



---

UNIVERSITÉ PARIS 8 - VINCENNES À SAINT-DENIS

**Master 1 Informatique**

**Mémoire projet tuteuré  
Qualification de caméras RGB-D**

**Yasmine BOUDJEMAÏ  
Mélanie DE JESUS CORREIA**

Date de soutenance : le 10/01/2020

Organisme : Université Paris 8 Vincennes-Saint-Denis  
Tuteur : Farès BELHADJ



# Dédicaces

À toutes les personnes qui nous sont chères.



# Remerciements

Nous souhaiterions adresser nos sincères remerciements à notre tuteur de projet **M.Farès BELHADJ** pour nous avoir accordées sa confiance et de nous avoir accompagnées tout au long de ce projet.

Toute notre gratitude et toute notre reconnaissance vont à l'encontre de nos deux familles respectives pour leur soutien quotidien inestimable.



# Résumé

Dès lors de sa sortie, de nombreux domaines ont connu un essor important à l'aide des technologies RGB-D. À titre d'exemple, on peut citer Natural-Pad<sup>1</sup> qui se sert de la Kinect pour la capture de mouvements pour des besoins de rééducation.

Dans des applications où la précision demeure nécessaire pour de meilleurs résultats, il est indispensable de savoir choisir le capteur qui convient le mieux aux exigences du contexte de l'application. Dans ce document, nous expliquons comment nous comptons réaliser notre application afin de qualifier un ensemble de caméras RGB-D et ce que nous avons pu développer entre-temps. Nous exposons les résultats produits par la première version de notre application, puis discutons de la fiabilité des résultats obtenus.

Dans le premier chapitre, nous proposons un état de l'art des technologies de la caméra RGB-D. Dans le second, nous proposons une maquette de notre application. En outre, nous détaillons les étapes par lesquelles nous sommes passées pour élaborer le prototype de l'outil répondant à la demande du projet tuteuré, ainsi que les résultats obtenus. Dans le troisième, nous montrons comment procéder pour utiliser deux modèles distincts de caméras afin de démarrer les tests avec ces derniers. Enfin, en conclusion, nous exposons succinctement le bilan du travail réalisé, de même que les perspectives envisagées pour parvenir à nos objectifs.

---

1. Entreprise, basée à Montpellier, spécialisée dans la création de jeux vidéo ludiques au service des besoins médicaux.





# Table des matières

<b>Introduction</b>	<b>1</b>
<b>1 État de l’art</b>	<b>3</b>
1.1 Description d’une caméra RGB-D . . . . .	3
1.2 Modèles existants . . . . .	3
1.3 Erreur de mesure . . . . .	4
1.3.1 Types et sources d’erreurs . . . . .	5
1.3.2 Instruments de mesure de distances . . . . .	5
1.3.3 Mesure d’erreurs par une approche théorique . . . . .	7
1.4 Technologies utilisées pour la depth . . . . .	8
1.4.1 Infrarouge . . . . .	8
1.4.2 Stéréo-vision . . . . .	9
1.5 Kinect V2 et Kinect V3 . . . . .	11
1.5.1 Kinect V2 . . . . .	11
1.5.2 Kinect V3 . . . . .	11
<b>2 Modèle logiciel pour la qualification</b>	<b>13</b>
2.1 Description de l’application . . . . .	13
2.2 Application développée . . . . .	14
2.2.1 Première partie : Affichage de la scène captée . . . . .	14
2.2.2 Seconde partie : Interface graphique . . . . .	14
<b>3 Cas pratiques de qualification</b>	<b>19</b>
3.1 Conception du modèle réel et du modèle virtuel . . . . .	19
3.1.1 Conception du modèle réel . . . . .	19
3.1.2 Conception du modèle virtuel . . . . .	19
3.2 Preuve de concept (Proof of concept - POC) . . . . .	20
3.3 Résultats et critique . . . . .	20
3.3.1 Résultats obtenus . . . . .	20
3.3.2 Critiques . . . . .	21

4 Conclusion et Perspectives	23
------------------------------	----

# Table des figures

1.1	Distancemètre laser Leica DISTO D8. . . . .	5
1.2	Odomètre vérifiable PCE-MW 2. . . . .	6
1.3	Distancemètre laser PCE-LRF 600. . . . .	6
1.4	Composants caméra RGB-D (Ici Asus Xtion Pro Live). . . . .	8
1.5	(De gauche à droite) image renvoyée par la caméra RGB, image IR, depth map (une seule caméra), depth map (deux caméras). (figure tirée de l'article de <i>F.Alhwarin, A.Ferrein et I.Scholl</i> ayant comme titre <i>IR Stereo Kinect : Improving Depth Images by Combining StructuredLight with IR Stereo</i> ). . . . .	9
1.6	Schéma illustrant la relation entre la distance focale et le champ de vision (fov). . . . .	10
2.1	Maquette globale de l'application. . . . .	13
2.2	Maquette de l'interface graphique de l'application. . . . .	13
2.3	Interface graphique du programme. . . . .	14
2.4	Partie du rapport généré par le programme. . . . .	17
3.1	Modèle-étalon. . . . .	19
3.2	Modèle 3D. . . . .	20



# Introduction

La caméra RGB-D dans ses débuts, notamment la Kinect, a beaucoup servi le monde du jeux-vidéo. Cependant, depuis un certain nombre d'années, elle contribue activement et fondamentalement dans le développement de multiples applications dans diverses disciplines telles que la médecine, la robotique et plus généralement l'aide à la personne.

Ce projet tuteuré a pour objet la conception d'outils pour la qualification de caméras RGB-D. Il vise à comparer différentes caméras afin de sélectionner la meilleure en fonction du besoin ciblé.

Tout d'abord, avant de passer à l'étape des qualifications, il nous faut concevoir un modèle-étalon sur lequel nous passerons nos tests de mesure. Ce dernier est fait principalement de polystyrène. Ensuite, nous fabriquons sa version virtuelle en 3D en ayant recours au logiciel *123D Design*.

Par la suite, nous exploitons certaines fonctionnalités fournies par OpenCV telles que `matchTemplate` afin de détecter le modèle-étalon.

Enfin, nous opposons les données OpenGL et les données des caméras que nous avons pu réunir à partir de l'aire du modèle détecté en se servant de méthodes différentes pour le calcul du taux d'erreurs tel que le RMSE (Root Mean Square Error)<sup>2</sup>. L'application produit automatiquement un rapport mettant en avant ces divergences.

