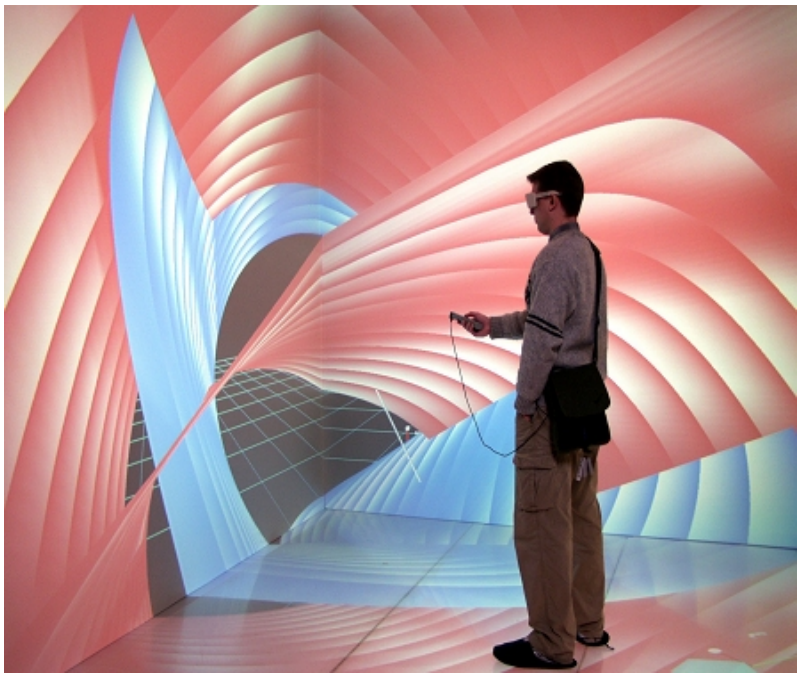
NTNU
Norwegian University of Science and Technology

UBFC
UNIVERSITÉ BOURGOGNE FRANCHE-COMTÉ

Project VMsIVR

Visualisation Multi spectrale Image in a Virtual Room



Master thesis report I2A 2015/2016
University of Burgundy
Directed by Karim MOULAY

Hosting Laboratory ColorLab NTNU – at Gjøvik
Supervisor : Pr. Jon Yngve HARDEBERG
Co-Supervisor : Sony GEORGE

Introduction :

Objective:

Develop a solution for the acquisition of parts / volumes / buildings from a shelf, and model the 3D environment. Use these 3D models for VR-virtual reality visits, or in AR-augmented reality.
Some 2D textures (paintings / art works ...) will be highlighted in particular. And for that, the works will be done with a multi-spectral camera, and their representations in the virtual environment will be replaced with photos from technical recombination of spectral images. The user can thus visualize the works according to their choice in multi-spectral recombinant RGB spectrum or spectrum by spectrum.

Start date : April 4th 2016
Duration : 13 weeks (3 month)
Place : ColorLab - NTNU - Gjøvik (Norway)

Speakers :

Pr. Jon HARDEBERG	: Supervisor
Sony GEORGE	: Co-Supervisor
Mariusz NOWOSTAWSKI	: Ass. Prof NTNU and speaker for TANGO PROJECT technology
Alaric COTTIN	: Intern from university of BURGUNDY (uB)

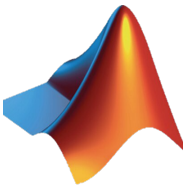
Tools and instruments avalaible :

Multi-spectral camera : Pixelteq Spectrocam --> with 8 filters wheel + laptop with software, and licence the camera driver.
Spectrometer : Minolta CS1000
Pad : Google Project Tango
Computer : personal computer
Software used :

- Matlab 2015
- Unity 3D game engine
- Ocean Thin Film – Software provided with the camera
- Android studio
- Blender 3D
- Web browser (to run the WebGL script)
- Text editor

Programming language :

- Matlab
- JavasScript (JS)/HTML5
- C/C++
- Language for Android



Project description before arriving:

The aim of this project consist of 3D reconstruction from non-calibrated images and videos and differents colours (i.e monochromatic, gray level, RGB(a), multi-spectral, etc.).
Thus I start from heterogeneous elements, and try to find reference marks, to be able to generate the 3D model reconstruction. It's a great challenge. But it totally fits to my expectation to work on such a project as well as perfecting my knowledge in multi spectral. And thus, carried by the the motivation I launch out in the adventure. The multi spectral camera has the advantage of collecting nuances that a classical RGB camera does not necessarily pick up or at least not correctly (like the yellow color for example, or what is hidden in IR (Red Infra) and UV (Ultra-violet). Thus, we can better collect and highlight nuances of hidden colors.

Arrived in the field:

First meeting with the supervisors. Discussed, debate and launching of the project.
The project consists in reproducing a building in 3D from images. The buildings can be museums, palaces or any other type of construction.
For the most invaluable objects, and most remarkable, acquisitions will be done with a multi-spectral camera, to better bring out the details.
The aim is to achieve a 3D scanner, and use the model to make the virtual tour, or even to virtual reality.
The works in multi-spectral can be observed on the wavelength of our choice (or at least according to what will propose the database we will provide).
A timeline can be considered, to visit the place at different times (e.g wallpaper has changed, some renovations ...).
On the final version of this project, it is even possible to interactively change the colors, and simulate future renovations.
The equipment available is a multispectral camera and other equipment that are pending.

It was proposed to me to work in collaboration with another trainee, Alaric COTTIN, a classmate. I suggested in this case, that our work remains quite distinct so that, each one can progress on its task at its rhythm.
The integration of our work, will be done progressively, according to an established planning in concert and by mutual agreement.
Supervisors have not give us specific instructions, so they let us total freedom of action, as well as the choice of the tools, programming language etc .....
Meeting with Alaric. He introduced me to the place and equipment available (as well as the city and and the places of conveniences).
We discussed about our respective goals. In my case, I wanted to take the opportunity during this internship to work the more
On the multi-spectral imaging (or even hyper-spectral if possible).
Our spots are therefore easily defined, I started working on acquisition and processing of multi-spectral image, and Alaric on 3D reconstruction.
I had already made research and had prepared documentation on this subject, in preparation for my internship, which I placed at the disposal.

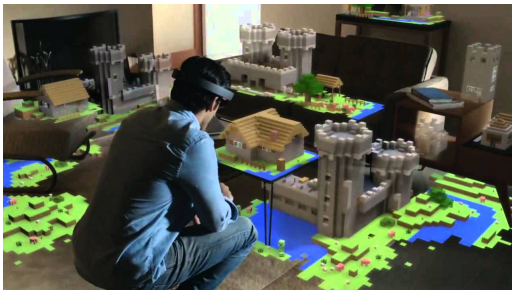
State of the art

The continuous evolutions on this project made difficult research to find out what exists so far or what is made. But in this realm, several projects that approximates what I have done, are ongoing or so have seen or will emerge. In my case, it is only a variant among others.

However, I must mention a few:

- **Microsoft - HoloLens** :

  AR (Augmented Reality) project, with interaction in a virtual world.



- **Google - Project Tango** :

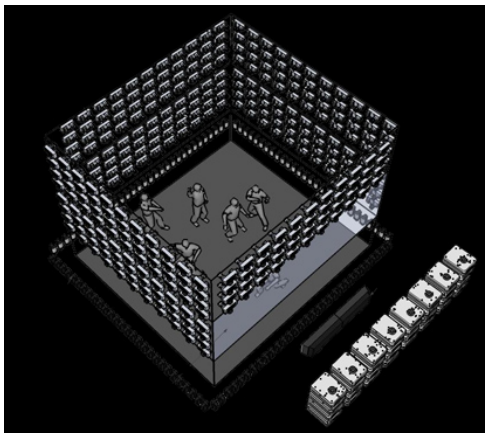
  Offers several things, including among other augmented reality to see the works of art

- **Sony – 3D holographic Football**

In 2010, Sony promises to offer the World Cup in 3D Holographic directly in major capitals of the world (in big stadiums), if Japan got the organization of the World Cup in 2022. I am personally not a football fan, but I have really wanted to see this project through, and emerge.

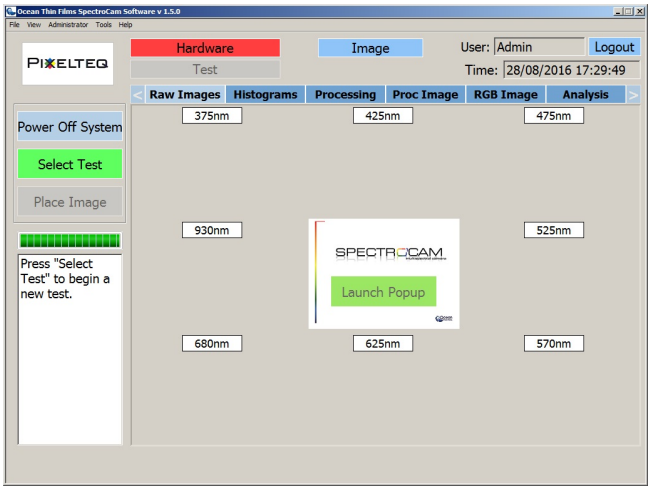
- **Carnegie Mellon university – Virtualizing Engine** (Pittsburgh, Pennsylvania, United States)

1,000 cameras to film directly, every angle some combat sports.
From these images, the whole scene is reproduced in 3D in real time. The result can be use for different purposes.



