



ÉCOLE CENTRALE LYON

UE PRO

PROJET D'APPLICATION EN RECHERCHE N°146

---

# Benchmark et analyse de données de capteurs de mouvement

---

***Élèves :***

Fernanda IROKAWA  
Pedro PINHEIRO

***Tuteurs :***

Romain VUILLEMOT  
Stéphane DERRODE

***Conseilleur :***

Arnaud BREARD

17 avril 2018

# Table des matières

|          |  |           |
|----------|--|-----------|
| <b>1</b> | <b>Introduction</b>  | <b>3</b>  |
| 1.1      | Contexte . . . . .   | 3         |
| 1.2      | Objectifs . . . . .  | 3         |
| 1.2.1    | Prise en main des capteurs de mouvement . . . . .  | 3         |
| 1.2.2    | Réaliser la synchronisation et sa représentation graphique pour des trajectoires générées . . . . .                        | 4         |
| 1.2.3    | Enregistrer les mouvements avec différents capteurs et les représenter graphiquement pour une trajectoire réelle . . . . . | 5         |
| 1.2.4    | Création d'une vidéo avec les données synchronisées . . . . .  | 5         |
| <b>2</b> | <b>Motion Capture (MoCap)</b>  | <b>6</b>  |
| 2.1      | Le principe . . . . .  | 6         |
| 2.2      | La plateforme AMIGO . . . . .  | 7         |
| 2.3      | Capture de données . . . . .   | 7         |
| <b>3</b> | <b>Accéléromètre</b>   | <b>8</b>  |
| 3.1      | Le principe . . . . .  | 8         |
| 3.2      | Le capteur Shimmer3 . . . . .  | 8         |
| 3.3      | Capture de données . . . . .   | 9         |
| <b>4</b> | <b>Comparaison des Capteurs</b>  | <b>10</b> |
| 4.1      | MoCap . . . . .  | 10        |
| 4.2      | Accéléromètre . . . . .  | 10        |
| <b>5</b> | <b>Méthodes d'analyse de données</b>   | <b>11</b> |
| 5.1      | Dynamic Time Warping . . . . .   | 11        |
| 5.2      | Generalized Time Warping . . . . .   | 13        |
| <b>6</b> | <b>Expérimentations</b>  | <b>16</b> |
| 6.1      | Données synthétiques . . . . .   | 16        |
| 6.1.1    | Trajectoire rectangulaire . . . . .  | 16        |
| 6.1.2    | Simulation de trajectoire polynomial 3D . . . . .  | 21        |
| 6.2      | Données réelles . . . . .  | 24        |
| <b>7</b> | <b>Interface graphique</b>   | <b>27</b> |
| <b>8</b> | <b>Conclusion et perspectives</b>  | <b>29</b> |
|          | <b>Références</b>  | <b>30</b> |

## Résumé

Ce rapport présente les méthodes, l'analyse et les résultats du travail développé au sein du Projet d'Application en Recherche 146 ayant pour objectif l'analyse des différents capteurs de mouvement (notamment le Motion Capture et l'accéléromètre). D'abord, l'étude et la prise en main des capteurs de mouvement sont faites pour pouvoir capturer et analyser les données de trajectoires enregistrées par eux. Ensuite, nous nous intéressons à l'étude de différentes méthodes d'analyse de données, notamment le Dynamical Time Warping (DTW) et d'une variante de ce méthode, le Generalized Time Warping. Nous réalisons des expérimentations avec ces méthodes, en utilisant des données synthétiques et des données réelles, et nous ouvrons l'horizon pour d'autres analyses à effectuer. Enfin, nous élaborons un outil de visualisation des données sur lesquelles nous avons travaillé.

# 1 Introduction

## 1.1 Contexte

Actuellement, nous disposons de plusieurs méthodes d'enregistrement de mouvements et de trajectoires, qui fournissent différents types d'information concernant ces trajectoires. Par exemple, les téléphones portables qui ont le GPS intégré et fournissent la position géographique de son utilisateur, les accéléromètres qui enregistrent l'accélération de mouvements, les gyroscopes qui fournissent les changements d'orientation, et encore plusieurs autres.

Dans le cas de l'enregistrement d'une trajectoire en utilisant différents capteurs, un problème fondamental qui apparaît est relié à l'assimilation de ces différentes données. Qu'il soit les différentes dimensions des données (position dans l'espace, accélération selon les trois axes) ou même l'écart temporel qui provient des différentes fréquences de capture, une synchronisation de ces données est nécessaire.

Nous nous intéressons finalement à la problématique de la synchronisation de données des différents capteurs : le motion capture (MoCap), l'accéléromètre et la caméra vidéo. Ce type d'étude a déjà été réalisé par Zhou et De la Torre [1], [2], [3], mais en s'intéressant à la synchronisation d'un même mouvement avec différents capteurs réalisée par différents personnes. Pour cela, ils s'appuient sur différents méthodes de synchronisation, spatiale et temporelle, et explorent plus précisément le « Generalized Canonical Time Warping ».

Pour notre projet, nous envisageons réaliser la synchronisation des données d'un mouvement simple capturé avec les différents capteurs. Pour cela, il est nécessaire une étude préalable des méthodes de synchronisation existants, qu'ils soient temporels, spatiaux ou le deux.

## 1.2 Objectifs

On a divisé notre projet en quatre parties principales. Chaque partie a été séparée en sous-objectifs. Ces objectifs ont été décidés en utilisant la technique SMART (Spécifique, Mesurable, Atteignable, Réaliste et Temporellement défini).

### 1.2.1 Prise en main des capteurs de mouvement

#### MoCap

Dans cette étape, notre objectif est prendre en main l'utilisation de l'appareil de motion capture situé dans la salle Amigo. Cela implique : maîtriser l'installation, le calibrage et l'utilisation des équipements et l'exportation de données. Pendant cette étape, on réalise l'acquisition des données de mouvements simples ainsi que la rédaction d'une [méthodologie](#) pour l'acquisition de ces données. Cette étape sera faite en suivant un [tutoriel](#) sur l'acquisition de mouvements rédigé par les anciennes élèves qui ont déjà utilisé l'équipement.

## Accéléromètres

De façon similaire à la prise en main du MoCap, on veut prendre en main l'utilisation des accéléromètres. Cela implique, encore une fois, maîtriser l'installation, le calibrage, l'utilisation des équipements et l'exportation de données. Pendant cette étape, on réalise aussi la rédaction d'un [tutoriel](#) pour la calibration et l'utilisation des accéléromètres ainsi que l'acquisition des données de mouvements simples. Pour cela, on utilise les logiciels et tutoriels qui sont disponibles dans le site web de l'entreprise qui produit les accéléromètres [4]. Pour la partie de calibration, on utilise le logiciel « Shimmer 9Dof Calibration » [5]. Pour les autres parties, on utilise le logiciel « Consensys PRO » [6].

### 1.2.2 Réaliser la synchronisation et sa représentation graphique pour des trajectoires générées

#### Génération de trajectoires pour une première synchronisation

Pendant la deuxième partie de notre projet, on vise, tout d'abord, à réaliser la génération computationnelle de trajectoires simples pour faire une première synchronisation avant de passer à l'acquisition d'une trajectoire réelle. Pour réaliser cette tâche, on crée un code python, avec lequel on peut générer graphiquement les trajectoires simples qui simulent l'enregistrement réel.

#### Synchronisation des données générées

Après la génération de trajectoires, on vise à réaliser la synchronisation, avec des codes en python et en MatLab, des trajectoires simples générées. Pour faire la synchronisation, on essaie de regarder les points d'intérêt des trajectoires (une altération de direction ou un changement de rythme par exemple), le taux d'erreur entre la trajectoire et sa référence ainsi que faire la correction des bruits (signal parasite qui génère des points qui ne correspondent pas à la trajectoire attendue).

#### Test de la robustesse

Ici, on essaye de comparer les résultats obtenus pour la synchronisation de trajectoires plus complexes par rapport à ceux des trajectoires simples. Pour ce test, on génère des trajectoires plus complexes et on regarde les résultats graphiques pour voir comment les méthodes sont affectées par la complexité de la trajectoire analysée (l'échelle, le bruit, le type de donnée utilisée).

#### Comparaison des différents méthodes de synchronisation

Dans cette étape, notre objectif est de comparer les résultats obtenus pour différentes méthodes de synchronisation de trajectoires. Pour comparer les méthodes utilisées, on regarde les résultats graphiques pour voir quelle méthode de synchronisation a été plus adaptée pour chaque type de trajectoire analysée.

### **1.2.3 Enregistrer les mouvements avec différents capteurs et les représenter graphiquement pour une trajectoire réelle**

#### **Enregistrement de données d'une trajectoire réelle**

Pour pouvoir synchroniser les données d'une trajectoire réelle enregistrées avec différents capteurs, on utilise le MoCap et les accéléromètres pour enregistrer une trajectoire réelle simple.

#### **Synchronisation des données pour une trajectoire réelle**

Cette étape a pour but de réaliser l'exportation et la synchronisation des données enregistrées pour les trajectoires réelles. Après les avoir enregistré, on exporte les données référents à chaque capteur et on les synchronise en utilisant les méthodes déjà utilisées et codés en python et MatLab.

### **1.2.4 Création d'une vidéo avec les données synchronisées**

Pour montrer nos résultats, on vise à faire une vidéo montrant les données déjà synchronisées. Pour accomplir cet objectif, on utilise un code en python pour reproduire les graphiques qui montrent les données synchronisées.

## 2 Motion Capture (MoCap)

### 2.1 Le principe

La technique de motion capture est la plus utilisée pour faire les animations et les effets spéciaux dans les films. Cette technique fonctionne en utilisant les différents capteurs, qui enregistrent les données des trajectoires et les transmettent à un logiciel capable d'interpréter ces données brutes et reconstruire les trajectoires 3D en temps réel dans un environnement virtuel. Cette technologie est utilisée pour l'étude des mouvements humains, mais aussi dans les domaines de la santé et de l'anatomie.

La capture peut être faite de différentes façons (optique, gyroscopique, mécanique ou magnétique). Dans ce rapport on ne s'intéressera qu'à la capture optique, disponible dans la salle amigo et aussi la plus utilisée.

La capture optique, à son tour, peut être faite de deux façons (avec marqueurs et sans marqueurs). La technologie utilisée par le Kinect par exemple n'utilise pas des marqueurs. L'appareil de motion capture situé dans la salle AMIGO, utilise les marqueurs pour capturer les données et, par conséquent, on ne parlera que de ce type de capture.

