
24	100,00%	100,00%	100,00%	100,00%	100,00%	100,00%

## Martin Tazlari

25	0,00%	100,00%	100,00%	100,00%	100,00%	100,00%
26	100,00%	100,00%	100,00%	100,00%	100,00%	100,00%
27	100,00%	100,00%	100,00%	100,00%	100,00%	100,00%
28	100,00%	100,00%	100,00%	100,00%	100,00%	100,00%
29	100,00%	100,00%	100,00%	100,00%	100,00%	100,00%
30	100,00%	100,00%	100,00%	100,00%	100,00%	100,00%
31	100,00%	100,00%	100,00%	100,00%	100,00%	100,00%
32	100,00%	100,00%	100,00%	100,00%	100,00%	100,00%
33	100,00%	100,00%	100,00%	100,00%	100,00%	100,00%
34	100,00%	100,00%	100,00%	100,00%	100,00%	100,00%
35	100,00%	100,00%	100,00%	100,00%	100,00%	100,00%
36	100,00%	100,00%	100,00%	100,00%	100,00%	100,00%
37	100,00%	100,00%	100,00%	100,00%	100,00%	100,00%
38	100,00%	100,00%	100,00%	100,00%	100,00%	100,00%
39	100,00%	100,00%	100,00%	100,00%	100,00%	100,00%
40	100,00%	100,00%	100,00%	100,00%	100,00%	100,00%
41	100,00%	100,00%	100,00%	100,00%	100,00%	100,00%
42	100,00%	100,00%	100,00%	100,00%	100,00%	100,00%
43	100,00%	100,00%	100,00%	100,00%	100,00%	100,00%
44	100,00%	100,00%	100,00%	100,00%	100,00%	100,00%
45	100,00%	100,00%	100,00%	100,00%	100,00%	100,00%
46	100,00%	100,00%	100,00%	100,00%	100,00%	100,00%
47	100,00%	100,00%	100,00%	100,00%	100,00%	100,00%
48	100,00%	100,00%	100,00%	100,00%	100,00%	100,00%
<b>Moyenne</b>	<b>95,83%</b>	<b>97,92%</b>	<b>95,83%</b>	<b>100,00%</b>	<b>100,00%</b>	<b>100,00%</b>

Tableau 9 : Moyenne du taux d'erreurs des algorithmes de classification sur 48 processus lancés avec cross-validation

« Tableau de l'auteur »

Les résultats avec cross-validation sont moins précis puisque 3 algorithmes ont une moyenne de 100% d'erreurs et n'ont donc jamais réussi à classer l'enregistrement sélectionné correctement.

### 7.3 Workflow d'analyse fréquentielle se basant sur le point minimum et maximum

Cette nouvelle analyse arrive pour tenter d'améliorer les performances des résultats en utilisant toujours les fréquences. Cette analyse a pour but d'essayer de cibler un peu plus les données importantes de mon dataset lors de l'analyse. C'est pourquoi j'ai choisi de ne garder que les points minimum et maximum pour chaque coordonnée (X, Y, Z).

	A	B	C	D	E	F	G	H
1	<b>Activity</b>	<b>Sujet</b>	<b>Max_X</b>	<b>Min_X</b>	<b>Max_Y</b>	<b>Min_Y</b>	<b>Max_Z</b>	<b>Min_Z</b>
2	Jump	4	7,0883	0,28457	1,5922	0,046299	1,7706	0,61447
11	Run	1	5,5368	0,28192	2,0355	0,019265	1,4376	0,082396
22	Run	12	7,3509	0,64774	1,8705	0,056588	1,2234	0,029708
23	Walk	1	7,9489	0,0025521	2,1396	0,0067116	1,2933	0,002421
30	Walk	8	10,123	0,40201	1,9969	0,021061	1,0339	0,11464
34	Walk	12	5,2275	0,098043	1,67	0,030612	1,6485	0,43637
46	Sit	12	10,087	0,10175	2,536	0,52506	6,2184	4,567

Figure 26 : Extrait du dataset min-max pour Knime « Figure de l'auteur »

Ce dataset m'a permis de relancer mes algorithmes de décisions et de définir des modèles plus précis.

### 7.3.1 Résultats des algorithmes

<b>Exécution</b>	<b>Decision Tree</b>	<b>K-NN 4</b>	<b>K-NN 3</b>	<b>K-NN 2</b>	<b>K-NN 1</b>	<b>Fuzzy Rules</b>
1	78,26%	73,91%	86,96%	43,48%	56,52%	95,24%
2	82,61%	78,26%	78,26%	34,78%	73,91%	80,95%
3	86,96%	73,91%	65,22%	43,48%	69,57%	94,44%
4	78,26%	73,91%	69,57%	34,78%	73,91%	90,00%
5	82,61%	78,26%	69,57%	21,74%	78,26%	77,27%
6	86,96%	69,57%	65,22%	26,09%	73,91%	80,95%
7	73,91%	69,57%	60,87%	47,83%	60,87%	69,57%
8	73,91%	65,22%	56,52%	43,48%	69,57%	82,61%
9	91,30%	78,26%	69,57%	39,13%	73,91%	85,71%
10	82,61%	73,91%	73,91%	30,43%	78,26%	89,47%
<b>Moyenne</b>	81,74%	73,48%	69,57%	36,52%	70,87%	84,62%

Tableau 10 : Moyenne du taux d'erreurs des algorithmes de classification sur 10 processus lancés « Tableau de l'auteur »

Les résultats sont sensiblement meilleurs surtout pour le K-NN avec les deux plus proches voisins.

