

Projet VMsIVR

Visualisation Multi spectrale Image in a Virtual Room



Rapport de stage master I2A 2015/2016
Université de Bourgogne
Réalisé par Karim MOULAY

Réalisé au ColorLab NTNU – à Gjøvik - Norvège
Superviseur principal :
Pr. Jon Yngve HARDEBERG
Superviseur : Sony GEORGE

Project Tango :	4
<i>Objectif:</i>	4
<i>Les intervenants :</i>	4
<i>Outils et instruments mis à disposition :</i>	4
<i>Description du projet avant d'arriver :</i>	5
<i>État de l'art</i>	6
Multi spectrale :	7
Project Tango :	11
Réalité virtuelle :	17
Futurs directions :	27
Autre :	27
Conclusion :	28

Introduction :

Objectif:

Développer une solution pour l'acquisition de pièces/volumes/bâtiments à partir d'une tablette, et de modéliser l'environnement en 3D. Utiliser ces maquettes 3D pour des visites en réalité virtuelle, ou en réalité augmentée.

Certaines textures 2D (des tableaux/œuvres artistiques...) seront mise particulièrement en valeur. Et pour cela, les œuvres seront saisies avec une caméra multi spectrale, et leurs représentations dans l'environnement virtuel seront remplacées par les photos issues de technique recombinaison des images spectrales.

L'utilisateur pourra ainsi visualiser les œuvres selon son choix en RGB recombinaison multi spectrale, ou spectre par spectre.

Date du début : 04 Avril 2016

Durée : 13 semaines (3mois)

Lieu : ColorLab - NTNU - Gjøvik (Norvège)

Les intervenants :

Pr. Jon HARDEBERG : Superviseur

Sony GEORGE : Superviseur

Mariusz NOWOSTAWSKI : Professeur au NTNU et notre conseiller en la technologie TANGO PROJECT

Alaric COTTIN : Stagiaire de l'université de BOURGOGNE (uB)

Outils et instruments mis à disposition :

Caméra multi spectrale : Pixelteq Spectrocam --> équipé de 8 filtres + ordinateur portable avec logiciel dédié, et licence pour piloté la caméra.

Spectromètre : Minolta CS1000

Tablette : Google Project Tango

Ordinateur : mon ordinateur portable personnel

Logiciels utilisés:

Matlab 2015

Moteur de jeu Unity 3D

Ocean Thin Film – Logiciel fourni avec la caméra

Android studio

Blender 3D

Navigateur web (pour l'exécution de script WebGL)

Éditeur de texte

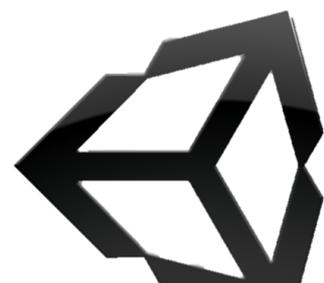
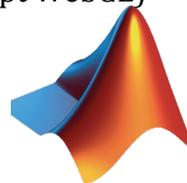
Langage de programmation :

Matlab

JavaScript (JS)/HTML5

C/C++

HTML5 Langage pour Android



Description du projet avant d'arriver :

Le projet consiste à créer une construction 3D à partir d'images et de vidéos, **non calibrées**, et de différentes couleurs (monochrome, niveau de gris, RGB(a)¹, multi spectrales, etc.). Ainsi je pars d'éléments hétérogènes, et j'essaye de retrouver des repères, pour pouvoir générer le modèle 3D de la construction.

Le challenge est de taille. Mais il correspond totalement à mes attentes de travailler sur un tel projet ainsi que de parfaire mes connaissances en multi spectrale. Et donc, emporté par les ailes de la motivation je me lance dans l'aventure.

La caméra multi spectrales offre l'avantage de capter des nuances qu'une caméra classique de base RGB ne capte pas forcément ou du moins pas correctement (comme la couleur jaune par exemple, ou encore ce qui se cache dans les IR (Infra Rouge) et les UV (Ultra Violet)). Ainsi, nous pouvons mieux capter et mettre en évidence des nuances de couleurs cachées.

Arrivée sur le terrain :

Première rencontre avec les superviseurs. Discussions, débat et lancement du projet.

Le projet consiste à reproduire un bâtiment en 3D, depuis des images. Les bâtiments peuvent être des musées, des palaces ou tout autre type de construction.

Pour les objets les plus précieux, et les plus remarquables, les acquisitions se feront avec une caméra multi spectrale, à fin de mieux faire ressortir les détails. L'objectif est donc de réaliser une forme de scanner 3D, et d'utiliser la maquette pour faire de la visite virtuelle, voir même, de la réalité virtuelle.

Les œuvres en multi spectrales peuvent être observer sur la longueur d'onde de notre choix (ou du moins selon ce que proposera la base de données que nous fournirons).

Une timeline peut être envisagée, pour visiter le lieu à différentes époques (exemple le papier peint a changé, il y a eu des rénovations, ...). Sur une version aboutit du projet, il est même envisageable de changer interactivement les couleurs, et de simuler les futures rénovations.

L'équipement mis à disposition, est une caméra multi spectrale, en attente d'autres équipements.

Il m'a été proposé de travailler en collaboration avec un autre stagiaire, Alaric COTTIN, qui est un camarade de classe. J'ai suggéré dans ce cas, que nos travaux restent bien distincts. Ainsi, chacun peut progresser sur sa tâche à son rythme. L'intégration de nos travaux, se fera au fur et à mesure, selon un planning établi de concert et d'un commun accord. Les superviseurs ne nous ont donné aucunes consignes particulières, et nous ont laissé une liberté totale d'action, ainsi que le choix des outils, le langage de programmation etc.

Rencontre avec Alaric. Il m'a présenté les lieux et l'équipement mis à disposition (ainsi que la ville, et les lieux de commodités). Nous avons discuté de nos objectifs respectifs. Dans mon cas, je voulais profiter de l'opportunité de ce stage pour travailler le maximum sur les l'imagerie multi spectrales (voir même hyper spectrale si possible).

Nos taches se sont donc facilement définies, je commence à travailler sur tout ce qui est acquisition et traitement d'image multi spectrale, et Alaric lui sur la construction 3D. J'avais déjà fait des recherches et préparé de la documentation à ce sujet, en préparation pour mon stage, que j'ai mis à disposition.

¹ RGB : Red Green Blue (Rouge Vert Bleu), et si l'on gère la transparence on rajoute « A » pour Alpha

État de l’art

Les évolutions continues du projet ont rendu difficile la recherche pour savoir ce qu’il existe ou ce qu’il se fait. Mais dans le domaine, plusieurs projets qui se rapproche de ce que j’ai fait, sont en cours, ou alors ont vu ou vont voir le jour.

Dans mon cas, je ce n’est qu’une variante parmi d’autres.

Toute fois, je me dois d’en citer quelques uns :

- **Microsoft - HoloLens :**

Projet de réalité augmentée, avec interaction dans le monde virtuel



- **Google - Project Tango :**

Propose plusieurs choses, dont entre autre de la réalité augmentée pour consulter les œuvres d’art



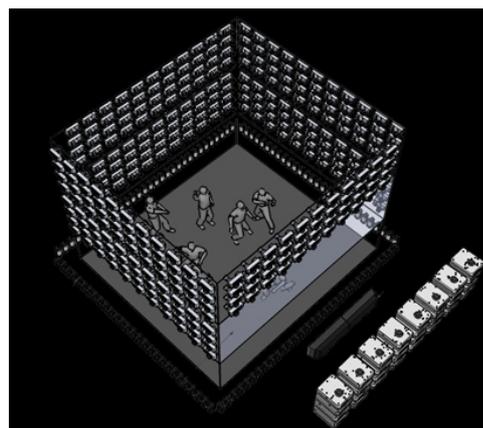
- **Sony – 3D holographic Football**

En 2010, Sony promet de proposer la coupe du monde en 3D holographique en directe dans les plus grandes capitales du monde (dans les grands stades), si le Japon obtenait l’organisation de la coupe du monde en 2022. Personnellement, je ne suis pas fan de foot, mais j’aurai vraiment voulu voir ce projet aboutir, et voir le jour.



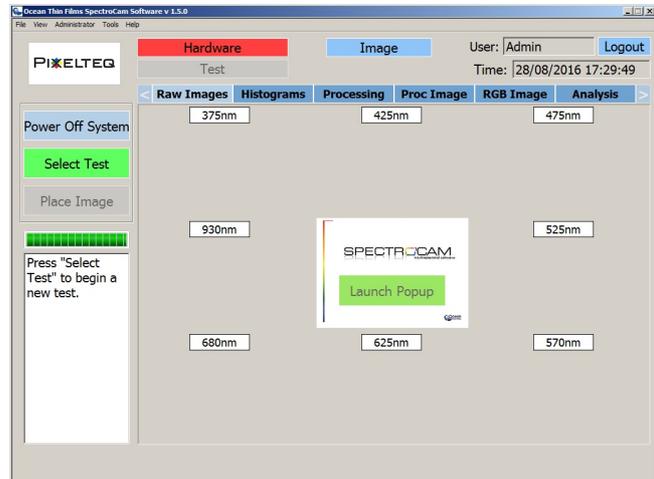
- **Carnegie Mellon university – Virtualizing Engine**
(Pittsburgh, Pennsylvania, United States)

1.000 caméras, pour filmer en directe, sous tous les angles certains sports de combat. De ces images, toute la scène est reproduite en 3D en temps réel. Le résultat peut être exploiter pour différentes fins.



Multi spectrale :

Nous avons à disposition une caméra multi spectrale «**Pixelteq Spectrocam**». Avec 8 filtres installés, allant de 370 à 930nm (longueur d'onde en NanoMètre). Un ordinateur portable dédié à la caméra est fourni aussi, le logiciel qui pilote la caméra y est installé, avec la licence. Alaric étant arrivé 3 semaines plus tôt, il me présente le matériel et le logiciel «**Ocean Thin Film**».

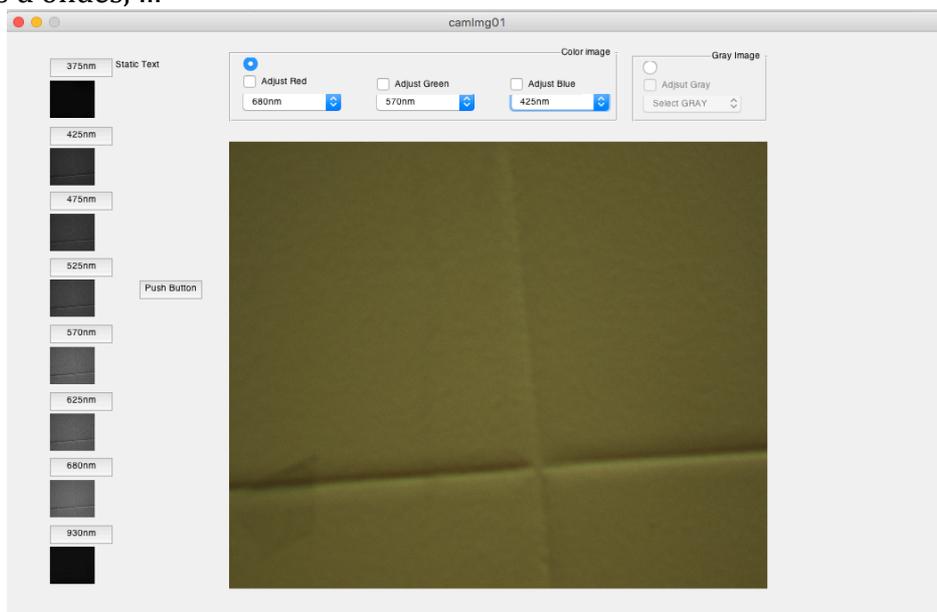


Je découvre l'interface, me familiarise avec les fonctionnalités, le produit, les acquisitions, ...