Multi spectrale :

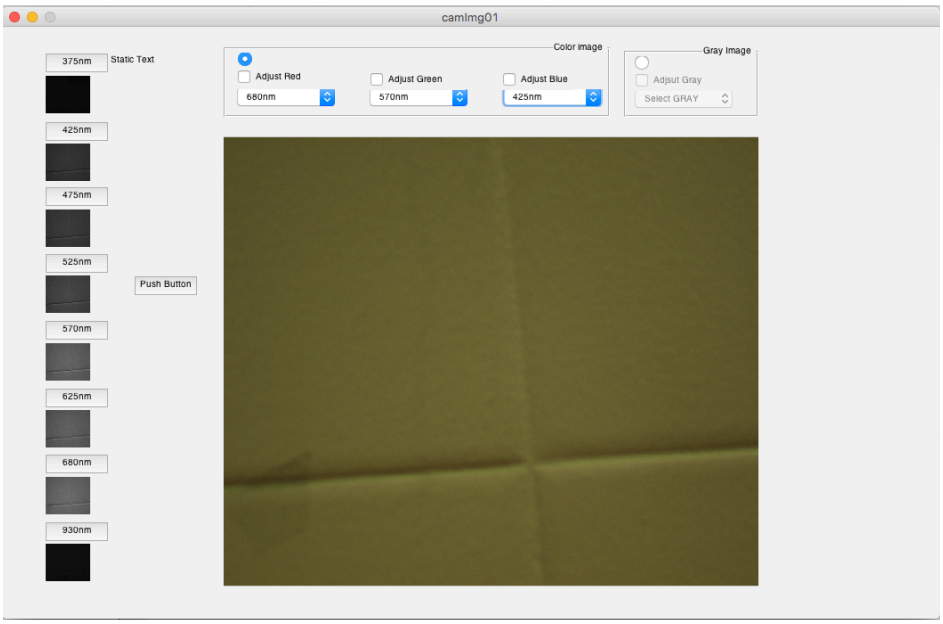
We have had a multi-spectral camera "Pixelteq Spectrocam". With 8 filters wheels, ranging from 370 to 930nm (wavelength in nanometers). A laptop dedicated to the camera is provided as the driver that the camera software is installed there, with the license. Alaric had arrived three weeks earlier, so he introduced me the hardware and software "Ocean Thin Film."



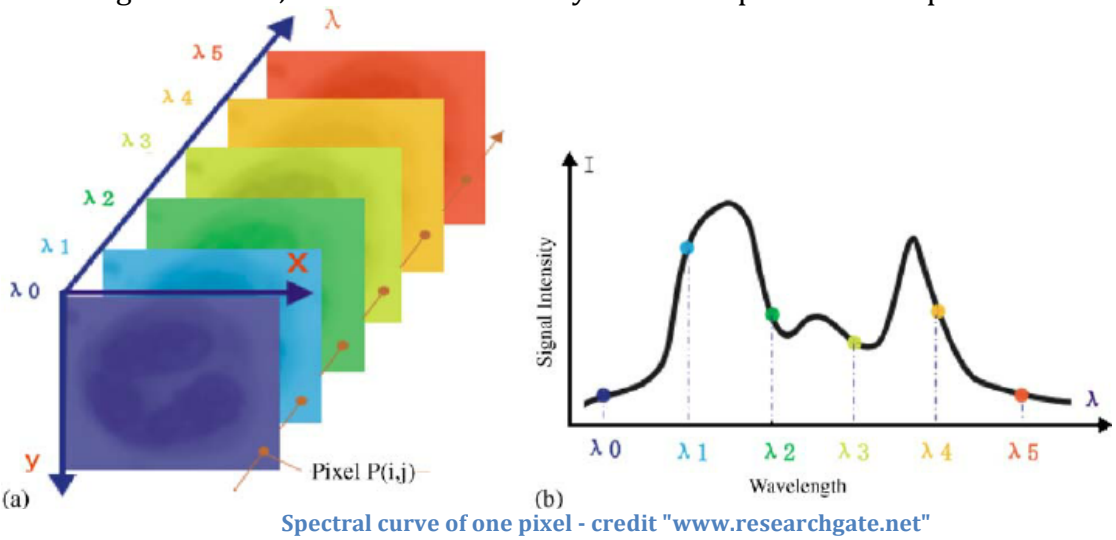
I have found the interface, familiarize myself with the features, the product, acquisitions, ...

First observation, image manipulations can be done only in real-time. The camera must be connected and operating. Otherwise, it does not work! It was a little disturbing to me because I might need to do tests with me, in case I have an idea comes into my head. It is possible to save the acquisitions, standard image format. An grayscale image (Graylevel) / wavelength. So I developed a Matlab program that will offer all the features (or at least as much as possible) offered by the manufacturer of the software for use outside connections.

The features are the RGB image reconstruction, by wavelength combination for each component, the logical operations between two images of two wavelengths, ...



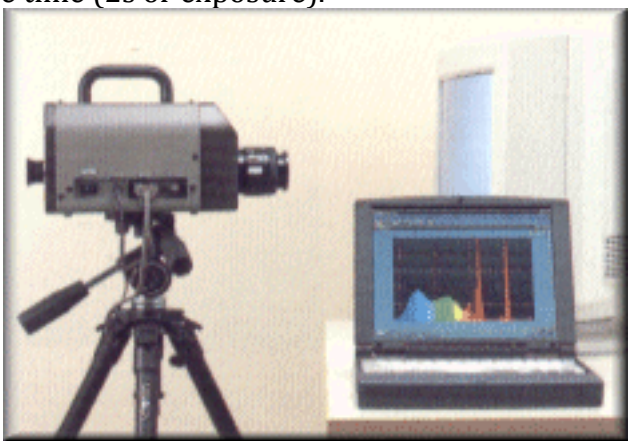
By writing the program, I started to add options, such as histogram equalization because of extreme (high and low frequencies, IR and UV), the images are very dark, black sometimes. I also started to think about a solution to rebuild an RGB image from wavelengths of our choice.
To do this, I was inspired by the principle of curves, so draw the spectral curve of each pixel (or at least calculate the wavelength of my choice), and thus, according to the wavelength selected, have the the intensity of the component in the pixel.



Spectral curve of one pixel - credit "www.researchgate.net"

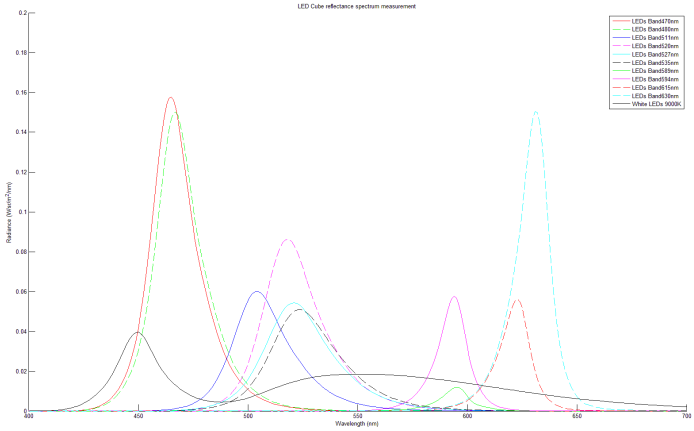
The principle is simple, I use the wavelengths and the intensities like check-point of the curve, and thanks to that, I build the spectral curve of the pixel. From there, I recover the intensity of a pixel over a wavelength of my choice.
With this intention, one can choose what type of curve to use, and a degree (or order).
During my course, I studied several types of modeling of different curves.
In my research, I saw that one could use the curves of LAGRANGE. Not to limit me, I made the choice to make curves NURBS (Non-Uniform Rational Basics Splines) cubic.
Why this choice? Cubic, because during my university course, it was shown to me that it is useless to use curves with a degree higher than 3 (of order 4). The difference will not be significant, but especially the complexity of the computing time will be exponentially high. When with the NURBS, they are the general case of the curves. From there, one can deduce B-Splines, that they even give Bezier curves, Hermit, and many others. Thus, thanks to this programming, I will be able easily to test and compare the quality of curves (time of construction, effectiveness, etc).
As the filters installed on the camera do not have a regular step (nonuniform sequence), in my case, the camera had 8 filters of which the wavelengths are - 375nm, 425nm, 475nm, 525nm, 570nm, 625nm, 680nm and 930nm), the use of the NURBS was an obviousness.
So if I want blue on the wavelength of 435nm for example, I must above all be an inverse function (through a solver that I wrote, the computation is done upstream.), which with a length wave (which can be called X), it gives me the value which will be called T, which is the position of the point on the curve. Indeed, NURBS gives us a time T (which corresponds to displacement of the point on the curve), homogeneous coordinates X and Y (it will be said to result Y is the intensity axis). So I use the invNURBS (X) function to find T, and from there the NURBS (T) function for Y. I applied the operation for each pixel on each of the three RGB components.