### 7.3.2 Résultats des algorithmes avec Cross-validation

<i>Exécution</i>	<b>Decision Tree</b>	<b>K-NN 4</b>	<b>K-NN 3</b>	<b>K-NN 2</b>	<b>K-NN 1</b>	<b>Fuzzy Rules</b>
1	100,00%	100,00%	100,00%	100,00%	100,00%	100,00%
2	100,00%	100,00%	100,00%	100,00%	100,00%	100,00%
3	100,00%	100,00%	100,00%	100,00%	100,00%	100,00%
4	100,00%	100,00%	100,00%	100,00%	100,00%	100,00%
5	100,00%	100,00%	100,00%	100,00%	100,00%	100,00%
6	100,00%	100,00%	100,00%	100,00%	0,00%	0,00%
7	100,00%	0,00%	0,00%	100,00%	0,00%	100,00%
8	0,00%	0,00%	0,00%	100,00%	100,00%	100,00%
9	100,00%	100,00%	100,00%	100,00%	100,00%	100,00%
10	100,00%	100,00%	100,00%	100,00%	100,00%	100,00%
11	100,00%	100,00%	100,00%	100,00%	100,00%	100,00%
12	100,00%	100,00%	100,00%	100,00%	100,00%	100,00%
13	100,00%	100,00%	100,00%	100,00%	100,00%	100,00%
14	100,00%	100,00%	0,00%	0,00%	0,00%	100,00%
15	100,00%	100,00%	100,00%	100,00%	0,00%	100,00%
16	100,00%	100,00%	100,00%	100,00%	100,00%	100,00%
17	100,00%	100,00%	100,00%	100,00%	100,00%	100,00%
18	100,00%	0,00%	0,00%	100,00%	100,00%	100,00%
19	100,00%	100,00%	100,00%	100,00%	100,00%	100,00%
20	100,00%	100,00%	100,00%	100,00%	100,00%	100,00%
21	100,00%	0,00%	0,00%	100,00%	100,00%	0,00%
22	100,00%	100,00%	0,00%	100,00%	0,00%	100,00%
23	100,00%	0,00%	0,00%	100,00%	100,00%	100,00%
24	100,00%	100,00%	100,00%	100,00%	100,00%	100,00%
25	100,00%	100,00%	100,00%	100,00%	100,00%	100,00%
26	100,00%	0,00%	0,00%	0,00%	100,00%	100,00%
27	100,00%	100,00%	100,00%	100,00%	100,00%	100,00%
28	100,00%	100,00%	100,00%	100,00%	100,00%	100,00%



29	100,00%	100,00%	100,00%	100,00%	0,00%	100,00%
30	100,00%	100,00%	0,00%	0,00%	0,00%	100,00%
31	100,00%	100,00%	100,00%	100,00%	100,00%	100,00%
32	100,00%	100,00%	100,00%	100,00%	100,00%	100,00%
33	100,00%	100,00%	100,00%	100,00%	100,00%	100,00%
34	100,00%	100,00%	100,00%	100,00%	100,00%	100,00%
35	100,00%	100,00%	100,00%	100,00%	100,00%	0,00%
36	0,00%	100,00%	0,00%	100,00%	100,00%	0,00%
37	100,00%	0,00%	0,00%	0,00%	0,00%	100,00%
38	100,00%	0,00%	100,00%	0,00%	0,00%	100,00%
39	100,00%	100,00%	100,00%	100,00%	100,00%	100,00%
40	100,00%	100,00%	100,00%	100,00%	100,00%	100,00%
41	100,00%	100,00%	100,00%	100,00%	100,00%	100,00%
42	100,00%	100,00%	100,00%	100,00%	100,00%	100,00%
43	100,00%	100,00%	100,00%	100,00%	100,00%	100,00%
44	0,00%	100,00%	0,00%	100,00%	100,00%	100,00%
45	0,00%	0,00%	0,00%	0,00%	0,00%	100,00%
46	100,00%	100,00%	100,00%	100,00%	100,00%	100,00%
47	100,00%	100,00%	100,00%	100,00%	100,00%	100,00%
48	100,00%	100,00%	100,00%	100,00%	100,00%	100,00%
<b>Moyenne</b>	<b>91,67%</b>	<b>81,25%</b>	<b>72,92%</b>	<b>87,50%</b>	<b>79,17%</b>	<b>91,67%</b>

Tableau 11 : Moyenne du taux d'erreurs des algorithmes de classification sur 48 processus lancés avec cross-validation  
« Tableau de l'auteur »

La cross-validation n'apporte pas des résultats très précis (jusqu'à 91% d'erreurs). Là encore, il faudrait proposer un plus grand set de données afin de pouvoir espérer améliorer le résultat.

## 8. Création d'un regroupement hiérarchique

J'ai également tenté de déterminer s'il était possible de classer hiérarchiquement mes données en 2 catégories distinctes. J'imaginai la possibilité que des actions similaires, comme marcher et sauter, pourraient se retrouver classées dans la même catégorie. Ainsi je pourrai réessayer mes modèles sur les deux regroupements, afin de vérifier si le taux d'erreurs peut être diminué.

### 8.1 Qu'est-ce que le regroupement hiérarchique

Il s'agit d'une méthode de classification automatique de données. Cette notion de regroupement fait appel à différentes méthodes de clustering. Le principe est de répartir les données en deux clusters (groupes) différents et de pouvoir ensuite segmenter l'analyse sur chaque groupe. <sup>[44]</sup>

Ces clusters peuvent souvent être illustrés sous forme de Dendrogramme (comme l'image ci-contre)