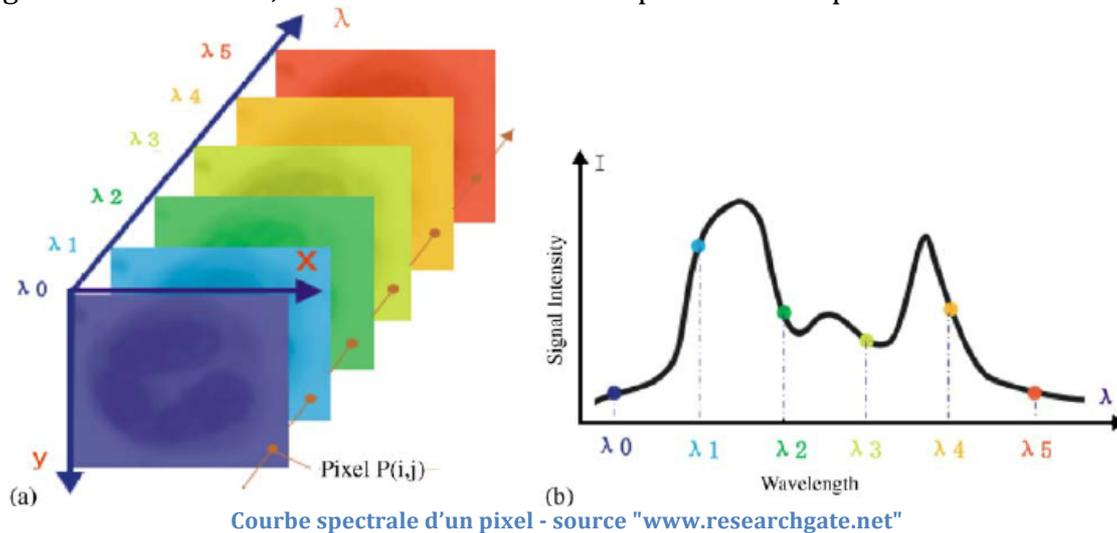
Première constatation, les manipulations d'images ne peuvent se faire qu'en direct. La caméra doit être connectée et en fonctionnement. Sinon, ça ne marche pas ! C'était un peu dérangent pour moi, car je pourrais avoir besoin de faire des tests chez moi, dans le cas où j'ai une idée qui me passe par la tête.

Il est possible de sauvegarder les acquisitions, en format image standard. Une image en niveau de gris (NG)/longueur d'onde. J'ai donc développé un programme sous Matlab qui proposera toutes les fonctionnalités (ou du moins le plus possible) qu'offre le logiciel du constructeur, pour une utilisation hors connections.

Les fonctionnalités sont la reconstruction d'image RGB, par association de longueur d'onde pour chaque composante, les opérations logiques entre deux images de deux longueurs d'ondes, ...



En écrivant le programme, j'ai commencé à ajouter des options, comme par exemple étaler l'histogramme, car sur les extrêmes (les hautes et basses fréquences ; IR et UV), les images sont très sombres, voir noir. J'ai aussi commencé à réfléchir à une solution pour reconstruire une image RGB à partir de longueur d'onde de notre choix. Pour ce faire, je me suis inspiré du principe des courbes, pour ainsi dessiner la courbe spectrale de chaque pixel (ou du moins calculer la longueur d'onde de mon choix), et ainsi, selon la longueur d'onde choisie, avoir l'intensité de la composante en ce pixel.



Le principe est simple, j'utilise les longueurs d'ondes et les intensités comme point de contrôle de la courbe, et grâce à cela, je construis la courbe spectrale du pixel. De là, je récupère l'intensité d'un pixel sur une longueur d'onde de mon choix.

Pour ce faire, il faut choisir quel type de courbe utiliser, et un degré (ou ordre). Lors de mon cursus, j'ai étudié plusieurs types de modélisation de courbes différentes. Dans mes recherches, j'ai vu que l'on pouvait utiliser les courbes de LAGRANGE. Pour ne pas me limiter, j'ai fait le choix de faire des courbes NURBS (Non-Uniform Rational Basics Splines) cubiques. Pourquoi ce choix ? Cubique, car lors de mon cursus universitaire, il m'a été démontré qu'il est inutile d'utiliser des courbes avec un degré supérieur à 3 (d'ordre 4). La différence ne sera pas significative, mais surtout la complexité du temps de calcul sera exponentiellement élevé. Quand au NURBS, elles sont le cas général des courbes. De là, on peut entre autre en déduire des B-Splines, qui elles même donnent les courbes de Bézier, Hermite, et bien d'autres. Ainsi, grâce à cette programmation, je pourrai facilement tester et comparer la qualité des courbes (temps de construction, efficacité, etc.). Comme les filtres installés sur la caméra n'ont pas un pas régulier (séquence non uniforme), dans mon cas, la caméra disposait de 8 filtres dont les longueurs d'ondes sont : 375nm, 425nm, 475nm, 525nm, 570nm, 625nm, 680nm et enfin 930nm), l'utilisation des NURBS fut une évidence.

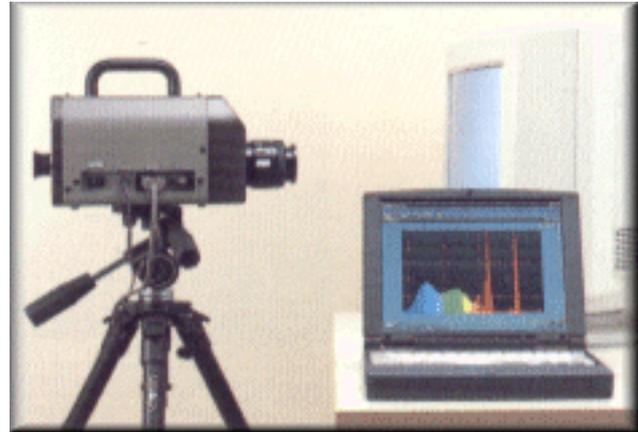
Ainsi, si je veux le bleu sur la longueur d'onde 435nm par exemple, je dois avant tout faire une fonction inverse (grâce à un solveur que j'ai écrit, le calcul est fait en amont), qui grâce à une longueur d'onde (que l'on peut appeler X), il me donne la valeur que l'on appellera T, qui est la position du point sur la courbe. En effet, les NURBS nous donne pour un instant T (qui correspond au déplacement du point sur la courbe), les coordonnées homogènes X et Y (on dira pour la suite que Y est l'axe des intensités). Donc j'utilise la fonction invNURBS(X) pour trouver T, et de là, la fonction NURBS(T) pour avoir Y. J'applique l'opération pour chaque pixel, sur chacune des 3 composantes RGB.

Multi spectrale – Spectromètre

Le programme que j'ai écrit fonctionne bien, mais... son temps d'exécution est bien trop long. Sur mon ordinateur personnel, le programme est écrit sous Matlab, avec peu d'optimisation du code (entre autre ajout d'affichage pour suivre l'évolution du script pour le débogage et autre), pour une image de 8 spectres de dimensions 2456×2058 , le temps de calcul est estimé à 3 jours plein. Pour cet exemple, j'ai testé 3 courbes de Bézier par pixel ; une courbe cubique par composante (pas de continuité entre les courbes). Naturellement quand je parle de courbe, je ne parle que du point d'intensité qui m'intéresse, je ne dessine pas toute la courbe.

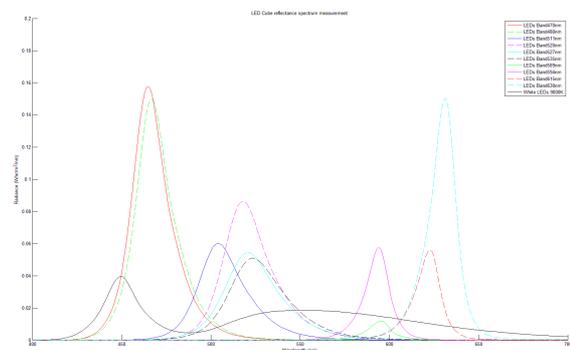
L'utilisation de palette de couleurs entre autre pourra réduire le temps de calcul. Mais partis de ce résultat, je me suis rendu compte qu'il sera impossible de proposer à l'utilisateur le choix des longueurs d'ondes pour avoir en temps réel une image couleur ! Du moins pas de cette façon.

Pour ce qui est de l'acquisition, en particulier quand on tend vers les extrémités (IR et UV), les images sont très sombres. Lors de l'acquisition, on peut choisir le temps d'exposition pour chaque filtre (longueur d'onde). Il était donc nécessaire de prolonger le temps d'exposition pour ces deux filtres. N'ayant pas de repère, je me contentais de choisir le temps d'exposition maximal (soit 2s d'exposition).



Spectromètre Minolta CS1000 – "www.scientec.fr"

Entre temps, j'ai fait la connaissance d'un ingénieur Mohamed BOUZIDI ancien diplômé de l'uB en électronique, qui développait 2 projecteurs à matrice LED, contrôlé par ordinateur. Il est possible grâce à ces projecteurs, de choisir quelles LED allumer et quelles LED éteindre, ainsi que leurs degré d'intensité (comme le fait une dalle de téléviseur OLED). J'ai donc commencé, à voir m'intéresser à son travail. J'ai ainsi découvert un équipement bien utile, le spectromètre, « **Minolta CS1000** ». Cet appareil permet de connaître la courbe spectrale d'un point. J'ai remarqué que la courbe spectrale de la lumière artificielle n'est pas uniforme. Elle est accentué sur certaines fréquences, et creuse dans d'autres. C'est un mélange des 3 composantes, mais qui laisse des creux entre les pics.



Multi spectrale – Spectromètre

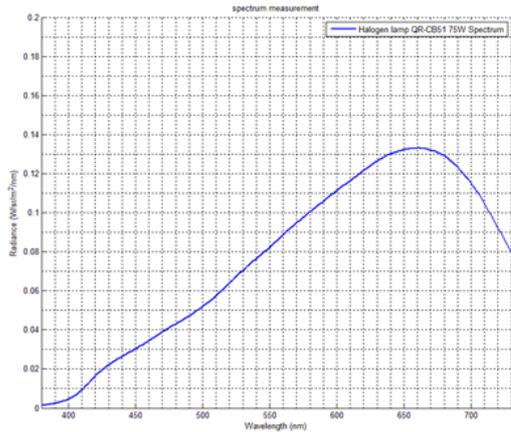


Figure: Plot of the Halogen lamp QR-CB51 75W Spectra

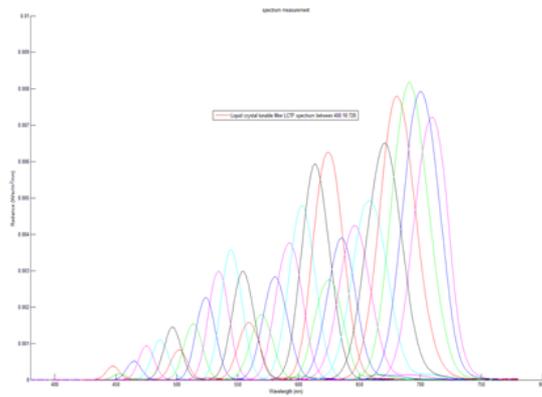


Figure: LCTF bands using the Halogen lamp QR-CB51 75W as a light source.