The program I wrote works well, but ... its very time consuming. On my PC, the program is written in Matlab, with little code optimization (among other display adding to monitor the script for debug and other), for an image size of 8 × 2456 spectra 2058, the calculation time is estimated at 3 full days. For this example, I tested 3 Bezier curve per pixel ; a cubic curve by component (no continuity between the curves).
Of course when I speak curve, I speak only from the point of intensity that interests me, I do not draw any curve.
Use palette of colors between different may reduce the computation time. But departed from this, I've realized that it is impossible to offer the user the choice of wavelengths for real-time color image ! At least not that way.
In terms of the acquisition, especially when tending towards the ends (IR and UV), the images are very dark.
Upon acquisition, we just choose the exposure time for each filter (wavelength).
It was therefore necessary to extend the exposure time for these two filters. With no mark, I merely choose the maximum exposure time (2s or exposure).



Minolta CS1000 Spectrometer – "www.scientec.fr"

Meanwhile, I met a engineer Mouhamed BOUZIDI graduated in electronic at University of Burgundy, who developed 2 LED matrix projectors, confrolled by a computer.
It is possible with these projectors, LED light to choose what and what LED off and their degree of intensity (as does a TV OLED panel). So I started to see my interest in his work.
I discovered a very useful equipment, spectrometer, "Minolta CS1000". This device allows to know the spectral curve of a point.



I noticed that the spectral curve of the artificial light is not uniform. It is accentuated on certain frequencies and in other hollow. It is a mixture of three components, but leaves troughs between the peaks.

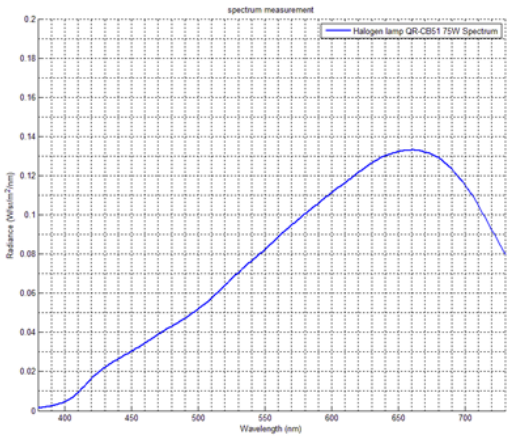


Figure: Plot of the Halogen lamp QR-CB51 75W Spectra

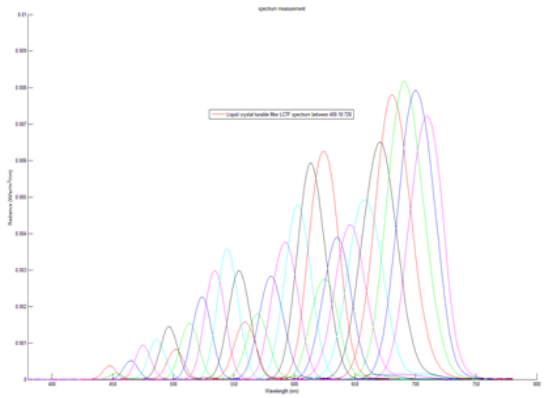


Figure: LCTF bands using the Halogen lamp QR-CB51 75W as a light source.

Extract of Mohamed BOUZIDI's works

So I bought the IR and UV LEDs to add to his projector, and I offered to write him a script that will integrate its light regulator.
The goal is either to obtain uniform spectral curve (a horizontal line, almost similar to that of the sun) either for each wavelength selected, make the mix of colors and adjust the intensity for each wavelength of the desired light intensity.
The multi-spectral camera provided, allows the filter acquisitions filter (or wavelength by wavelength) automatically or manually.
The interest of this is not having to randomly choose the exposure time to make acquisitions, but have unique exposure time, the fact that the light intensity for each wavelength is the same. Nevertheless, the exposure time remains an ambiguous item. The images taken wavelength by wavelength, are grayscale.
A too long exposure time, saturate the colors, and a too short exposure, erase the darkest colors. There are solutions to reduce this gap, such as the HDR (High Dynamic Range Imaging- high-dynamic-range imaging). But things can quickly become complex. The multi-spectral imaging is only part of my project. So I made a choice, the easiest to start. I decided to make color palettes that I added to the scene at the time of acquisition, to have a reference of each color, and I standardizes pixels compared to this reference that I have chosen. For red or white, like all other colors, which can tell what true white or true red? Everything is relative, and it is a perspective that depends on each one. This is the reason why I use a reference. And it is this reference which decides for me.

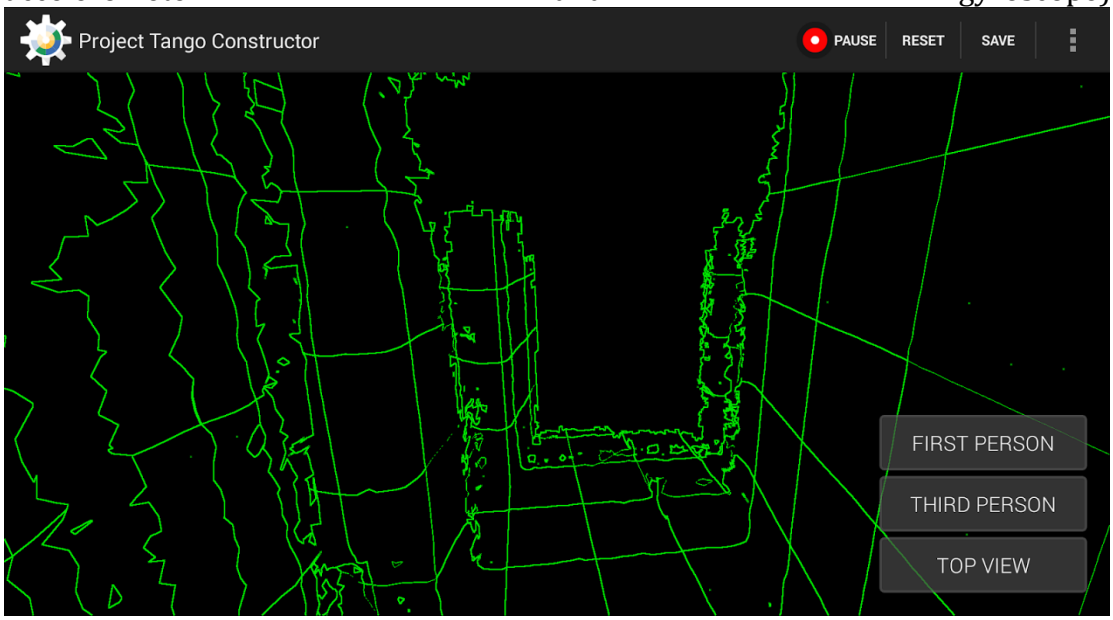
Project Tango :

We received new equipment, the tablet "Project Tango" Google, in the second week after the beginning of my internship. I knew the product through forums and articles I've read, but this is the first time I had one in my hands.

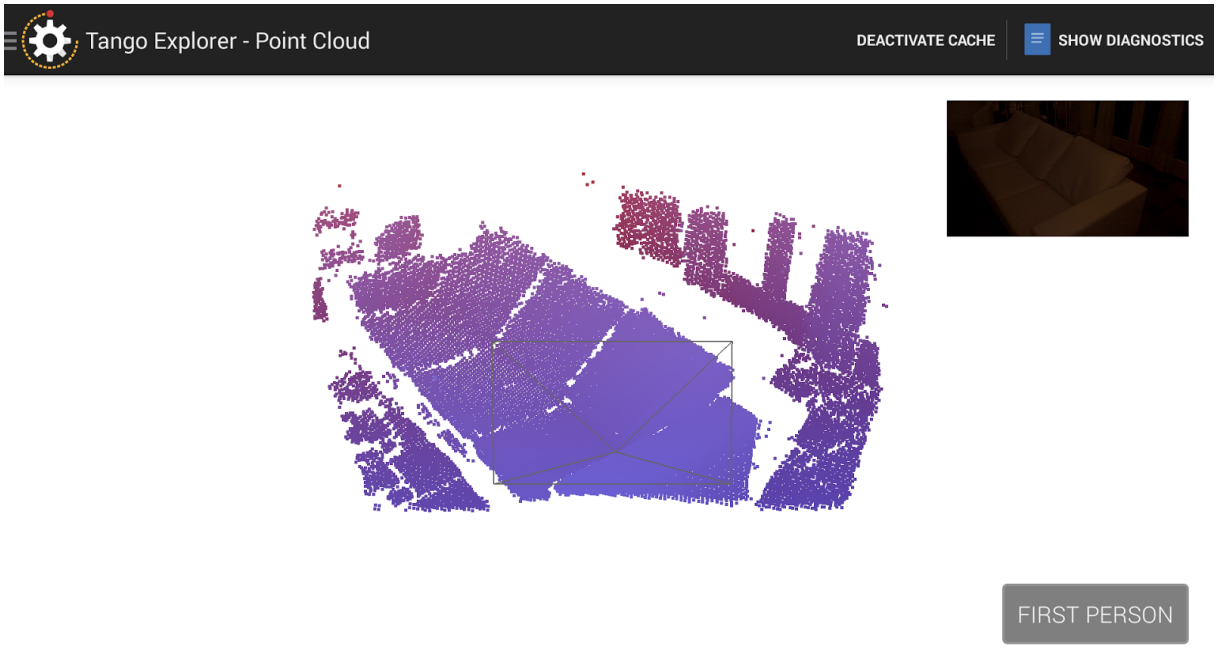
We had instructions as to use the pad in priority to make augmented reality.
I had to stop everything I was doing on imaging, and therefore focus on the tablet. The aim is that the user who moves into an art gallery for example, can see actually work increased through the tablet, it can view it on the wavelength of choice, and benefit all options previously provided for virtual reality.
Although in theory, I know the principle, neither I nor Alaric knew how to make AR (augmented reality). There's a first time for everything ! So I thought I could continue working with my basics and my preparation for this course, I find myself launching into space.
As I already mentioned above, supervisors have given me great autonomy in terms of both schedules and rythm of work, choices, and other tools. I did not have to go to daily report (or even weekly). I just had to notify them via email on what I was working and go back any significant progress. Such autonomy and freedom is much appreciated.
But in every situation there is a flip side, I'm alone, and supervisors do not have time for me, at least no more than respond to one question in writing. That said, we must start. And first thing to do is to discover the tablet.