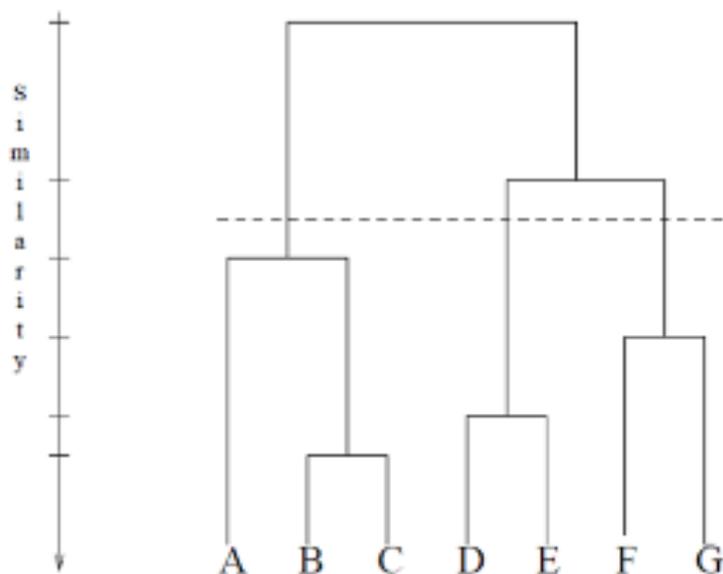
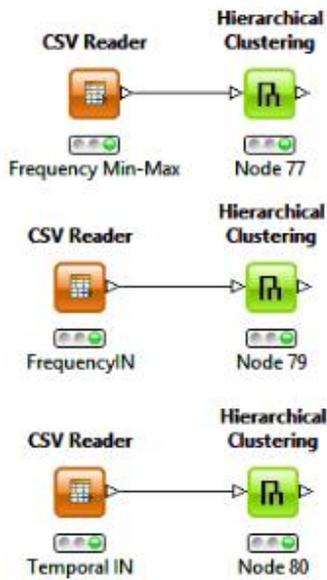


Figure 27 : Exemple de dendrogramme

## 8.2 Workflow de regroupement hiérarchique



J'ai utilisé les 3 fichiers de données de départ :

- Séries temporelles
- Séries fréquentielles
- Séries fréquentielles minimum, maximum

Mon but était de voir si des activités très ressemblantes les unes des autres se retrouvaient dans le même cluster (groupe) afin de pouvoir ensuite les analyser séparément.

Figure 28 : Workflow de regroupement hiérarchique « Figure de l'auteur »

Mon idée théorique était de retrouver les deux groupes suivants :

- Cluster 1 : Marche, Course
- Cluster 2 : Assis, Saut

En pratique aucun cluster significatif n'est ressorti. Cela peut venir du choix des données en input de mon workflow ou simplement du fait que mes données soient trop similaires afin de pouvoir être rassemblées significativement.

J'ai donc choisi de laisser cette possibilité en suspens et de chercher une nouvelle technique pour améliorer la précision de mon modèle de données.

## 9.0 Analyse se basant sur des histogrammes normalisés

Pour aller plus loin, il est possible d'améliorer la qualité des séries temporelles. En effet, comme énoncé plus haut, les séries temporelles utilisées sont légèrement faussées, cela est dû au fait qu'elles n'ont pas exactement le même nombre de points pour chaque série.

Dans un premier temps j'ai décidé de prendre uniquement le nombre minimum de points par série soit 275. Cependant il existe une autre possibilité de traitement des données qui permettra normalement d'améliorer la qualité des résultats.

### 9.1 Transformation des données

J'ai donc transformé mes données en utilisant le programme Matlab, déjà utilisé lors de la transformation en fréquence. Le but de cette transformation était d'utiliser l'algorithme du K-means sur mes données enfin de créer des clusters (groupes) puis de comparer chacun de mes points aux centres de chaque groupe pour en récupérer la distance. J'additionne ensuite la distance de tous les points face à chaque centre. Vous retrouvez, en annexe, le script ainsi que la fonction utilisé dans Matlab.

### 9.2 K-means

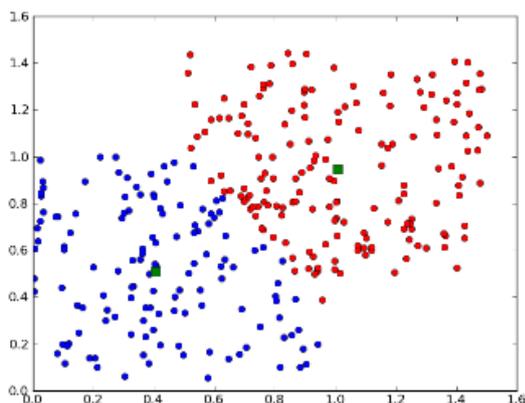


Figure 29 : Deux clusters K-means avec leur centroïdes

L'algorithme de K-means est très utilisé en statistique ainsi que pour l'apprentissage automatique. Cet algorithme se base sur le partitionnement des données en clusters dans lesquelles chaque élément appartient à la partition avec la moyenne la plus proche. <sup>[45]</sup>

### 9.3 Centroïdes

Les centroïdes correspondent aux centres de chaque cluster dans une analyse faite avec K-means. Pour mon analyse, j'ai décidé dans un premier temps de comparer mes données sur 400 centroïdes et de former ainsi 400 clusters. Ce choix a été motivé par la moyenne du nombre de points de chaque activité qui se trouve proche de 400.

### 9.4 Histogrammes

Une fois mes données transformées j'ai pu afficher pour chaque activité un histogramme, afin de mettre toutes mes données sur la même échelle j'ai normalisé les données et ainsi transformé tous mes points pour qu'ils se retrouvent entre 0 et 1. Les histogrammes sont très ressemblant, mais l'échelle des X change afin d'être normalisée.

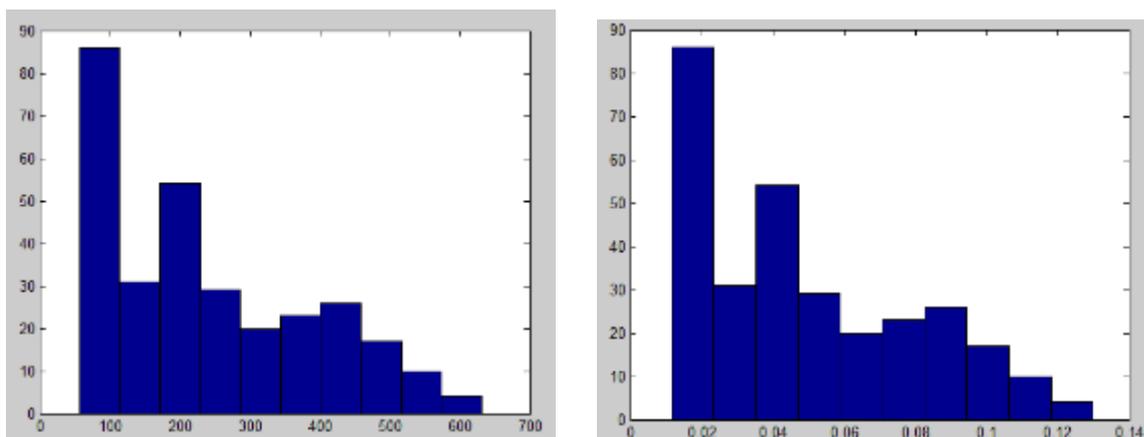


Figure 30 : A gauche : histogramme d'une activité non normalisée, à droite : histogramme d'une activité normalisée « Figure de l'auteur »