Extrait des travaux de Mohamed BOUZIDI

J'ai donc acheté des LEDs IR et UV pour les ajouter à son projecteur, et je lui ai proposé de lui écrire un script qu'il intégrera à son régulateur de lumière. Le but est soit d'obtenir une courbe spectrale uniforme (une ligne horizontale, presque similaire à celle du soleil), soit pour chaque longueur d'onde choisi, faire le mélange de couleurs et régler l'intensité pour obtenir pour chaque longueur d'onde l'intensité lumineuse souhaitée. La caméra multi spectrale mise à disposition, permet les acquisitions filtre par filtre (ou longueur d'onde par longueur d'onde) de manière automatique ou manuelle. L'intérêt de cela, est de ne plus avoir à choisir aléatoirement le temps d'exposition pour faire les acquisitions, mais avoir un temps d'exposition unique, du fait que l'intensité lumineuse pour chaque longueur d'onde sera le même.

Malgré tout, le temps d'exposition reste un point ambigu. Les images prises longueur d'onde par longueur d'onde, sont en niveau de gris. Un temps d'exposition trop long, saturera les couleurs, et une exposition trop courte, effacera les couleurs les plus sombres. Il y a des solutions, pour atténuer cette lacune, comme par exemple le HDR (High Dynamic Range imaging : imagerie à grande gamme dynamique). Mais les choses peuvent vite devenir complexe. L'imagerie multi spectrale n'est qu'un élément de mon projet. J'ai donc fait un choix, au plus simple pour commencer. J'ai décidé de prendre des palettes de couleurs, que j'ajoute à la scène au moment de l'acquisition, pour avoir une référence de chaque couleur, et je normalise les pixels par rapport à cette référence que j'ai choisie. Car le rouge ou le blanc, comme toutes les autres couleurs, qui peut dire ce qu'est le vrai blanc, ou le vrai rouge ? Tout est relatif, et c'est un point de vue qui dépend de chacun. C'est pourquoi j'utilise une référence. Et c'est cette référence qui décide pour moi.

Project Tango :

Nous avons reçu un nouvel équipement, la tablette « Project Tango » de Google, dans les 2^{èmes} semaines après le début de mon stage. Je connaissais le produit de par les forums et les articles que j'ai pu lire, mais c'est la première fois que j'en avais une entre les mains.



Nous avons comme consignes d'utiliser la tablette en priorité pour faire de la réalité augmentée. Je devais stopper tout ce que je faisais sur l'imagerie, et donc me concentrer sur la tablette. L'objectif, est que l'utilisateur qui se déplace dans une galerie d'arts par exemple, puisse voir une œuvre en réalité augmentée à travers la tablette, il peut alors la visualiser sur la longueur d'onde de son choix, et bénéficier de toutes les options préalablement prévus pour la réalité virtuelle.

Bien qu'en théorie, je connais le principe, ni moi, ni Alaric ne savons comment faire de la réalité augmentée.

Il y a un début à tout ! Alors que je pensais pouvoir continuer à travailler avec mes bases ainsi qu'à ma préparation en vue de ce stage, je me trouve lancer dans le vide.

Comme je l'ai déjà signalé plus haut, les superviseurs m'ont donné une grande autonomie tant au niveau des horaires et rythme de travail, les choix, outils et autres. Je n'avais pas à rendre de compte rendu quotidien (ni même hebdomadaire). Je devais juste les informer par email sur quoi je travaillais et leur remonter toute progression significative.

Une telle autonomie et liberté est très appréciable.

Mais à chaque situation il y a un revers de médaille, je suis seul, et les superviseurs n'ont pas de temps à me consacrer, du moins pas plus que pour répondre à une seule question par écrit. Ceci étant, il faut se lancer. Et première chose à faire, c'est de découvrir la tablette.

Au jour d'aujourd'hui, une tablette n'a rien de sorcier en soit, encore moins pour un informaticien comme moi. Mais la tablette que j'appellerai tout simplement Tango, n'est pas comme les autres. Il suffit de voir la photo de l'arrière de l'appareil, pour ce rendre compte déjà qu'elle est différente. Je ne vais pas rentrer dans la description technique du produit, car le fournisseur la décrit très bien sur son site (store.google.com -> Project Tango). De plus, il faut prendre en compte que le produit est en phase de développement, il n'est donc pas encore fini. Plusieurs outils utilisés ou développés au moment du stage, peuvent s'avérer obsolète, surtout après les annonces du Google I/O courant Mai. Beaucoup d'innovations, surtout les applis, pas encore rendu public.

Pour commencer, il fallait se familiariser avec le produit, le découvrir ainsi ses que fonctionnalités... Nous y sommes mis à deux, Alaric et moi. Et ce que l'on peut dire, c'est que ça a pris du temps. 2 à 3 jours pour installer et résoudre les bugs des applis développées spécifiquement par Google pour une initiation et découverte, ainsi que pour installer l'environnement de développement sur nos ordinateurs personnels d'où nous développons (il y a beaucoup de paquets et beaucoup de dépendances).

Enfin, nous pouvions commencer à exploiter l'appareil. Deux applications (APK) ont particulièrement retenu notre attention. L'apk "Tango Constructor ²" et l'apk "Tango Explorer³".

La encore, sur internet vous pouvez trouver des démos et des explications de ces deux applications. Mais je vais tout de même les décrire brièvement pour expliquer ce qui a retenu mon attention.

- Tango Constructor, est un APK qui permet de faire une construction rapide en 3D, à partir de ce qui ressemblerai à de la stéréovision, mais à partir d'une caméra, et de mouvements, un peu comme la Kinect de Microsoft. C'est ce que l'on appelle « Shape from Motion ». La valeur ajoutée de Tango, est qu'elle utilise des capteurs de mouvements intégrés à l'appareil pour corriger les déplacements (entre ce qu'elle voit et ce qu'elle subit; accéléromètre et Gyroscope).

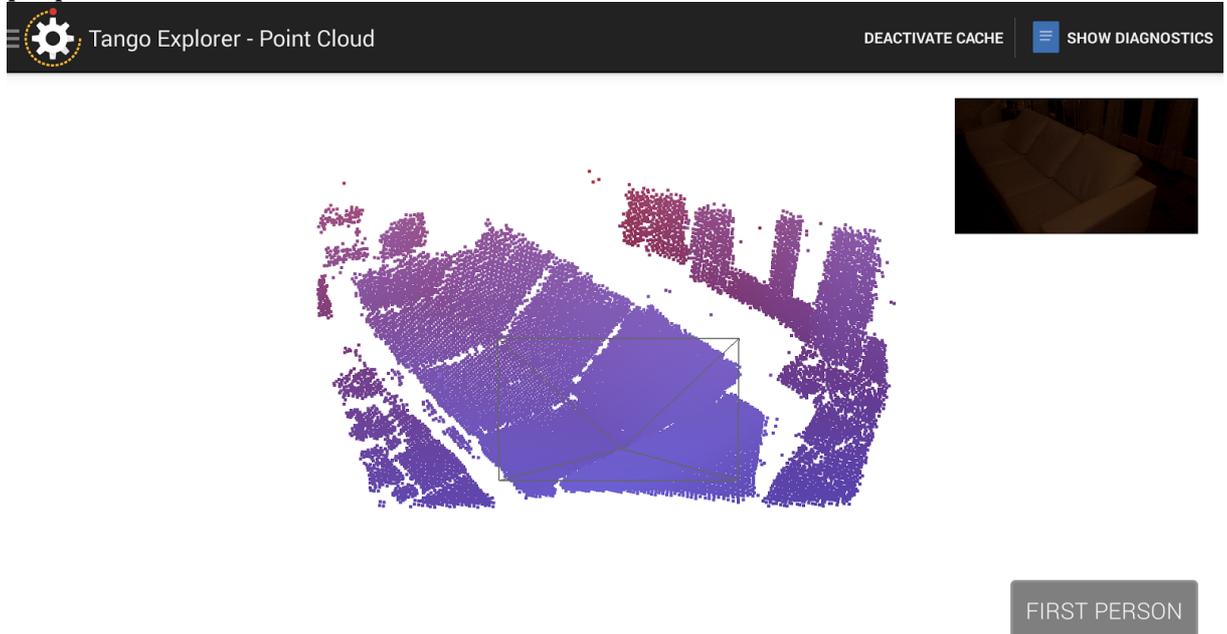


² <https://play.google.com/store/apps/details?id=com.projecttango.constructor>

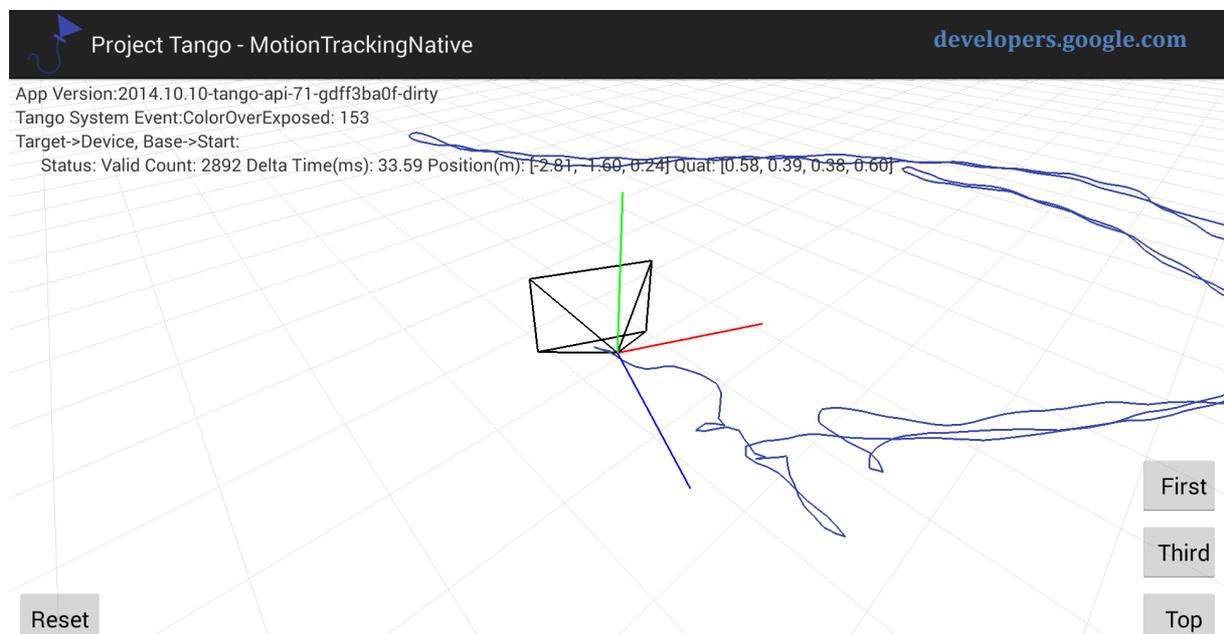
³ <https://play.google.com/store/apps/details?id=com.projecttango.tangoexplorer>

Project Tango – Initiation aux APIs Tango

- Tango Explorer, lui permet de faire un apprentissage d'un volume (une pièce, un endroit), et de pouvoir par la suite arriver à se localiser dans cette pièce, quelque soit le point de départ, la tablette arrive à se situer. Pour se faire, l'application utilise un nuage de point 3D (point cloud). Ces points mettent en évidence les différences de profondeur de l'environnement. Donc une surface lisse et perpendiculaire à la tablette serait blanche à l'intérieur de ces bords.



