

Rapport du projet Scanner 3D

Axel Jacquot

M1 SER





Table des matières

Présentation du sujet :	2
Scanner 3D :	3
Choix de la solution :	3
Kinect v2 :	3
Intel RealSense:	3
Solution choisie :	4
Travaillant restant à faire pour cette partie :	4
Application 3D :	5
Processeur Graphique ou GPU :	5
Choix de l'API graphique :	5
OpenGL :	5
Vulkan :	5
Travaillant restant à faire pour cette partie :	6
Logiciel :	7
Conclusion :	8



Présentation du sujet :

Pour ce projet, on nous a demandé de concevoir un scanner 3D qui sera en capacité de pouvoir scanner un buste d'être humain. Après avoir fait un état de l'art et analysé les différentes solutions existantes, faire un choix de technologie ou de croisement de technologie à utiliser pour avoir un scan 3D automatisé d'objets d'une taille réglable par l'utilisateur avec un maximum fixé. Ce scanner devra embarquer une carte microcontrôleur pour effectuer les calculs nécessaires à l'analyse de la surface du sujet et la reconstitution du nuage de points, dans un format 3D au choix par l'utilisateur. Dans mon cas je m'occuperai du Scannage 3D , de l'application 3D et du logiciel d'utilisation

Scanner 3D :

Choix de la solution :

Nous avons plusieurs technologies pour effectuer un scan 3D, que nous avons expliqué dans le rapport de groupe.

Lors du choix de la solution, nous avons eu plusieurs choix à faire pour pouvoir effectuer le travail demandé, ces choix sont le prix, la mise en œuvre de la solution, le type de la technologie, la communauté travaillant sur cette solution. Différents types de technologies ont été présenté dans le rapport de groupe.

Nous avons sélectionné de prendre une caméra de profondeur comme technologies. Dans cette technologie nous avons deux produits qui ressortent qui sont la Kinect et les caméra RealSense de Intel.

Kinect v2 :



La Kinect est une caméra de profondeur développée par Microsoft. Microsoft a développé une Application pour ordinateur, pour le traitement des données de la Kinect. Sur la Kinect nous retrouvons une résolution pour la caméra de profondeur de 512 x 424 pixels et de 1920 * 1080 pour la caméra couleur. Elle a une distance de profondeur de 4.5m au maximum et 50cm au minimum. Elle est capable de prendre 30 images par seconde. Elle a un prix de 100€ et en plus il nous faut un adaptateur pour pouvoir la brancher qui est dans les 30€, ce qui nous fait un total de 130€.

Intel RealSense:

Les caméras RealSense de Intel quant à eux pour la gamme D4. Cette gamme a un avantage, cet avantage est le fait que le traitement 3D se fait directement à l'intérieur de la caméra, ce qui va nous permettre d'alléger la charge de travail au niveau de la carte microcontrôleur principal.

	D415	D435
FOV	63.4° * 40.4°	85.2° * 58°
Sensor Technology	Rolling Shutter	Global Shutter
Résolution de la caméra de profondeur	1280 * 720 pixels	1280 * 720 pixels
Images par secondes (fps)	90	90



Distance minimale pour la profondeur	0.16m	0.11m
Distance maximale pour la profondeur	10m	10m
Résolution de la caméra couleur	1920 * 1080 pixels	1920 * 1080 pixels
Prix	179\$ soit 156.213€	149\$ soit 130.032€

Global Shutter (Obstruteur Globale) et Rolling Shutter (Obstruteur Déroulant) :

La différence entre les deux se trouve dans la méthode de captures d'une image. Pour le Global Shutter l'image est prise en une fois ce qui permet qu'il n'y pas d'effet déformant sur l'image qui a été prise. Pour le Rolling Shutter, l'image est prise ligne de pixels par ligne de pixels, ce qui créer des effets déformants sur l'image.

Solution choisie :

Pour la solution, nous avons fais le choix de prendre la caméra RealSense D435 de Intel. Nous avons fait ce choix pour plusieurs raisons :

- Global Shutter : ça permettra de prendre des images nettes lors de la rotation de la cible sur la plateforme et ainsi lorsque le sujet bouge lors du scan. Cela va nous permettre aussi de mettre moins de temps pour faire un tour car la plateforme marchera en continue.
- FOV : Cette caméra à l'angle de vision le plus grand ce qui va permettre de réduire la longueur du bras et la partie mécanique.

Travaillant restant à faire pour cette partie :

- Communication avec la caméra
- Configuration de la caméra
- Conversion du fichier 3D

Application 3D :

Pour la partie traitement 3D, nous devons choisir une méthode qui nous permettra de faire des calculs graphiques pour pouvoir créer la cible sous forme 3D. Pour faire ceci il existe différentes API qui existent comme OpenGL ou Vulkan (qui est basé sur OpenGL). Ce type d'API, nous permettra aussi de déplacer les différents calculs graphiques sur un processeur graphique et ainsi d'améliorer la durée des calculs. Les processeurs graphiques sont aussi plus adaptés à la gestion des calculs 3D.

Processeur Graphique ou GPU :

Un processeur graphique permet d'assurer les calculs d'affichage et 3D. Il s'agit d'une structure parallèle formé de plusieurs cœurs permettant d'effectuer plusieurs tâches en même temps. Ces cœurs ne sont pas aussi performants que les CPUs mais plus nombreux donc ils sont utilisés pour effectuer des calculs 3D ou d'affichages car les calculs sont moins lourds mais plus nombreux.

Choix de l'API graphique :

OpenGL :

OpenGL est une API graphiques se trouvant dans la plupart des temps déjà intégré au GPU, elle permet d'utiliser le processeur graphique pour effectuer les différents calculs pour le traitement 3D et pour l'affichage.

Vulkan :



Vulkan est une API graphique, il s'agit de l'API qui succède à OpenGL. Elle a plusieurs avantages comparer à OpenGL :

- Meilleur Gestion du multi-processing
- API plus facile d'utilisation
- Meilleur Gestion de la mémoire
- Calcul non graphique pouvant être fait sur le GPU

Après différents tests avec les deux API, nous avons choisi que pour l'affichage 3D, nous utiliserons Vulkan pour effectuer cette tâche. Ce choix pourra changer selon la carte embarquée que nous choisirons.



Travaillant restant à faire pour cette partie :

- Appréhender la méthode de programmation de Vulkan (en cours)
- Programmer l'application



Logiciel :

Pour la partie logicielle, nous pensions à faire en sorte que l'utilisateur puissent régler le type de fichier 3D qu'il souhaite avoir en sortie, ainsi que le lancement du scan de la cible. A la fin du scan, nous souhaitons afficher le résultat du scan donc pour ce faire nous devons faire un logiciel compatible avec OpenGL.



Conclusion :

Après des recherches sur les caméras nous nous sommes aperçus que la caméra faisait les traitement 3D par elle-même ce qui nous a permis de retirer une partie du travail à effectuer. Il me reste du travail.