

# MI12 - Rapport de projet

## Synchronisation de données Android-Kinect

*Bache Cédric*

*Chateau Quentin*

*Wozny Virgile*

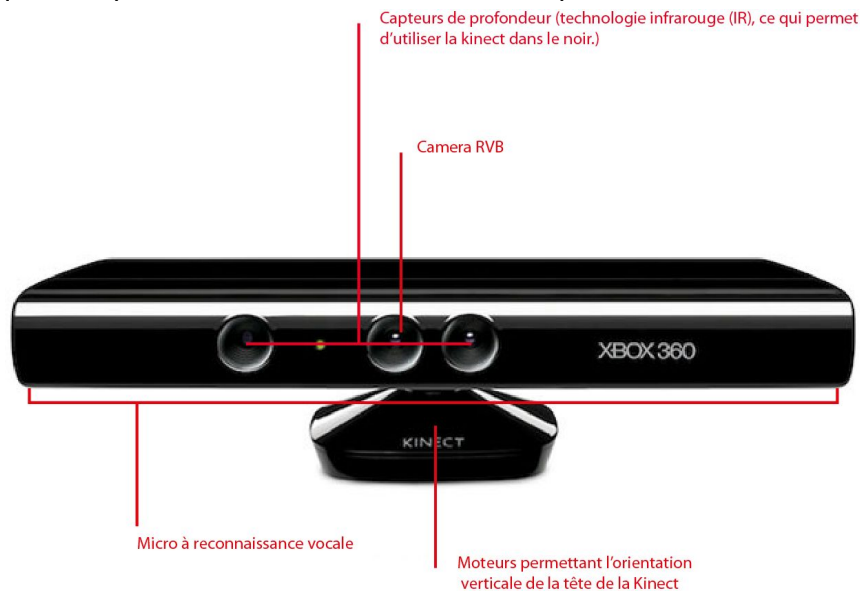
# Sommaire

1. Introduction
2. Synchronisation
3. Application android
4. Application Kinect
5. Programme de mise en correspondance
6. Conclusion

# 1. Introduction

Le but de ce projet est d'aborder les contraintes temps réel dans un projet concret mettant en oeuvre plusieurs capteurs, contenus d'une part dans un téléphone Android, d'autre part dans une Microsoft Kinect.

La Kinect est un capteur RGB-D, c'est à dire un capteur qui permet non seulement d'acquérir des données d'images couleur comme une caméra classique, mais également des données de profondeur d'image. De plus, ce capteur possède une suite logicielle très développée qui nous permet de l'utiliser de manière simple et efficace.



*Capteurs d'une Microsoft Kinect*

Un smartphone Android contient de très nombreux capteurs (accéléromètres, gyroscopes, GPS, etc...), et constitue donc un élément intéressant pour découvrir les contraintes liées à l'acquisition de données.



*Capteurs d'un téléphone Android*

L'objectif de notre projet s'oriente autour de la synchronisation des données entre ces deux équipements. Ainsi, nous avons pour objectif de mettre en place un système permettant de détecter dans quelle main une personne se tenant la Kinect tient le téléphone Android.

Pour réaliser cela, nous allons avoir besoin d'étudier les mouvements des mains détectées par la Kinect et les mouvements détectés par le téléphone. Cette étude nécessite une synchronisation précise des données et une mise en correspondance de celles-ci.

Le système est constitué de quatre programmes :

- Une application Android
- Une application Kinect
- Un serveur de synchronisation
- Un programme de mise en correspondance des données