Nowaday, a tablet, it is not something mysterious even less for a computer scientist that I am. But the tablet I just call Tango, is not like the others. Just seeing the photo of the back of the device, to go already that she is different. I will not return in the product technical specification, because the provider describes very well on the website (store.google.com -> Project Tango).
Also, take into account that the product is in development phase, so it is not finished yet. Several tools used or developed during the course, may be obsolete, especially after the advertising from Google I / O stream in May. Many innovations, especially the apps, not yet made public.
To start, we had become familiar with the product, and discover its capabilities as ... We have worked on it, Alaric and I.
And what we can say is that it took time. 2 to 3 days to install and fix bugs apps specifically developed by Google for initiation and discovery, as well as to install the development environment on our personal computers where we develop (there are many packages and many dependencies).
Finally, we could begin to operate the device. Two applications (APK) particularly caught our attention. The apk "Tango Constructor" and apk "Tango Explorer."
Again, on the internet you can find demos and explanations of these two applications.
But I will still describe briefly to explain what caught my attention.

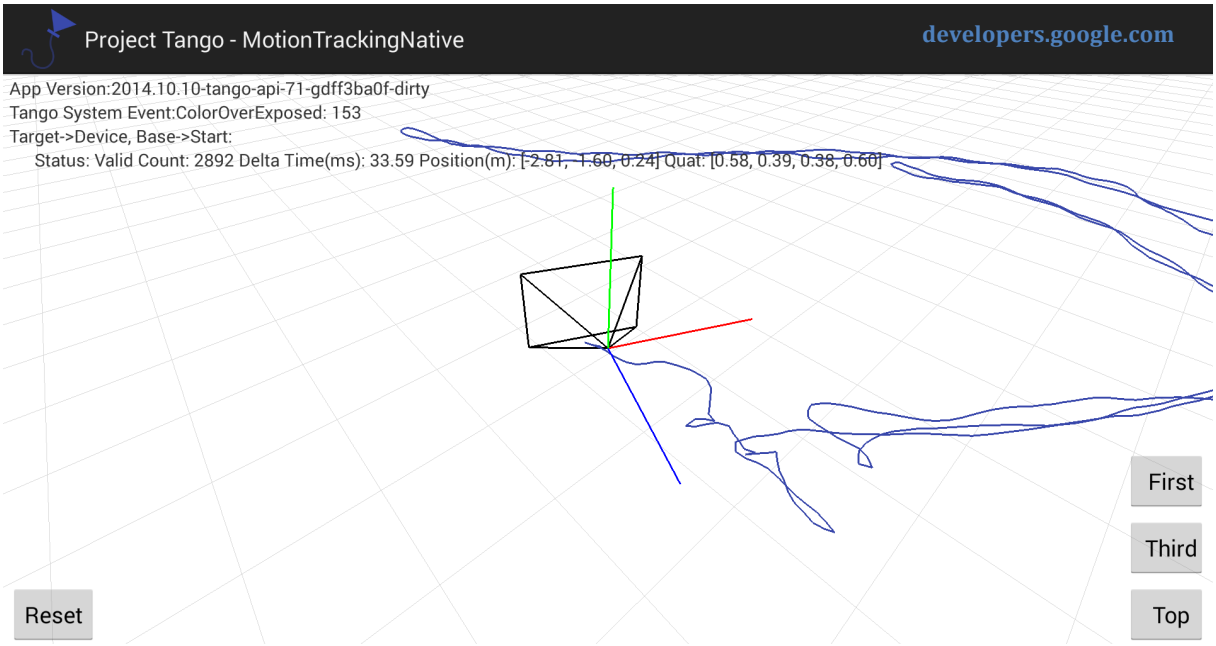
- Tango Constructor is an APK that enables a rapid 3D reconstruction, from what would look like stereovision, but from a camera, and motion, much like the Microsoft Kinect. This is what is called "Shape from Motion."
  The added value of Tango, is that it uses motion sensors built into the device for correcting displacements (between what it sees and what it suffers; accelerometer and gyroscope).



- Tango Explorer, allows to make learning a volume (a room, a place), and to subsequently reach locate in this room, whatever the starting point, the tablet can be situated. To do so, the application uses a 3D point cloud (cloud point).
  These points bring into evidence the differences in the depth of the environment. Therefore a smooth surface and perpendicular to the tablet is white inside these edges.



It's time to get serious. Explorer is a very useful tools to locate, yet he must know how to use. Google has thought of everything.
It offers APIs that precisely. Explorer uses two things, a path, and a set point. My research led me to understand one thing, I associated it with the constellations. Learning to record the path of the tablet, and seeks together the strengths, markers (typically folds, corners, changes of colors, ...). Thanks to it later, simply move and locate themselves in space as a browser in the seas following the stars and constellations in the sky.



I suggested to Alaric again which is separated our respective spots. with experience in odometry, I volunteered to handle this part.
I've created a small robot "GeekEnstein" that I presented at uB in 2016.

My ETIC project - Exploration in Unknown Earth (unknown to the robot) - Exploration in known Land (known to us).
For my robot, I use the same location method and the same components as those in the Tango tablet, or even more (compass, ...). Unlike here, we must develop Android.
Something new for me (and Alaric), that I never developed Android. So I had to start with a small self-training to get to operate and use the APIs.
It was laborious, but I will not spread myself on the subject.
The goal is to write a program to exploit the data provided by the API Motion Tracking.
It is fairly documented, but simply put, it gives access to a large number of variables to track sudden changes in real-time tablet. Thereby localize in space during or following a motion. The advantage of this tool is, for example, environmental learning is actually a museum or art gallery, moving in and trying to observe the most points and viewpoints as possible. This happens naturally when there is none.
But when using it are people and movement. It may then be difficult to find all the markers that we had previously recorded. During the meeting with Professor Mariusz NOWOSTAWSKI, he advised that for 3D reconstruction, it was necessary to add objects (usually cubes with QR codes) to assess distances and depths, and locate between two images. But with what we had, we were looking to do in another way.
And indeed, once located, thanks to odometry, though visually, the tablet loses these visual cues (along a smooth wall), it continues to localize. We appreciated his advice, but there was still much work, and it does not looks like something we really know.
The main issues I raised are how to define the work area to change, how to handle the difference in brightness, color, (etc.) while conditions of brightnesses are never the same, how the recognition to identify the position of the table for example to overlay augmented reality (it's not like adding a free object in the scene, as if tracing exactly on object and confused)? And the more I advanced, the more questions were added.
With a little practice, I began to understand how to get the API data.
Thus, I could find out where and how high is the tablet (i.e the camera), it looks (gyroscope) ... From there I would draw inspiration from what I learned during my trainning, i.e shaders and ray launched. It was a track, a first idea anyway. The goal is that if I imagine myself in a completely virtual environment, depending on where I am,
I can limit the generation of the environment to only what is visible. Starting from there,
I can know my position, look what I'm supposed to find.

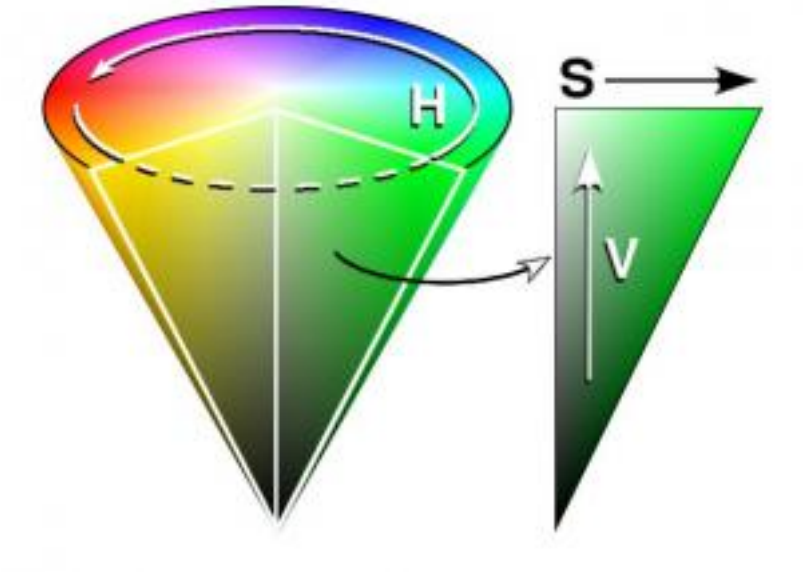
Augmented reality:

I attended several lectures and presentations, including that of Dr. Shida University of Siegen (Germany).
Dr. Shida made a presentation on how to capture the original colors of the material. If we look at an object, in our view angle, and the influences of the environment around, the color will not necessarily identical (it rarely is in reality).
So she developed with team have a method that identifies and separates the color of the material, as well as shadows and reflections, and can easily change the color of an object, then restores it shadows and reflections that depend on the new background color.