C'est le moment de passer aux choses sérieuses. Explorer est un outils très utile pour se localiser, encore faut il savoir l'utiliser. Google a pensé à tout. Elle propose des APIs pour ça justement. Explorer utilise deux choses, un chemin, et un ensemble de point. Mes recherches m'ont conduit à comprendre une chose, j'ai associé cela à des constellations. L'apprentissage permet d'enregistrer le parcours de la tablette, et cherche en même temps des points forts, des marqueurs (c'est généralement des plis, des coins, des changements de couleurs, ...). Grace à ça, par la suite, il suffit de se déplacer et de se localiser dans l'espace tel un navigateur dans les mers qui suit les étoiles et les constellations dans le ciel.



J'ai proposé à Alaric là encore, que l'on sépare nos tâches respectives. Ayant déjà une expérience dans l'odométrie, je me suis proposé de gérer cette partie. J'ai déjà créé un petit robot "GeekEnstein", que j'ai présenté lors d'une présentation en Anglais à l'uB en 2016. Mon projet ETIC : **Exploration en Terre Inconnue** (inconnu pour le robot) – **Exploration en Terrain Connu** (connu pour nous). Pour mon robot, j'utilise le même procédé de localisation et les mêmes composants que ceux dans la tablette Tango, voir même plus (boussole, ...). À la différence qu'ici, il faut développer sous Android.

Petite nouveauté pour moi (et Alaric), je n'ai jamais développé sous Android. Il me fallait donc commencer par une petite auto-formation pour arriver à exploiter et utiliser les APIs. Ce fut laborieux, mais je ne vais pas m'étaler sur le sujet. Le but est de pouvoir écrire un programme, pour pouvoir exploiter les données fournies par l'api Motion Tracking¹. Elle est assez documentée, mais pour faire simple, elle donne accès à un grand nombre de variables, qui permettent de suivre les variations subites par la tablette en temps réel. Ce qui permet de se localiser dans l'espace au cours ou suite à un mouvement.

L'intérêt de cet outil, est par exemple, on fait l'apprentissage de l'environnement d'un musée ou une galerie d'art, en ce déplaçant, et essayant d'observer le plus de points et de points de vue possible. Cela se fait naturellement quand il n'y a personne. Mais lors de l'utilisation, il a des gens et du mouvement. Il peut alors être difficile de retrouver tous les repères que l'on avait déjà enregistrés.

Le professeur Mariusz NOWOSTAWSKI que nous sommes allés voir, Alaric et moi (avant de recevoir la tablette) nous avait conseillé que pour la construction 3D, il fallait ajouter des objets (généralement des cubes, avec des codes QR²) pour pouvoir évaluer les distances et les profondeurs, et se localiser entre deux images. Mais avec ce que nous avons, nous cherchions à faire autrement. Et justement, une fois localisé, grâce à l'odométrie, même si visuellement, la tablette perd ces repères visuels (le long d'un mur lisse par exemple), elle continue à se localiser.

Nous avons apprécié ses conseils, mais il restait beaucoup de travail encore, et nous n'avions rien qui nous parle vraiment. Les principaux problèmes que j'ai levés, sont comment définir la zone de l'œuvre à modifier, comment gérer la différence de luminosité, couleurs, (etc.) alors que les conditions de luminosités ne sont jamais les mêmes, comment faire la reconnaissance pour identifier la position du tableau par exemple pour y superposer la réalité augmentée (c'est pas comme ajouter un objet libre dans la scène, comme se calquer exactement sur un objet et les confondre)? Et plus j'avancais, plus les interrogations s'ajoutaient.

Avec un peu de pratique, j'ai commencé à comprendre comment récupérer les données de l'API. Ainsi, je pouvais savoir où se trouve et à quelle hauteur est la tablette (donc la caméra), ce qu'elle regarde (gyroscope)... De là, j'ai voulu m'inspirer de ce que j'ai appris en cours, à savoir les shaders et les lancés de rayons. C'était une piste, une première idée du moins. Le but, est que si je m'imagine dans un environnement entièrement virtuelle, selon l'endroit où je me trouve, je peux limiter la génération de l'environnement à uniquement ce qui est visible. En partant de là, je peux en sachant ma position, rechercher ce que je suis censé trouver.

¹ <https://developers.google.com/tango/overview/motion-tracking>

² Quick Reponse ; code barre à deux dimensions

J'ai assisté à plusieurs conférences et présentations, dont entre autre celle du Dr. Shida BEIGPOUR de l'université de Siegen (Allemagne). Le Dr. Shida a fait une présentation sur comment capter les couleurs originales de la matière. Si l'on regarde un objet, selon notre angle de vue, ainsi que les influences de l'environnement autour, la couleur ne sera pas forcément identique (elle l'est rarement en réalité). Ainsi, elle a mis au point avec son équipe, une méthode qui identifie et sépare la couleur de la matière, ainsi que les ombres et les reflets, et peut changer facilement la couleur d'un objet, puis, lui restitue les ombres et les reflets, qui dépendront de la nouvelle couleur de fond.



Voiture trop claire par rapport à l'environnement

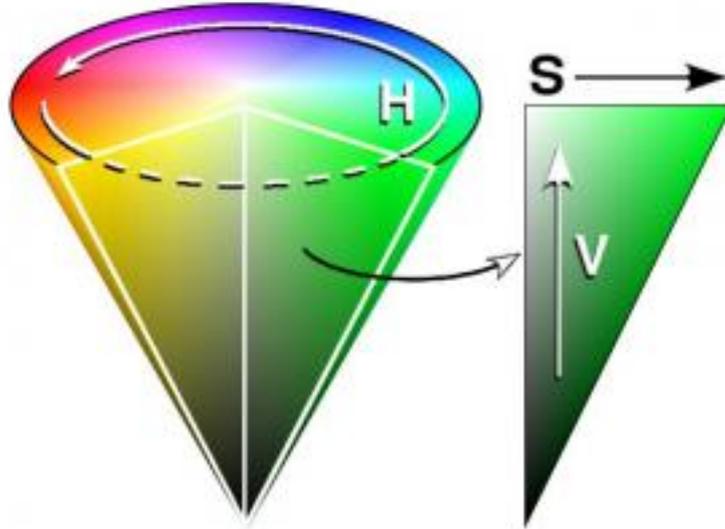
J'ai donc discuté avec elle de ce procédé qui est assez complexe, pour voir si je peux l'intégrer facilement dans mon projet. En effet, l'un des principaux problèmes de la réalité augmentée, est que les objets ajoutés doivent s'intégrer dans l'environnement réel. Je ne peux pas ajouter un objet pris avec une lumière parfaite, et dans des conditions optimales, et l'ajouter dans un environnement sombre. Je lui ai donc exposé les différentes pistes que j'ai suivies, à savoir entre autre calculer la luminosité de la pièce par le capteur photosensible intégré à Tango, ou encore faire une interpolation linéaire de quelques pixels de références pour connaître ce qui pourrait être le coefficient de luminosité, ...

Le Dr. Shida c'est donc intéressée au projet, et nous avons organisé une réunion (avec elle, Alaric et moi), pour présenter nos travaux, et ainsi profiter de ces conseils¹. Nous avons alors très vite compris que ses solutions, sont à un niveau très élevé, qui allaient nous embarquer sur un sujet de thèse, alors que nous n'avons que 3 mois (voir moins, vu que le stage est déjà entamé). Pour ce qui est du problème de précision, elle a plus ou moins appuyé mon idée de shader, sur le principe du moins. Elle m'a proposé de remplacer les pixels de l'œuvre pixel par pixel. Si nous avons la même image, prise du même angle, et avec la même résolution (ou du moins les mêmes ratios), c'est jouable, mais ici, il va falloir s'adapter, et en temps réel qui plus est. Avec une puissance de calcul qui est celui d'une tablette.

Bien qu'Alaric se sentait le courage de faire ce travail, j'ai préféré faire quelque chose de plus modeste, pour arriver à finir dans les temps.

¹ <http://colorlab.no/news/archive/shida>

Il faut résoudre les problèmes un par un, étape par étape. Je me suis donc concentré sur ce que je dois faire, si possible avec ce que je sais faire. Pour ce qui est des images et de leurs intégrations, j'ai donc pensé à travailler dans l'espace colorimétrique HSV (Hue Saturation Value ; aussi appelé TSV : Teinte Saturation Valeur), plutôt que le RGB. C'est ma seconde solution après avoir testé l'interpolation linéaire.



Espace colorimétrique HSV - " wikipedia.org "

Mon idée, est si je génère une image en RGB depuis des images multi spectrales (on appellera cette image **Is**), que je prends la même scène en photo avec une caméra RGB classique (image **Ib**). Je convertie les deux images en HSV, et de là, je remplace la composante H (teinte) de l'image Ib par celle de l'image Is. L'objectif, est de donner des couleurs plus précises, mais en préservant la saturation et la valeur. J'ai donc fait des tests, mais uniquement depuis deux images toutes les deux du même appareil (caméra RGB), mais chacune avec une luminosité différente. Les résultats ne sont pas particulièrement spectaculaires, mais j'ai déjà développé au moins deux solutions.

Alaric a apporté un outil fort utile, c'est un paquet Tango (développé par Google) que l'on intègre à un projet sous Unity 3D, et qui permet d'afficher des objets virtuels dans le monde réel par la réalité augmentée. C'est un petit pas pour l'homme, mais un grand pas pour notre projet.

Ayant déjà des notions sous Unity, j'ai vite compris ce qu'il fallait faire. On peut créer nos objets dans le monde virtuel à l'échelle du monde réel, et, on peut se déplacer dans l'environnement virtuel depuis le monde réel, à la façon d'un FPS¹, mais plutôt que d'utiliser un clavier et une souris pour se déplacer, nous utiliserons respectivement nos propres déplacements ou du moins ceux de la tablette dans l'espace, et les rotations qu'elle fait pour connaître le point de vue.

Il ne restait plus qu'à se localiser dans l'espace, pour faire fusionner correctement les deux mondes. C'est le travail d'Alaric de me localiser dans l'espace, et par rapport à ces résultats, je calibre l'environnement virtuel sur le réel. J'ai toutefois proposé pour commencer de se localiser en partant d'une position définie par un code QR sur la scène. Il peut y avoir plusieurs balises QR dans la pièce pour démarrer d'où l'on veut.

¹ "First Person Shooter ; désigne un genre de jeu vidéo, appelé en français jeu de tir à la première personne", source wikipedia

Réalité virtuelle :

Cela fait déjà 1 mois et demi que je suis en stage. Les superviseurs demandent à faire une réunion, pour faire le point. Nous avons donc présenté Alaric et moi nos travaux respectifs et communs. C'est alors que le professeur HARDEBERG séduit par nos résultats, demande à revenir sur l'idée d'une visite virtuelle, ainsi que l'ajout de nouvelles possibilités pour la réalité augmentée, à savoir la possibilité de changer le papier peint des murs par exemple, ...