---

2. Méthode permettant de calculer l'écart-type des erreurs enregistrées. Se référer à cette page web pour davantage de renseignements : <https://www.statisticshowto.datasciencecentral.com/rmse/>



# Chapitre 1

## État de l'art

Dans ce chapitre, nous allons définir ce qu'est une caméra RGB-D, nous énumérons certains des différents modèles existants, nous abordons brièvement les différentes techniques utilisées pour la récupération de la profondeur par une caméra. Enfin, nous discutons les différences recensées sur la **kinect** V2 et V3.

### 1.1 Description d'une caméra RGB-D

La caméra RGB-D, aussi appelée capteur RGB-D, est une caméra fournissant en même temps une image couleur et une carte de profondeur caractérisant la distance des objets vus dans l'image. Cela est rendu possible grâce à un capteur RGB et un capteur de profondeur (D pour Depth). C'est principalement ces captures qui vont nous intéresser tout au long des qualifications.

### 1.2 Modèles existants

Il existe différents modèles de caméras RGB-D. Parmi elles, nous pouvons citer la Kinect et ses différentes versions, Asus Xtion Pro Live, BlasterX Senz3D, Orbbec, Intel RealSens D415, ...

La **Kinect** a fait son apparition en septembre 2008. Elle a été conçue par Microsoft et était destinée pour la console de jeu XBox 360. Elle permettait aux utilisateurs d'interagir avec la console à l'aide d'une NUI<sup>1</sup> en utilisant les mouvements gestuels et une reconnaissance vocale. Elle sera plus tard

---

1. Natural User Interface (Interface Utilisateur Naturelle), se réfère à une interface utilisateur invisible.

utilisée dans les domaines de la recherche et du développement pour différents secteurs comme le domaine de la médecine, l'industrie automobile, la robotique, l'éducation, ....

**Asus Xtion Pro Live** est le modèle de référence que nous utilisons afin d'effectuer les qualifications. Elle utilise la technologie PrimeSense<sup>2</sup> pour la détection de mouvements.

**BlasterX Senz3D** a fait son apparition en Septembre 2016. Conçue par Creative, elle a été présentée comme une webcam intelligente basée sur ce qu'il y a de mieux en matière de savoir-faire et de technologie. Elle possède trois lentilles pour capturer les données visuelles : une caméra RGB, une caméra infra-rouge et un projecteur laser. Ces dernières collaborent avec la technologie Intel RealSense pour réagir aux expressions faciales et aux gestes corporels des utilisateurs.

**Orbbec Astra Pro** fait partie de la série Astra. Elle offre une vision par ordinateur qui permet des dizaines de fonctions telles que la reconnaissance des visages, la reconnaissance des gestes, le suivi du corps humain, la mesure tridimensionnelle, la perception de l'environnement et la reconstruction de cartes en trois dimensions. De plus, elle offre une réactivité haut de gamme, une mesure de la profondeur, des dégradés fluides et des contours précis ainsi que la possibilité de filtrer les pixels de profondeur de faible qualité.

**Intel RealSense D415** a un champ de vision standard bien adapté aux applications de haute précision telle que la numérisation 3D. Elle comprend le processeur Intel RealSense Vision D4 offrant une résolution en profondeur élevée, des capacités de longue portée, une technologie d'obturation globale et un large champ de vision. Grâce à ces deux dernières, la caméra offre une perception précise de la profondeur lorsque l'objet est en mouvement ou que l'appareil est en marche. De plus, elle couvre un champ de vision plus large, minimisant ainsi les angles morts.

### 1.3 Erreur de mesure

L'erreur de mesure, autrefois appelée l'erreur absolue, est un processus qui permet d'évaluer l'écart entre la valeur mesurée et la valeur de référence qui est soit exacte ou connue. L'objectif de la mesure d'erreurs est de jauger à quel degré les deux valeurs sont proches. Dans cette section, nous citons les types ainsi que quelques sources d'erreurs, ensuite, nous faisons un détour sur

---

2. Connu principalement pour sa licence de conception matérielle et de puce employée dans le mécanisme de détection de mouvements de la Kinect XBox360. Pour plus d'informations, le lecteur peut se référer à ce lien <https://www.crunchbase.com/organization/primesense#section-web-traffic-by-similarweb>.



quelques instruments de mesure de distances ainsi que les moyens théoriques employés à cet effet.

### 1.3.1 Types et sources d'erreurs

Il existe deux types d'erreurs : aléatoires et systématiques.

Les erreurs aléatoires sont des erreurs dont les valeurs sont incohérentes et imprévisibles même en répétant les observations et en conservant les mêmes paramètres.

Les erreurs dites systématiques, à l'inverse des erreurs aléatoires, sont des erreurs reproductibles, elles demeurent constantes tant que les paramètres restent inchangés.

Ces erreurs peuvent résulter de différentes sources ; elles peuvent être dues à des facteurs environnementaux, à la résolution de l'instrument utilisé, au calibrage de l'appareil employé ou même à des erreurs humaines.

### 1.3.2 Instruments de mesure de distances

Diverses instruments de mesures de distance sont mis à porté des professionnels tels que les ingénieurs en génie civil, les architectes, les métreurs, les dessinateurs, les topographes, les menuisiers, les maçons, les agents immobiliers, les électriciens, les décorateurs, les experts. Nous nommons certains d'entre eux ci-après.

**Distancemètre laser Leica DISTO D8** : cet appareil est pourvu d'un localisateur numérique, d'un écran couleur haute résolution 2,4", d'un capteur d'inclinaison 360° et de la technologie Bluetooth.



FIGURE 1.1 – Distancemètre laser Leica DISTO D8.

Parmi ses caractéristiques techniques, on peut citer :

- Une portée allant de 0.05m jusqu'à 200m.
- Une précision de  $\pm 1mm$ .
- Des opérations arithmétiques ainsi que des fonctions mathématiques sont intégrées à l'instrument telles que la fonction Pythagore, la fonction Trapèze ou encore la fonction triangle.
- Image en temps réel
- Un Plug-In d'AutoCAD embarqué dans l'appareil qui permet de dessiner et de planifier avec le logiciel d'AutoCAD.
- Il inclut un logiciel de transfert de données pour le distancemètre **Leica Disto Transfert**.
- Avec la technologie Bluetooth intégrée, l'utilisateur peut transférer les mesures effectuées vers un ordinateur portable ou un PC sans avoir recours aux câbles. Les données ainsi transférées peuvent être éditées à l'aide de logiciels tels que Word, Excel, AutoCAD, ....

**Odomètre vérifiable PCE-MW 2** : Instrument permettant de mesurer différentes zones (ie : chantiers, zones industrielles, zones sportives, ...), des câblages, des tuyauteries, ....



FIGURE 1.2 – Odomètre vérifiable PCE-MW 2.