Too ligh colour of the car compared to the environnement

So I discussed with her this process is quite complex, to see if I can easily integrate into your project. Indeed, one of the main problems of augmented reality is that the added objects must fit into the real environment. I can not add an object taken with perfect light, and in optimal conditions, and add in a dark environment. So I exposed him different tracks that I followed, namely among others calculate the brightness of the room by the light sensor integrated with Tango, or make a linear interpolation of some reference pixels to know what could be the brightness coefficient, ...
Dr. Shida is therefore interested in the project, we organized a meeting (with her, Alaric and me) to present our work, and take advantage of these tips. We then quickly realized that its solutions are at a very high level, who would embark on a thesis topic, while we only have 3 months (or even less, as the course is already underway).
In terms of the accuracy problem, she more or less supported my idea of shader, the least in principle.
She proposed to replace the pixels of the work pixel by pixel. If we have the same picture, taken from the same angle, and with the same resolution (or at least the same ratios), it is playable, but here, we have to adapt, in real time at that.
With computing power, which is that of a tablet. Although Alaric felt the courage to do this work, I decided to do something more modest, to get to finish in time.

We must solve the problems one by one, step by step. So I focused on what I should do, if possible with what I can do. As for the images and the integration, so I thought working in the color space HSV (Hue Saturation Value, also called HSV: Hue Saturation Value) rather than RGB. This is my second test solution after the linear interpolation.



Color space HSV – " wikipedia.org"

My idea is if I generate an image from RGB multi spectral image (we call this Is), I take the same scene acquired with a conventional RGB camera (picture Ib).
I converted both images in HSV, and from there, I replace the H component (hue) of the image Ib by that of the image Is.
The goal is to provide more accurate colors, but preserving the saturation and value.
So I did the tests, but only from two images both of the same device (RGB camera), but each with a different brightness.
The results are not really spectacular, but I already developing at least two solutions.

Alaric has provided a useful tool, it is a Tango package (developed by Google) which is integrated into a project using Unity 3D and for displaying virtual objects in the real world through augmented reality. That's one small step for man, one giant step for our project already have notions in Unity, I quickly understood what to do.
We can create our objects in the virtual world to the scale of the real world, and we can move in the virtual environment from the real world, in the manner of a FPS, but instead of using a keyboard and mouse to move, we will use respectively our own movements or at least those of the pad on space, and rotations she makes to know the point of view.
It only remained to localize in space to properly fuse the two worlds.
It is the job of Alaric to locate me in space, and from these results, I gauge the virtual environment to the real. I have, however, proposed to begin to locate starting from a position defined by a QR code on the scene. There may be several QR tags in the room to start from where we want.

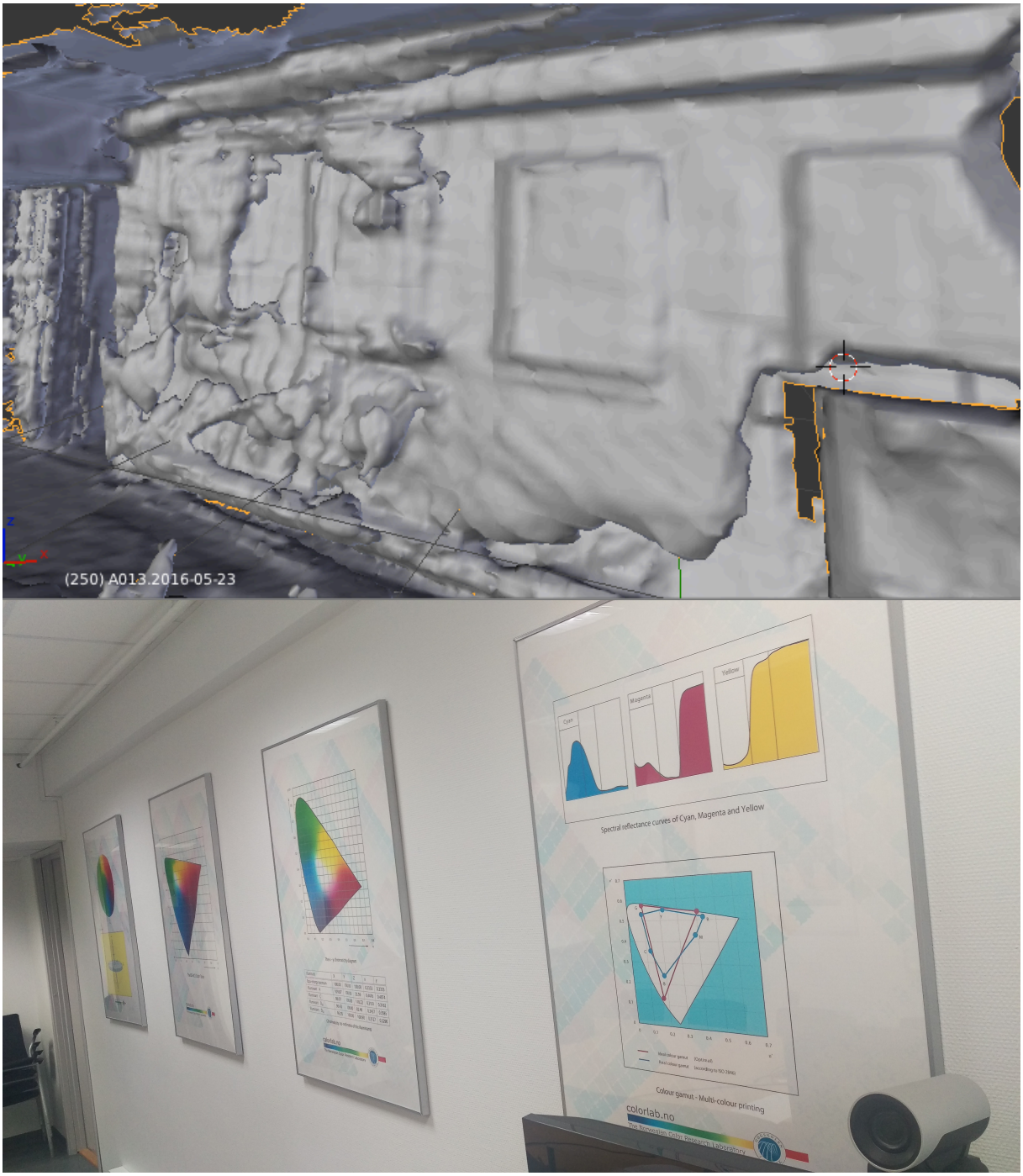
VR – Virtual reality :

It's been one month and a half I am in training. Supervisors to ask a meeting to take stock.
So we presented Alaric and I and our respective joint work. That's when the Pr. Hardeberg seduced by our results, requests to change the idea of a virtual tour, as well as adding new possibilities for AR (augmented reality), namely the ability to change the wallpaper walls for example, ...
By mutual agreement, we agreed Alaric and I take different paths. I take VR (virtual reality), and Alaric continued on AR (augmented reality). As I said before, we had a developed specifically for Tango APK, APK Constructor. This one builds 3D forms from scans performed with the tablet. The 3D format is kind of .PLY or OBJ

In both cases, the binary files. That's why I prefer to take the .obj file, and then generate a file or .PLY .STL ascii, with Blender 3D. I'm already familiar with these two formats, often used for 3D printers. For me it is more simple and natural to work from ASCII files as binary. Hence the choice.
Then I, in both cases, facets that I can see shortly that I write a small program to do it.
During my studies in first year of Master program, I studied several types of 3D modeling. I saw how to model regular polyhedra in OpenGL, as well as WebGL (at an educational project), thanks to the boundary representation model.
On a small scale, simply draw the facets one by one, and add listeners who will listen and interact with the environment. So I made the choice to use this knowledge and various scripts I had written before.