## 9.5 Output de données utilisées

Une fois cette opération effectuée pour chaque activité, je récupère le fichier suivant en sortie :

Row ID	S Activity	D C1	D C2	D C3	D C4	D C5	D C6	D C7
Row0	Sit	0.036	0.01	0.012	0.091	0.072	0.077	0.02
Row1	Sit	0.033	0.008	0.009	0.088	0.07	0.074	0.02
Row2	Sit	0.032	0.01	0.01	0.091	0.071	0.077	0.017
Row3	Sit	0.033	0.009	0.011	0.09	0.072	0.076	0.018
Row4	Sit	0.037	0.01	0.013	0.088	0.071	0.075	0.021
Row5	Sit	0.031	0.01	0.011	0.086	0.067	0.072	0.02
Row6	Sit	0.035	0.009	0.011	0.089	0.07	0.075	0.02
Row7	Sit	0.032	0.008	0.009	0.089	0.07	0.075	0.018
Row8	Sit	0.031	0.01	0.011	0.086	0.068	0.073	0.021
Row9	Sit	0.031	0.01	0.01	0.087	0.068	0.073	0.02
Row10	Sit	0.039	0.013	0.016	0.085	0.069	0.072	0.024
Row11	Sit	0.032	0.011	0.012	0.084	0.067	0.071	0.022
Row12	Jump	0.031	0.038	0.038	0.086	0.065	0.073	0.024
Row13	Jump	0.023	0.029	0.027	0.084	0.062	0.069	0.019
Row14	Jump	0.037	0.032	0.033	0.089	0.068	0.075	0.016
Row15	Jump	0.043	0.026	0.03	0.081	0.064	0.068	0.019
Row16	Jump	0.028	0.029	0.028	0.09	0.067	0.075	0.013
Row17	Jump	0.027	0.037	0.036	0.082	0.06	0.068	0.024
Row18	Jump	0.031	0.027	0.026	0.087	0.065	0.072	0.016
Row19	Jump	0.03	0.038	0.037	0.079	0.058	0.065	0.024
Row20	Jump	0.024	0.039	0.037	0.075	0.053	0.061	0.026

Figure 31 : Extrait du fichier d'entrée pour les analyses Knime « Figure de l'auteur »

## 10. Analyse

La première analyse a été effectuée sur 400 Centroides, ce chiffre m’a servi de base pour la première itération.

### 10.1 Résultats des analyses sans validation croisée

<b>Exécution</b>	<b>Decision Tree</b>	<b>K-NN 4</b>	<b>K-NN 3</b>	<b>K-NN 2</b>	<b>K-NN 1</b>	<b>Fuzzy Rules</b>
1	20,83%	25,00%	16,67%	12,50%	29,17%	21,74%
2	33,33%	33,33%	29,17%	12,50%	25,00%	26,32%
3	20,83%	25,00%	12,50%	25,00%	20,83%	20,83%
4	25,00%	25,00%	29,17%	16,67%	25,00%	20,83%
5	20,83%	29,17%	29,17%	16,67%	25,00%	21,74%
6	29,17%	25,00%	25,00%	16,67%	20,83%	30,43%
7	25,00%	20,83%	16,67%	20,83%	25,00%	33,33%
8	29,17%	25,00%	16,67%	12,50%	25,00%	13,64%
9	29,17%	25,00%	29,17%	4,17%	29,17%	28,57%
10	25,00%	29,17%	29,17%	16,67%	25,00%	26,09%
11	25,00%	20,83%	20,83%	8,33%	29,17%	20,83%
12	20,83%	20,83%	25,00%	8,33%	25,00%	16,67%
13	33,33%	29,17%	37,50%	16,67%	37,50%	34,78%
14	29,17%	20,83%	25,00%	0,00%	29,17%	29,17%
15	25,00%	25,00%	29,17%	12,50%	29,17%	30,43%
16	29,17%	16,67%	16,67%	16,67%	29,17%	25,00%
17	62,50%	25,00%	33,33%	8,33%	37,50%	26,09%
18	29,17%	25,00%	25,00%	20,83%	25,00%	28,57%
19	20,83%	25,00%	25,00%	16,67%	29,17%	19,05%
20	29,17%	20,83%	25,00%	16,67%	16,67%	21,74%
<b>Moyenne</b>	<b>28,13%</b>	24,58%	24,79%	<b>13,96%</b>	26,88%	24,79%

Tableau 12: Moyenne du taux d’erreurs lancé sur 20 processus sans validation croisée « Tableau de l’auteur »

Les résultats sont meilleurs que lorsque l'on utilise les séries temporelles sans les avoir corrigées. On obtient un taux d'erreur minimum de 13,96% avec l'algorithme K-NN avec les deux voisins les plus proches. Tous les algorithmes sont en dessous des 30% d'erreurs.