Parmi ses caractéristiques, on peut mentionner :

- Un poids de 2200g
- Mesure pouvant aller jusqu'à 99999,9 mètres.
- Permet d'effectuer une addition en se déplaçant vers l'avant et une soustraction en bougeant vers l'arrière.
- Une précision de  $\pm 1\%$
- La roue est en plastique et permet de résister aux huiles.

**Distancemètre laser PCE-LRF 600** : Le design de cet instrument est bien conçu puisqu'il est léger et il permet de le maintenir avec une seule main. De plus, il est facile d'utilisation.



FIGURE 1.3 – Distancemètre laser PCE-LRF 600.

De ses caractéristiques, on peut évoquer :

- Un poids de 165g.
- Une portée allant de 15m à 600m.
- Une précision de  $\pm 1m / \pm 0.1\%$ .
- Un laser de type classe 1.
- Une résistance aux éclaboussures d'eau
- Une mesure des distances dans diverses situations météorologiques.

### 1.3.3 Mesure d'erreurs par une approche théorique

À ce jour, on recense de multiples méthodes exploitées à cet effet. Nous nous contentons de citer ci-dessous celles qui nous paraissent être les plus pertinentes.

**RMSE/RMSD** : Root Mean Square Error ou encore Root Mean Square Deviation que l'on pourrait traduire par ; la racine carrée de la moyenne des erreurs au carré. Elle demeure parmi les plus fréquentes en la matière. C'est une mesure de l'exactitude, qui permet de comparer les erreurs de prévision de différents modèles pour un ensemble de données précis et non entre des ensembles de données. Sa formule s'exprime comme suit :

$$\sqrt{\frac{1}{N} \cdot \sum_{i=1}^N (x_i - \hat{x}_i)^2} \quad (1.1)$$

De par la formule, on peut déduire que le résultat sera toujours positif et qu'obtenir une valeur de 0 (qui n'est généralement non atteint en

pratique), témoignerait d'une exactitude parfaite. Ainsi, obtenir des valeurs avoisinant le zéro est positif pour les mesures.

**MSE** : Mean Square Error que l'on pourrait traduire par ; l'erreur quadratique moyenne. Sa formule est comme suit :

$$\frac{1}{N} \cdot \sum_{i=1}^N (x_i - \hat{x}_i)^2 \quad (1.2)$$

**MAE** : Mean Absolute Error qui pourrait être traduit par ; erreur moyenne absolue. Son équation est comme ci-dessous :

$$\frac{1}{N} \cdot \sum_{i=1}^N |x_i - \hat{x}_i| \quad (1.3)$$

**MAPE** : Mean Absolute Percentage Error qui pourrait être traduit par ; le pourcentage de l'erreur moyenne absolue. Elle fait partie également des moyens de mesure d'erreurs les plus fréquents. A noter que la valeur observée ne doit être aucunement nulle. La formule est comme ci-après :

$$\frac{1}{N} \cdot \sum_{i=1}^N \left| \frac{x_i - \hat{x}_i}{x_i} \right| \quad (1.4)$$

Avec :

$x_i$ , la valeur observée à la ième observation.

$\hat{x}_i$ , la valeur exacte à la ième observation.

$N$ , le nombre d'observations.

## 1.4 Technologies utilisées pour la depth d'une caméra RGB-D

Des méthodes employées pour la récupération de la depth, on peut en citer deux principales.

### 1.4.1 Infrarouge

Le projecteur infrarouge de certaines caméras le possédant qu'on peut voir sur la figure 1.4 projette un spectre infrarouge sur la scène captée. Le motif produit sur cette dernière sera capté par la caméra infrarouge et sera

par la suite comparé à la base de données de motifs de référence stockés au préalable dans la caméra. Ces derniers seront indispensables pour la mesure de la profondeur de chaque pixel.

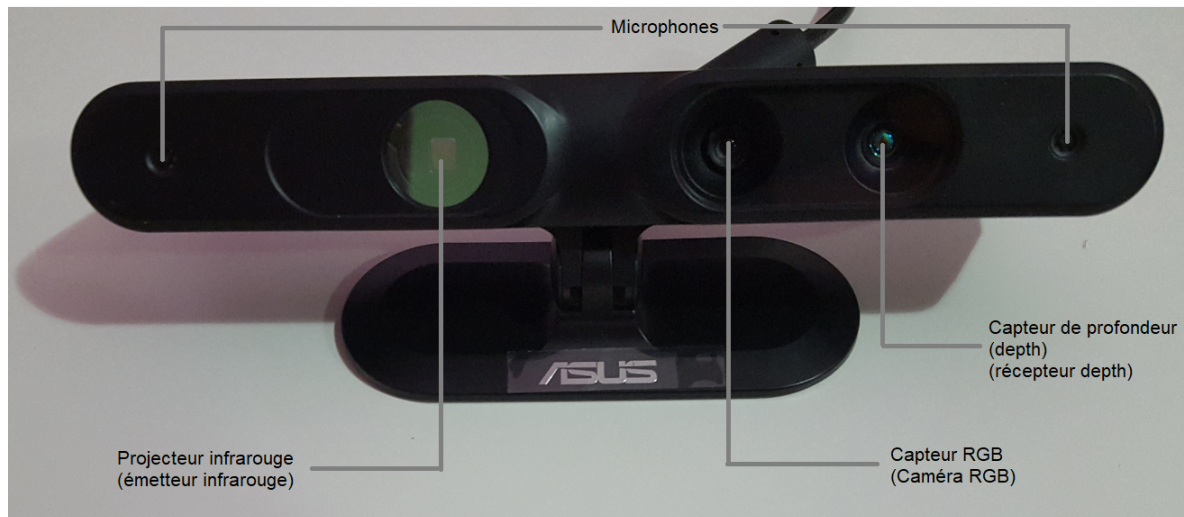


FIGURE 1.4 – Composants caméra RGB-D (Ici Asus Xtion Pro Live).

Par la suite, les valeurs obtenues seront corrélées à un capteur RGB qu'on peut apercevoir sur la figure. Ces données pourront être représentées par un nuage de points<sup>3</sup>.

Toutefois, ce type de caméras possède certaines restrictions. Parmi ces dernières, on peut citer :

- Distances de mesure limitées.
- Problèmes de calculs des informations de profondeur à l'encontre de surfaces brillantes, très mates, transparentes, réfléchissantes ou encore envers des objets absorbants.
- Interférence des motifs (patterns) infrarouges si présence de plusieurs caméras RGB-D de même type. En effet, chaque capteur visualisera ses propres motifs ainsi que ceux des autres caméras présentes et ne saura distinguer les siens des autres qui se chevauchent. De cela découle une perte considérable d'informations de profondeur comme nous pouvons l'observer sur la figure 1.5.