In the case of STL files, the file contains the description of facets (triangle) and the direction of the normal.
All this, in a structure that looks like XML. Advantage, there are already parsers.
For PLY, here alsoo it is interesting. The structure is different.
In the simplest version of the format, we have a list of contact points (one line per point) which will constitute the form, followed by a list of forms, consisting of the list of items in order, (a polygon described by line, and each point is described by its number in ascending order lines). Firstly, I made the choice to make a WebGL rendering, available online, its the JavaScript (JS) code.
In addition, as part of one of my previous projects, I had already developed a pseudo game engine. Soon the limits of rendering appeared. I do not control enough illumination, textures ... As Alaric continued with Unity 3D, I came to this program, and in addition, it offers among other compilation in WebGL.
Having come back to VR - virtual reality, was not ultimately a bad thing for me, in fact, thanks to the experience for AR- augmented reality, I've realized that I can ultimately combine these two worlds.
Indeed, if one part of what I did in the beginning, I placed works in a virtual world I layered to emulate them in the real world. Because I have digital image (a model) of the real, if I place my works in the virtual, and I erase walls, I have plan for augmented reality. Let me explain, for augmented reality, I have to add objects (panels) about which I will post my picture.
These panels should be exactly where the real work is located (because we have no graduation on the work, so we localize us in relation to the environment, and not in relation to what we have in front of us). When I place the works on the model, it is enough to hide everything to turn.
Another advantage, I can more easily change the color of the walls, wallpaper, and more easily represent the timeline (one of the proposed ideas, the ability to compare a building several times, following a renovation or other) previous experience with Unity 3D, especially in virtual reality (among other ETIC for my project, I have not struggled, because most of the work was already done. Therefore, I only had to integrate my mesh in my project, and for the rest just comes alone. So, I'm more concentrated on the mesh. Constructor is a good way to start, and to get a first preview of the potential of the project. But this is not really the point. The advantage of the Tango tablet, as I said above, is that it combines two technologies to locate, software localization, optical (IR and RGB) to a modeling "Shape from Motion" and odometry to confirm the position with a hardware location. And one must correct the other.

Voici un résultat :

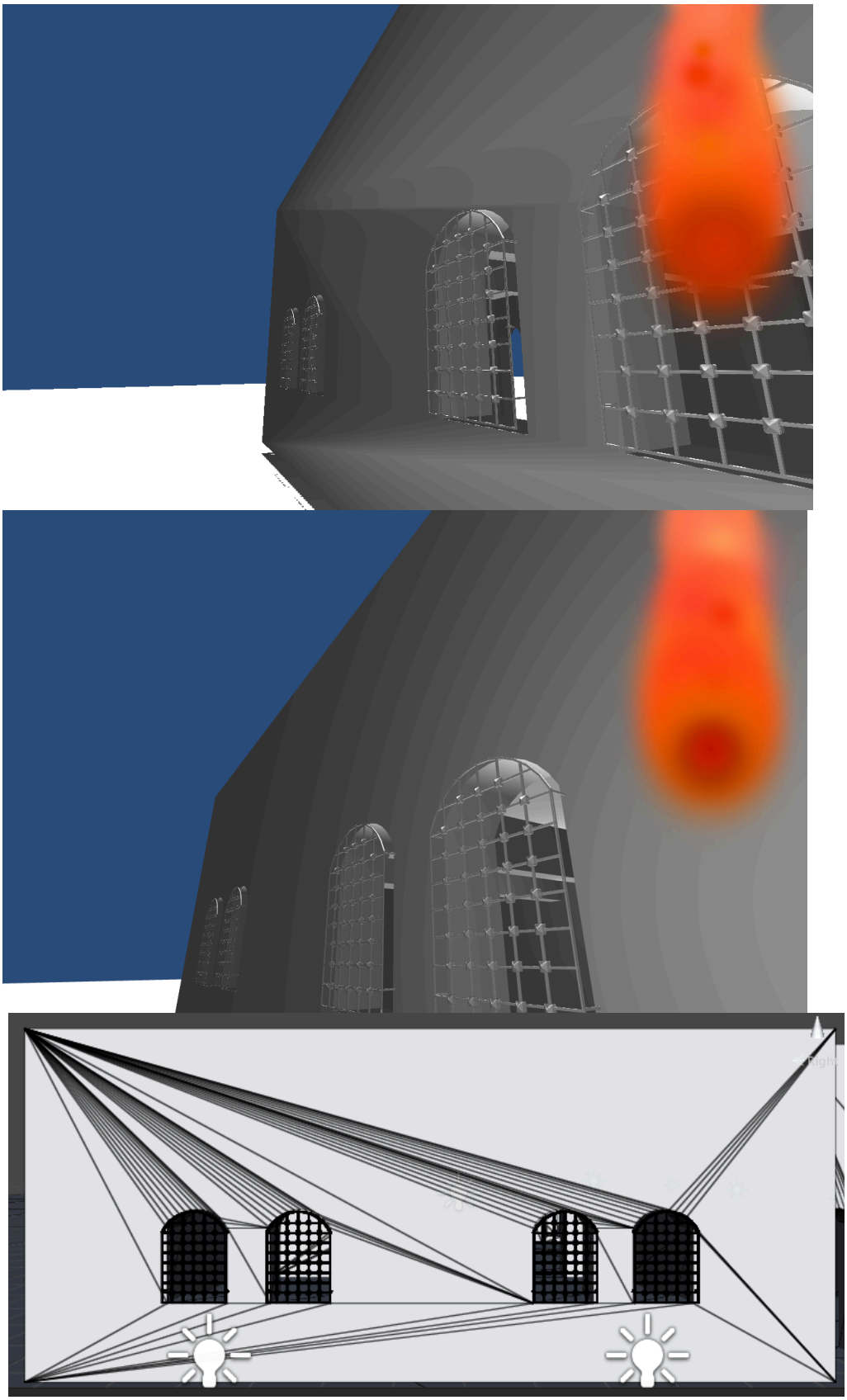


One may asked about the result. So to begin with, like any 3D scanner, nowaday we cannot scan which is transparent /shiny.
Therefore, the ceiling lights, because of their diffusers stainless steel, can not be taken.
Behind the TV screen with the corner shown (bottom right) is too dark. And one of the properties of light and color, is that one sees only the wavelengths reflected by the material, and black is a lack of reflection. Therefore, the more the system has difficulty assessing the distance that it considers too far.
And finally, there are on the left side of the mesh burrs, and other imperfections. These distortions are reduced on the right side of the mesh.

These modeling defaults are due to the tare on hardware. If we go too fast, the tablet can have losses of informations, because it did not capture some shifting. If we go too slowly, it will depend on the sensitivity of the sensors.
Thus, the extended scan which allows for the details, can have negative consequences on the quality of the scan, because calibration is lost.
And yet I'm only talking to a wall. He had to have spent time on the product to make more tests thorough. But I personally preferred not to dwell on these details (which are nevertheless important for quantifying the quality of results).
I consider the Tango tablet is a prototype, so a few weeks or months, this work has been obsolete.
In addition, the choice of this product is only to make a first approach, a compact object, which includes any one, but the day today, I do not quite consider the results to devote too much time.
In the future, it is likely other instruments to be used. That is the reason why I preferred to concentrate on my job which I found most useful and durable.
So I concentrated on the mesh. We must take into account several parameters.
On one hand, the desired level of detail, the quality of the mesh, and the weight of the file, which will involve subsequently the type of use.
Depending on whether the product is intended for local use by a powerful computer with a large computational power, a big graphics card for optimal rendering (reflections, shadows, ...), and memory, it is a tower or a single tablet, that consultation is done on the internal disk of the machine, or local area network (LAN), see online (WAN).
So I made my first demostration to present to supervisors. And their request which direction to take. Here is my proposal :
I have knowledge in WebGL in Unity 3D and Unreal Engine.

- WebGL is an easy solution compatible on almost all platforms. No need portability. In addition, it is easy to use. From the server side, you need a server to host the file online (or at least on the target network). Does not require large equipment. Everything is done from the host side, subject to having enough bandwidth to serve all potential simultaneous hosts. From the host side, all you need is a web browser on his computer, and an internet connection (or at least be on the same network as the server). The record will depend on the bandwidth of the host, the compatibility of its browser with WebGL, and computing power. Use few graphics card, so, the renderings are less impressive.
- Unity 3D is a cross-platform solution. Once the work is finished, you can compile it to generate the files on almost all known platforms.
  In general, it offers proper rendering, enough to be both supported on phones and tablets that despite progress in the area remain underperforming compered to a real computer.
  But also by the most advanced computers. Inconvenient, no hand on any rendering. The mesh can be changed, the mode of illumination (for color rendering), is the choice of the renderer.

Thus, for the same purpose, the angle of the camera in motion, and the angle of the light source, my rendering presents changes in illumination on surfaces that have the same direction of their normal:



These images are from one of my personal project work 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 is known for the quality of the renderings. Given the resources used for the project (multi-spectral camera, and many other tools) is to have a perfected, and not approximate.
  The inconvenient is rather intended to computers with a good graphic. To my knowledge, it is more oriented games consoles and computers running Windows (where Gamers are located), and lack of flexibility compared with Unity 3D. However, my knowledge and skills in this environment are very basic (not zero, but not successful either).

Supervisors do not want yet set a direction to follow, they want me to put forth a little more the potential of the project initially. So I made the choice to continue with Unity 3D, for which, if necessary, shared my work with Alaric. At this level, I did a little focus, to know what I was going to keep working. So I decided to manage the display of images, simplify and smooth the mesh and ease the selection and addition of publishing area (choose a table, a wall, etc.)