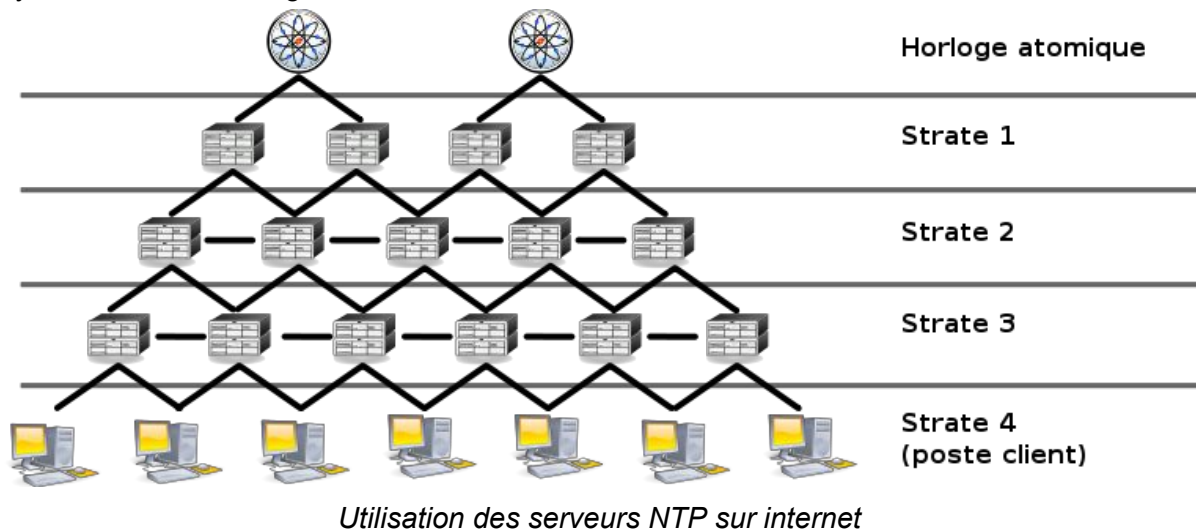
Le serveur de synchronisation, l'application Kinect et le programme de mise en correspondance des données seront exécutés sur un ordinateur portable. L'application Android sera bien entendu exécutée sur un téléphone Android.

Ces deux parties matérielles du système communiqueront par Wi-Fi, en utilisant le protocole classique TCP/IP.

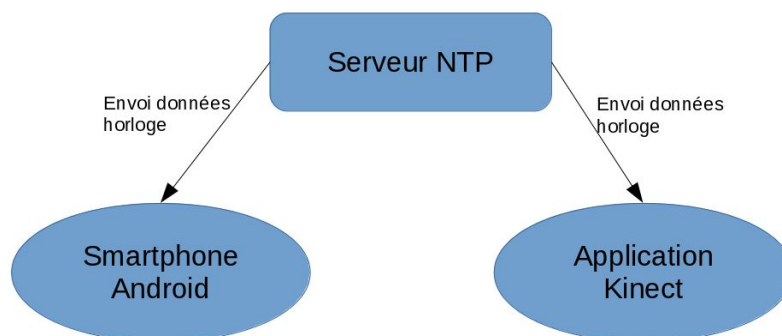
## 2. Synchronisation

Afin de mettre en correspondance les informations des deux programmes d'acquisition de données, il est nécessaire que ces données soient datées. De plus, les dates utilisées par chacun des deux programmes doit être identique. Pour cela nous utilisons un serveur de synchronisation d'horloges, qui permet aux deux programmes d'utiliser la même date.

Le serveur de synchronisation utilisé est un serveur NTP, pour Network Time Protocol. Ce protocole de synchronisation d'horloges est utilisé à très grande échelle sur internet pour synchroniser les horloges de différentes machines.



Son utilisation nous permet d'avoir accès un protocole dédié à notre usage, utilisé depuis 1985 et toujours d'actualité. Il permet d'obtenir une précision de synchronisation de l'ordre de quelques millisecondes.