### 10.2 Résultats des analyses avec validation croisée

<i>Exécution</i>	<b>Decision Tree</b>	<b>K-NN 4</b>	<b>K-NN 3</b>	<b>K-NN 2</b>	<b>K-NN 1</b>	<b>Fuzzy Rules</b>
1	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%
2	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%
3	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%
4	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%
5	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%
6	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%
7	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%
8	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%
9	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%
10	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%
11	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%
12	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%
13	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%
14	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%
15	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%
16	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%
17	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%
18	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%
19	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%
20	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%
21	100,00%	0,00%	0,00%	0,00%	0,00%	0,00%
22	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%
23	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%
24	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%

25	0,00%	100,00%	100,00%	100,00%	100,00%	100,00%
26	100,00%	0,00%	0,00%	100,00%	100,00%	0,00%
27	100,00%	100,00%	100,00%	100,00%	100,00%	100,00%
28	0,00%	0,00%	0,00%	100,00%	100,00%	100,00%
29	100,00%	100,00%	100,00%	100,00%	100,00%	100,00%
30	100,00%	100,00%	100,00%	100,00%	0,00%	100,00%
31	0,00%	100,00%	100,00%	100,00%	100,00%	100,00%
32	100,00%	100,00%	100,00%	100,00%	0,00%	100,00%
33	0,00%	100,00%	100,00%	100,00%	0,00%	100,00%
34	0,00%	100,00%	100,00%	100,00%	100,00%	0,00%
35	0,00%	0,00%	0,00%	0,00%	0,00%	100,00%
36	0,00%	0,00%	0,00%	100,00%	0,00%	0,00%
37	0,00%	0,00%	0,00%	100,00%	100,00%	0,00%
38	0,00%	100,00%	100,00%	100,00%	100,00%	100,00%
39	0,00%	0,00%	0,00%	0,00%	0,00%	100,00%
40	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%
41	100,00%	100,00%	100,00%	100,00%	100,00%	100,00%
42	100,00%	100,00%	100,00%	100,00%	0,00%	100,00%
43	0,00%	100,00%	100,00%	100,00%	100,00%	100,00%
44	0,00%	100,00%	0,00%	0,00%	0,00%	0,00%
45	0,00%	100,00%	100,00%	100,00%	100,00%	0,00%
46	100,00%	0,00%	0,00%	0,00%	0,00%	0,00%
47	100,00%	0,00%	0,00%	100,00%	0,00%	0,00%
48	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%
<b>Moyenne</b>	<b>20,83%</b>	<b>29,17%</b>	<b>27,08%</b>	<b>37,50%</b>	<b>25,00%</b>	<b>29,17%</b>

Tableau 13 : Moyenne du taux d'erreurs lancé sur 48 processus avec validation croisée « Tableau de l'auteur »

Les résultats sont également bien meilleurs avec les données temporelles corrigées. On peut relever qu'avec la validation croisée le meilleur algorithme est le « Decision Tree », et le moins bon est le « K-NN avec les deux voisins les plus proches », c'est l'inverse de l'analyse sans validation croisée.

## 11. Analyse sur différents nombres de centroïdes

Pour aller plus loin, j'ai également tenté de modifier le nombre de centroïdes lors de ma classification K-means. Le fait de modifier le nombre de centroïdes peut améliorer, ou non, mon modèle de données.

### 11.1 Résumé des résultats sans validation croisée

<b>Nombre de centroïdes</b>	<b>Decision Tree</b>	<b>K-NN 4</b>	<b>K-NN 3</b>	<b>K-NN 2</b>	<b>K-NN 1</b>	<b>Fuzzy Rules</b>
100	28,54%	24,38%	23,33%	13,33%	23,54%	24,70%
200	30,00%	21,25%	23,54%	15,00%	23,75%	25,82%
300	26,46%	22,92%	24,38%	16,67%	25,00%	27,10%
400	28,13%	24,58%	24,79%	13,96%	26,88%	24,79%
500	28,75%	20,83%	21,25%	15,83%	24,38%	20,70%

Tableau 14 : Moyenne du taux d'erreurs des algorithmes avec un nombre de centroïdes variable (sans validation croisée)

« Tableau de l'auteur »

Les résultats sont intéressants, on voit qu'avec seulement 100 centroïdes, on obtient déjà des résultats très proches du 100% avec un taux d'erreur de seulement 13,33%. Il faut également relever que le nombre de centroïdes le plus performant, tout algorithme confondu est 400, comme mon hypothèse de départ.

## 11.2 Résumé des résultats avec validation croisée

<b>Nombre de centroïdes</b>	<b>Decision Tree</b>	<b>K-NN 4</b>	<b>K-NN 3</b>	<b>K-NN 2</b>	<b>K-NN 1</b>	<b>Fuzzy Rules</b>
100	39,58%	29,17%	25,00%	35,42%	22,92%	25,00%
200	37,50%	29,17%	25,00%	35,42%	22,92%	29,17%
300	31,25%	29,17%	27,08%	35,42%	25,00%	25,00%
400	20,83%	29,17%	27,08%	37,50%	25,00%	29,17%
500	41,67%	29,17%	25,00%	35,42%	22,92%	22,92%

Tableau 15 : Moyenne du taux d'erreurs des algorithmes avec un nombre de centroïdes variable (avec validation croisée)

« Tableau de l'auteur »

Avec la validation croisée, les résultats sont légèrement moins performants, cependant ils restent bons. On remarque, quel que soit le nombre de centroïdes, que les résultats de l'algorithme K-NN avec les 4 voisins les plus proches ne changent pas. Cet algorithme reste donc constant indépendamment du nombre de centroïdes définis. On remarque également que l'algorithme le plus précis est le « Decision Tree » avec 400 centroïdes mais que celui-ci devient également le pire résultat avec 500 centroïdes.