Select an image, consist of choosing the spectral images which are previously generated and stored in a database or a directory. The application is accessed remotely to retrieve and display (depending on the desired wavelength). For the demonstration, I only change the image. But this is the same at the end. As I headed towards a solution WebGL, so I have chosen to work with a repertory, which contains all images. When the user browses in the model, as soon as he approaches a work, he will be in a field that he has own. Then the overall lighting is switched off and its a dedicated spot that will illuminate the picture. The program scan the folder (dedicated to the work), and it is possible to choose the image to display. I started working on the Bump Map (bump mapping), to better manage the colors and the reflection of light because the colors change depending on the point and angle of view of the camera.

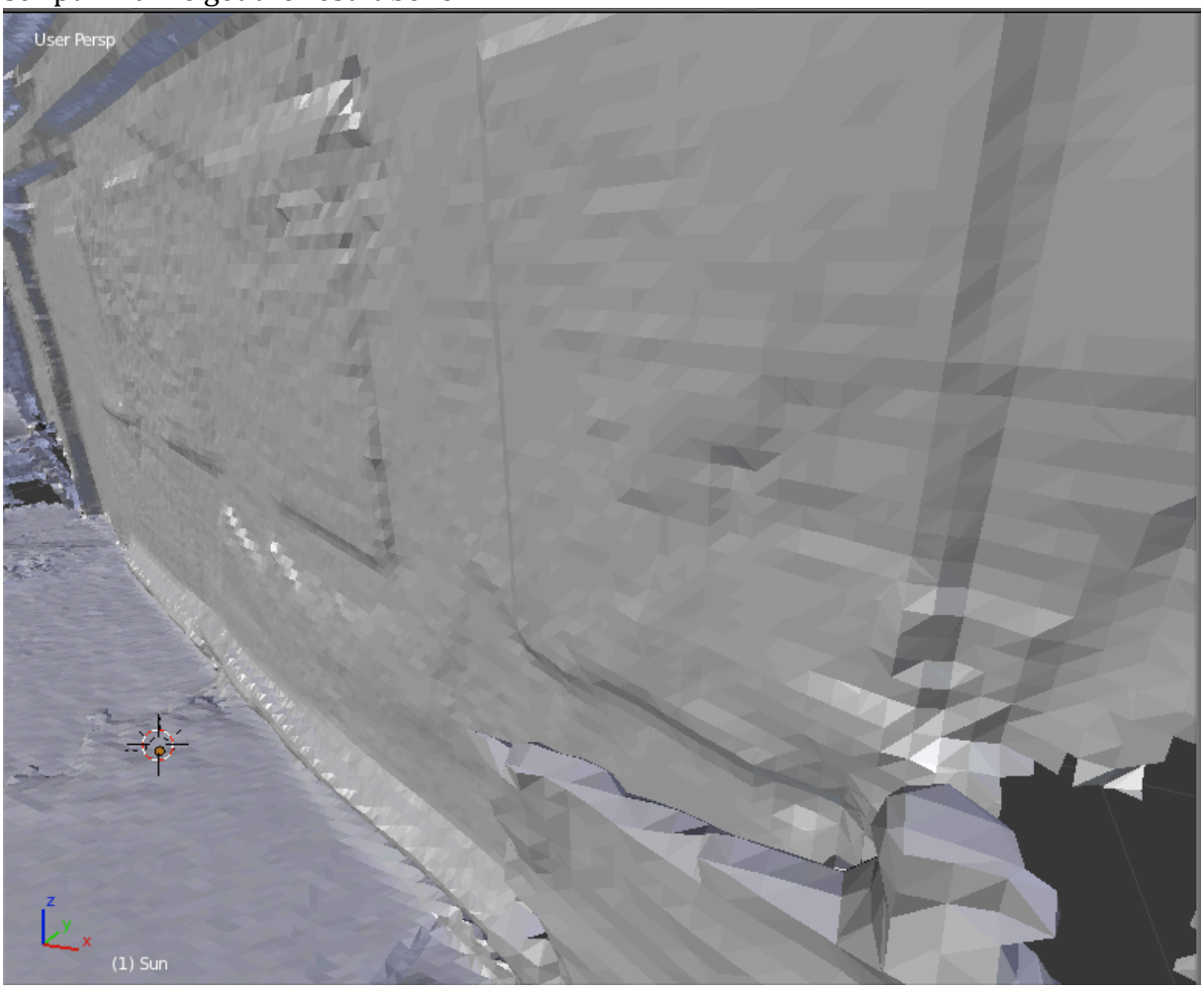
I have also met Professor Mariusz, on solutions to optimize and simplify the mesh.
My idea is to convert the facilities that belong to the same plane, in a single polygon.
Of course, as there are many distortions, I consider as belonging to the same plane, all the facilities that belong to a bounding volume (which is a plane with a thickness).
Then I run a set point for creating a convex envelope of a plane (2D of course).

Professor Mariusz offered to contact one of his student interns, who also works with the Tango tablet who developed a code on optimization for flat surfaces.
That one works directly from the point cloud (similar to what Explorer does).
To generate surfaces, it compares the normal od two facets that share the same stop.
If normal are inclined from each other less than 6°, it merges. Otherwise not.

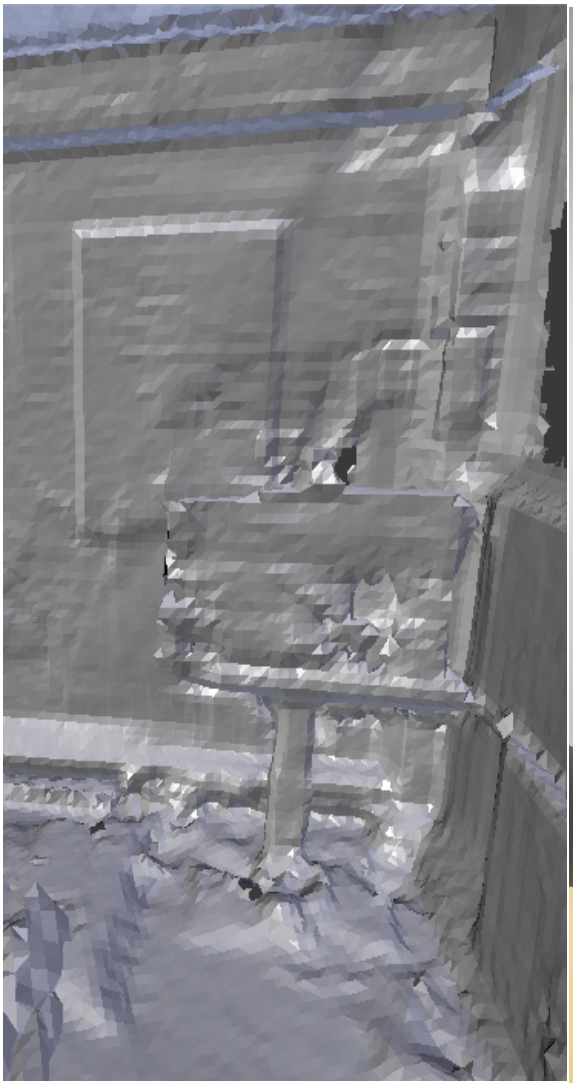
So I get contact with the student and asked him an appointment to see a demo. But he has never responded to my request.
Having studied in my course the convex hulls, and even carried out a project with various algorithms (2D and 3D)

I considered to be able to do this myself. Also, I'm already at the end of my 3 month internship.
The work is done in several steps :
• I generate surface polygons
• I transform the triangle polygons
• I optimizes triangles until the desired result.

To do so, I started from a .PLY mesh in ASCII format. I randomly take one facet. From there, I take the three facets that are terraced.
Of these facets, I generates the bounding volume (of a thickness equivalent to 1 cm in the real world). Then I begin to spread recursively my volume, and As I refocused over to the average of the points contained.
Once at the border, when I can not add more points to be contained in the volume, I have a point list (thanks to the structure of .PLY file list is already done).
Then we must therefore projected points on the new plan, and from there I have only to launch my convex hull algorithm. In theory it's simple, but when we go to practice ... there are always special cases, and it always takes a long time to identify.
I admit that a few small arrangements are made by hand, to achieve and complete the script. And we get the result bellow:



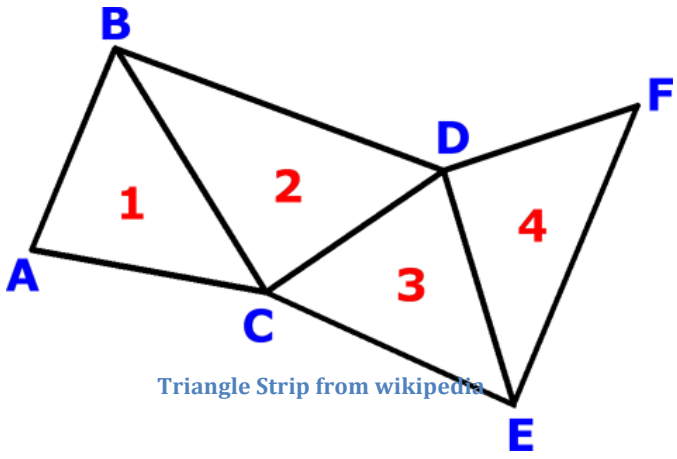
The result is not yet optimal, but this is one of the consequences from detail. If one want a high level of detail, few nuisance (or deformations, imperfections) will be deleted. And reciprocally.