*Serveur NTP dans notre système*

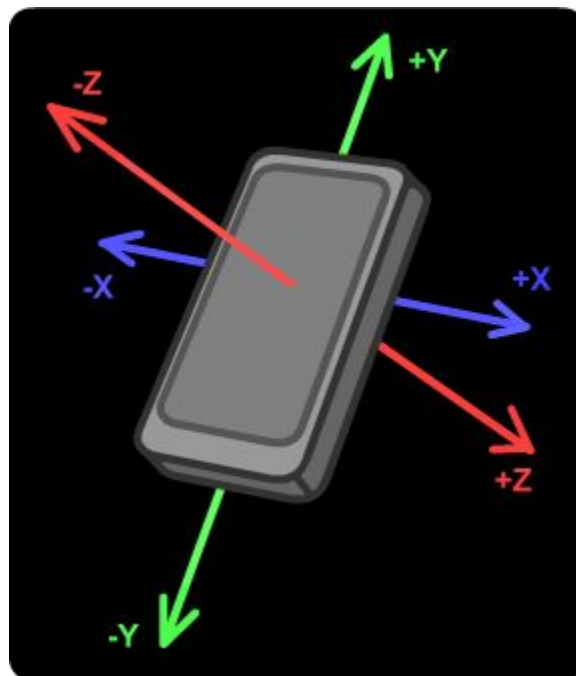
3 Il faut mettre en opposition la précision de nos horloges et la précision nécessaire au bon fonctionnement du systèmes. Quelques millisecondes de décalage peuvent sembler faible mais être tout de même trop élevé dans de nombreuses utilisations.

Notre système possède deux capteurs principaux : la centrale inertielle (accéléromètres et gyroscopes) du smartphone Android et la caméra RGB-D de la Kinect. La centrale inertielle Android renvoi des données à une fréquence de 100Hz, ce qui nous permet d'obtenir des données toutes les 10 millisecondes. La caméra Kinect permet d'obtenir des données à 30Hz, soit toutes les 33 millisecondes.

Nous pouvons donc en déduire qu'un incertitude de quelques millisecondes dans la synchronisation de nos horloges est acceptable puisqu'elle demeure inférieure à la période d'échantillonnage de nos données.

### 3. Application android

L'application Android a pour but d'acquérir les données des accéléromètres et des gyroscopes du téléphone. Avec ces données, nous sommes capables de détecter les mouvements du téléphone. Pour acquérir ces données, nous utilisons la centrale inertielle du smartphone, qui nous permet d'obtenir des données d'accélération et de vitesse de rotation suivant chacun des trois axes du téléphone

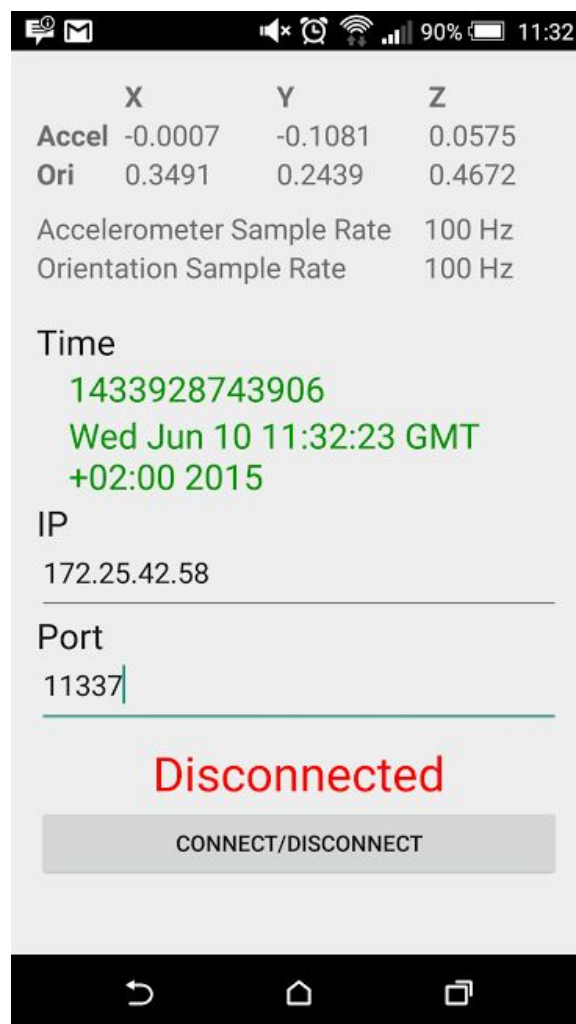


*Axes de mesure d'un smartphone*

Pour acquérir les données, nous utilisons l'API Android, qui nous donne un accès simple aux capteurs et à différentes fonctions de calcul de mouvement. Cependant, les données des capteurs ne sont pas exploitables directement, ces données sont très bruitées et les mesures sont effectuée dans le référentiel du téléphone.