---

3. Est une représentation des points de coordonnées tridimensionnelles dont chacun peut avoir des attributs qui lui sont propres.



FIGURE 1.5 – (De gauche à droite) image renvoyée par la caméra RGB, image IR, depth map (une seule caméra), depth map (deux caméras). (figure tirée de l'article de *F.Alhwarin, A.Ferrein et I.Scholl* ayant comme titre *IR Stereo Kinect : Improving Depth Images by Combining StructuredLight with IR Stereo*).

Cependant, des travaux de recherches ont été conduits afin d'y remédier. Parmi eux, on peut citer le travail réalisé par l'équipe de *Rafibakhsh* ([?]) qui recommande de laisser un angle de  $35^\circ$  entre deux caméras suspendues à la même hauteur en considérant les scènes captées dans de bonnes conditions et une interférence très faible. *Maimone* et *Fuchs* [?] proposent un algorithme de remplissage et de lissage en modifiant le filtre médian aux zones trouées à l'exception des bords. Quant à *F. Kenton Musgrave, Craig E.* et *Robert S. Mace* [?], ils appliquent une certaine quantité minimale de mouvements (en utilisant des composants matériels supplémentaires) à certains capteurs de sorte que chacun puisse voir son propre motif infrarouge de façon nette et une version floue des motifs de ses voisins.

### 1.4.2 Stéréo-vision

Une caméra stéréoscopique est un appareil qui contient deux voire plus de capteurs d'images. Ceci nous rappelle la vision binoculaire humaine. En effet, ce mécanisme permet au système nerveux central de percevoir simultanément les images issues de chaque œil envoyées sous forme de signaux. Ainsi, il sera en mesure de se servir de ces différences (entre les deux images) pour permettre une vision stéréoscopique pour la perception de relief et une mesure des distances en utilisant la triangulation<sup>4</sup>. Le concept que nous venons d'expliquer est appelé disparité stéréoscopique<sup>5</sup>.

4. Approche géométrique permettant une mesure des distances. Le lecteur peut consulter cette page web pour de plus amples informations : <https://fr.wikipedia.org/wiki/Triangulation>.

5. Différence dans la localisation d'un objet perçu par l'œil gauche et l'œil droit résultant de la séparation horizontale des yeux dont certaines caméras essaient de s'approprier la technique afin de récupérer la profondeur.

La stéréo-vision sert principalement à reconstituer la scène observée sous forme de modèle 3D.

### Carte de disparités

Le principe de la disparité est une approche du mécanisme humain vu précédemment. Il consiste en la différence entre les coordonnées pixels d'un point bidimensionnel d'une image et celles de son correspondant (présent sur une autre image prise au même moment). Ainsi, en appliquant le même traitement sur tous les pixels correspondants, on obtient la carte des disparités. Une carte est dite éparsée lorsqu'une disparité est associée à quelques pixels et lorsque cette dernière est associée à chaque pixel, on dit d'elle qu'elle est dense.

Une formule pour calculer la profondeur en fonction de la disparité s'obtient comme suit :

$$z = \frac{B \cdot f}{d} \quad (1.5)$$

Avec :

$z$  représentant la profondeur.

$B$  *Baseline* représente la distance séparant les deux capteurs.

$f$  La distance focale en pixels calculée comme montré ci-après en nous servant de la figure 1.6 .

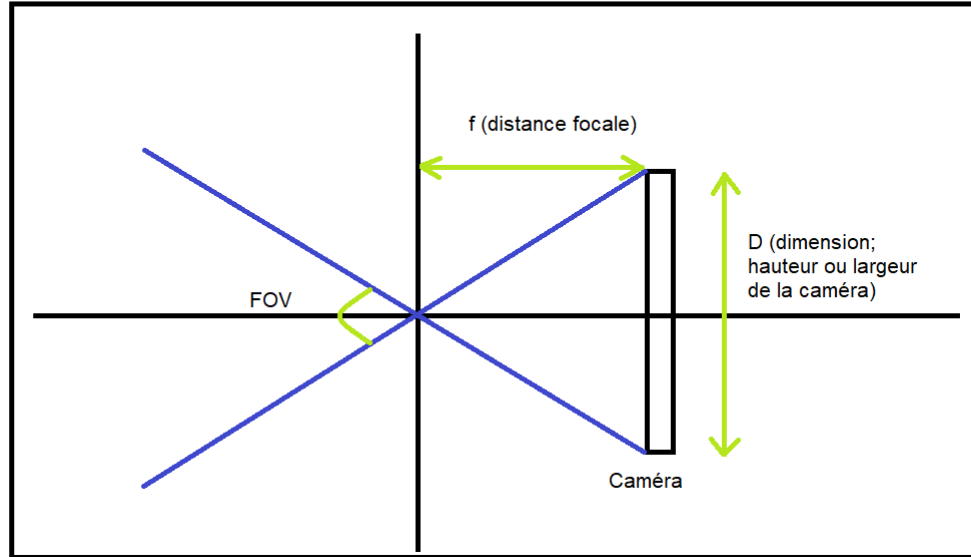


FIGURE 1.6 – Schéma illustrant la relation entre la distance focale et le champ de vision (fov).

$$\tan\left(\frac{fov}{2}\right) = \frac{D}{2.f} \Leftrightarrow f = \frac{D}{2 \cdot \tan\left(\frac{fov}{2}\right)} \quad (1.6)$$

Avec :

$D$  dimension qui peut représenter soit la largeur soit la hauteur en fonction du fov de la caméra employé.

$fov$  (Field Of View) représente l'angle du champ de vision (soit horizontal ou vertical).

$f$  représente la distance focale.



## 1.5 Kinect V2 et Kinect V3

### 1.5.1 Kinect V2

**Kinect V2** se sert de la méthode TOF (ie : Time Of Flight ou autrement dit Temps De Vol) pour générer la carte de profondeur. Cette technique se base sur la différence de temps entre l'émission d'un faisceau lumineux et son retour après réflexion sur un objet.

La distance est calculée comme suit :

$d = c \cdot \frac{\Delta t}{2}$ , avec  $c$  la vitesse de la lumière dans l'air.

Elle permet une bien meilleure précision même dans le noir que sa version précédente (**Kinect V1**).

### 1.5.2 Kinect V3

**Kinect V3**, baptisée **Microsoft Azure Kinect V3**, tout comme la précédente version, se base aussi sur la technologie TOF. Elle comprend un capteur RGB de 12 Mp, un capteur de depth de 1Mp avec un fov réglable en large ou réduit ainsi que 7 microphones intégrés.