Ce type de capture est fait en plaçant des marqueurs sur la personne ou l'objet qui ira réaliser le mouvement (figure 1). Ces marqueurs sont capables de réfléchir le rayonnement infrarouge. Les caméras émettent un rayonnement infrarouge et captent sa réflexion. Comme les caméras ne sont sensibles qu'à les longueurs d'onde dans le domaine infrarouge, les caméras n'enregistrent pas l'image visible par l'être-humain, mais si la réflexion des marqueurs sur forme de points. Il faut dire que les marqueurs sont de format arrondi pour que la réflexion soit faite pour n'importe quel angle d'incidence des rayonnements.

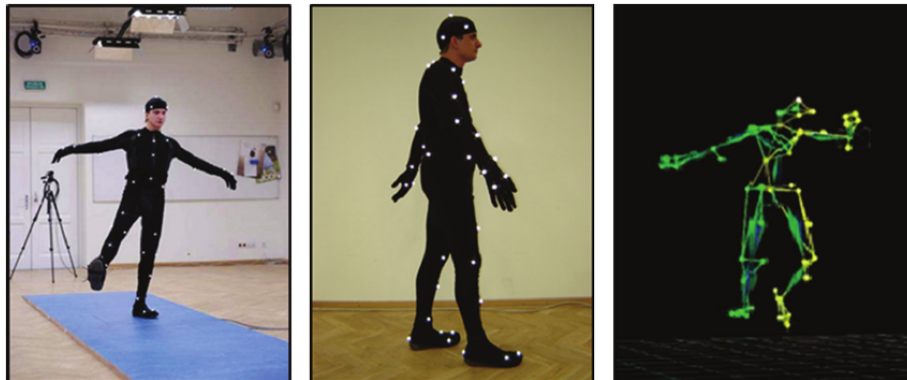


FIGURE 1 – Capture de mouvement en utilisant le Motion Capture. [7]

Vu qu'une caméra toute seule n'est pas capable d'observer 3 dimensions, il faut au moins deux caméras pour que le logiciel puisse calculer la position du point dans l'espace par triangulation. Pour la rotation il faut au moins deux capteurs qui doivent être liés manuellement à partir du logiciel. Ces deux points seront donc traités comme une trace dans la visualisation virtuelle. Cela permet le logiciel de faire le calcul et la représentation de la rotation dans l'environnement virtuel.

## 2.2 La plateforme AMIGO

À l'École Centrale Lyon, nous avons eu accès à la salle AMIGO située au deuxième étage du bâtiment W1. AMIGO est une plateforme d'analyse de mouvement. Cette salle est équipée de 6 caméras infrarouges Vicon, de nombreux marqueurs réfléchissants, des vêtements spéciaux sur lesquels les marqueurs peuvent être collés, d'une croix de calibration de l'équipement et d'un ordinateur avec le logiciel Vicon installé et prêt à l'emploi. Un tutoriel sur l'utilisation de l'équipement était également disponible.

Selon la référence [8], l'erreur de capture du système de capture de mouvement Vicon est négligeable dans l'axe Z par rapport à l'erreur dans les composants XY. De plus, l'erreur est réduite à des vitesses plus élevées et à des marqueurs de plus petit diamètre. L'erreur peut être réduite jusqu'à 40 % pour le même objet si la vitesse est augmentée. Pour une capture statique, l'erreur de positionnement moyenne est de 0,15 mm et pour un objet en mouvement, l'erreur est inférieure à 2 mm.

## 2.3 Capture de données

Avant la capture de mouvement, une calibration simple doit être effectuée en utilisant une croix de calibration. Tous les objets pouvant refléter l'infrarouge doivent être retirés de la zone de capture. Après avoir retiré tous les objets réfléchissants, les caméras peuvent toujours détecter les faux points de réflexion. Afin que ces points n'interfèrent pas avec le calibrage et la capture de mouvement, le logiciel permet à l'utilisateur de sélectionner les zones qui devraient être négligées pendant la capture de mouvement. Cette sélection est faite pour chaque caméra individuellement.

Les marqueurs réfléchissants doivent être positionnés sur les parties clés de l'objet. Par exemple, pour capturer un mouvement d'un bras, il est important que nous ayons un marqueur sur le coude, un dans la main et un dans l'épaule. Ceci est important car après la capture, il est possible de relier ces points et donc de reconstituer le mouvement capturé (figure 1). Enfin, le programme permet la visualisation en plus d'exporter les données de la capture (par exemple la position dans l'espace de chaque marqueur), de forme simple et dans divers formats.



## 3 Accéléromètre

### 3.1 Le principe

Quand on parle d'un accéléromètre, en réalité, on parle d'un système trois accéléromètres. Chacun de ses accéléromètres mesure l'accélération linéaire dans une des trois axes de directions orthogonaux. Le principe peut être illustré par un système masse-ressort. Si on fait un mouvement dans un système de ce type, la masse essaie de rester dans sa position initiale à cause de l'inertie. Le ressort est alors allongé ou comprimé et par la loi fondamentale de la dynamique pour une ressort,  $F = K \times x$ , mais aussi  $F = m \cdot a$ . Alors, si on sait la différence de position de la masse ( $\Delta x$ ), la masse  $m$  et la constante de rappel du ressort  $K$ , on peut calculer le module et la direction de l'accélération dans une axe. Avec trois systèmes, on peut observer l'accélération dans les trois dimensions et, en observant les changements d'accélération dans les 3 axes, on peut avoir la rotation d'accéléromètre.



FIGURE 2 – Accéléromètre du type Shimmer3. [4]

### 3.2 Le capteur Shimmer3

Dans notre projet, on utilise l'accéléromètre Shimmer3 (figure 2). Pour exporter et pour traiter les données pour cet accéléromètre de façon simple et pour configurer le Shimmer3 (changer la fréquence d'enregistrement par exemple), on peut utiliser le logiciel Consensus PRO [6]. Pour la calibration des accéléromètres on utilise le logiciel Shimmer 9DoF Calibration [5].

Selon la référence [9], le Shimmer3 a une densité de seulement  $45 \mu g / \sqrt{Hz}$ . Avec une résolution de 16 bits, Shimmer3 présente une représentation plus fiable du signal original. Les erreurs moyennes et les écarts-types des gyroscopes et des accéléromètres dans chaque axe sont représentés sur les figures 3 et 4. Il est important de noter que le calibrage 9DoF élimine le biais, les erreurs de sensibilité et les erreurs d'alignement de l'appareil.

| (Unit: $m/s^2$ ) | Average | Standard Deviation |
|------------------|---------|--------------------|
| Accelerometer X  | -0.06   | 0.02               |
| Accelerometer Y  | -0.08   | 0.02               |
| Accelerometer Z  | 0.03    | 0.03               |

FIGURE 3 – Erreur moyenne et écart-type - Accéléromètres. [9]

| (Unit: deg/sec) | Average Error | Standard Deviation |
|-----------------|---------------|--------------------|
| Gyroscope X     | 0.02          | 0.08               |
| Gyroscope Y     | 0.01          | 0.10               |
| Gyroscope Z     | 0.03          | 0.09               |

FIGURE 4 – Erreur moyenne et écart-type - Gyroscopes. [9]

### 3.3 Capture de données

Pour effectuer la capture de données à l'aide de plusieurs Shimmer3, les accéléromètres doivent être synchronisés. La capture de données peut être effectuée à distance en utilisant un ordinateur qui recevra les données capturées par Bluetooth ou en utilisant les propres accéléromètres qui ont un bouton qui doit être pressé pour démarrer la capture et à nouveau pressé pour la terminer. La capture sera ensuite enregistrée sur la carte mémoire du Shimmer3. Pour l'exportation de données, le programme Consensys PRO [6] permet de les exporter dans un format simple et dans plusieurs formats. Pour une explication plus détaillée de l'étalonnage et de l'utilisation des accéléromètres, nous avons créé un [tutoriel](#) qui est disponible en-ligne.

## 4 Comparaison des Capteurs

### 4.1 MoCap

#### Avantages

L'appareil de capture de mouvement optique a une calibration facile avec une croix de calibration. Il faut juste mouvementer la croix dans la salle. Aussi, on a une représentation compréhensible et virtuelle de la trajectoire en temps réel. L'exportation des données enregistrées peut être faite de façon très simple avec le logiciel.

#### Désavantages

L'appareil présente des désavantages liés à l'optique. L'infrarouge n'est pas capable de traverser les objets et alors nous avons la possibilité de cacher un ou plus marqueurs pendant quelques instantes. Les marqueurs doivent, alors, être placés de façon à qu'ils soient visibles durant toute l'acquisition. Aussi, la capture peut être perturbée à cause des points parasites dus aux objets qui sont aussi capables de réfléchir l'infrarouge. Un deuxième problème est lié à l'espace d'utilisation de l'appareil. Le mouvement doit être fait dans l'espace délimitée par la vision des caméras. L'appareil est, donc, fixé et limite les types de mouvement que peuvent être fait. Un troisième problème est le prix des appareils qui est très élevé. Celui de la salle AMIGO vaut au tour de 40.000 euros par exemple.

### 4.2 Accéléromètre

#### Avantages

Le logiciel Consensus PRO pour l'accéléromètre Shimmer3 permet d'exporter et traiter les données de façon simple et rapide. L'accéléromètre est portable et donc permet d'analyser mouvements sans une contrainte d'espace. La calibration est aussi simple et rapide avec le logiciel Shimmer 9DoF. Il est beaucoup moins cher que les appareils MoCap.

#### Désavantages

L'accéléromètre ne donne pas directement la position spatiale. Pour cela, il faut utiliser des algorithmes qui essaient de calculer cette position en utilisant les accélérations, ce qui normalement ne donne pas des résultats précis.

## 5 Méthodes d'analyse de données

Il existe de plusieurs méthodes pour effectuer la synchronisation de séquences de données. Dans notre projet, nous nous sommes concentrés sur l'analyse de deux méthodes, le Dynamical Time Warping [1], [2], [3] et le Generalized Time Warping [3]. Ces méthodes ont été bien étudiées et développées et constituent donc une bonne base pour l'étude de la synchronisation des données.