D'un commun accord, nous avons convenu Alaric et moi de prendre des chemins différents. Moi je prends la réalité virtuelle, et Alaric lui continue sur la réalité augmentée. Comme je l'ai précisé avant, nous avons un APK développé spécifiquement pour Tango, l'APK Constructor. Celui ci construit des formes 3D à partir de scans effectués avec la tablette. Le format 3D en sortie est du .PLY¹ ou .OBJ



Dans les deux cas, les formats sont binaires. C'est pourquoi je préfère prendre le fichier .obj, et de là, générer un fichier .PLY ou .STL² ascii, grâce à Blender 3D. Je suis déjà familier de ces deux formats, souvent utilisé pour les imprimantes 3D.

Pour moi, il est plus simple et plus naturel de travailler à partir de fichiers ASCII, que des binaires. D'où ce choix. Ensuite, j'ai dans les deux cas, des facettes que je peux afficher, sous peu que j'écris un petit programme pour le faire. Lors de mon cursus, en M1, j'ai étudié plusieurs modèles de modélisations 3D. J'ai ainsi vu comment modéliser les polyèdres réguliers en OpenGL, ainsi qu'en WebGL (lors d'un projet pédagogique), grâce au modèle B-Rep³.

À petite échelle, il suffit de dessiner les facettes une par une, et d'ajouter des écouteurs, qui vont écouter et interagir sur l'environnement. J'ai donc fait le choix d'exploiter ce savoir, ainsi que différents scripts que j'avais déjà écrit avant.

¹ Polygon File Format, dit aussi Format de Fichier de Polygones

² STereoLithography : stéréographie

³ B-Rep : **B**oundary **R**epresentation ou représentation par les bords en français

Dans le cas des fichiers STL, le fichier contient la description des facettes (triangle), ainsi que la direction de la normale. Le tout, dans une structure qui ressemble au XML. Avantage, il existe déjà des parseurs.

Pour ce qui est du PLY, là aussi c'est intéressant. La structure est différente. Dans la version simple du format, nous avons la liste des coordonnées de points (un point par ligne) qui vont constituer la forme, suivie de la liste des formes, composé de la liste des points dans l'ordre, (un polygone décrit par ligne, et chaque point est décrit par son numéro dans l'ordre croissant des lignes).

J'ai dans un premier temps, fait le choix de faire un rendu en WebGL, accessible en ligne, c'est du JavaScript (JS). De plus, dans le cadre d'un de mes anciens projets, j'avais déjà développé un pseudo moteur de jeu.

Très vite, les limites du rendu sont apparues. Je ne maîtrise pas assez l'illumination, les textures, ... Comme Alaric continuait avec Unity 3D, je suis revenu à ce logiciel, et en plus, il propose entre autre la compilation en WebGL.

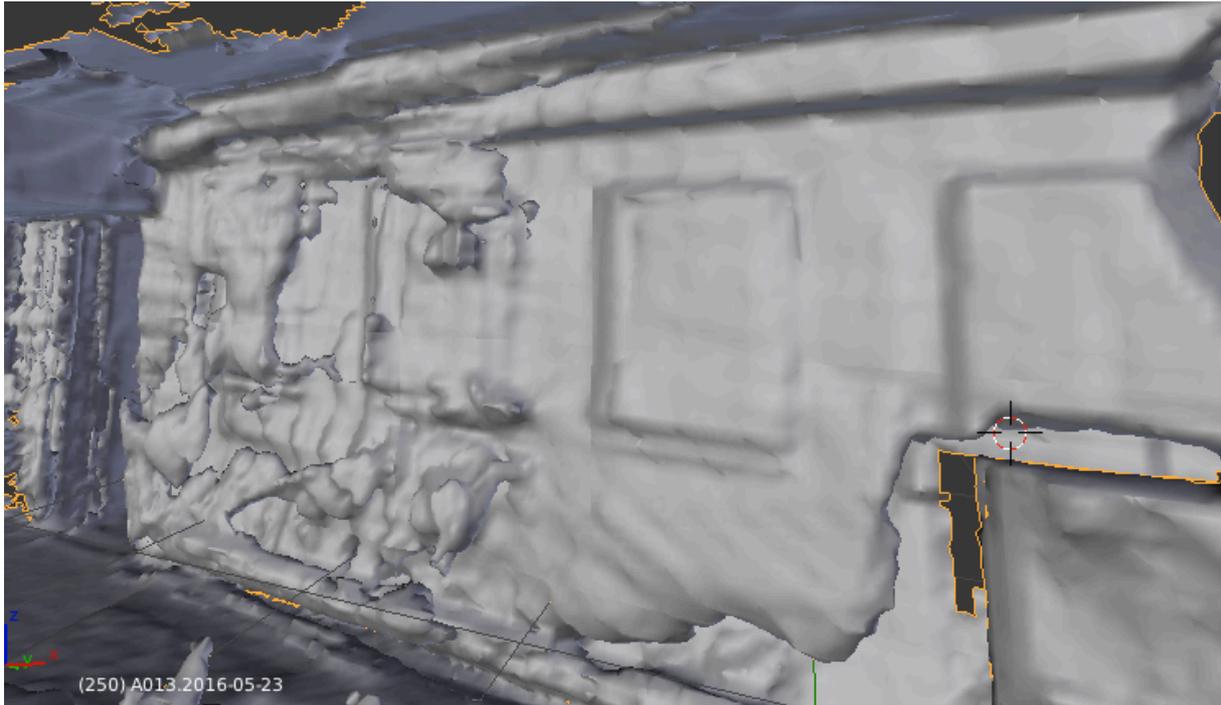
Le fait de revenir à la réalité virtuelle, ne fut au final pas une si mauvaise chose pour moi, en effet, grâce à l'expérience acquise pour la réalité augmentée, je me suis rendu compte que je peux au final combiner ces deux mondes. En effet, si l'on part de ce que j'ai fait au début, je plaçais des œuvres dans un monde virtuel que je superposais pour les calquer sur le monde réel. Du fait que j'ai une image numérique (une maquette) du réel, si je place mes œuvres dans le virtuel, et que j'efface les murs, j'ai mon plan pour la réalité augmentée.

Je m'explique, pour faire de la réalité augmentée, je dois ajouter des objets (panneaux) sur lesquels je vais afficher mon tableau. Ces panneaux, doivent se trouver exactement là où se trouvent les œuvres réelles (car nous n'avons aucun repère sur l'œuvre, nous nous localisons donc par rapport à l'environnement, et non par rapport à ce que l'on a en face de nous). Lorsque je place les œuvres sur la maquette, il me suffit de cacher tout ce qui est au tour. Autre avantage, je peux aussi plus facilement changer la couleur des murs, le papier peint, et représenter plus facilement la timeline (une des idées proposé, la possibilité de comparer un bâtiment à plusieurs moments, suite à une rénovation ou autre).

Ayant déjà une expérience avec Unity 3D, et en particulier dans la réalité virtuelle (entre autre pour mon projet ETIC, je n'ai pas eu beaucoup de mal, car le plus gros du travail était déjà fait. De ce fait, je n'avais plus qu'à intégrer mon maillage dans mon projet, et pour le reste, ça ce fait tout seul. Du coup, je me suis plus concentré sur le maillage.

Constructor est une bonne solution pour le démarrage, et pour avoir un premier aperçu du potentiel du projet. Mais, ce n'est pas encore vraiment au point. L'avantage de la tablette Tango, comme je l'ai dit plus haut, est qu'elle combine deux technologies pour se localiser, la localisation logiciel, optique (IR et RGB) pour faire une modélisation « Shape from motion », et l'odométrie pour confirmer la position grâce à une localisation matériel. Et l'une doit corriger l'autre.

Voici un résultat :



On peut se poser des questions sur mon résultat. Donc pour commencer, comme tout scanner 3D, à ce jour, nous ne pouvons pas scanner ce qui est transparent/brillant. De ce fait, les éclairages au plafond, du fait de leurs diffuseurs en inox, ne peuvent pas être pris. Derrière l'écran TV dont on voit le coin (en bas à droite), c'est trop sombre. Et une des propriétés de la lumière et des couleurs, est que l'on ne voit que les longueurs d'ondes réfléchies par la matière, et le noir est une absence de réflexion. C'est pourquoi, la encore, le système a du mal à évaluer la distance qu'il considère au final trop éloignée. Et pour finir, on distingue sur le côté gauche du maillage des bavures, et autre imperfections. Ces déformations sont atténuées sur le côté droit du maillage.

Ces défauts de modélisations sont dus à la tare qu'il y a sur le matériel. Si l'on va trop vite, la tablette peut avoir des pertes, car elle n'a pas capté certains déplacements. Si l'on va trop doucement, cela va dépendre de la sensibilité des capteurs. Ainsi, le scan prolongé qui permet d'avoir le plus de détails, peut avoir des conséquences négatives sur la qualité du scan, car le calibrage sera perdu. Et pourtant je ne parle que d'un mur.

Il aura fallu passé du temps sur le produit pour faire des tests plus approfondit. Mais j'ai personnellement préféré ne pas m'attarder sur ces détails (qui sont pourtant importants pour la quantification de la qualité des résultats). Je considère que la tablette Tango, est un prototype, donc d'ici quelques semaines ou quelques mois, ce travail aura été obsolète. De plus, le choix de ce produit, est uniquement pour faire une première approche, c'est un objet compact, qui englobe tout en un, mais qui au jours d'aujourd'hui, je ne le considère pas assez aboutit pour y consacrer trop de temps. Dans le futur, c'est probablement d'autres instruments qui seront utilisés. C'est pourquoi, j'ai préféré concentré mon travail sur ce qui m'a semblé plus utile et durable.