## 12. Comparaison des différentes techniques d'analyses

Tableau 16: Comparatif des techniques d'analyses « Tableau de l'auteur »

<b>Caractéristiques</b>	<b>Série temporelle</b>	<b>Série fréquentielle</b>	<b>Série temporelle corrigée</b>
<i>Nécessite un nombre de points fixes</i>	Il faut que chaque série ait le même nombre d'images pour l'analyse	Non, les points sont transformés en fréquence	Non, seules les distances face aux centroïdes sont conservées
<i>Contraintes des prises de vues</i>	Chaque prise de vues vidéo doit être effectuée dans les mêmes conditions avec la caméra au même endroit	Les prises de vues sont plus flexibles, la caméra ne doit pas obligatoirement toujours filmer le sujet sous le même angle	Chaque prise de vues vidéo doit être effectuée dans les mêmes conditions avec la caméra au même endroit
<i>Robustesse de l'analyse</i>	Les contraintes de départ étant plus fortes, le modèle d'analyse sera moins facile à fausser que celui des fréquences	Possibilité d'effectuer un mouvement différent qui aura la même fréquence, ce qui trompera le modèle d'analyse	Les contraintes de départ étant plus fortes, le modèle d'analyse sera moins facile à fausser que celui des fréquences
<i>Meilleur algorithme d'analyse (sans v-c)</i>	K-NN 2 voisins	K-NN 2 voisins	K-NN 2voisins
<i>Meilleur algorithme d'analyse (avec v-c)</i>	K-NN 1 voisin	K-NN 3 voisins	Decision Tree

### 12.1 Tableau comparatif des algorithmes

Le tableau suivant résume le résultat des algorithmes selon toutes les techniques d'analyses utilisées. On remarque que les algorithmes fonctionnent moins bien en analyse fréquentielle avec validation croisée et fonctionne mieux sur les séries temporelles corrigées.

		<b>ANALYSE</b>	<b>Decision Tree</b>	<b>K-NN 4</b>	<b>K-NN 3</b>	<b>K-NN 2</b>	<b>K-NN1</b>	<b>Fuzzy Rules</b>
Série temporelle	Dataset complet (275 images)		39,99%	51,24%	42,06%	17,46%	28,33%	26,81%
	Dataset complet (275 images) avec Cross-Validation		33,33%	37,50%	31,25%	35,42%	27,08%	29,17%
Série fréquentielle	Dataset complet		81,25%	77,08%	78,33%	52,08%	86,25%	87,74%
	Dataset complet avec Cross-Validation		95,83%	97,92%	95,83%	100,00%	100,00%	100,00%
	Min-Max Dataset		81,74%	73,48%	69,57%	36,52%	70,87%	84,62%
	Min-Max Dataset avec Cross-Validation		91,67%	81,25%	72,92%	87,50%	79,17%	91,67%
Série temporelle corrigée	Dataset complet		28,13%	24,58%	24,79%	13,96%	26,88%	24,79%
	Dataset complet avec Cross-Validation		20,83%	29,17%	27,08%	37,50%	25,00%	29,17%

Tableau 17 : Comparatif des résultats d'algorithmes « Tableau de l'auteur »



*A noter : Ce tableau compare les techniques d'analyses sur l'ensemble des dataset, il ne contient pas les informations concernant les analyses des séries temporelles ayant moins de 275 images (voir analyse temporelle), ni les analyses temporelles corrigées ayant différents nombres de centroïdes (voir analyse temporelle corrigée).*

### 13. Hypothèses d'utilisations réelles

Dans quel cas est-il préférable d'utiliser une analyse fréquentielle, ou une analyse utilisant des séries temporelles ?

Les analyses fréquentielles ont la possibilité de s'affranchir de la position du sujet face à la caméra et sont plus flexibles, malheureusement elles restent moins précises et nécessitent de transformer les données temporelles.

Les analyses se basant sur les séries temporelles sans correction des données grâce au K-means, sont assez rapides et simples à utiliser. On pourrait imaginer ce genre d'analyse effectué même sur un smartphone.

Mon avis est qu'il est encore difficile de créer des applications de détections d'activité très précises, en effet, les résultats d'analyses ont démontré qu'en l'état, il était impossible de détecter un mouvement avec 100% de réussite.

Il faudrait continuer à chercher d'autres méthodes d'analyses en cherchant à diminuer le nombre d'images nécessaires pour la détection du mouvement. En effet, si on imagine une application smartphone qui serait capable de détecter une activité par une simple vidéo, il ne faudrait pas que le sujet soit obligé d'effectuer l'action pendant 15 secondes alors qu'en moyenne un simple saut en dure moins de deux.

## 14. Bilan du travail et améliorations possibles

Si je devais refaire ce travail depuis le début, j'essaierais de modifier plusieurs éléments.

Premièrement, je me concentrerais plus sur le type de données que je récupère afin de mieux comprendre leur importance. Si on prend l'exemple des prises de vues vidéo, j'essayerais d'obtenir un environnement peut-être un peu moins fermé, en modifiant la position de la caméra afin d'observer les modifications que cela pourrait engendrer.

Deuxièmement, je changerais aussi des choses dans ma méthodologie de travail, durant ces 3 mois de travail, j'ai souvent eu recours à des petites astuces pour retoucher mes fichiers d'input et d'output, ne sachant la forme exacte qu'ils devraient avoir. C'est pourquoi je commencerais par mieux définir la structure de mes fichiers afin de les générer directement de la façon la plus adaptée à l'analyse.

De plus, j'essayerais de plus automatiser mon processus de travail, en effet, entre la capture vidéo et l'analyse finale je dois passer par quatre programmes différents afin d'obtenir un résultat significatif.

Il existe également de nombreuses améliorations possibles à ce travail particulièrement au niveau de l'analyse :

- Possibilité d'analyser les mouvements non plus sur le barycentre mais sur un autre point du corps
- Amélioration de l'analyse fréquentielle en essayant d'obtenir des mouvements plus linéaires.
- Possibilité d'amélioration de la méthodologie de travail en créant une application qui effectuerait toutes les actions nécessaires qu'effectuent les quatre autres programmes

En me lançant dans ce projet, je ne savais pas vraiment quels seraient les attentes et les objectifs de ce travail de Bachelor et j'appréhendais vraiment que cela puisse être au-dessus de mes capacités.

Martin Tazlari

Cependant, plus j'avancais, plus les objectifs devenaient clairs et précis et grâce à l'aide des assistants du projet, j'ai pu les atteindre sans encombre. Pour moi, ce travail était un challenge et je suis vraiment fier du résultat. J'espère avoir la chance de retravailler dans le domaine de la recherche un jour.