### 5.1 Dynamic Time Warping

Le Dynamic Time Warping (DTW) [1], [2], [3] a été étudié en plusieurs cas de synchronisation de trajectoires. Il est une méthode simple qui normalement est incorporé à d'autres méthodes de synchronisation plus puissants et plus complexes. En dehors du contexte de trajectoires, il a été déjà utilisé pour réaliser la reconnaissance vocale, la correspondance de signatures. Il consiste à synchroniser deux séries de données en utilisant la distance minimale entre chaque point de chaque trajectoire (figure 5).

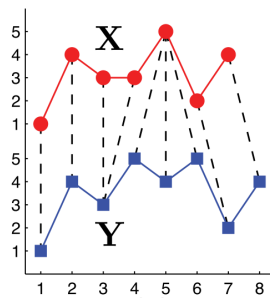


FIGURE 5 – Deux trajectoires X et Y. Les lignes pointillées représentent la synchronisation en utilisant la méthode DTW. [1]

#### Le principe

A partir de deux séries temporelles de données spatiales  $X = [x_1, \dots, x_{n_x}] \in \mathbb{R}^{2 \times n_x}$  et  $Y = [y_1, \dots, y_{n_y}] \in \mathbb{R}^{2 \times n_y}$ , la synchronisation des données doit minimiser la distance entre les points alignés.

$$\min(J_{dtw}(P)) = \sum_{t=1}^m \|x_{p_t}^x - y_{p_t}^y\|^2 \quad (1)$$

Où  $P$  est la matrice de correspondance  $P = [p^x, p^y]^T \in \mathbb{R}^{2 \times m}$  et  $m$  est le nombre d'indices nécessaires pour réaliser la correspondance entre les séries. Les deux vecteurs qui composent cette matrice de correspondance sont  $p_x \in \{1 : n_x\}^{m \times 1}$  et  $p_y \in \{1 : n_y\}^{m \times 1}$ . La matrice de correspondance  $P$  doit satisfaire les contraintes :

- Condition de frontière :  $p_1 = [1, 1]^T$  et  $p_m = [n_x, n_y]^T$ , c'est-à-dire, le premier et le dernier frame des séries doivent être alignés.

- Condition de continuité :  $0 \leq p_t - p_{t-1} \leq 1$

- Condition de monotonie :  $t_1 \geq t_2 \Rightarrow p_{t_1}^x \geq p_{t_2}^x$  et  $p_{t_1}^y \geq p_{t_2}^y$

Le  $i_{eme}$  frame de la série  $X$  sera aligné au  $j_{eme}$  frame de la série  $Y$  s'il existe un  $p_t$  tel que  $p_t = [p_t^x, p_t^y] = [i, j]$ .

## Implémentation

Pour pouvoir implémenter la méthode du DTW, il était nécessaire d'envisager quelles structures seraient fondamentales pour résoudre le problème. La synchronisation des données est faite en trouvant le chemin optimal (par rapport à la distance entre les points de chaque série) pour arriver du premier point au dernier. La figure 6 montre une représentation du chemin qu'on veut trouver (en rouge). En effet, à chaque pair  $(i, j)$  de cette matrice, on peut trouver la distance entre les deux points  $x_i$  et  $y_j$ . Le chemin va donc passer par les points où cette distance est minimale, en respectant toujours le sens qu'il peut suivre (soit à gauche, soit en bas ou soit à gauche et en bas – figure 7).

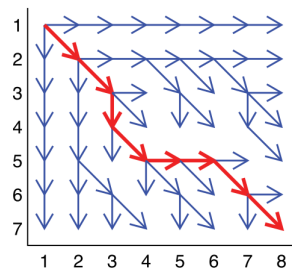


FIGURE 6 – Représentation graphique méthode du DTW pour trouver le chemin optimal de synchronisation des deux séries temporelles X et Y. [1]

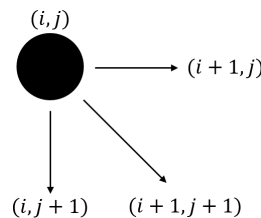


FIGURE 7 – Représentation des chemins possibles pour un point  $(i, j)$ .

L'implémentation passe donc par une matrice de distances où chaque élément  $(i, j)$  de la matrice correspond à la distance entre ces deux points  $x_i$  et  $y_j$  (Figure 8). On peut aussi retrouver les données synchronisées en partant du dernier élément de la matrice représentée par la figure 8, c'est-à-dire, le point  $\{n_x, n_y\}$  (condition de frontière) et en réalisant le chemin inverse pour y arriver.

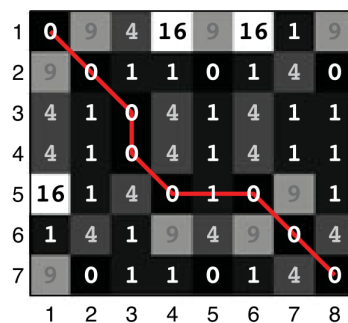


FIGURE 8 – Représentation de la matrice de distances pour deux trajectoires X et Y. [1]

La représentation graphique de cette synchronisation (figure 9) permet de montrer l'association entre ces points et sert aussi à savoir si tous les conditions aux limites ont été respectées. En effet, à chaque point  $p_t$ , l'élément  $(p_t^x, p_t^y)$  de la matrice représenté sur la figure assume la valeur 1. Ce type d'illustration sera utilisé pour une analyse de nos résultats.

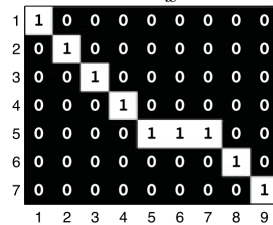


FIGURE 9 – Représentation de la matrice de correspondance. [1]

## 5.2 Generalized Time Warping

Une deuxième méthode étudiée est le Generalized Time Warping (GTW) [3]. Cette méthode est une variante du Dynamical Time Warping et a été développée par Zhou et De la Torre pour pouvoir réaliser l'alignement de données spatio-temporales de différents capteurs de mouvement (comme le MoCap, les accéléromètres et la caméra vidéo) d'un même mouvement réalisé par des individus différents (figure 10). L'avantage du GTW par rapport au DTW est la possibilité de réaliser le recalage des trajectoires qui n'ont pas les mêmes dimensions (données dites "multimodales").

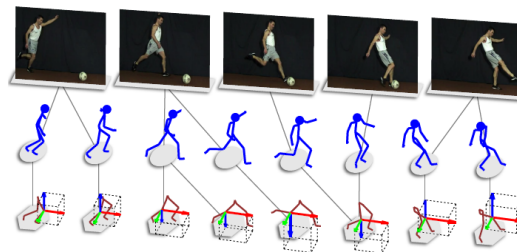


FIGURE 10 – Alignement temporel de trois séquences de sujets différents battant une balle. Enregistré avec différents capteurs (vidéo en haut, capture de mouvement au milieu et accéléromètres en bas). [3]

La méthode DTW présente trois problèmes principaux : En utilisant la programmation dynamique pour trouver le meilleur chemin possible, elle a une complexité quadratique ( $O(n_x, n_y)$ ) dans le temps et l'espace et n'est donc pas adaptée à une grande séquence de données. La méthode n'est pas efficace pour plus de deux séquences puisque le nombre de chemins possibles augmenterait considérablement. Un alignement, utilisant la programmation dynamique, de  $m$  séquences implique une complexité de  $O(\prod_{i=1}^m n_i)$  dans le temps et l'espace. De plus, la méthode DTW n'applique aucun type de mécanisme de pondération aux données traitées.

La GTW prend en compte les différentes dimensions des signaux par l'analyse des corrélations canoniques et réalise une déformation temporellement flexible basée sur des fonctions monotones simples. Il utilise aussi un algorithme de Gauss-Newton, qui, au lieu d'une complexité quadratique, a une complexité linéaire basée sur la taille de la séquence de données.

## Le Principe

Pour une collection de  $m$  séries  $\{X_1, \dots, X_m\}$ , avec  $X_i = [x_1^i, \dots, x_{n_i}^i] \in \mathbb{R}^{d_i \times n_i}$ . GTW trouve pour chaque  $X_i$ , une transformation temporelle non linéaire  $W_i \in \{0, 1\}^{n_i \times l}$  et une intégration de dimension faible  $V_i \in \mathbb{R}^{d_i \times d}$  de sorte que la séquence  $V_i^T X_i W_i \in \mathbb{R}^{d \times l}$  est alignée avec les autres. La GTW cherche à minimiser la somme des distances des paires :

$$J_{gtw}(\{W_i, V_i\}) = \sum_{i=1}^m \sum_{j=1}^m \frac{1}{2} \|V_i^T X_i W_i - V_j^T X_j W_j\|_F^2 + \left( \sum_{i=1}^m \psi(W_i) + \phi(V_i) \right),$$

où  $W_i \in \Psi$  et  $V_i \in \Phi$ ,  $\forall i \in \{1 : m\}$  (2)

On alterne entre la solution de  $W_i$  en utilisant un algorithme de Gauss-Newton et l'optimisation de  $V_i$  par l'analyse des corrélations canoniques. Ces étapes alternatives diminuent  $J_{gtw}$  et convergent vers un point critique.

Pour réduire la complexité de calcul, GTW approche le chemin de déformation  $p \in \{1 : n\}^l$  à une combinaison linéaire de fonctions monotones  $q \in [1, n]^l$  :

$$p \approx \sum_{\bar{c}=1}^{\bar{k}} \bar{a}_{\bar{c}} \bar{q}_{\bar{c}} + \sum_{\acute{c}=1}^{\acute{k}} \acute{a}_{\acute{c}} \acute{q}_{\acute{c}} = \bar{Q} \bar{a} + \acute{Q} \acute{a} = Q a \quad (3)$$

$a = [\bar{a}, \acute{a}]$  est le vecteur de pondération et  $Q = [\bar{Q}, \acute{Q}]$  est l'ensemble de base qui contient les fonctions constantes  $\bar{Q} = [\bar{q}_1, \dots, \bar{q}_{\bar{k}}]$  et les fonctions croissantes monotones  $\acute{Q} = [\acute{q}_1, \dots, \acute{q}_{\acute{k}}]$ .

Comme dans le DTW, le GTW a contraintes qui doivent être satisfaites par la pondération  $a$  telle que  $p = Qa$ .