### **Kinect V2 Vs Kinect V3**

**Kinect V3** est beaucoup plus légère et plus petite que la **V2**. Elle possède entre autre plus de microphones que la précédente. Elle a été conçue afin d'être principalement utilisée avec **Azure** , le service cloud de **Microsoft**. C'est sans doute ce qui la démarque le plus de sa prédécesseure. En effet, ce service lui permet d'effectuer une partie des calculs. En outre, elle bénéficie des **Cognitive Services**, autrement dit de l'intelligence artificielle pourra être incluse dans les applications créées.



# Chapitre 2

## Modèle logiciel pour la qualification de caméra RGB-D

Dans ce passage, nous allons décrire notre outil que nous avons pu concevoir pour qualifier les caméras.

### 2.1 Description de l'application

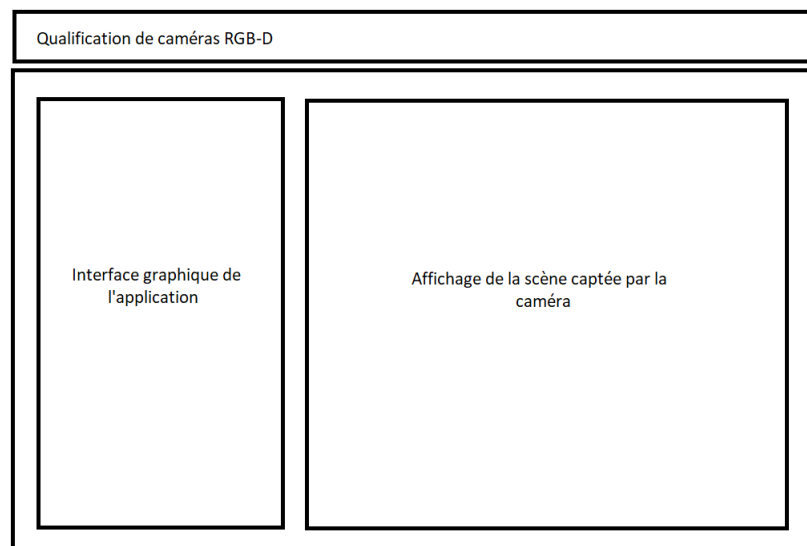


FIGURE 2.1 – Maquette globale de l'application.

La figure 2.1 représente l'esquisse globale de l'application mise en œuvre.

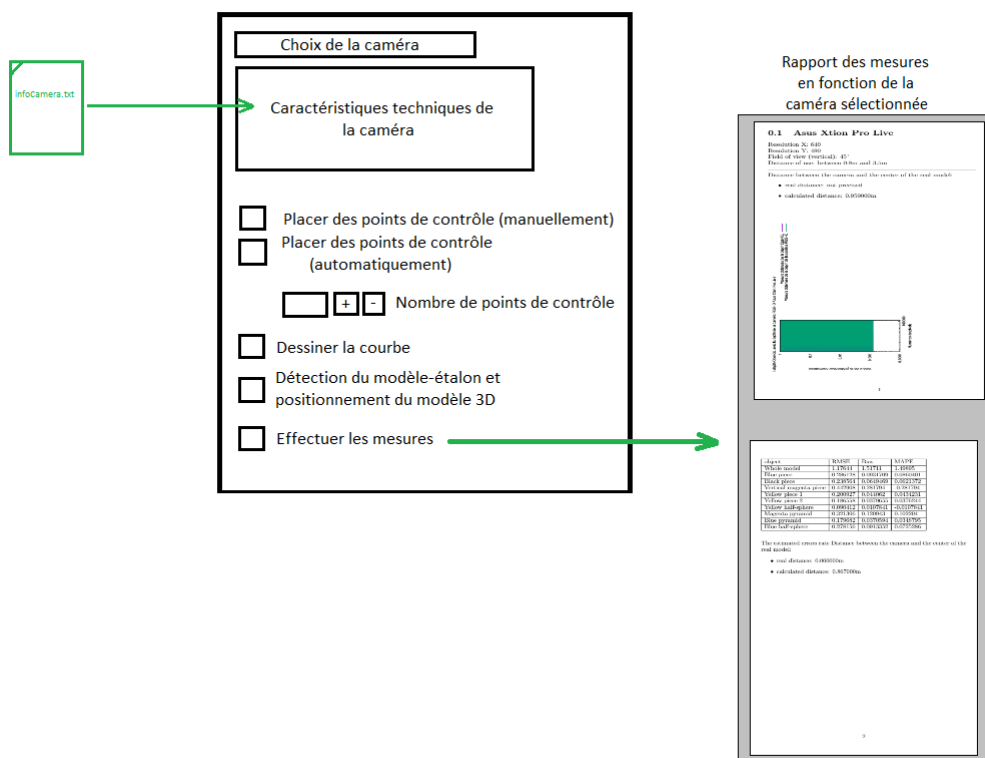


FIGURE 2.2 – Maquette de l'interface graphique de l'application.

La figure 2.2 expose le squelette de l'interface graphique avec ses éléments la composant.

## 2.2 Application développée

Notre application est composée d'une fenêtre qui comprend une partie affichage de la scène filmée en temps réel ainsi qu'une partie interface graphique qui nous permet d'interagir avec le programme. Nous parlerons un peu plus en détail de ces parties dans les sous-sections à venir.

### 2.2.1 Première partie : Affichage de la scène captée

Pour cet effet, nous sommes partis d'un programme fourni par *M. Farès BELHADJ* que nous avons complété par d'autres lignes de code pour la conception de l'outil. Il comprend l'utilisation de la bibliothèque OpenGL, GL4Dummies ainsi que la bibliothèque OpenNI2 pour établir la connexion avec la caméra RGB-D pour nous fournir par la suite la carte de couleurs et la carte de profondeurs.

### 2.2.2 Seconde partie : Interface graphique

Pour l'interface graphique, nous avons exploité la bibliothèque Dear ImGui disponible sur GitHub sur le lien suivant <https://github.com/ocornut/imgui>. C'est une bibliothèque assez facile d'utilisation une fois l'étape de l'intégration dans le contexte OpenGL effectué. Cette interface contient un certain nombre de composants. Parmi les composants principaux, nous pouvons citer une liste déroulante qui nous permet de sélectionner le modèle de caméra sur laquelle nous souhaitons réaliser les tests de mesures. Ainsi, une fois le choix de caméra effectué, le programme met automatiquement à jour tous les paramètres présents dans les champs suivants la liste en récupérant les données à partir d'un fichier que nous avons nommé *infoCameras.txt*. Ainsi, ce dernier pourra être exploité pour tout rajout de nouvelles caméras et/ou modification sur les données uniquement. Autres les composants cités ci-dessus, nous noterons la présence de deux cases à cocher comme montré dans la figure 2.3. La première permet une détection du modèle et l'autre de passer les tests de mesures. Ces étapes seront détaillées dans les sous-sections à venir.