Ainsi, le smartphone attend de recevoir des données de la part des gyroscopes et des accéléromètres du téléphone, effectue une agrégation de ces données pour obtenir un vecteur d'accélération tridimensionnel orienté dans le référentiel terrestre, effectue un filtrage de ces données pour supprimer le bruit intrinsèque aux capteurs et date les données.

L'application peut ensuite envoyer la mesure datée au programme de mise en correspondance des données par le réseau Wi-Fi en utilisant TCP/IP. En effet, nous nous permettons d'utiliser ce protocole de communication dont les délais de délivrance d'un paquet ne sont pas garantis car les données sont datées et pourront donc être exploitée par le programme de mise en correspondance indépendamment de leur date d'arrivée.

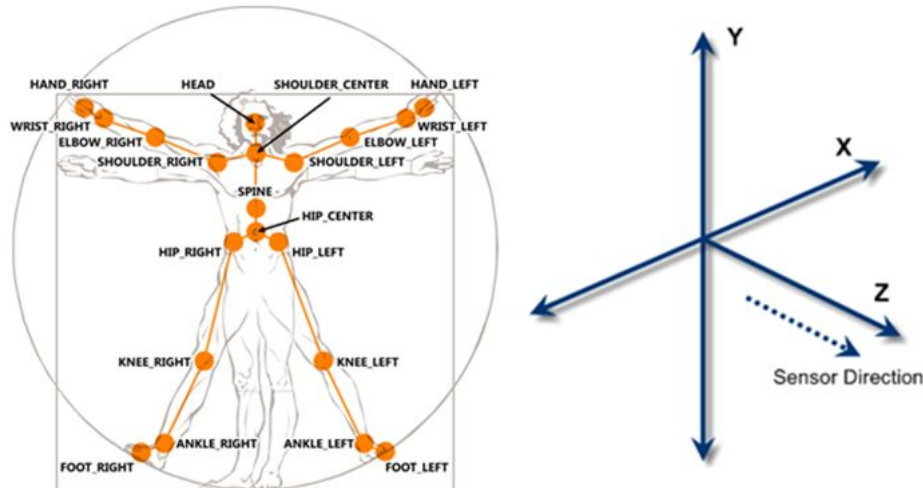


*Capture d'écran de l'application Android*

## 4. Application Kinect

La Kinect est un capteur complexe et de nombreuses utilisations peuvent en être faites dû au caractère très générique de ses informations. Elle permet de reconstituer en trois dimensions la scène vers laquelle la caméra est pointée.

La Kinect s'utilise donc avec l'API fournie par Microsoft qui comprend de très nombreuses fonctions d'exploitation de ces données. Tout particulièrement, la Kinect est très utilisée pour détecter les personnes se tenant face à elle, et permet de reconstituer un squelette.