- Condition de frontière : la position de le premier cadre  $p_1 = q^{(1)}a \geq 1$  et la position du dernier cadre  $p_l = q^{(l)}a \leq n$ .  $q^{(1)}$  et  $q^{(l)}$  sont la première et la dernière ligne de la matrice de base  $Q$  respectivement. GTW permet, avec les contraintes d'inégalité, de indexer par  $p$  une sous-partie de  $X$ .
- Condition de continuité : Pour approcher la contrainte de continuité du DTW ( $0 < p_t - p_{t-1} < 1$ ), on applique une condition sur la courbure ( $F$  est l'opérateur de la première dérivée) :

$$\sum_{t=1}^l \|\nabla q^{(t)} a\|_2^2 \approx \|FQa\|_2^2 \quad (4)$$

- Condition de monotonicité : Pour avoir  $t_1 < t_2 \implies p_{t_1} < p_{t_2}$ , on fait que le poids soit positive :  $\acute{a} > 0$ .

## Implémentation

Un exemple de fonction de gauchissement est montré sur la figure 11, où la fonction de gauchissement  $p$  est la combinaison linéaire de deux fonctions monotones croissantes et une fonction constante  $q$ .

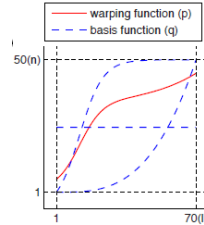


FIGURE 11 – Exemple de fonction de déformation  $p$ . [3]

Au début, on initialise les pondérations  $q_i$  par un alignement uniforme des séquences (GN-Init sur la figure : 12). La longueur du chemin de déformation est choisie comme :  $l = 1.1 \times \max_{i=1}^m (n_i)$ . Si la longueur  $l$  est très élevée, une condition doit être ajoutée pour assurer un résultat stable.

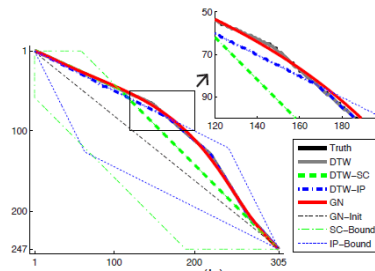


FIGURE 12 – Comparaison des chemins  $p$  pour différentes méthodes. [3]

Pour 100 paires de séquences aléatoires 1-D, si on compare le GTW (GN sur la figure 13) et le DTW, on voit que le DTW a une erreur inférieure au GTW (figure 13a), mais un temps de calcul plus élevé (figure 13b). Le GTW est plus efficace parce que il a une complexité linéaire en relation à la longueur de la séquence. On peut voir aussi que le nombre de fonctions de base réduit l'erreur (6, 10 et 14 sur la figure 13).

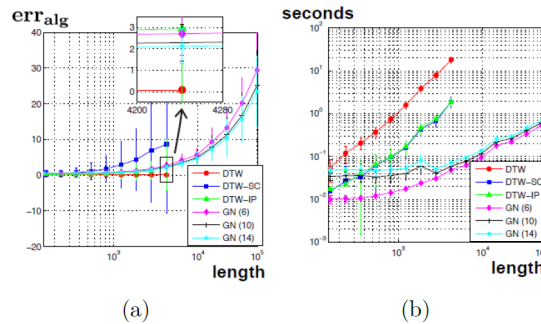


FIGURE 13 – (a) Comparaison des erreurs pour différentes méthodes et (b) Comparaison des temps de calcul pour différentes méthodes. [3]



## 6 Expérimentations

Pour pouvoir étudier la performance des méthodes d'analyse de données, l'utilisation de différents jeux de données est nécessaire. On commence d'abord avec l'étude de données synthétiques, c'est-à-dire, des trajectoires qu'on a simulé et dont les paramètres on pouvait contrôler. En partant sur des trajectoires très simples, on augmente le niveau de complexité (en ajoutant des imprécisions ou de bruit, par exemple) pour s'approcher de la simulation des trajectoires enregistrées par les capteurs. Cela nous permet finalement d'analyser le comportement des méthodes d'analyse face à des données réelles enregistrés lors des sessions de capture à la plateforme Amigo et avec des accéléromètres.

### 6.1 Données synthétiques

#### 6.1.1 Trajectoire rectangulaire

On s'intéresse à tracer une trajectoire de référence (idéale) et une trajectoire "aléatoire", qui sera basée sur cette trajectoire de référence. A cette trajectoire aléatoire on ajoute des imprécisions, bruits et/ou changement d'échelles.

##### Trajectoire de référence

D'abord, on construit la trajectoire de référence : un rectangle 50x100, avec une segmentation de 200 points (figure 14).

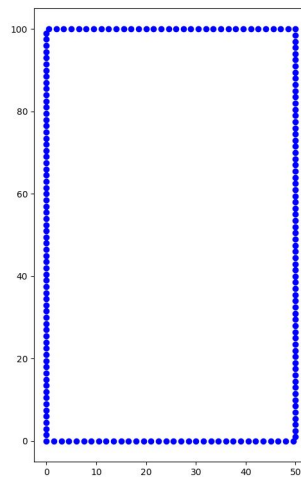


FIGURE 14 – Trajectoire de référence : rectangle 50x100 décrit par 200 points.

##### Trajectoire aléatoire sans bruit

A partir de la trajectoire de référence, on construit une trajectoire aléatoire, un rectangle 50x100 de segmentation de 100 points auxquels on ajoute une imprécision. Cette imprécision est caractérisée par une loi uniforme continue, d'intervalle  $[-5, 5]$ .

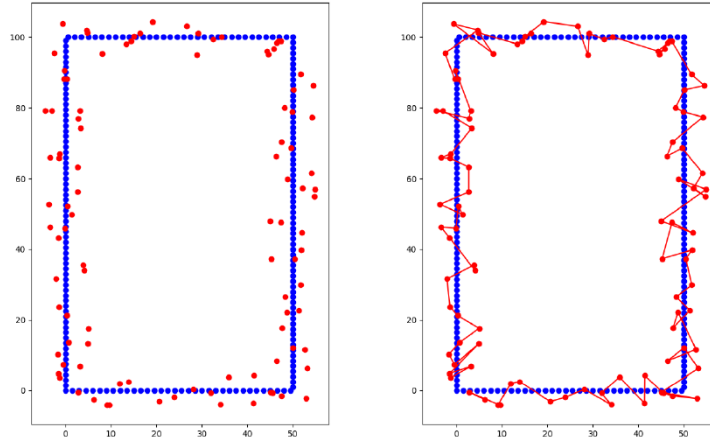


FIGURE 15 – Trajectoire de référence et trajectoire aléatoire avec imprécision uniforme continue d'intervalle  $[-5, 5]$ .

Avec cette première trajectoire aléatoire, il est finalement possible d'utiliser le DTW pour tester la correspondance des points des trajectoires. On réalise donc la simulation et on obtient la matrice de correspondance (représentée par la figure 16a) et le recalage entre les deux trajectoires (figure 16b). Les traits noirs montrent la correspondance entre les points de la trajectoire de référence avec ceux de la trajectoire aléatoire.

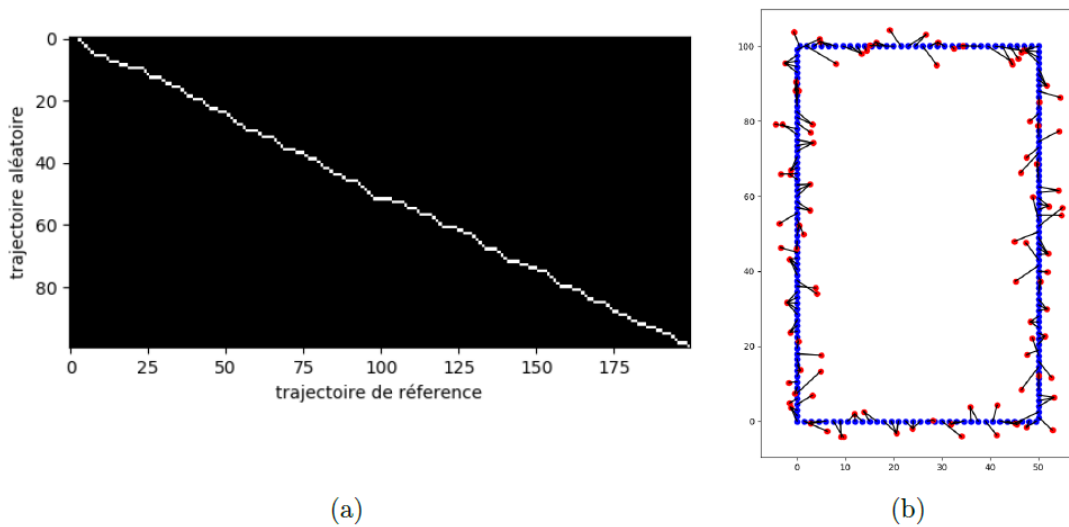


FIGURE 16 – (a) Matrice de correspondance pour la trajectoire réelle et la trajectoire aléatoire et (b) Recalage (en noir) entre les points des trajectoires de référence (en bleu) et aléatoire (en rouge).

On voit que la synchronisation tend vers la diagonale qui lie le premier ( $p_1$ ) et le dernier point ( $p_m$ ). De plus, on obtient une matrice de correspondance de taille  $m = 200$ . Cela veut dire qu'à chaque point de la trajectoire de référence on associe un et seulement un point de la trajectoire « aléatoire ». Par contre, les points de cette dernière s'associent à un ou plusieurs points de la référence. Il est possible d'observer cela à partir de la largeur du rectangle en blanc à chaque ligne de la figure 16a. Ce résultat est attendu car la trajectoire « aléatoire » a été créée en utilisant les points de la référence.

### Trajectoire aléatoire avec bruit impulsif

A la trajectoire aléatoire traitée précédemment on ajoute un bruit impulsif, dont la probabilité d'occurrence est définie en 10% et qui va donc introduire un bruit de loi uniforme d'intervalle  $[30, 50]$  pour  $x$  et  $[-30, -50]$  pour  $y$ . On réalise à nouveau la synchronisation avec le DTW, la matrice de correspondance et le recalage de points étant représentés sur la figure 17.