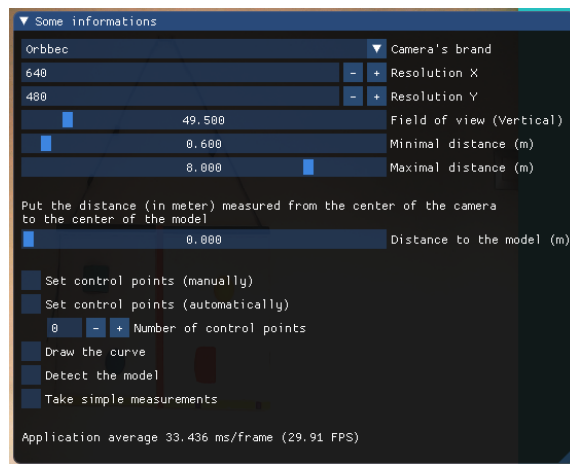


FIGURE 2.3 – Interface graphique du programme.

## Détection du modèle réel

Pour cette partie, une fonction *detectObject* est prévu à cet effet afin de renvoyer la localisation du modèle réel ainsi que celles des éléments le composant (demi-sphères, "pyramides", ...), ce qui fait un total de 10 objets à identifier. Une base de données d'images est créée pour servir de templates (images modèles) afin de pouvoir les distinguer.

Pour chacun d'entre eux, un appel à la fonction *callMatchTemplate* est effectué, qui, comme son nom l'indique, appelle la fonction *matchTemplate*<sup>1</sup>. Dans cette dernière, nous transformons notre image source (image récupérée de la scène captée par la caméra RGB-D) ainsi que notre image template en niveaux de gris et ensuite nous les passons sous le filtre Canny afin de rehausser les contours pour une meilleure détection<sup>2</sup>. Ce processus qui comprend l'utilisation des filtres est appliqué uniquement pour la détection du modèle en entier et non sur ses composants. En effet, la couleur importe pour ces derniers puisque la forme n'est pas unique (présence de deux demi-sphères, ...).

1. Fonction OpenCV permettant de trouver à partir d'une image source les zones de correspondances avec l'image *template* (image modèle). La documentation sur cette fonction est disponible sur le lien suivant : [https://docs.opencv.org/2.4/doc/tutorials/imgproc/histograms/template\\_matching/template\\_matching.html](https://docs.opencv.org/2.4/doc/tutorials/imgproc/histograms/template_matching/template_matching.html)

2. Nous nous sommes fortement inspirés du travail réalisé par Adrian ROSEBROCK, Ph.D en vision par ordinateur, disponible sur le lien suivant : <https://www.pyimagesearch.com/2015/01/26/multi-scale-template-matching-using-python-opencv/>.

Cette étape est assez conséquente en terme de temps. En effet, suivant la distance à laquelle est éloignée la caméra du modèle réel, le programme doit redimensionner la scène filmée avec une certaine échelle et avec un nombre précis de fois avant d'appeler la fonction *matchTemplate*, qui, combinée avec *minMaxLoc*<sup>3</sup>, nous permet de renvoyer la localisation du modèle (coordonnées en  $x$  et  $y$  ainsi que la largeur et la hauteur de l'aire détectée). De ce fait, une fois les coordonnées récupérées, nous pouvons positionner notre modèle virtuel dont nous discuterons les détails de conception un peu plus loin dans ce mémoire. Ces coordonnées seront naturellement converties en coordonnées OpenGL. La formule employée à cet effet (permettant une conversion d'un système de coordonnées vers un autre) est comme suit :

$$value_1 = \frac{value - min}{max - min} \cdot (MAX - MIN) + MIN$$

tel que  $value \in [min; max]$  (intervalle d'entrée) et  $value_1 \in [MIN; MAX]$  (intervalle de sortie)

## Tests de mesures et rédaction de rapport

Pour celle-ci, nous récupérons les valeurs de la carte de profondeur d'OpenGL ainsi que les valeurs de profondeur de la caméra RGB-D. les valeurs de la depth OpenGL seront par la suite linéarisées<sup>4</sup> et ensuite normalisées<sup>5</sup>. Celles renvoyées par la caméra seront uniquement normalisées. Les valeurs des deux côtés étant normalisées, les tests de mesures peuvent débuter.

Pour mesurer le taux d'erreur entre les valeurs des deux profondeurs, nous avons opté pour trois méthodes distinctes ; RMSE<sup>6</sup>, MAPE<sup>7</sup> et enfin Biais<sup>8</sup>. Une fois ces valeurs calculées pour chacun des éléments du modèle ainsi que

---

3. Fonction OpenCV qui renvoie les valeurs minimales et maximales d'une *Mat*. Cette dernière est une sorte de vecteur ou plus communément une matrice.

4. Technique employée pour « approcher par une fonction linéaire ».

5. Standardiser les données afin qu'elles appartiennent à l'intervalle  $[0; 1]$ .

6. Root Mean Square Error est une méthode permettant de calculer l'écart-type des erreurs enregistrées. Se référer à cette page web pour davantage de renseignements : <https://www.statisticshowto.datasciencecentral.com/rmse/>

7. Mean Absolute Percentage Error est une méthode permettant une mesure des erreurs tout comme le RMSE vu plus haut. Consulter cette page web pour un maximum d'informations : <https://www.statisticshowto.datasciencecentral.com/mean-absolute-percentage-error-mape/>

8. Tout comme les deux méthodes vu précédemment, elle permet un calcul du taux d'erreur. Le lecteur peut consulter ce lien la concernant <http://www.chups.jussieu.fr/polys/biostats/poly/POLY.Chp.10.html>

le modèle lui-même, nous les sauvegardons dans un tableau, tableau qui sera plus tard utilisé pour le rapport.

Par la suite, nous nous servons de Gnuplot<sup>9</sup> afin de tracer les graphes montrant les valeurs de depth des deux côtés en faisant attention à ne pas écraser un éventuel graphe déjà présent.