*Représentation d'un humain sous forme de squelette*

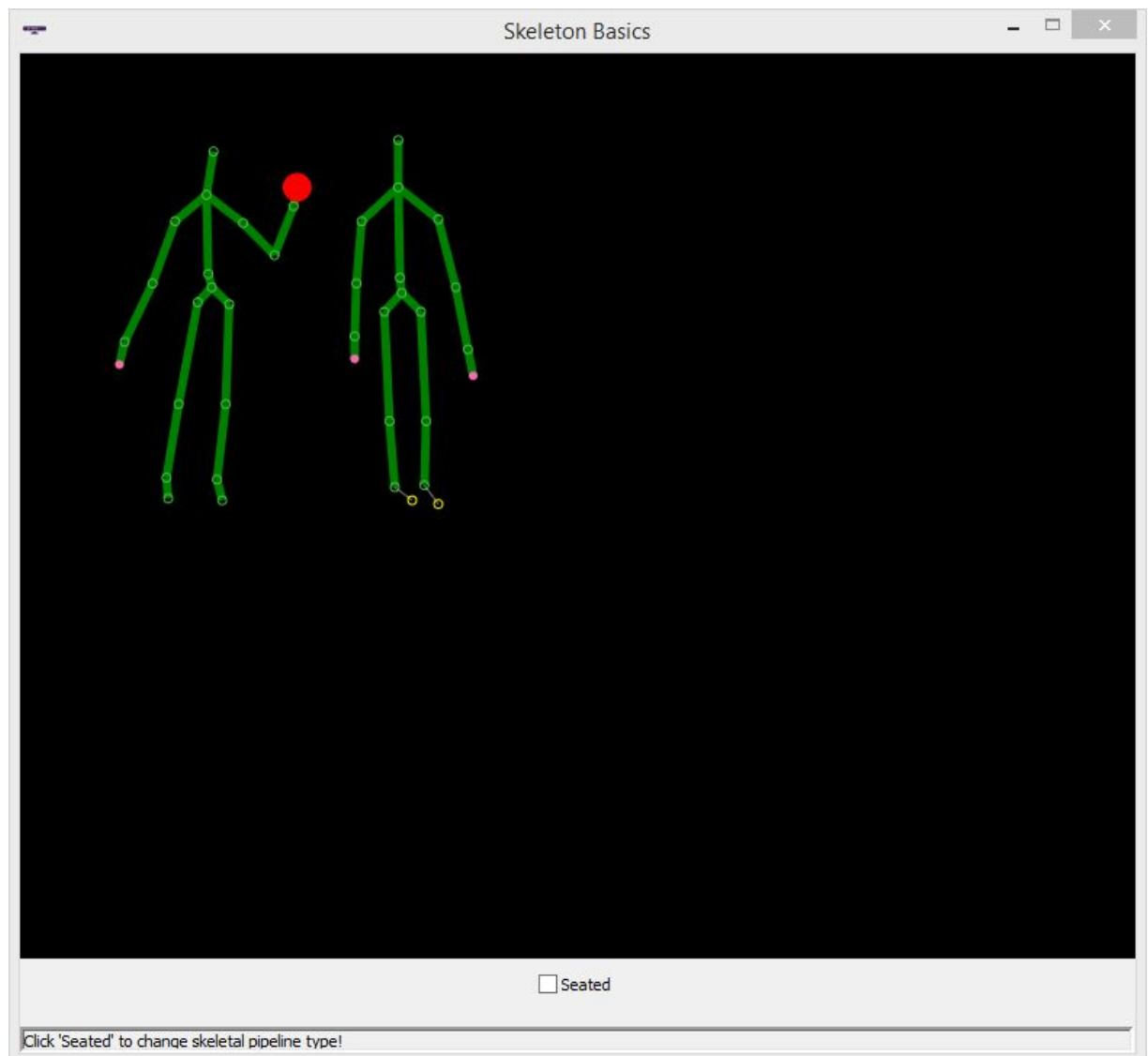
En utilisant l'API Microsoft, nous pouvons donc détecter le porteur du téléphone se tenant face à la Kinect. Une représentation en squelette est tout à fait adaptée pour reconnaître les mains du porteur et mesurer leur position tridimensionnelle.

Nous ne nous intéressons qu'à la position des mains du porteur afin de réduire la quantité de données à traiter. L'application Kinect effectue donc les opérations suivantes :

- Initialisation de la Kinect
- Récupération des données de représentation de la scène
- Détection d'un squelette dans la scène
- Suivi de la position des mains
- Calcul de mouvement des mains
- Datage des données
- Envoi des données au programme de mise en correspondance

De manière identique à l'application Android, l'application Kinect envoie ses données au programme de mise en correspondance par Wi-Fi en utilisant le TCP/IP pour les mêmes raisons.





*Impression d'écran de l'application Kinect*

## 5. Programme de mise en correspondance

Le programme de mise en correspondance des données reçoit les mesures datées de la part du smartphone Android et de l'application Kinect. Cependant ces données ne sont pas sous le même format : la Kinect permet de mesurer la position tridimensionnelle des mains et envoie donc cette position au programme de mise en correspondance. L'application Android permet d'obtenir l'accélération du téléphone dans le repère terrestre.