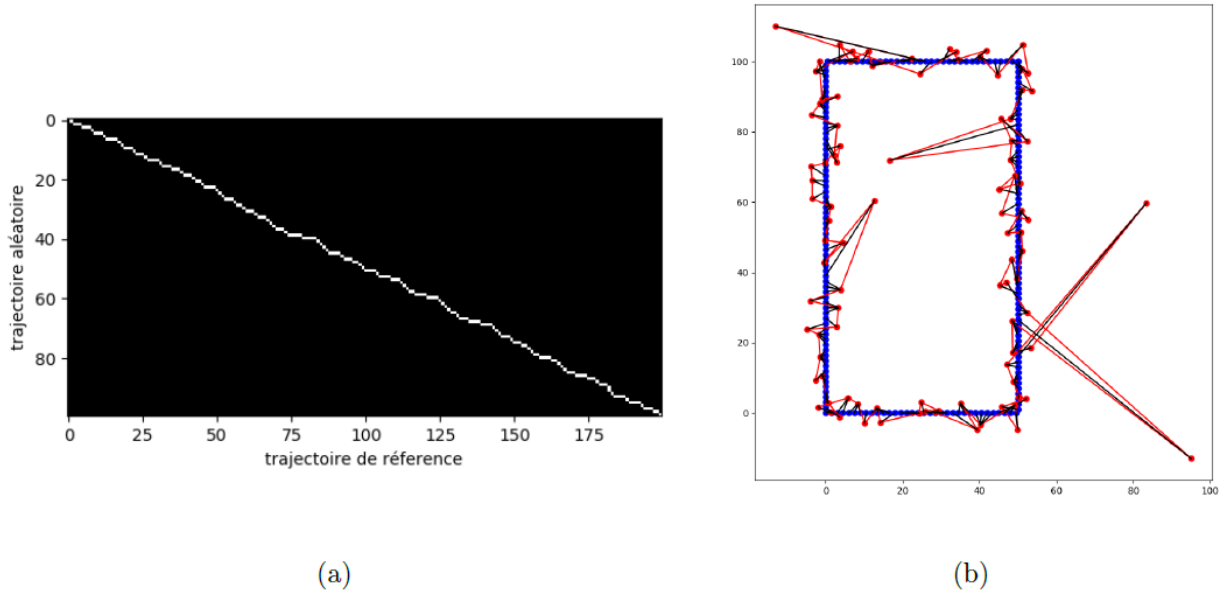


FIGURE 17 – (a) Matrice de correspondance pour la trajectoire réelle et la trajectoire aléatoire et (b) Recalage entre les points des trajectoires de référence (en bleu) et aléatoire avec bruit impulsif (en rouge).

### Trajectoire aléatoire avec bruit gaussien

Maintenant, on ajoute à la trajectoire un bruit gaussien, dont l'écart-type  $\sigma = 3$ . On génère des variables aléatoires vérifiant la loi gaussienne centrée. Pour cela, on utilise :

$$variable = \sigma \sqrt{-2 \ln(U_1)} \cos(2\pi U_2),$$

où  $U_1$  et  $U_2$  sont deux variables aléatoires uniformes sur l'intervalle  $[0,1]$

On réalise à nouveau la synchronisation avec le DTW, la matrice de correspondance et le recalage de points étant représentés sur la figure 18.

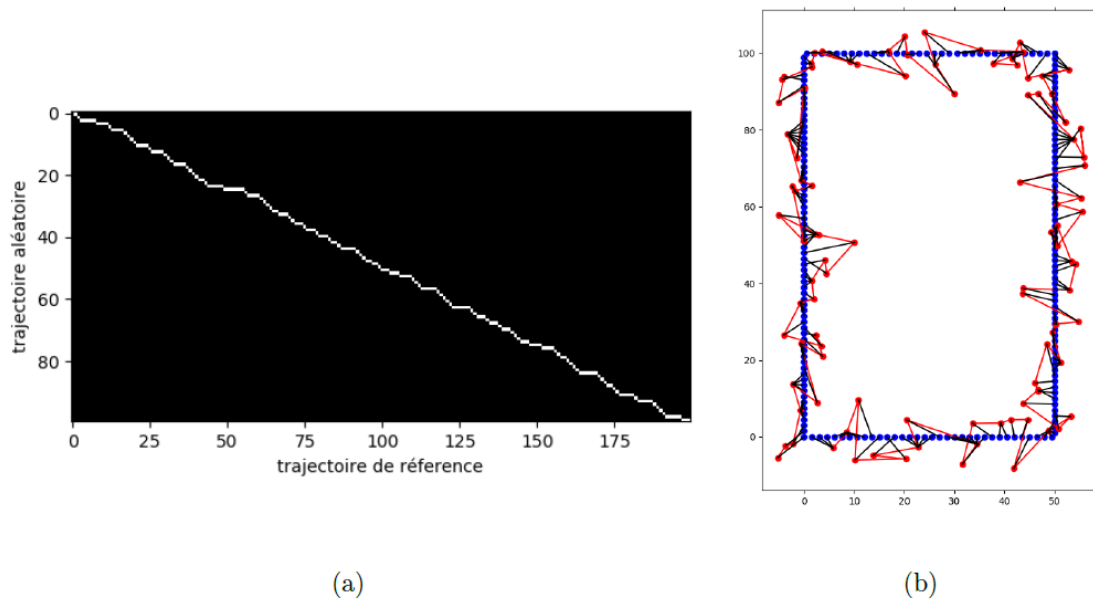


FIGURE 18 – (a) Matrice de correspondance pour la trajectoire réelle et la trajectoire aléatoire et (b) Recalage entre les points des trajectoires de référence (en bleu) et aléatoire avec bruit gaussien (en rouge).

On observe que même avec un bruit (gaussien ou impulsif), les conditions aux limites du DTW permettent de ne pas commettre grands erreurs dans les points qui suivent un point bruité. De plus, la matrice de correspondance de cette trajectoire bruitée tend vers la diagonale de la figure, sans présenter grands écarts.

### Trajectoire aléatoire avec changement d'échelle

Dans cette étape, nous avons décidé de tester l'efficacité de la méthode DTW pour les trajectoires avec des données à différentes échelles. Les trajectoires utilisées dans cette section sont générées de la même manière qu'avant, la seule différence étant que les dimensions de la trajectoire aléatoire ont été multipliées par deux. Deux tests ont été effectués : (1) Le point de départ des deux trajectoires est le même (figure 19) et (2) Les centres des trajectoires sont les mêmes (figure 20).

On observe sur les figures de recalage (figures 19b et 20b) que maintenant, on a beaucoup des points d'une trajectoire qui sont associés à des points non cohérents dans la seconde trajectoire. Cela est très évident dans le cas où l'origine est la même, mais reste toujours visible dans le cas où les centres sont les mêmes. De plus, la matrice de correspondance de cette trajectoire bruitée présente grands écarts. Le DTW est donc sensible à l'échelle des données analysées.

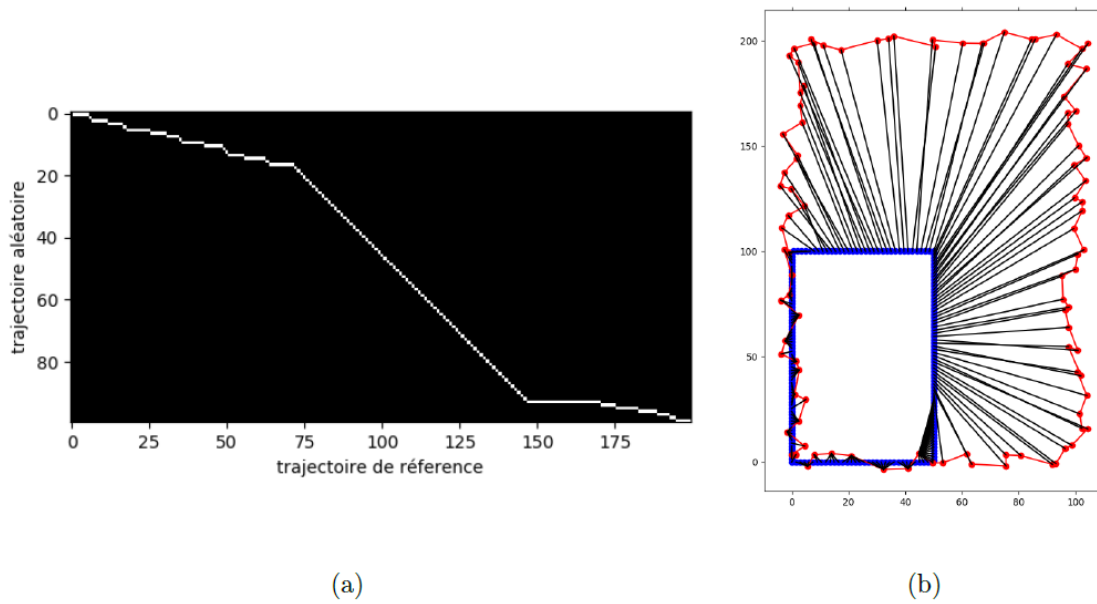


FIGURE 19 – (a) Matrice de correspondance pour la trajectoire réelle et la trajectoire aléatoire et (b) Recalage entre les points des trajectoires de référence (en bleu) et aléatoire avec bruit (en rouge).

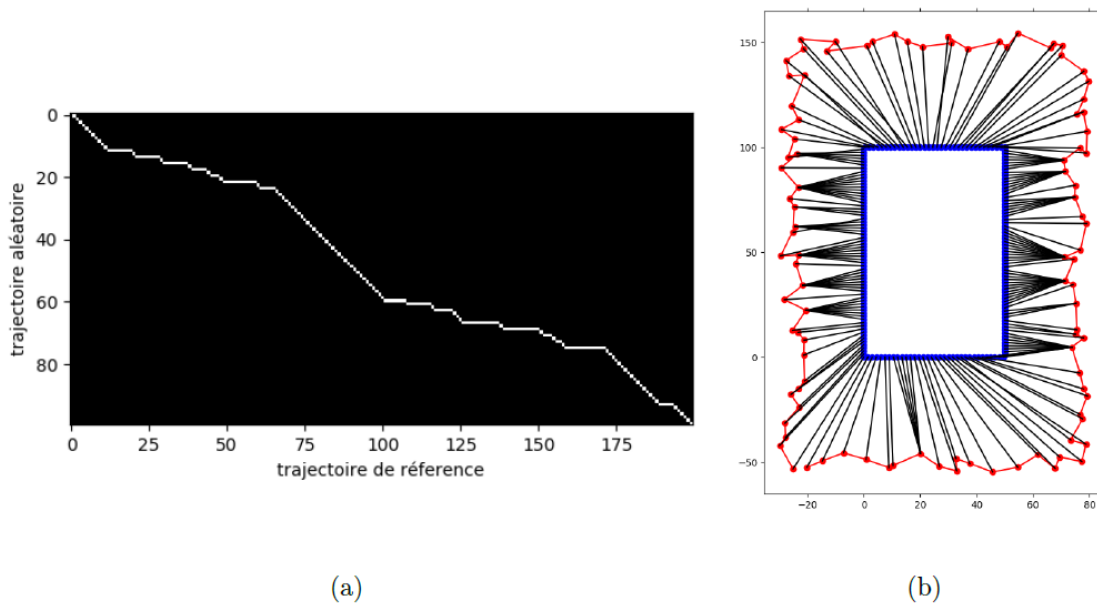


FIGURE 20 – (a) Matrice de correspondance pour la trajectoire réelle et la trajectoire aléatoire et (b) Recalage entre les points des trajectoires de référence (en bleu) et aléatoire avec bruit (en rouge).