De plus, un rapport est rédigé en L<sup>A</sup>T<sub>E</sub>X en mettant en avant les informations relatives à la caméra (comme le champ de vision, la résolution, ...). A chaque fois qu'on coche sur la case *Take simple measures* de l'interface graphique, le programme écrit ou réécrit sur un fichier que nous nommons *Report.tex* situé dans un dossier *Report* créé à cet effet. Si la caméra a déjà été testée, le programme rajoute dans la section la concernant (dans le rapport) les nouveaux résultats incluant graphe des valeurs de la depth, la distance calculée par le programme entre la caméra et le centre du modèle détecté ainsi qu'un tableau exposant les valeurs obtenues du taux d'erreurs pour les modèles et ses composants avec les trois méthodes citées plus haut. Et ce, si et seulement si la caméra n'a pas été d'ores et déjà testée avec cette distance calculée par le programme.

---

9. Un logiciel multiplateforme gratuit qui offre la possibilité de tracer des graphes en deux ou trois dimensions.



0.1 Asus Xtion Pro Live

Resolution X: 640  
Resolution Y: 480  
Field of view (vertical): 45°  
Distance of use: between 0.8m and 3.5m

Distance between the camera and the center of the real model:

- real distance: not precised
- calculated distance: 0.959000m

object	RMSE	Bias	MAPE
Whole model	1.17644	1.51711	1.49895
Blue piece	0.286128	0.0931709	0.0860401
Black piece	0.238564	0.0649469	0.0621372
Vertical magenta piece	0.442908	0.281794	-0.281794
Yellow piece 1	0.200927	0.044062	0.0434231
Yellow piece 2	0.186558	0.0379655	0.0376244
Yellow half-sphere	0.090412	0.0107841	-0.0107841
Magenta pyramid	0.321306	0.120943	0.102204
Blue pyramid	0.179682	0.0370594	0.0348795
Blue half-sphere	0.278156	0.0913352	0.0725286

The estimated errors rate Distance between the camera and the center of the real model:

- real distance: 0.000000m
- calculated distance: 0.867000m

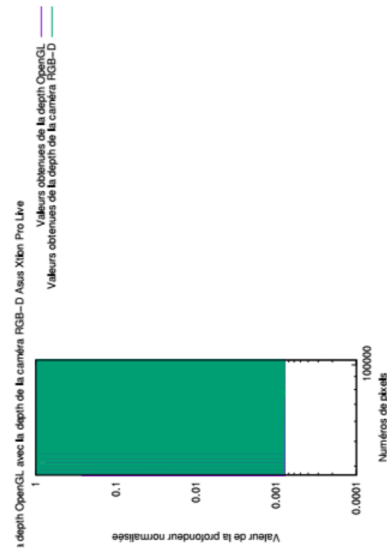


FIGURE 2.4 – Partie du rapport généré par le programme.



# Chapitre 3

## Cas pratiques de qualification

### 3.1 Conception du modèle réel et du modèle virtuel

#### 3.1.1 Conception du modèle réel

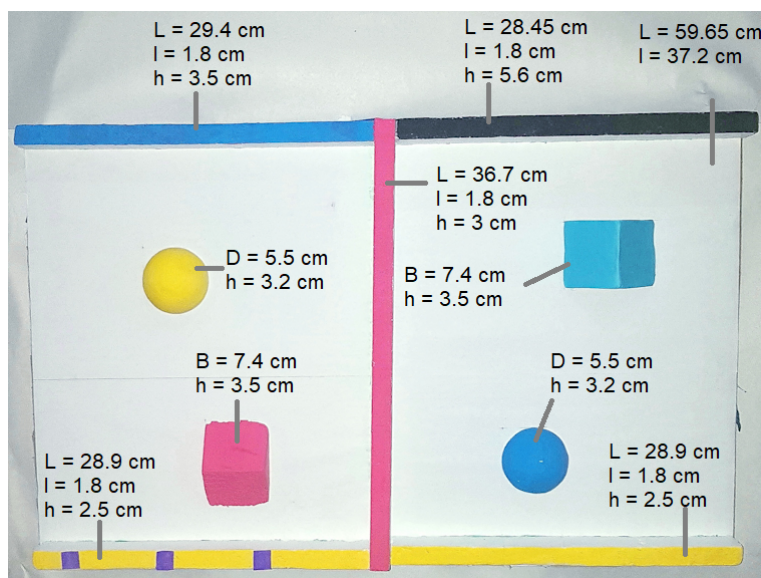


FIGURE 3.1 – Modèle-étalon.

Pour fabriquer le modèle réel, nous avons décidé d'opter pour le polystyrène pour les composants du modèle et du carton pour sa base. Cette

dernière sera par la suite recouverte de papier blanc. Les composants seront peints de couleurs distinctes suivant leur hauteur (les éléments qui possèdent une même hauteur seront peints de la même couleur plus particulièrement les rebords du modèle) sauf les demi-sphères et "pyramides".

Nous avons opté pour le polystyrène car c'est un matériau facilement maniable et que l'on peut trouver assez facilement.

### 3.1.2 Conception du modèle virtuel

La réalisation du modèle virtuel s'est faite à l'aide du logiciel 123D Design qui est assez facile d'utilisation. Chaque élément du modèle est conçu en fonction des vraies mesures faites sur le modèle réel.

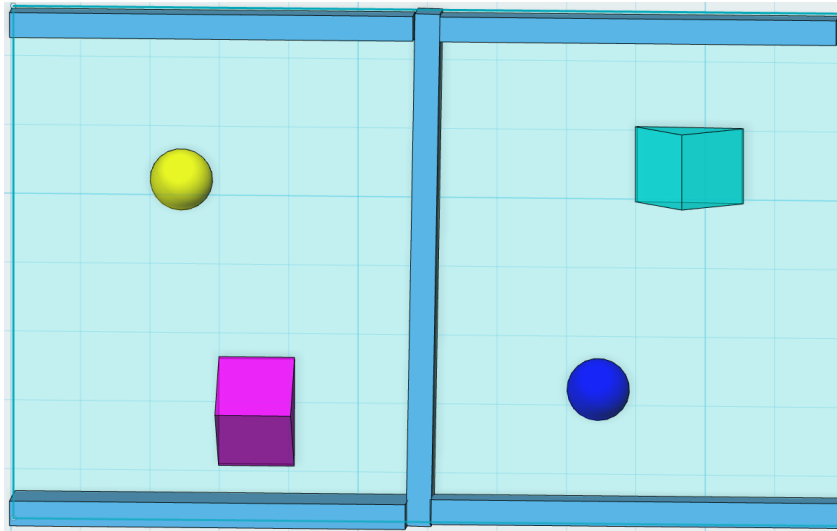


FIGURE 3.2 – Modèle 3D.

## 3.2 Preuve de concept (Proof of concept - POC)

Dans cette section, nous exposons la procédure suivie pour configurer les caméras *Asus Xtion Pro Live* et *Orbbec Astra Pro* afin que le programme puisse procéder aux tests de mesures. Tout d'abord, nous avons édité le fichier *infoCameras.txt* cité dans le chapitre précédent en vue d'y rajouter les informations relatives à ces caméras.