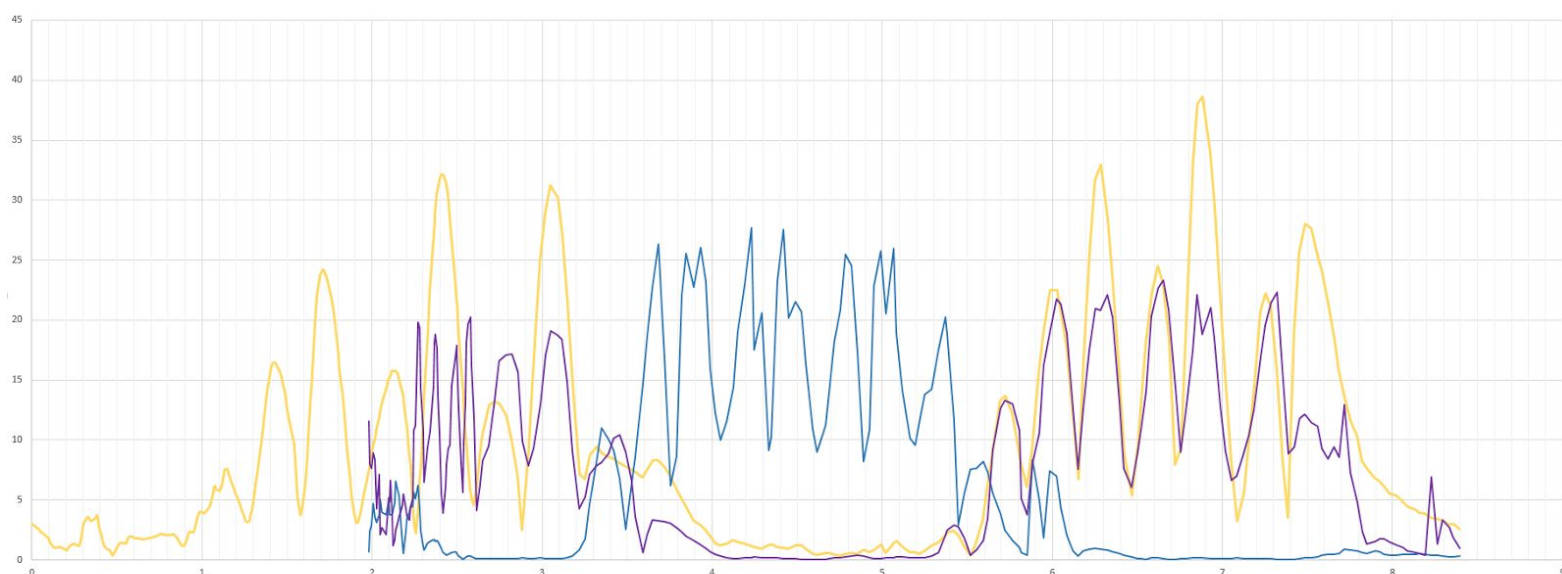
Afin de pouvoir comparer ces deux informations, il est donc nécessaire de les convertir afin qu'elles représentent la même information. Nous avons alors le choix entre dériver la position afin d'obtenir une accélération, ou intégrer l'accélération pour se ramener à une position.

La transformation de l'accélération en position passe par une double intégration. Ainsi l'erreur liée aux incertitudes de mesure devient quadratique par rapport au temps. Une telle opération n'est donc viable qu'à très court terme.