### 6.1.2 Simulation de trajectoire polynomial 3D

Pour pouvoir simuler une trajectoire de référence en trois dimensions, on prend comme base une fonction polynomiale pour décrire la position de chaque point de la trajectoire (équation 5).

$$P = (x(t), y(t), z(t)) = (t^5 - \frac{t^3}{2} - 3, \pi k^2, 10) \text{ avec } t \in [0, 1] \quad (5)$$

La simulation d'une trajectoire qui serait acquise avec le Motion Capture est faite en ajoutant une imprécision en chaque point de la trajectoire de référence. Pour pouvoir bien simuler cette imprécision, on fait référence à [8], où des expérimentations ont été menées pour pouvoir établir la précision de capture du système VICON [10] de Motion Capture. On utilise donc la moyenne et l'écart type définis dans la littérature :

$$\mu = 0,3863 \text{ et } \sigma = 0,2474 \text{ (en mm)} \quad (6)$$

On introduit donc à la trajectoire une imprécision qui suit une distribution gaussienne de paramètres  $\mu$  et  $\sigma$ . Les deux trajectoires simulées sont représentées dans la figure 21a et leur alignement est montrée par la figure 21b. On peut observer que pour des trajectoires dont le déplacement dans les trois axes est de l'ordre du mètre, les imprécisions de l'ordre du millimètre sont très faibles, ce qui ne pose pas de problème à l'algorithme DTW pour trouver la matrice de correspondance.

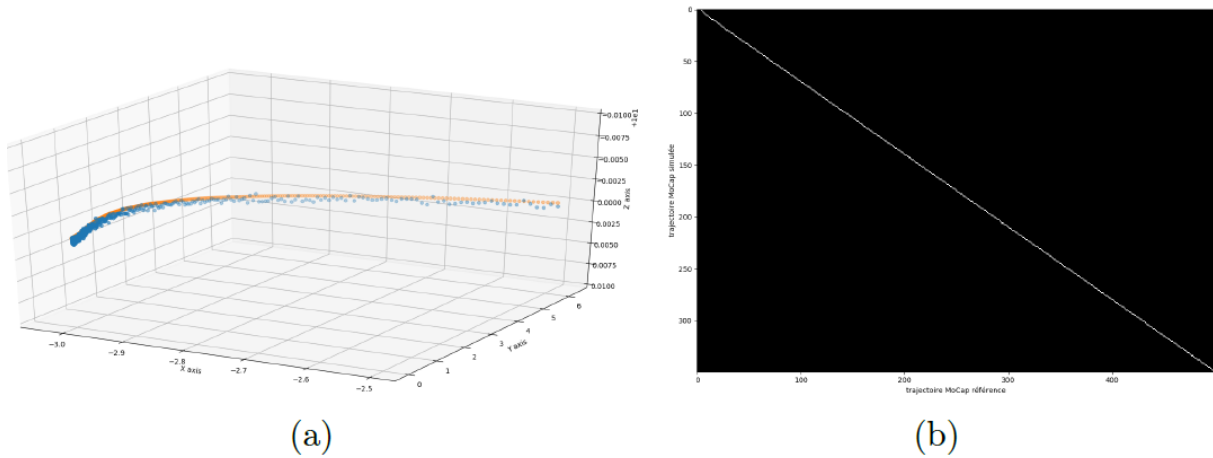


FIGURE 21 – (a) Trajectoire de référence polynomiale (500 points) et trajectoire MoCap simulée (350 points) et (b) Matrice de corrélation pour une trajectoire de référence 3D polynomiale et une trajectoire MoCap 3D simulée.

### Simulation de trajectoires multimodales

#### DTW

On s'intéresse à l'analyse du DTW face à des données multimodales qui décrivent une même trajectoire. Pour tester cela, on reprend la trajectoire polynomiale décrite par l'équation 5 pour simuler le MoCap. On utilise une approche très idéale pour décrire les accélérations : la deuxième dérivée de cette trajectoire polynomiale (équation 7). À l'axe z on utilise la valeur de la pesanteur ( $g = 9,81 \text{ m}^2/\text{s}$ ).

$$P = (x(t), y(t), z(t)) = (20t^2 - 3t; 2\pi; -9,81) \text{ avec } t \in [0, 1] \quad (7)$$

A cette trajectoire on introduit la performance du capteur, à partir des informations fournis par le fabricant du accéléromètre Shimmer3[4] . On ajoute à chaque point de la trajectoire une imprécision qui suit une distribution gaussienne de paramètres (en  $m/s^2$ ) :

- $\mu = -0,06$  and  $\sigma = 0,02$  dans l'axe  $x$
- $\mu = -0,08$  and  $\sigma = 0,02$  dans l'axe  $y$
- $\mu = 0,03$  and  $\sigma = 0,03$  dans l'axe  $z$

Finalement, les deux trajectoires simulant une capture avec le MoCap et avec un accéléromètre sont représentées sur la figure 22. On utilise une différente quantité de points pour chaque trajectoire pour pouvoir simuler une différence de fréquence d'enregistrement des deux capteurs.

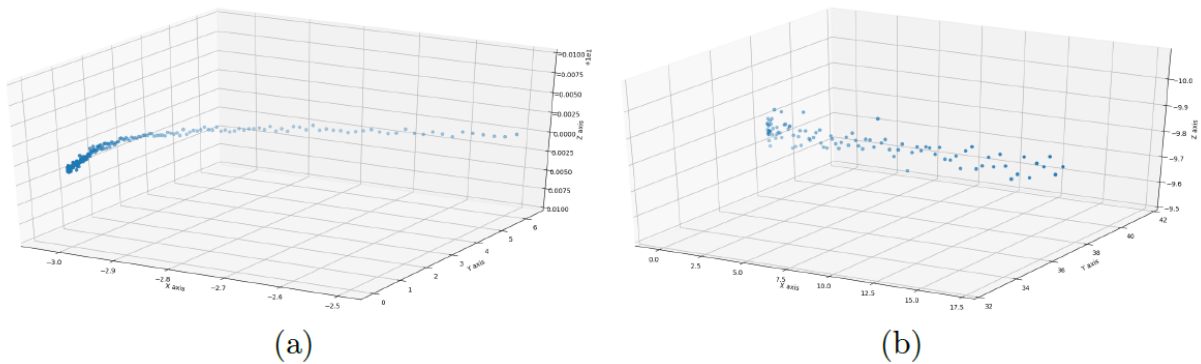


FIGURE 22 – Trajectoires générées pour (a) le MoCap (200 points) et (b) un accéléromètre (100 points).

Pour faciliter la visualisation, on ne s'intéresse qu'aux axes  $x$  et  $y$  dans un premier instant (car les trajectoires de référence sont les deux dans un plan  $z$ ). On utilise l'algorithme du DTW 2D pour réaliser l'alignement de ces deux trajectoires. Les résultats obtenus sont montrés par la figure 23. On observe un plateau très clair dans la matrice de correspondance qui correspond à l'association de 100 points de la trajectoire en rouge à un unique point de la trajectoire bleue. Sachant que nous avons utilisé le même intervalle  $t \in [0, 1]$  pour les deux trajectoires avec une discrétisation différente, on devrait avoir une correspondance d'environ deux points de la trajectoire rouge pour un point de la bleue, ce qui n'est pas du tout le cas. On peut donc conclure que la méthode du Dynamical Time Warping n'est pas adapté à l'utilisation sur données multimodales. Une méthode plus puissante est donc nécessaire pour pouvoir traiter ce type de jeux de données multimodales.

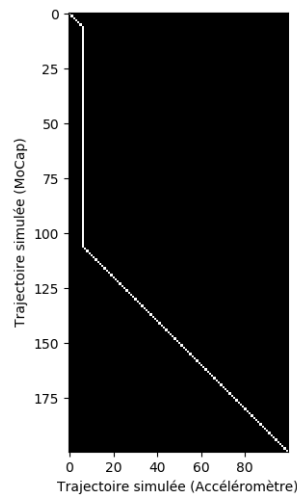


FIGURE 23 – Matrice de corrélation pour une trajectoire 3D polynomiale et une trajectoire MoCap 3D simulée.

## GTW

En utilisant les mêmes trajectoires générées pour le test du DTW, on s'intéresse maintenant à la performance de la méthode GTW face à des données multimodales. La figure 24 montre la synchronisation de l'axe  $x$  des deux trajectoires. On peut noter que le nombre de points à la fin de la synchronisation est supérieur à 200. Ce qui montre qu'il y a des points de la trajectoire du Motion Capture (respectivement de l'accéléromètre) qui ont été synchronisés avec plus d'un point de la trajectoire de l'accéléromètre (respectivement du Motion Capture).

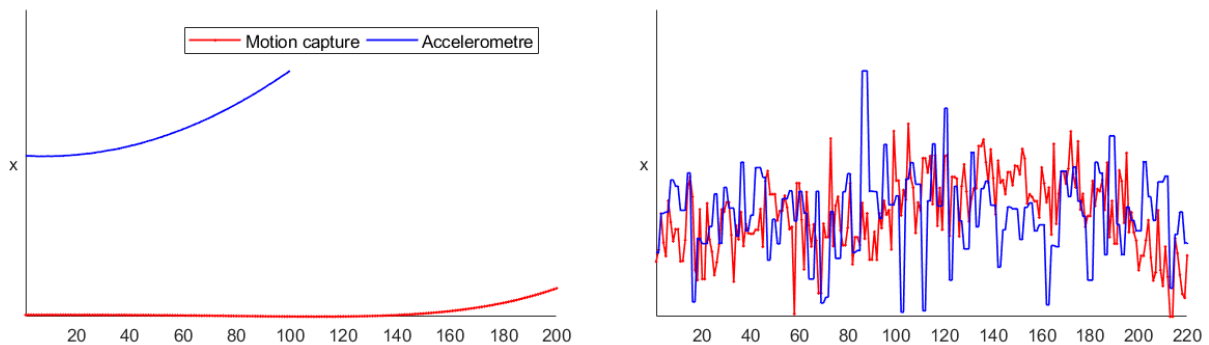


FIGURE 24 – Axe  $x$  des trajectoires générées pour simuler le MoCap et l'accéléromètre avant synchronisation à gauche et le résultat de la synchronisation avec la méthode GTW à droite.