Par la suite, nous n'avons pas eu besoin de rajouter des lignes de code pour la reconnaissance de la caméra *Asus Xtion Pro Live* (initialisation et

récupération de la carte de couleurs et de la carte de profondeurs) car ce modèle est reconnu par la bibliothèque OpenNI2. Bien que la caméra **Orbbec Astra Pro** soit aussi supportée par cette bibliothèque, il a quand même été indispensable de rajouter des lignes de code afin de récupérer la carte de couleurs. En effet, OpenNI2 ne permet pas de la capturer, par conséquent, nous avons dû exploiter certains aspects de la bibliothèque OpenCV pour se faire.

Enfin, il nous suffira de sélectionner, à partir de l'interface de l'application, le modèle de la caméra branché et cocher sur les cases *Detect the model* pour une détection du modèle-étalon et *Take simple measures* pour démarrer les tests de mesures et la rédaction automatique du rapport comme expliqué précédemment.

## 3.3 Résultats et critique

### 3.3.1 Résultats obtenus

Ces résultats sont extraits du rapport rédigé par le programme après réalisation des tests avec les caméras Asus Xtion Pro Live et Orbbec Astra Pro sur trois distances différentes exprimées en mètre (distance entre la caméra et le modèle). Nous avons créé la base de données d'images templates suivant une des distances.

	Distances (m)					
	<b>1.7</b>		<b>1.609</b>		<b>1.522</b>	
<b>object</b>	<b>Asus</b>	<b>Orbbec</b>	<b>Asus</b>	<b>Orbbec</b>	<b>Asus</b>	<b>Orbbec</b>
Whole model	51%	81%	90%	40%	91%	75%
Blue piece	55%	45%	91%	50%	94%	41%
Black piece	56%	46%	91%	50%	94%	42%
Vertical magenta piece	55%	46%	91%	50%	94%	41%
Yellow piece 1	28%	98%	93%	22%	99%	97%
Yellow piece 2	28%	98%	94%	48%	99%	97%
Yellow half-sphere	54%	45%	90%	51%	94%	37%
Magenta pyramid	53%	46%	90%	51%	94%	37%
Blue pyramid	53%	46%	91%	50%	94%	38%
Blue half-sphere	55%	45%	90%	52%	94%	37%
<b>Average rate</b>	<b>48.8%</b>	<b>59.6%</b>	<b>91.1%</b>	<b>46.4%</b>	<b>94.7%</b>	<b>54.2%</b>

On peut aisément observer un taux d'erreur moins important avec la Orbbec. Les taux restent néanmoins très élevés.

### 3.3.2 Critiques

Les taux anormalement élevés observés dans la section précédente nous prèlèvent des doutes sur deux points principaux qui pourraient être omis dans l'application.

- Fiabilité des valeurs de la depth OpenGL extraites.
- Aire de détection du modèle 3D non conforme à celle du modèle réel en terme de coordonnées.

Pour l'heure, à moins que nous ignorons d'autres points, nous ne voyons pas d'autres éléments qui pourraient impacter ces résultats.

## Chapitre 4

# Conclusion et Perspectives

Dans ce rapport, nous avons présenté l’outil que nous avons développé à ce jour pour la qualification de caméras RGB-D. D’après les résultats exposés précédemment, nous pouvons admettre que la **Orbbec Astra Pro** l’emporte sur la **Asus Xtion Pro Live** en terme de précision dans la mesure.

Cependant, comme nous avons pu le constater sur les deux caméras, cette application renvoyait des taux d’erreur excessivement élevés, ce qui nous a questionné sur sa fiabilité. Ainsi, nous avons établis deux éléments qui pourraient potentiellement être à l’origine de cette anomalie.

- Fiabilité de la depth OpenGL.
- Non conformité des coordonnées des zones détectées entre le modèle réel et le modèle virtuel.

En perspectives, nous souhaiterions vérifier de manière plus pointilleuse les données OpenGL extraites ainsi que la correspondance entre les deux aires de détection. Dans ce cas, la qualification pourra être effectuée dans des conditions permettant une meilleure comparaison entre les différents modèles de caméras.

Une fois les conditions réunies, nous pensons effectuer les tests de mesures sur la colonne vertébrale d’un vrai patient. Pour ce faire, il nous faut rechercher un modèle 3D de colonne vertébrale ou un modèle s’en rapprochant le plus possible.

L’application résultante devra être en mesure de détecter la colonne vertébrale de la personne posant de dos à la caméra RGB-D. Les personnes qui suscitent notre intérêt sont principalement ceux atteints de scoliose. Dans un premier lieu, la localisation de l’épine dorsale va être effectuée en se basant sur du Machine Learning. Par la suite, une fois la colonne vertébrale détectée, le programme devra être apte à bien positionner le modèle 3D de manière à ce qu’il corresponde à l’épine dorsale du patient observé. Enfin, les tests de

mesures pourront être lancés.



# Bibliographie

- [AFS14] Faraj Alhwarin, Alexander Ferrein, and Ingrid Scholl. Ir stereo kinect : Improving depth images by combining structuredlight with ir stereo. 12 2014.
- [AH15] A.Maimone and H.Fuchs. Encumbrance-free telepresence system with real-time 3d capture anddisplay using commodity depth cameras. 2015.
- [Har10] Jacques Harvent. *Mesure de formes par corrélation multi-images : application à l'inspection de pièces aéronautiques à l'aide d'un système multi-caméras*. Thèse en systèmes informatiques critiques, Université Paris 8, November 2010.
- [KSS05] John Kessenich, Graham Sellers, and Dave Shreiner. *OpenGL Programming Guide : The Official Guide to Learning OpenGL, Version 4.5 with SPIR-V*. Addison-Wesley, 2005.
- [Mat09] Eric Matheron. *Incidence des phories verticales sur le contrôle postural en vision binoculaire*. Thèse en sciences (neurosciences), dec 2009.
- [MKM12] F. Kenton Musgrave, Craig E. Kolb, and Robert S. Mace. Shake'n'sense : Reducing interference for overlapping structured light depth cameras. 2012.
- [MKM15] F. Kenton Musgrave, Craig E. Kolb, and Robert S. Mace. Ametrologicalcharacterizationofthekinectv2time-of-flightcamera. *ELSEVIER*, 2015.
- [NJMKSL15] N.RAFIBAKHSH, J.GONG2, C.GORDON Mohsin K. SIDDIQUI, and H.Felix LEE. Analysis of xbox kinect sensor data for use on construction sites : Depth accuracy and sensor interference assessment. 2015.