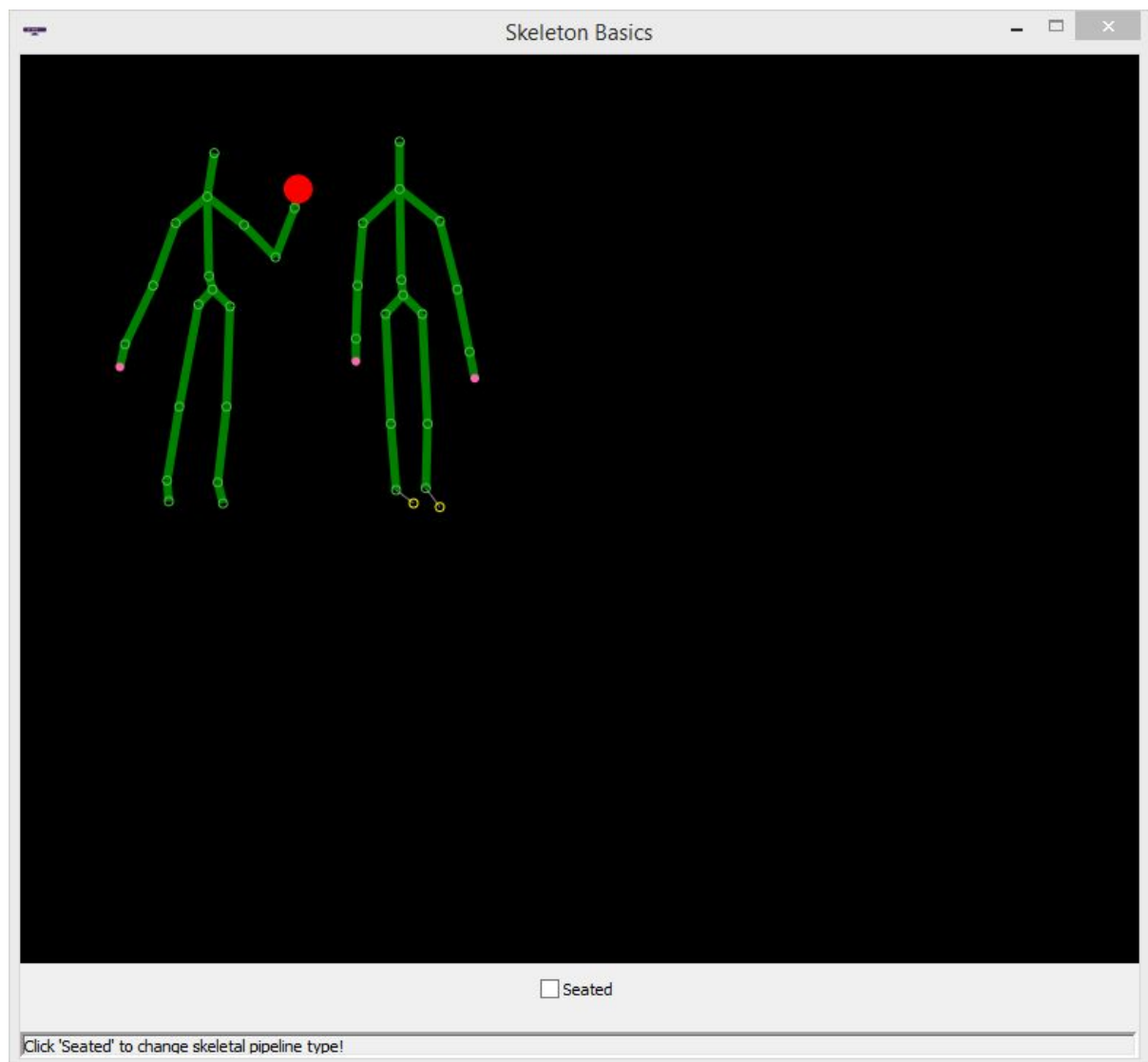
En revanche la transformation de la position en accélération ne possède pas ce problème et l'incertitude de mesure reste constante face au temps. C'est donc la solution que nous avons choisie.

Ainsi, le programme de mise en correspondance commence par dériver deux fois la position reçue par l'application Kinect. Il compare ensuite l'accélération du téléphone aux accélérations des différentes mains détectées dans la scène.

Un algorithme de choix détermine alors dans quelle main le téléphone a la plus de chance de se trouver. Il renvoie alors un message à l'application Kinect qui affiche graphiquement le résultat.



Courbes des accélérations (en  $m/s^2$ ): **Android** **Main1** **Main2**



*Impression d'écran de l'application Kinect avec la détection du téléphone*

## 6. Conclusion

Nous avons pu mettre en oeuvre un système de mesure et d'interprétation constitué de plusieurs éléments matériels et logiciels grâce à ce projet. Ceci fût un bon moyen de nous former aux contraintes imposées par cette déstructuration du système, notamment les contraintes de temps de réponse du système.

Nous avons pu utiliser des matériels et des langages différents pour mener ce projet à bien, ce qui rend sa conception d'autant plus intéressante. De plus, nous avons appris l'importance de la synchronisation et du datage des données dans un système comprenant plusieurs logiciels exécutés sur plusieurs machines.