A partir de la matrice de corrélation (figure 25), on peut remarquer que l'association des points des deux trajectoires tend vers la diagonale de la matrice. Ce résultat montre que la synchronisation a été bien réalisée et que la méthode GTW est donc plus adaptée que la méthode DTW dans le cas des trajectoires multimodales.



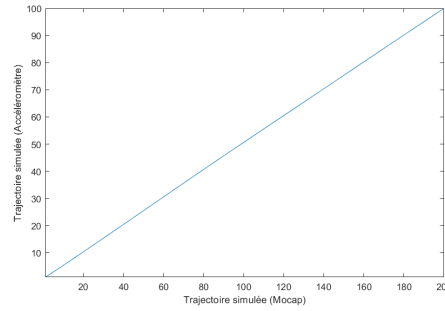


FIGURE 25 – Matrice de corrélation en utilisant la méthode GTW pour les trajectoires générées pour simuler le MoCap et l'accéléromètre.

## 6.2 Données réelles

### DTW pour une trajectoire réelle enregistrée avec le MoCap et une trajectoire de référence simulé.

Avant d'expérimenter les méthodes de synchronisation pour une trajectoire réelle enregistrée par les deux capteurs simultanément, nous avons tenté de synchroniser, avec la méthode DTW, une trajectoire de référence et une trajectoire réelle enregistrée avec MoCap. Pour cela, nous avons effectué l'enregistrement de une trajectoire rectangulaire avec le MoCap et, après la suppression des points que n'appartenait pas à la trajectoire, la trajectoire est représentée en rouge sur la figure 26a. La trajectoire de référence a été créée de façon à avoir le même échelle de la trajectoire du MoCap. Malgré quelques défauts, la matrice de corrélation (figure 26b) tend vers la diagonale de l'image et par conséquent, la synchronisation a été bien faite. Connaissant les limitations du DTW, le succès de la synchronisation par cette méthode montre que la trajectoire capturée par MoCap est proche de la trajectoire attendue et donc, ce capteur présente une excellente précision d'acquisition de données.

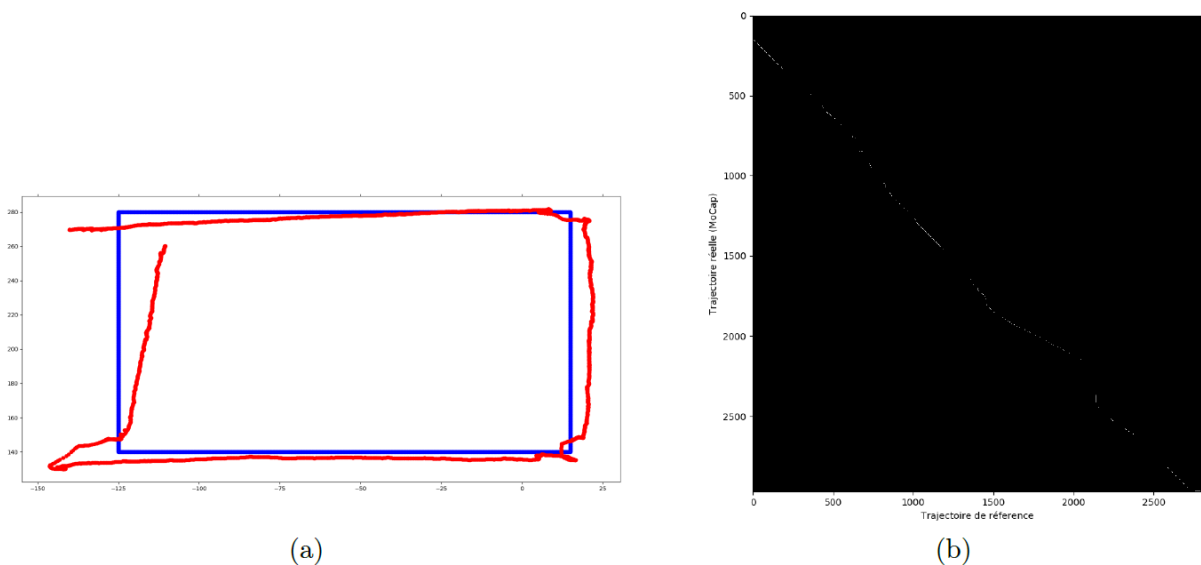


FIGURE 26 – (a) Trajectoire de référence (en bleu) et enregistrement d'une trajectoire réelle avec le MoCap (en rouge) et (b) Matrice de corrélation pour une trajectoire 3D polynomiale et une trajectoire réelle avec MoCap.

### Trajectoire réelle enregistrée par le MoCap et l'accéléromètre simultanément.

Finalement, on s'intéresse maintenant à la synchronisation de données d'une trajectoire réelle captée avec le MoCap et l'accéléromètre simultanément. Pour cette étape, la trajectoire choisie était un cercle dans le plan  $xy$  dans la référence MoCap. La position de cette trajectoire a été capturée par un MoCap en  $m/s$  et son accélération a été capturée par l'accéléromètre en  $m/s^2$ . Après la sélection des données qui se réfèrent réellement à la trajectoire analysée, les données concernant le MoCap (5501 points) et l'accéléromètre (2501 points) sont représentées sur la figure 27a.

#### DTW

En utilisant la méthode DTW pour synchroniser les trajectoires, nous pouvons voir que nous avons 5501 points à la fin de la synchronisation, ce qui implique que seules les données de l'accéléromètre ont été associées à plus d'une donnée du MoCap. De plus, la matrice de corrélation (27b) montre un écart important par rapport à la diagonale de l'image et possède un point dans la séquence de l'accéléromètre associée à plus de la moitié de la séquence de points du MoCap, ce qui ne représente en aucun cas une bonne synchronisation. Ceci est probablement dû à la multi-modalité et à la différence d'échelle des données, qui, comme déjà présenté auparavant, font que la méthode DTW ne fonctionne pas. Il est important de noter que le temps d'exécution de cette méthode pour ces données a été d'environ 32 secondes.

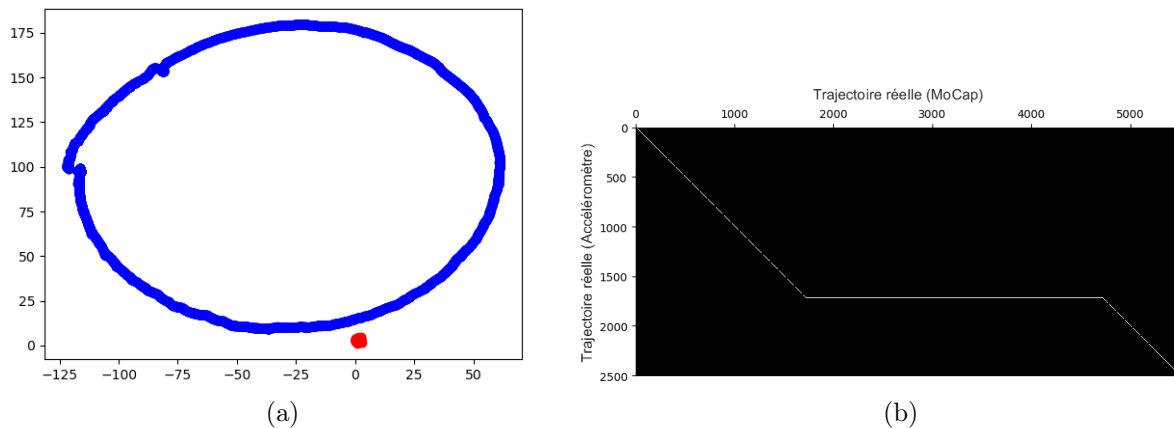


FIGURE 27 – (a) Trajectoire réelle enregistrée avec le MoCap (en blue) et avec l'accéléromètre (en rouge) et (b) Matrice de corrélation pour la méthode DTW pour une trajectoire réelle enregistrée avec le MoCap et avec l'accéléromètre simultanément.

#### GTW

On s'intéresse maintenant à la performance de la méthode GTW face à des données enregistrées pour la trajectoire réelle. La figure 28 montre la synchronisation de l'axe  $x$  des deux séquences de données. On peut noter que le nombre de points à la fin de la synchronisation est 6051, ce qui est supérieur à 5501 (nombre de points du MoCap). Ce qui montre qu'il y a des points de la séquence de données du Motion Capture (respectivement de l'accéléromètre) qui ont été synchronisés avec plus d'un point de la séquence de données de l'accéléromètre (respectivement du Motion Capture).

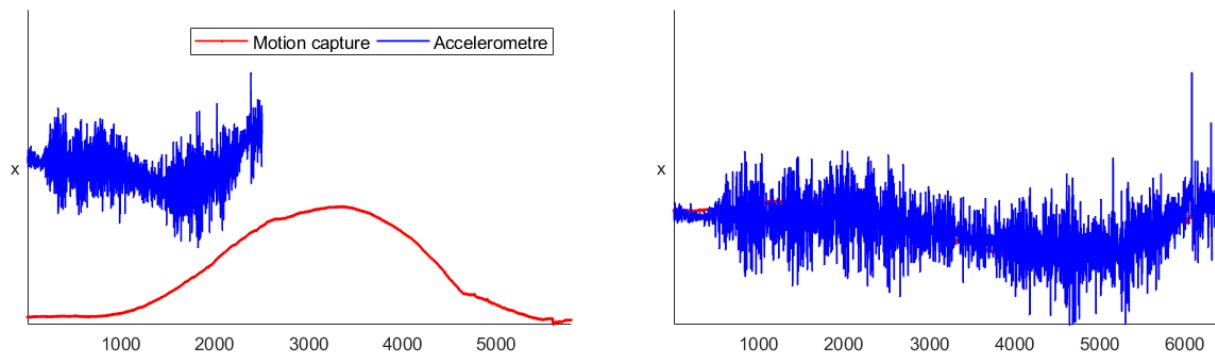


FIGURE 28 – Axe  $x$  de la trajectoire enregistrée avec le MoCap et l'accéléromètre avant synchronisation à gauche et le résultat de la synchronisation avec la méthode GTW à droite.