## 15. Annexes :

La totalité des documents en annexes sont disponibles sur le lien suivant :

[www.mega.co.nz](http://www.mega.co.nz) Login : [aldises@hotmail.fr](mailto:aldises@hotmail.fr) / Mot de passe : « Annexes4TB@MartinTazlari ».

Vous retrouvez l'arborescence suivante :

	Vidéo	Tous les fichiers des sujets enregistrés avec Kinect Studio
	Exportation Vidéos	Fichiers csv exportés depuis l'application développée
	Protocoles	Descriptifs des protocoles des activités effectuées par les sujets
	Knime Input	Fichiers d'entrées pour les Workflows Knime ainsi que les workflows des analyses
	Matlab Input	Fichiers d'entrée pour le script Matlab avec le script et la fonction créée
	Analyses	Tous les résultats des analyses triés par type avec et sans validation croisée
	Cahier des charges et planification	Documents administratifs

*A noter : les annexes sont également disponibles sur le CD ci-joint à l'exception des fichiers vidéo.*

## Références

1. Ashley, J. (2014, March 5). *Quick Reference: Kinect 1 vs Kinect 2*. Repris de <http://www.imaginativeuniversal.com/blog/post/2014/03/05/Quick-Reference-Kinect-1-vs-Kinect-2.aspx>.  
Dernier accès 21.07.2014 .
2. Dngoins (2014, February 25). *Kinect Hearth Rate Detector*. Repris de <https://k4wv2heartrate.codeplex.com/> .  
Dernier accès 21.07.2014.
3. Elsbree, J. (2013, November 18). *Kinect for Windows API*. Repris de <https://connect.microsoft.com> .  
Dernier accès 21.07.2014.
4. Gorman, C. (2013a, march 6). *Kinect Fusion Lets You Build 3-D Models of Anything*. Repris de <http://spectrum.ieee.org/video/consumer-electronics/audiovideo/kinect-fusion-lets-you-build-3d-models-of-anything> .  
Dernier accès 21.07.2014.
5. Gorman, C. (2013b, march 8). *Kinect + Brain Scan = Augmented Reality for Neurosurgeons*. Repris de <http://spectrum.ieee.org/automaton/robotics/medical-robots/microsoft-kinect-fusion-augmented-reality-neurosurgeons> .  
Dernier accès 21.07.2014.
6. Grossman, J. (2007, March 1). *Thermal Infrared vs. Active Infrared: A New Technology Begins to be Commercialized*. Repris de [http://www.irinfo.org/articles/03\\_01\\_2007\\_grossman.html](http://www.irinfo.org/articles/03_01_2007_grossman.html).  
Dernier accès 21.07.2014.
7. Heremo.com. *Modern rehabilitation with Kinect for Windows*. Repris de <http://heremo.com/>.  
Dernier accès 21.07.2014.

8. *Kinect for Windows Team (2013, May 9). Reflexion Health advancing physical therapy with Kinect for Windows. Repris de <http://blogs.msdn.com/b/kinectforwindows/archive/2013/05/09/reflexion-health-advancing-physical-therapy-with-kinect-for-windows.aspx> .  
Dernier accès 21.07.2014.*
9. *Microsoft. Kinect Fusion. Repris de <http://msdn.microsoft.com/en-us/library/dn188670.aspx> .  
Dernier accès 21.07.2014.*
10. *Microsoft (2013, October 14). Welfare Denmark, Municipalities Adopt Home Health Care Solution, Reduce Rehabilitation Costs. Repris de <http://www.microsoft.com/casestudies/Microsoft-Kinect-for-Windows/Welfare-Denmark/Municipalities-Adopt-Home-Health-Care-Solution-Reduce-Rehabilitation-Costs/710000003344> .  
Dernier accès 21.07.2014.*
11. *Microsoft (2014). Kinect for Windows features. Repris de <http://www.microsoft.com/en-us/kinectforwindows/discover/features.aspx> .  
Dernier accès 21.07.2014.*
12. *Moore, S. (2012, August 8). Microsoft's KinEtre Animates Household Objects. Repris de <http://spectrum.ieee.org/video/consumer-electronics/audiovideo/microsofts-kinetre-lets-you-possess-people-inanimate-objects> .  
Dernier accès 21.07.2014.*
13. *Reflexion Health. Repris de <http://reflexionhealth.com/>.  
Dernier accès 21.07.2014.*
14. *Shotton, J. et al. Real-Time Human Pose Recognition in Parts from Single Depth Images. Repris de <http://research.microsoft.com/pubs/145347/bodypartrecognition.pdf>.  
Dernier accès 21.07.2014.*

15. *Shpunt, A. et al. (2008, May 8). Depth-varying light fields for three dimensional sensing. Repris de <https://www.google.com/patents/US20080106746> .  
Dernier accès 21.07.2014.*
16. *Sigal, L. (Fall 2012). Lecture 4 Game Motion Capture. Repris de <http://www.cs.cmu.edu/~yaser/Lecture-4-Games.pdf> .  
Dernier accès 21.07.2014.*
17. *Stanchak, J. (2013, December 30). Microsoft CityNext: Turning Microsoft Kinect into a physical therapy tool. Repris de [http://blogs.msdn.com/b/microsoft\\_uk\\_health\\_blog/archive/2013/12/30/microsoft-citynext-turning-microsoft-kinect-into-a-physical-therapy-tool.aspx](http://blogs.msdn.com/b/microsoft_uk_health_blog/archive/2013/12/30/microsoft-citynext-turning-microsoft-kinect-into-a-physical-therapy-tool.aspx) .  
Dernier accès 21.07.2014.*
18. *Welfare Denmark. Repris de <http://www.welfaredenmark.dk/uk/home/> .  
Dernier accès 21.07.2014.*
19. *Weinhoffer, E. (2013, April 4). Gestual Interaction in the Operating Room with a Wii remote and Kinect. Repris de <http://makezine.com/2013/04/04/gestural-interaction-in-the-operating-room-with-a-wii-remote-and-kinect/> .  
Dernier accès 21.07.2014.*
20. *Oculus VR. Oculus Rift Virtual Reality Headset for 3D Gaming. Repris de <http://www.oculusvr.com/> .  
Dernier accès 21.07.2014.*
21. *Leap Motion. Mac & PC Motion Controller for Games, Design, & More. Repris de <https://www.leapmotion.com/> .  
Dernier accès 21.07.2014.*
22. *Google. Google Glass. Repris de <http://www.google.com/glass/start/> .  
Dernier accès 21.07.2014.*