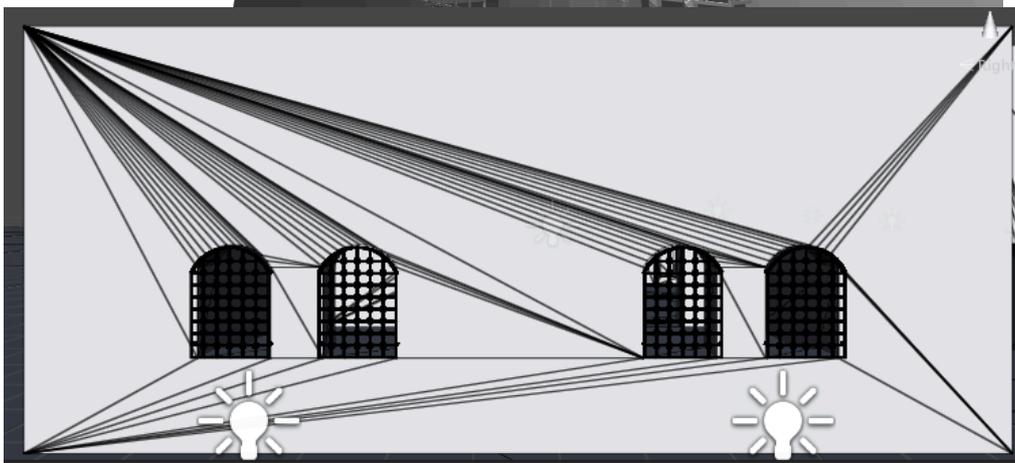
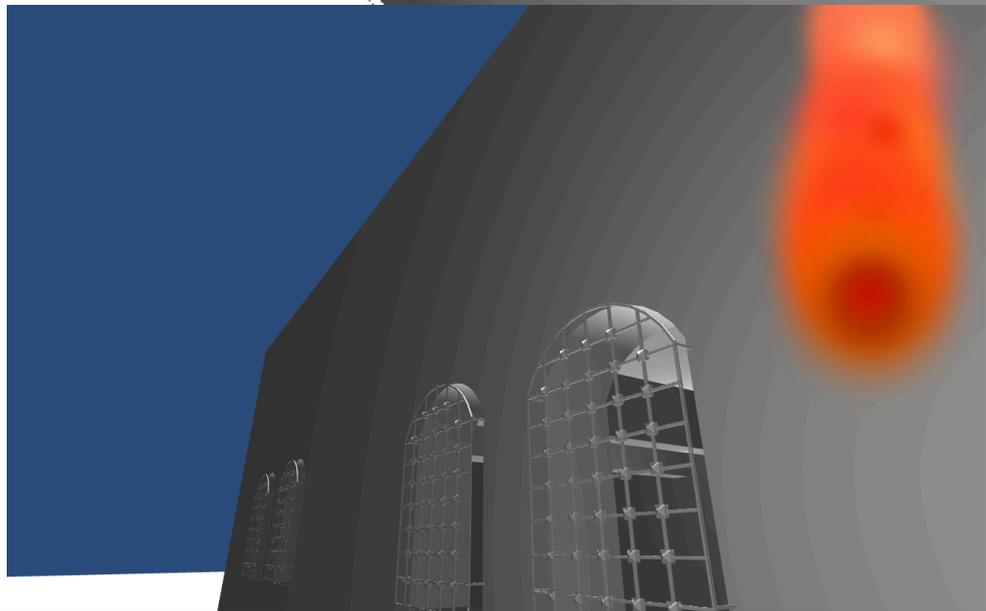
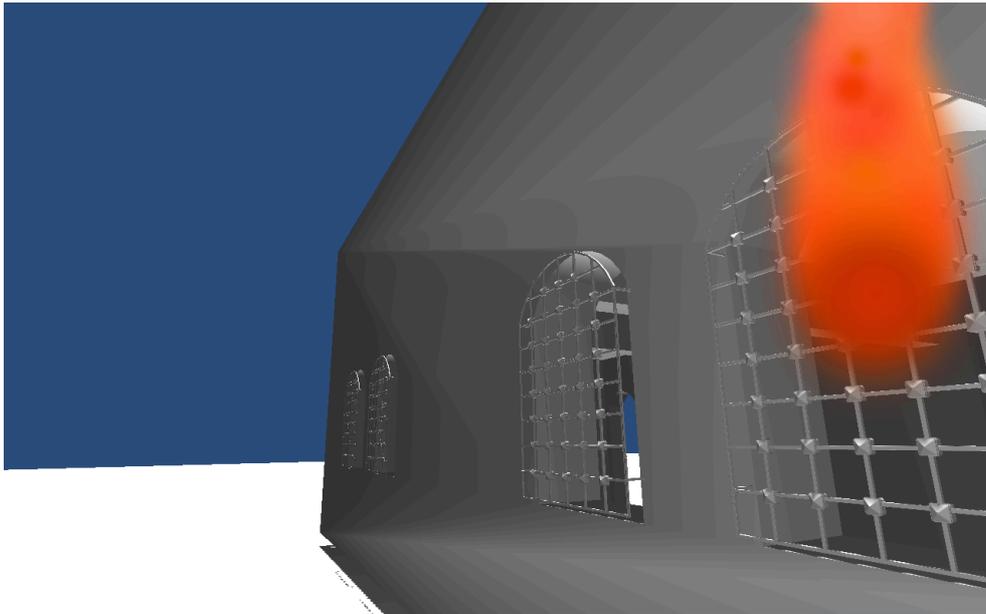
Je me suis donc concentré sur le maillage. Il faut prendre en compte plusieurs paramètres. D'une part, le niveau de détail souhaité, la qualité du maillage, ainsi que le poids du fichier, qui impliquera par la suite le type d'usage. Selon que le produit est destiné à une utilisation en local sur un ordinateur puissant, avec une grosse puissance de calcul, une grosse carte graphique pour un rendu optimal (reflets, ombres, ...), et de la mémoire, qu'il s'agit d'une tour ou d'une simple tablette, que la consultation se fait sur le disque interne de la machine, ou en réseau local (LAN), voir en ligne (WAN).

J'ai donc réalisé ma première démo¹, pour la présenter aux superviseurs. Et de la, leurs demander quelle direction prendre. Voici ma proposition :

J'ai des connaissances en WebGL, en Unity 3D et en Unreal Engine.

- WebGL est une solution facile et compatible sur presque toute les plateformes. Aucune portabilité nécessaire. De plus, elle est facile d'utilisation. Coté fournisseur, il faut avoir un serveur qui hébergera les fichier en ligne (ou du moins sur le réseau visé). Ne nécessite pas de gros matériel. Tout se fait coté client, sous réserve d'avoir une bande passante suffisante pour desservir tout les potentiels clients simultanés. Côté client, il suffit d'avoir un navigateur web sur sa machine, et une connexion internet (ou du moins être sur le même réseau que le serveur). Le rendu dépendra de la bande passante du client, de la compatibilité de son navigateur avec WebGL, et de la puissance de calcul. Utilise peu la carte graphique, du coup, les rendus sont moins impressionnants.
- Unity 3D est une solution multiplateforme. Une fois le travail fini, il est possible de le compiler pour générer les fichiers sur presque toutes les plateformes connues. De manière générale, il propose un rendu correct, assez pour être à la fois supporté par les téléphones et les tablettes qui malgré les progrès dans le domaine, restent peu performants face à un vrai ordinateur. Mais aussi par les ordinateurs les plus aboutis. Inconvenant, on n'a pas la main, sur tout le rendu. Le maillage peut être modifié, le mode d'illumination (pour le rendu des couleurs), est au choix du moteur de rendu. Ainsi, pour le même objet, selon l'angle de la caméra en mouvement, et selon l'angle de la source lumineuse, mon rendu présente des changements de l'illumination sur des surfaces qui ont la même direction de leur normale :

¹ Les démos sont encore en ligne pour une durée indéterminé sur mon serveur : ntnu.moulay.me



Les images sont issues d'un de mes projets personnels en rapport avec mon projet ETIC.

Naturellement, je ne suis pas expert Unity 3D, je ne suis pas infographiste, et il est donc possible que ce défaut de rendu soit gérable.

- Unreal Engine, est connu pour la qualité de ces rendus réalistes. Vu les moyens mis pour le projet (caméra multi spectrale, et bien d'autres outils), c'est pour avoir un rendu parfait, et non approximatif. L'inconvénient, c'est plutôt destiné à des ordinateurs gonflés aux hormones graphiques. A ma connaissance, il est plus orienté consoles de jeux et ordinateurs sous Windows (là où se trouvent les Gamers), et manque de souplesse comparé à Unity 3D. Toutefois, mes connaissances et compétences en cet environnement sont plutôt sommaires (pas nul, mais pas abouti non plus).

Les superviseurs ne veulent pour le moment définir une direction à suivre, ils veulent que je mette en avant un peu plus le potentiel du projet dans un premier temps. On verra par la suite.

J'ai donc fait le choix de continuer avec Unity 3D, entre autre pour si nécessaire, mutualisé mes travaux avec Alaric.

À ce niveau, j'ai fait une petite mise au point, pour savoir ce que j'allais continuer à faire. J'ai donc décidé de gérer l'affichage des images, de simplifier et lisser le maillage et de faciliter la sélection et l'ajout de zone de publication (choisir un tableau, un mur, etc.)

Choisir une image, consiste à choisir les images spectrales qui sont préalablement générées et stockées dans une base de données ou dans un répertoire. L'application y accède à distance, pour les récupérer et les afficher (selon la longueur d'onde souhaitée). Dans le cas de la démo, je ne fais que changer d'image. Mais ceci revient au même au final. Comme je me suis dirigé vers une solution WebGL, j'ai donc choisie de faire un répertoire par œuvre, qui contient toutes les images.

Lorsque l'utilisateur navigue dans la maquette, dès qu'il se rapproche d'une œuvre, il rentre dans la zone qui lui est propre. L'éclairage global alors s'éteint, et c'est un spot dédié qui va éclairer le tableau. Le programme scan le dossier (dédié à l'œuvre), et il est alors possible de choisir l'image à afficher. J'ai commencé à travailler sur le Bump Map (placage de relief), pour mieux gérer les couleurs et le reflet de la lumière, car les couleurs changent selon le point et l'angle de vue de la caméra.

J'ai parallèlement consulté le professeur Mariusz NOWOSTAWSKI, sur les solutions d'optimisation et de simplification du maillage. Mon idée, est de convertir les facettes qui appartiennent à un même plan, en un seul polygone. Naturellement, comme il y a beaucoup de déformations, je considère comme appartenant à un même plan, toutes les facettes qui appartiennent à un volume englobant (qui est un plan avec une épaisseur). Ensuite, je gère un ensemble de point pour créer une enveloppe convexe¹ sur un plan (2D naturellement).

Le professeur Mariusz m'a proposé de contacter un de ces étudiants en stage, qui travaille lui aussi avec la tablette Tango, et qui justement développe un programme sur l'optimisation de surfaces planes. Ce dernier travaille directement depuis le nuage de point (similaire à ce que fait Explorer). Pour générer les surfaces, il compare la normale de deux facettes qui partagent une même arête. Si les normales sont inclinées l'une de l'autre de moins de 6°, il les fusionne. Sinon non.

¹ « L'enveloppe convexe d'un objet ou d'un regroupement d'objets géométriques est l'ensemble convexe le plus petit parmi ceux qui le contiennent. » Source wikipedia

J'ai donc pris contact avec cet étudiant, et lui ai demandé un rendez-vous pour voir une démo. Mais celui ci n'a jamais donné suite à ma demande.