La matrice de corrélation (figure 29), montre que l'association des points des deux séquences de données tend vers la diagonale de la matrice. Ce résultat montre que la synchronisation a été bien réalisée. De plus, le temps d'exécution de cette méthode a été d'environ 15 secondes, ce qui est bien inférieur au temps d'exécution du DTW (environ 32 secondes). Cela montre qu'en plus d'être fonctionnel pour les séquences multimodales, GTW est aussi plus efficace que le DTW pour des séquences de données plus longues.

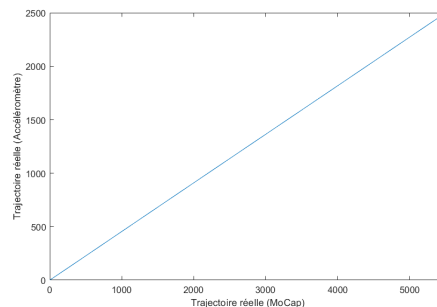


FIGURE 29 – Matrice de corrélation en utilisant la méthode GTW pour les séquences de donnée enregistrées avec le MoCap et l'accéléromètre.

## 7 Interface graphique

### Implémentation

Pour mieux visualiser la synchronisation, nous avons créé un code Python capable d'afficher les données synchronisées. Le code génère une animation des données dans les graphiques et montre quelles données des trajectoires ont été caractérisées comme équivalentes selon la méthode. Le code permet aussi de sauvegarder la animation en vidéo.

Pour que le code soit capable de sauvegarder la vidéo de l'animation, il est nécessaire d'installer FFmpeg [11]. L'animation fonctionne avec la bibliothèque matplotlib et est basée sur le code Jake Vanderplas [12] et sur l'exemple de code fourni par matplotlib [13].

Le principe de l'animation est de créer les graphiques où les données des trajectoires seront affichées. Après la création de la zone d'affichage, l'animation est faite en traçant une partie des données à tout moment. Cela génère le sentiment que les graphiques progressent progressivement. Ceci est fait en retirant le premier donnée et en faisant le prochain donnée devenir la première. De cette façon, on trace le même numéro de données à chaque fois, mais l'ordre de ces données est différente. La première donnée devient maintenant la dernière. Cela rend l'animation infinie. Comme si la trajectoire était répétitive.

La partie enregistrement de la vidéo est similaire. Le programme capture chaque image affichée à la fois et l'enregistre. La différence réside dans le fait que la vidéo ne se répète pas indéfiniment.

### Résultats

Un exemple d'animation peut être vu sur la figure 30. Il se réfère au cas représenté par la figures 16, où nous avons une trajectoire de référence et une trajectoire aléatoire sans bruit. On peut voir que la ligne rouge indique quelles données de la trajectoire aléatoire ont été associées à chaque donnée de la trajectoire de référence. Pour cet exemple les données ont été synchronisées correctement en utilisant le DTW.

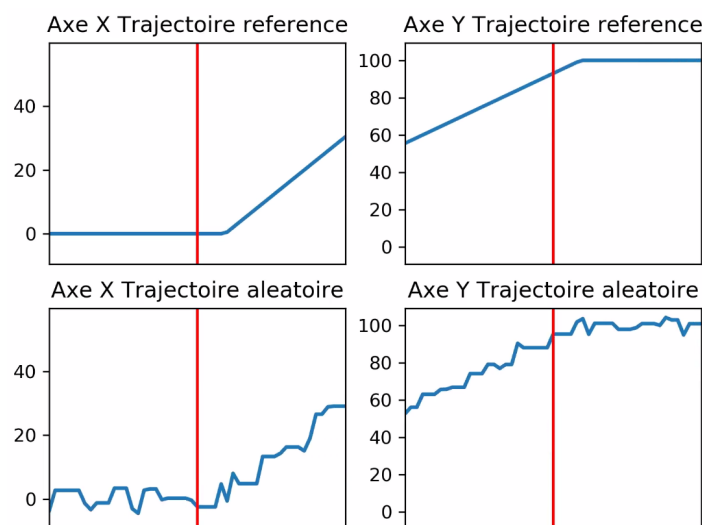


FIGURE 30 – Exemple de vidéo qui montre une bonne synchronisation avec le DTW.

Un deuxième exemple est montré sur la figure 31 pour le cas où nous avons une trajectoire de référence et une trajectoire aléatoire avec un changement d'échelle représenté sur la figure 19. Dans ce cas, nous pouvons voir que, comme prévu, la méthode DTW n'est pas efficace.

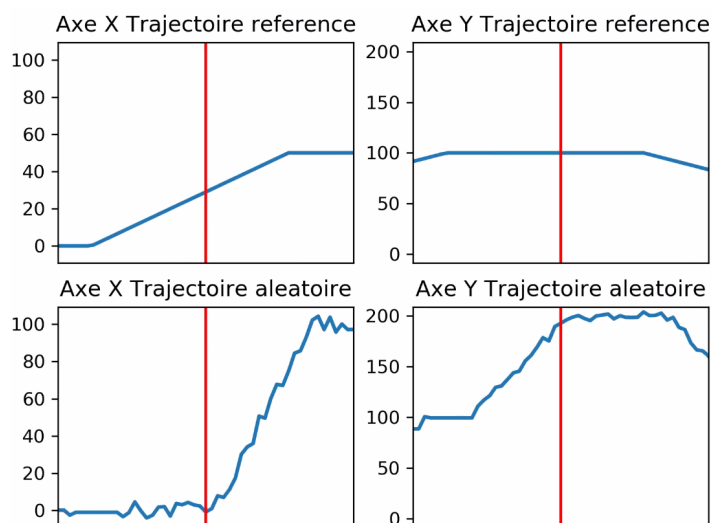


FIGURE 31 – Exemple de vidéo qui montre une mauvaise synchronisation avec le DTW.

## 8 Conclusion et perspectives

Les capteurs de mouvement qui existent aujourd'hui sont nombreux et chacun présente des caractéristiques particulières qui sont plus ou moins adaptées à chaque type de situation. Notre étude portait surtout sur le Motion Capture et les Accéléromètres. Les données enregistrées par ces deux types de capteurs se distinguent principalement par la dimension des données (accélérations ou positions dans l'espace) et par leurs contraintes d'enregistrement. L'analyse portée aux jeux de données visait principalement pouvoir réaliser la synchronisation entre les deux types de données. D'abord, la méthode du Dynamical Time Warping est très simple et se montre utile lorsqu'on travaille sur des données de même dimensions. Cependant, il n'est plus capable de donner des résultats satisfaisants si on a, par exemple, un changement d'échelle. En souhaitant explorer la synchronisation des données multimodales, l'utilisation de la méthode du Generalized Time Warping s'est montrée efficace, même avec une plus grande complexité du code et de la méthode elle-même. Finalement, la synchronisation des données multimodales reste encore peu étudiée par la bibliographie et de nouvelles méthodes doivent encore être développées dans ce domaine dans l'avenir. En ce qui concerne l'analyse de données multimodales, l'utilisation du DTW peut encore être envisagée pour réaliser la classification des trajectoires [14].

## Références

- [1] F. Zhou and F. De la Torre. Generalized Canonical Time Warping. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(2) :279–294, February 2016.
- [2] Feng Zhou and Fernando De la Torre. Canonical Time Warping for Alignment of Human Behavior. In *Advances in Neural Information Processing Systems 22 - Proceedings of the 2009 Conference*, pages 2286–2294, January 2009.
- [3] F. Zhou and F. De la Torre. Generalized time warping for multi-modal alignment of human motion. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1282–1289, June 2012.
- [4] Wearable Sensor Technology | Wireless IMU | ECG | EMG | GSR. Disponible sur : <http://www.shimmersensing.com/>. [Consulté le : 18-janv-2018].
- [5] Shimmer 9dof Calibration Application | IMU Calibration. Disponible sur : <http://www.shimmersensing.com/products/shimmer-9dof-calibration>. [Consulté le : 18-janv-2018].
- [6] ConsensysPRO Software | Collect and analyze biometric and motion data from Shimmer sensors. Disponible sur : <http://www.shimmersensing.com/products/consensys>. [Consulté le : 18-janv-2018].
- [7] Henryk Josiński, Daniel Kostrzewa, Agnieszka Michalczyk, and Adam Świtoński. The Expanded Invasive Weed Optimization Metaheuristic for Solving Continuous and Discrete Optimization Problems. *TheScientificWorldJournal*, 2014 :831691, March 2014.
- [8] Pierre Merriaux, Yohan Dupuis, Rémi Boutteau, Pascal Vasseur, and Xavier Savatier. A Study of Vicon System Positioning Performance. *Sensors (Basel, Switzerland)*, 17(7), July 2017.
- [9] Understanding Signal Calibration, Quality and Fidelity. Disponible sur : <https://www.shimmersensing.com/about/news/understanding-signal-calibration-quality-and-fidelity>. [Consulté le : 09-avril-2018].
- [10] VICON. Motion Capture Systems. Disponible sur : <http://www.vicon.com>. [Consulté le : 09-avril-2018].
- [11] FFmpeg. A complete, cross-platform solution to record, convert and stream audio and video. Disponible sur : <https://www.ffmpeg.org/download.html>. [Consulté le : 12-avril-2018].
- [12] Jake Vanderplas. Matplotlib animation tutorial. Disponible sur : <https://jakevdp.github.io/blog/2012/08/18/matplotlib-animation-tutorial/>. [Consulté le : 12-avril-2018].
- [13] matplotlib. animation example code : moviewriter.py. Disponible sur : <https://matplotlib.org/examples/animation/moviewriter.html>. [Consulté le : 12-avril-2018].
- [14] Martin Wöllmer, Marc Al-Hames, Florian Eyben, Björn Schuller, and Gerhard Rigoll. A multidimensional dynamic time warping algorithm for efficient multi-modal fusion of asynchronous data streams. *Neurocomputing*, 73(1) :366 – 380, 2009.