23. *Wikipedia Kinect for Xbox One*. Repris de <http://www.ign.com/wikis/xbox-one>  
Dernier accès 21.07.2014
24. *Xbox One Features*. Repris de [http://www.ign.com/wikis/xbox-one/Xbox\\_One\\_Features](http://www.ign.com/wikis/xbox-one/Xbox_One_Features) Dernier accès 21.07.2014
25. *Time of Flight camera*. Repris de <http://web.archive.org/web/20090612071500/http://www.3dvsystems.com/technology/3D%20Imaging%20in%20the%20studio.pdf>  
Dernier accès 21.07.2014
26. *Kinect Xbox One Features*. Repris de [http://www.ign.com/wikis/xbox-one/Xbox\\_One\\_Kinect](http://www.ign.com/wikis/xbox-one/Xbox_One_Kinect)  
Dernier accès 21.07.2014
27. *Kinect Fitnect*. Repris de : <http://www.fitnect.hu/>  
Dernier accès 21.07.2014
28. *Kinect Use Cases*. Repris de <http://www.kinecthacks.com/kinect-kiwiban-interactive-wall/>  
Dernier accès 21.07.2014
29. *Kinect House automation*. Repris de [http://blogs.codesources.com/aqaltier/archive/2014/04/16/kinect-speech-recognition-edomus-dommy.aspx?utm\\_source=twitter&utm\\_campaign=5081397727860842&utm\\_medium=social&utm\\_roi=BringrImpact](http://blogs.codesources.com/aqaltier/archive/2014/04/16/kinect-speech-recognition-edomus-dommy.aspx?utm_source=twitter&utm_campaign=5081397727860842&utm_medium=social&utm_roi=BringrImpact)  
Dernier accès 21.07.2014
30. *Micosoft Speech SDK*. Repris de <http://www.microsoft.com/en-us/download/details.aspx?id=10121>  
Dernier accès 21.07.2014
31. *Kinect Studio*. Repris de : <http://msdn.microsoft.com/en-us/library/hh855389.aspx> Dernier accès 21.07.2014  
Dernier accès 21.07.2014

32. Kinect Studio. Repris de : <http://dgoins.wordpress.com/2014/03/30/exploring-the-kinect-studio-v2/>  
*Dernier accès 21.07.2014*
33. Visual Studio 2013. Repris de : <http://www.visualstudio.com/fr-fr/explore/application-development-vs>  
*Dernier accès 21.07.2014*
34. Knime. Repris de : <http://www.knime.org/>  
*Dernier accès 21.07.2014*
35. Repris de : <http://www.faq-logistique.com/Barycentre.htm> *Dernier accès 21.07.2014*  
*Dernier accès 21.07.2014*
36. Format xef . Repris de <http://pcsupport.about.com/od/fileextensions/f/xeffile.htm>  
*Dernier accès 21.07.2014*
37. Filtre avec Kinect 2.0. Repris de : <http://msdn.microsoft.com/en-us/library/jj131024.aspx> *Dernier accès 21.07.2014*  
*Dernier accès 21.07.2014*
38. Filtre avec Kinect 2.0. Repris de : <http://msdn.microsoft.com/en-us/library/jj131429.aspx>  
*Dernier accès 21.07.2014*
39. Cross-validation. Repris de : <http://www.statsoft.fr/concepts-statistiques/glossaire/v/validation-croisee.html>  
*Dernier accès 21.07.2014*
40. Fuzzy Rules. Repris de [http://www.doc.ic.ac.uk/~nd/surprise\\_96/journal/vol4/sbaa/report.fuzrules.html](http://www.doc.ic.ac.uk/~nd/surprise_96/journal/vol4/sbaa/report.fuzrules.html)  
*Dernier accès 21.07.2014*

41. Transformation de Fourier. Repris de : <http://www.futura-sciences.com/magazines/mathematiques/infos/dico/d/mathematiques-transformee-fourier-11880/>  
*Dernier accès 21.07.2014*
42. Série Temporelle. Repris de : [http://labomath.univ-lille1.fr/~jacques/Download/Cours/ST\\_Cours.pdf](http://labomath.univ-lille1.fr/~jacques/Download/Cours/ST_Cours.pdf)  
*Dernier accès 21.07.2014*
43. Fréquence. Repris de : <http://www.futura-sciences.com/magazines/matiere/infos/dico/d/physique-frequence-3940/>  
*Dernier accès 21.07.2014*
44. Regroupement hiérarchique. Repris de :  
<http://rstat.ouvaton.org/?article10/classification-hierarchique>  
*Dernier accès 21.07.2014*
45. K-means. Repris de : <https://www.youtube.com/watch?v=aiJ8II94qck>  
*Dernier accès 21.07.2014*
46. Domotique. Repris de : <http://domotique-info.fr/>  
*Dernier accès 21.07.2014*

## Déclaration sur l'honneur

Je déclare, par ce document, que j'ai effectué le travail de Bachelor ci-annexé seul, sans autre aide que celles dûment signalées dans les références, et que je n'ai utilisé que les sources expressément mentionnées. Je ne donnerai aucune copie de ce rapport à un tiers sans l'autorisation conjointe du RF et du professeur chargé du suivi du travail de Bachelor, y compris au partenaire de recherche appliquée avec lequel j'ai collaboré, à l'exception des personnes qui m'ont fourni les principales informations nécessaires à la rédaction de ce travail et que je cite ci-après :

- M. Yann Bocchi
- M. Stefano Bromuri
- M. Gianluca Rizzo

Martin Tazlari