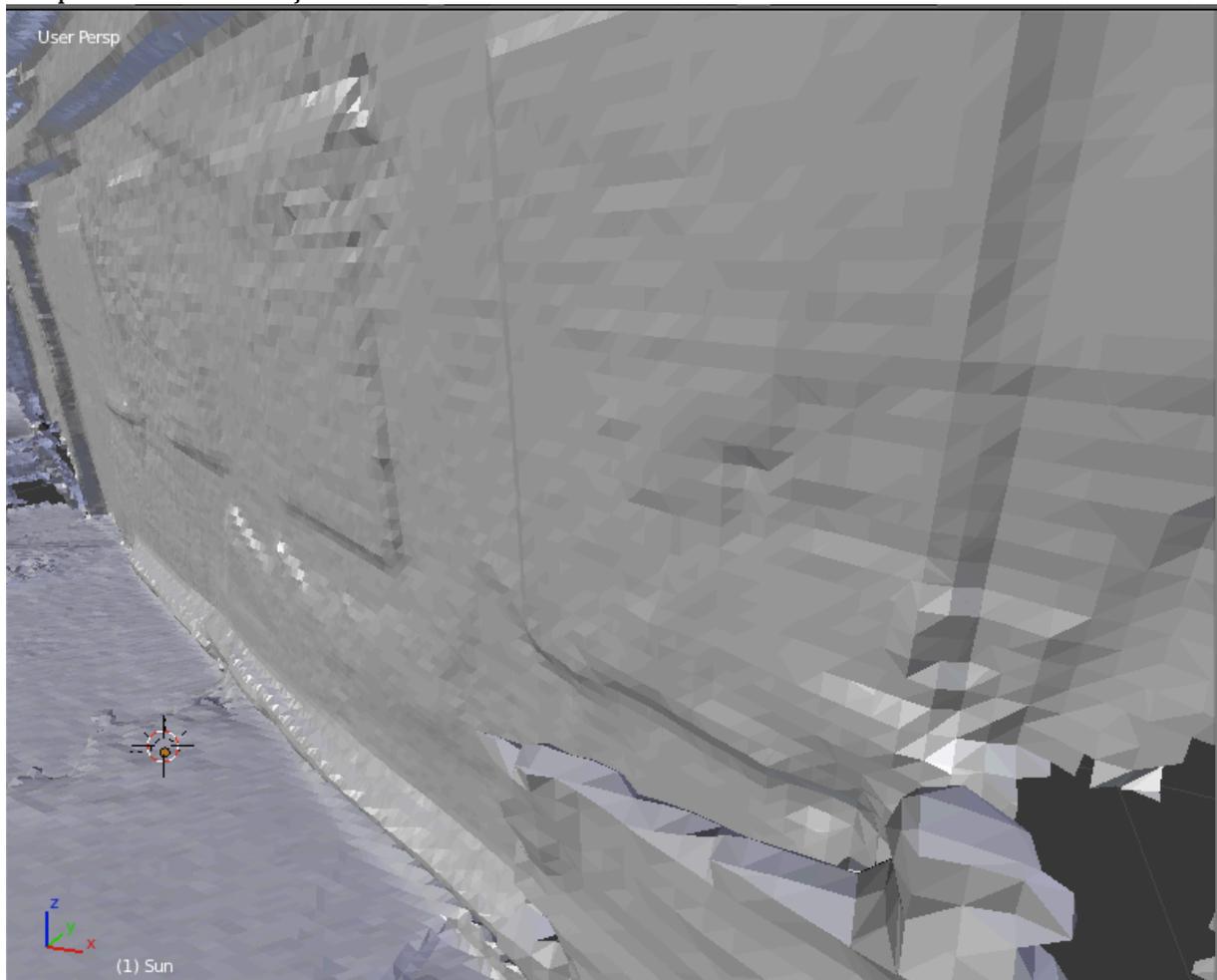
Ayant déjà étudié dans mon cursus les enveloppes convexes, et même réalisé un projet avec différents algorithmes (2D et 3D), j'ai considéré pouvoir être capable de faire cela moi même. De plus, j'en suis déjà à la fin de mes 3 mois de stage.

L'opération se fait en plusieurs étapes :

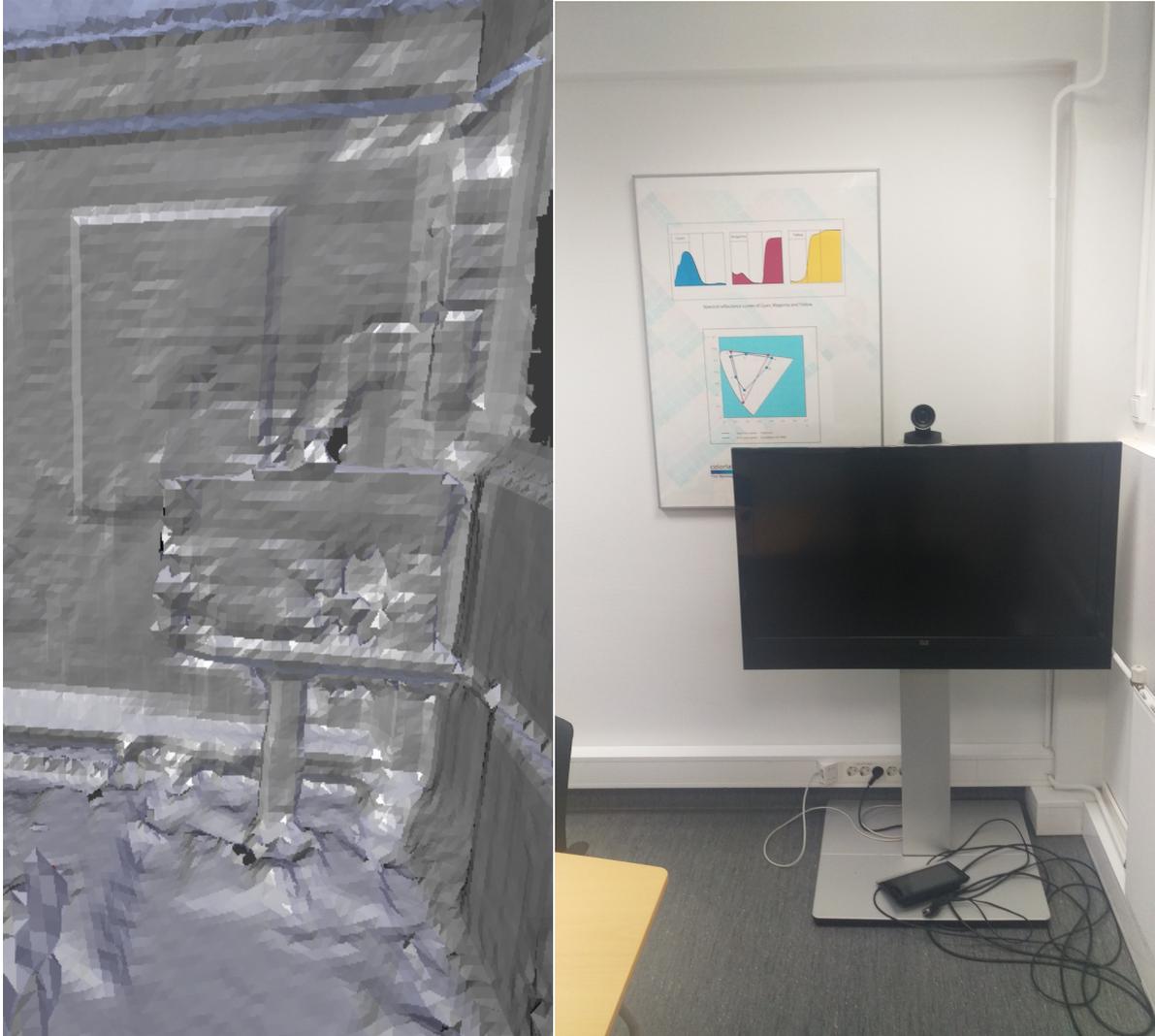
- Je génère les polygones de surface
- Je transforme les polygones en triangle
- J'optimise les triangles jusqu'à obtenir le résultat souhaité.

Pour se faire, je pars d'un maillage .PLY au format ASCII. Je prends aléatoirement une facette. De là, je prends les 3 facettes qui lui sont mitoyennes. De ces facettes, je génère le volume englobant (d'une épaisseur qui équivaut à 1cm dans le monde réel). Ensuite je commence à répandre récursivement mon volume, et au fur et à mesure, je le recentre par rapport à la moyenne des points qu'il contient. Une fois arrivé à la limite, quand je ne peut plus ajouter de points qui seront contenu dans le volume, j'ai une liste de point (grâce à la structure des fichiers .PLY, la liste est déjà faite). Il faut donc projeter les points sur ce nouveau plan, et de là je n'ai plus qu'à lancer mon algorithme d'enveloppe convexe. En théorie c'est simple, mais quand on passe à la pratique... il y a toujours des cas particuliers, et ça prends toujours beaucoup de temps à identifier (lorsque j'y arrive).

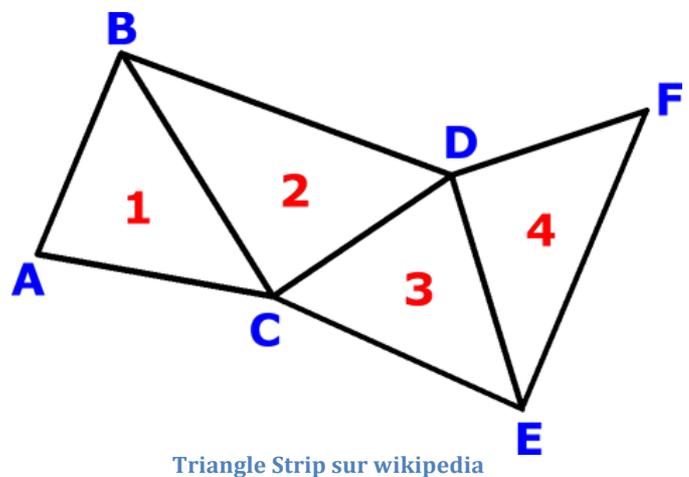
J'avoue qu'il y a quelques petits arrangements fait à la main, pour parvenir à boucler le script. Et on arrive à ça :



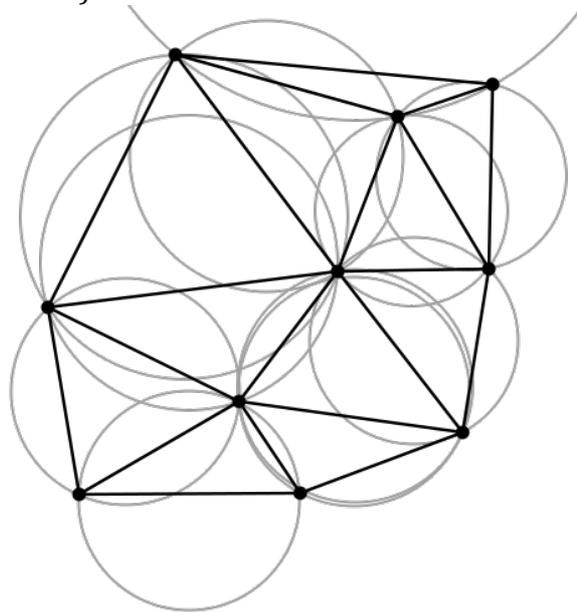
Le résultat n'est pas encore optimal, mais ceci est une des conséquences du niveau de détail. Si l'on veut un niveau de détail élevé, peu de nuisances (ou déformations, imperfections) seront supprimés. Et réciproquement.



Pour ce qui est de l'enveloppe convexe, j'ai préféré rester sur ce qui est le plus souvent utilisé, j'ai utilisé le parcours de Graham. Généralement (et il est même conseillé), on évite les polygones (de plus de 3 points je veux dire) en modélisation 3D, pour éviter qu'un point qui appartient à la forme, mais qui n'appartient pas au même plan que les autres points cause erreurs d'affichage. C'est pourquoi, je reviens à des facettes (triangulaires). Ainsi, je découpe les polygones grâce à un algorithme « triangle strip »



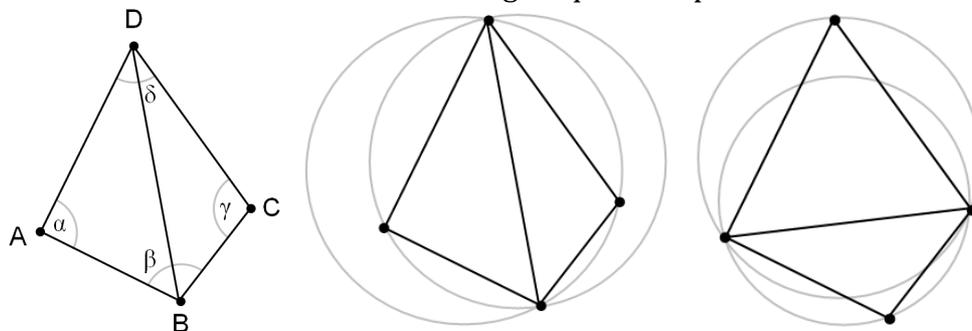
L'inconvénient de ce découpage, est qu'il peut donner des triangles très petit à coté de triangles de plus grandes tailles. C'est pourquoi, il est nécessaire d'appliquer l'algorithme de triangulation de Delaunay (que j'avais utilisé aussi lors de mon ancien projet d'enveloppe convexe).



Triangulation de Delaunay sur wikipedia

Pour rappel, voici un exemple de ce que fait cet algorithme :

Soit un cercle qui passe par les 3 sommets d'un triangle, si un autre triangle est contenu à l'intérieur de ce cercle, on supprime l'arête commune aux deux triangles et on en crée une autre entre les autres sommets des triangles qui n'ont pas été touchés.



A ce niveau, le travail n'est pas encore fini. En effet, comme je l'ai montré pour le rendu avec Unity 3D, le fait d'avoir des facettes trop longues, et trop grandes, selon le mode d'illumination local, avec les ombres, sur un carré par exemple composé de deux facettes, on peut se trouver avec deux couleurs différentes, car chacune des deux facettes a une moyenne d'illumination différente. Ce qui donne le problème vu avec Unity 3D.

Pour un rendu rapide, l'illumination est calculée uniquement par rapport aux sommets, et non pas par rapport aux pixels. Pour avoir un meilleur contrôle sur ce qui peut devenir un défaut d'affichage, je préfère contrôler la longueur des arêtes. Donc si sur un triangle, une des arêtes est au moins 2 fois plus longue que l'une des deux autres, il y a subdivision. J'utilise une solution SDS (Sub-Division of Surface), mais à ma façon, destinée à mes besoins.

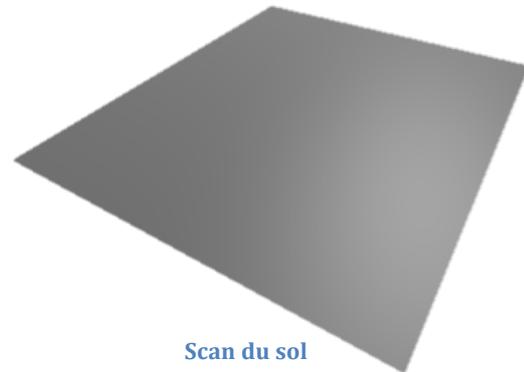
L'autre objectif est aussi d'alléger le poids du maillage. La mission est accomplie grâce ce que j'ai fait. En effet, je suis parvenu à avoir le scan d'un mur et de quelques objets qui

fait le même poids (en Ko naturellement) que celui de la pièce entière optimisé. L'optimisation ne peut pas être réellement quantifiée, car elle est très efficace sur les surfaces planes, mais demande beaucoup de temps de calcul pour les autres parties plus complexes.

Exemple sur une forme simple :

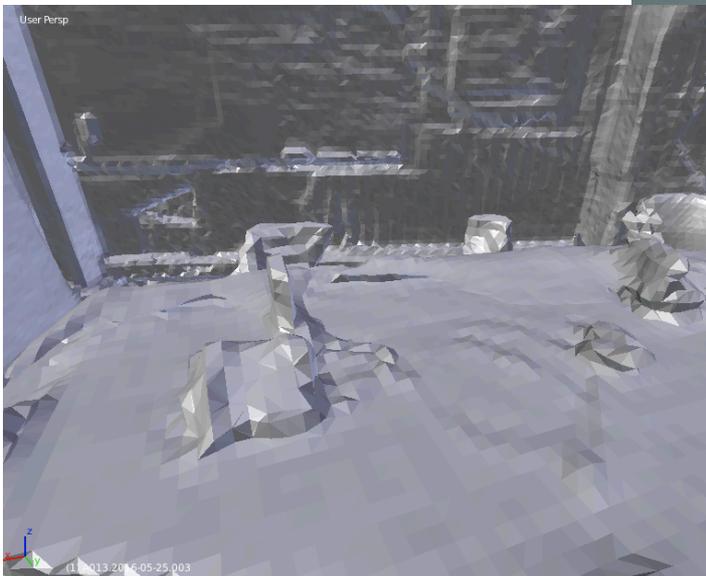
~20Mo avant optimisation

Moins de 500Ko après optimisation (plus que deux facettes, et 4 points)



Au final, on obtient un maillage plus ou moins uniforme. Il reste encore plusieurs défauts à rectifier, tel que la forme qui n'est pas un polyèdre fermé, l'application de texture est un point complexe si l'on veut être fidèle à l'original (et à quel point souhaitons l'être). Comment gérer les mises à jours, car chaque scan génère un maillage différent, ce qui implique une forme différente, qui ne peut pas se fusionner avec une autre !

Mais en 3 mois, on ne peut pas tout faire. Surtout que la grosse difficulté, est que je n'avais aucun (où plutôt disons presque aucun) encadrement, ou support. Pour chaque chose, je devais me référer à mes acquis, ou faire des recherches et de m'auto-former (souvent dans l'urgence).



Futurs directions :

Il reste encore beaucoup à faire ou à refaire dans ce projet. J'ai ouvert une voie. Nous pouvons voir le potentiel si l'on veut poursuivre le développement. J'ai plusieurs idées que j'ai suggérées pour la suite, à savoir :

- Tester le scan avec d'autres solutions du marché (comme Kinect par exemple). J'ai aussi proposé et commencé à développer et tester le scan basé sur le « shape from shading ».
- L'acquisition multi spectrale d'objet 3D, comme une sculpture par exemple. Il sera possible de prendre la texture d'un objet avec une camera multi spectrale.
- Pour optimiser la taille du fichier (maillage), il serait intéressant d'utiliser une librairie d'objets, pour les plus récurrents. Exemple les chaises, les poignées de tiroirs, ... Les avantages en serait multiple :
 - L'objet sera modélisé qu'une seule fois, stocké en mémoire qu'une fois, mais serait appelé et affiché plusieurs fois.
 - La modélisation de cet objet peut être parfaite si elle est faite manuellement. L'avantage à cela, est mis à part la qualité de son rendu, il sera déjà optimisé.
 - Les objets tels que les chaises par exemple, peuvent comporter des parties réfléchissantes (métalliques, verni, etc.), ou la matière est trop sombre, mate. Ils seront donc incomplets. D'où l'intérêt d'une modélisation spécifique et indépendante.

Pour se faire, il faut dans un premier temps identifier les objets de la scène manuellement, et les remplacer. Puis si possible automatisé l'opération.

- Je manque cruellement de maîtrise des moteurs de rendu que j'ai utilisé. C'est indéniable. En même temps, on ne peut pas être bon en tout, et surtout tout maîtrisé en si peu de temps. C'est pourquoi, je pense qu'il est peut être utile de développer un moteur de rendu spécifique, qui sera optimisé à nos besoins. Il est toutefois possible que ce soit faisable avec Unity ou Unreal. Mais je ne les pratique pas assez pour les maîtriser.
- Les images RGB depuis le multi spectral sont générées au préalable. Ainsi, l'utilisateur peut choisir les spectres un par un, mais ne peut pas choisir lui même les longueurs d'ondes pour construire son image (cela lui prendra trop de temps). Avec le progrès de la technologie, et de l'optimisation de temps de calcul, cela pourra être envisageable dans l'avenir.

Autre :

Mon stage ne fut pas qu'un travail isolé dans le laboratoire. En effet, j'ai pu assister à plusieurs présentations de travaux dans le domaine de l'image. Des chercheurs du monde entier sont venus présenter leurs travaux (dont des français, et entre autre de l'uB). J'ai découvert un vaste monde de savoir dans le domaine des couleurs, comme la reproduction, l'identification, la perception... beaucoup de ces recherches sont très intéressantes. Mais il ne fallait se perdre dans toutes ces opportunités et idées, car le stage ne dure que 3 mois. Et en si peu de temps, il m'a semblé audacieux de ce sentir capable d'intégrer toutes ces solutions à mon projet. Même si toutes ces personnes sont ouvertes et motivées à nous aider Alaric et moi, mais à force, très vite, lorsque chacun ajoute son petit plus, ça se transforme en un mélange d'idée qui ne tien plus la route.

Conclusion :

Pour conclure, ce fut un bon projet, bien qu'assez ambitieux. Je l'ai fini (difficilement en autre par manque de temps), même s'il n'en est qu'à son début, car ce projet n'est qu'une voie de départ pour de futurs développements d'avenir.

Malgré quelques désaccords sur certains choix avec Alaric, nous arrivions toujours à trouver un terrain d'entente. Les changements dans le projet qui surviennent au cours du développement sont souvent effrayants. Mais je pense avoir réussi malgré tout à toujours répondre aux exigences de mes superviseurs.

L'autonomie dans un projet est bien, mais il faut assumer ces choix, et prendre une mauvaise direction, peut conduire à l'échec. C'est pourquoi, j'ai préféré ne pas me lancer dans de trop grandes choses d'un coup, mais plutôt me fixer des objectifs intermédiaires, pour toujours avoir quelque chose au final, mais surtout se rendre compte au plus tôt que l'on se trouve sur la mauvaise voie.

Quant aux conditions de stage, rien de particulier à dire. Mis à part mon ordinateur, que j'ai utilisé, pour le reste, j'avais toujours accès aux équipements dont j'avais besoin. L'absence d'encadrement, est parfois difficile, car on part d'une page vierge, et on doit réaliser un travail en se lançant dans l'inconnu. Mais ça permet d'acquérir plus d'autonomie.

La visite des professeurs de l'uB (professeur Alamin MANSOURI et Jean Batiste THOMAS) était bien, bien que tardive. En effet leurs conseils et opinions du projet, en absence d'encadreur furent réconfortants, mais en même temps, ils sont arrivés à la fin du stage, et les cartes étaient déjà jouées. Heureusement, j'ai joué les bonnes cartes (ou du moins, les moins mauvaises).

L'absence de rémunération, et de bourse pour le séjour, sont difficiles à assumer. Vivre plusieurs mois dans l'un des pays les plus riches et les plus chers au monde, n'est pas chose aisée. De retour en France, on se sent plus riche, dans plusieurs sens du terme, mais surtout riche d'une expérience et de connaissance.

Dans le cadre d'une thèse, ou de recherche, les laboratoires sont très bien équipés et le déroulement de mon stage dans un tel environnement était très appréciable. Il y a régulièrement des conférences pour présenter des nouvelles découvertes... La barrière de la langue est surmontable dans la majorité des cas (la majorité des gens parle l'anglais, et avec le temps on commence à comprendre le norvégien).

Ainsi, je ne suis pas déçu de mon choix. C'est un projet que j'ai choisi de faire, car je travaille déjà sur ce type de réalisation. Je compte le poursuivre en privé ou dans un cadre professionnel. C'est à voir. En tout cas, se fut pour moi une opportunité de concilier plusieurs points, d'une part la vie et la découverte de la Scandinavie, d'autre part la validation d'un cursus universitaire, mais surtout, l'opportunité de travailler sur un projet qui me tenait à cœur dans un cadre professionnel de renommée.

Contact et bibliographie

Projet réalisé par

Karim MOULAY – karim@moulay.me

En collaboration avec

Alaric COTTIN – alaric_cottin@etu.u-bourgogne.fr

Supervisé par

Jon Yngve HARDEBERG - jon.hardeberg@ntnu.no

&

Sony GEORGE - sony.george@ntnu.no

Organisme d'accueil

ColorLab NTNU Gjøvik (Norvège)

Sources :

wikipedia.org

developer.google.com

docs.unity3d.com

Image de couverture : daum.net

Pensez à consulter les images en annexe



The Norwegian
Colour and Visual Computing
Laboratory



NTNU

Norwegian University of
Science and Technology

UBFC



UNIVERSITÉ
BOURGOGNE FRANCHE-COMTÉ