P In terms of the convex hull, I preferred to stay on what is most often used, I used the Graham Scan.

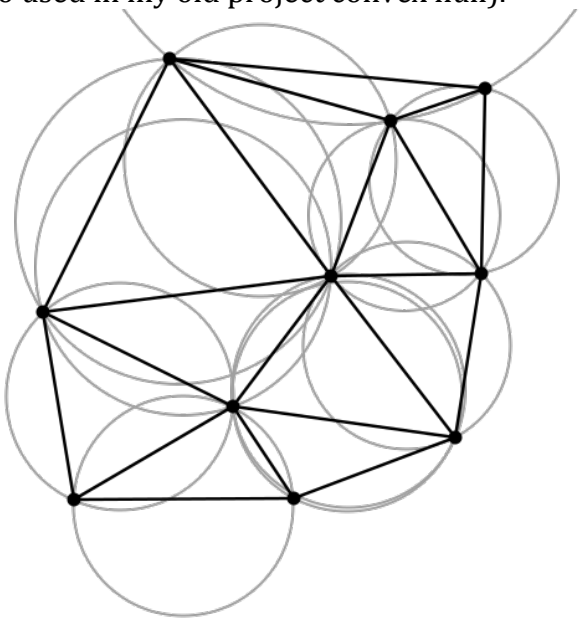
Generally (and it is even recommended), the polygons are avoided (over 3 points I mean) 3D modeling, to avoid a point that belongs to the form, but that is not in the same plane as the other points because display errors.

That's why I come back to the facets (triangular). So I cut the polygons with a "triangle strip" algorithm.



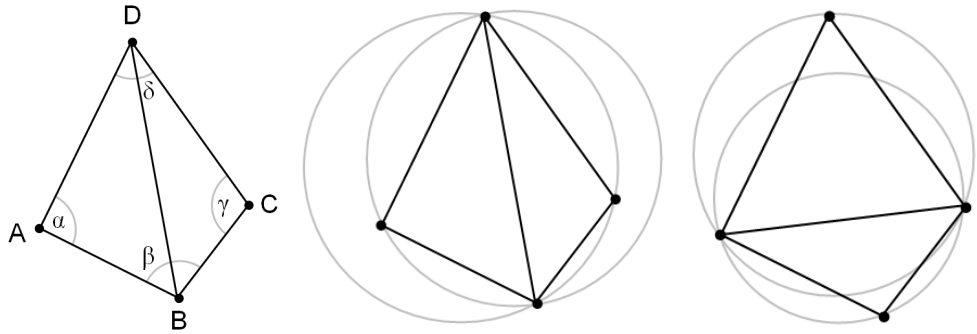
Triangle Strip from wikipedia

The disadvantage of this cutting is that it may give very small triangles next to this one larger sizes triangles. Therefore, it is necessary to apply the Delaunay triangulation algorithm (which I also used in my old project convex hull).



Delaunay triangulation from wikipedia

As a reminder, here is an example of what does this algorithm :
Given a circle passing through the three vertices of a triangle, if another triangle is contained within this circle one can remove the edge common to the two triangles and creates another between the other vertices of the triangles that were not affected.



At this level, the work is not finished yet. Indeed, as I have shown for the record with Unity 3D, having too many facets, and too large, according to the local illumination mode, with shadows, on a square, for example, consists of two facets, we can find two different colors because each of the two sides has an average of different illumination.
That gives the problem seen with Unity 3D.
For fast rendering, the lighting is calculated only from the highs, not relative to the pixels. For a better control over what can become a display defect, I prefer to control the length of the edges. Therefore if on a triangle, one of the edges is at least 2 times longer than the one of the other, there is a subdivision. I am using SDS solution (Sub-Divion of Surface), but in my way, and for my needs.

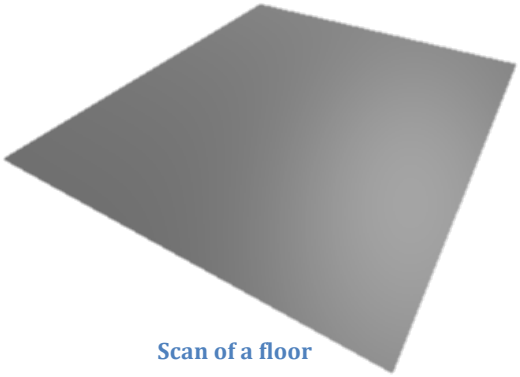
The other objective is also to reduce the weight of the mesh. The task is accomplished through what I did.
Indeed, I managed to have the scan of a wall and a few objects that the same weight (in KB naturally) than the entire room optimized.

Optimization cannot really be quantified because it is very effective on flat surfaces, but its time consuming for computation of more complex parts.

Exemple of a simple shape :

~20MB before optimisation

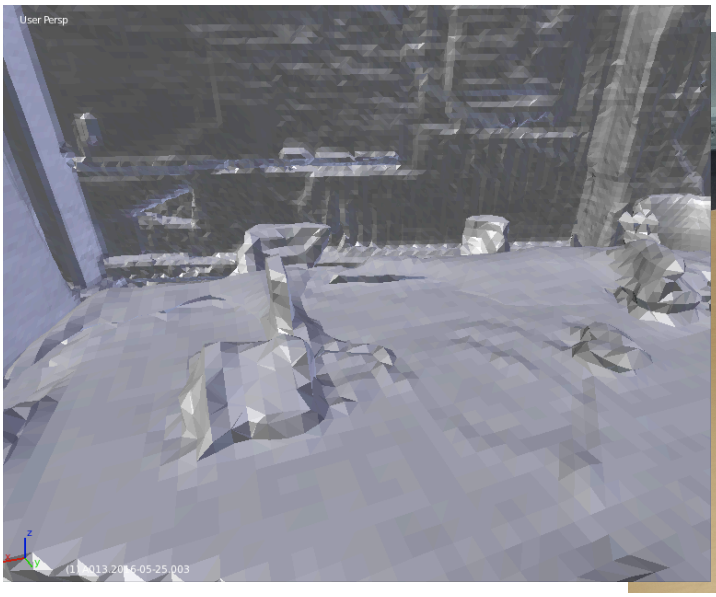
Less than 500KB after optimisation (more than 2 facets, and 4 points)



Scan of a floor

In the end, we get a more or less uniform mesh. There are still several default to correct, as the form is not a closed polyhedron, the texture mapping is a complex point if we want to be faithful to the original (and how want to be). How to manage updates, since each scan generates a different mesh, which implies a different shape, which cannot merge with another !

But in 3 months, we cannot do everything. Especially the big problem is that I had no (or rather say almost none) coaching or support.
For everything, I had to refer to my achievements, or do research and self-educate myself (often in emergency).



Futures directions :

There is still much to do or redo this project. I have opened a path. We can see the potential if we want to continue developing. I have several ideas that I have suggested for the following, such that :

• Test the scan with other solutions on the market (like Kinect for example). I also proposed and started to develop and test the scan based on the "shape from shading".

• The multi-spectral acquisition of 3D object like a sculpture for example. It would be possible to take the texture of an object with a multi-spectral camera.

• To optimize the file size (mesh), it would be interesting to use an object library for most recurrent. Example chairs, drawer handles, ... The advantages would be multiple :

- The object will be modeled only one time, stored into memory one time also, but would be called and displayed several times.
- The modeling of this object can be perfect if done manually. The advantage to this is apart from rendering quality sound, it will already be optimized.
- The objects such as chairs, for example, may include reflective parts (metal, polished, etc.), or the subject is too dark, matt.

They will be incomplete. Hence the interest of a specific and independent modeling.

To do so, we must firstly identify the objects in the scene manually, and replace them. Then if possible automate the operation.

- I sorely lacking mastery of renderers that I used. This is undeniable. At the same time, we can not be good at everything, and above all mastered in a short time. Therefore, I think it may be useful to develop a specific rendering engine, which will be optimized for our needs. It is possible that this is feasible with Unity or Unreal. But I do not practice enough to master them.
- RGB images from the multi-spectral is generated beforehand. Thus, the user can select the spectra one by one, but can not choose himself wavelengths to build his image (it will take too much time). With the progress of technology, and optimization of computing time, it may be possible in the future.

Other :

My internship was not an isolated work in the laboratory. Indeed, I was able to attend several presentations of works in the area of the image. Researchers from around the world came to present their work (including French, and from the uB). I discovered a wide world of knowledge in the field of colors, such as reproduction, identification, perception ... a lot of this research are very interesting. But he had lost in all these opportunities and ideas because the internship was only for 3 months. And so little time, it seemed daring that to feel able to integrate all these solutions to my project. Even if all these people are open and motivated to help Alaric and I, but when everyone adds his little something, it turns into a mixture of idea no longer holds the road.

Conclusion :

In conclusion, this was a good project, although quite ambitious. I finished (with difficulty by lack of time), even if it is only at its beginning, because this project is only a starting path for future developments of the future. Despite some disagreements on certain choices with Alaric, we always manage to find common ground.
Changes in the project that arise during development are often frightening. But I think I still managed to always fit the demands of my supervisors. Autonomy is a project, but we must assume these choices, and taking a wrong turn, can lead to failure.
Therefore, I preferred not to get into too many big things at once, but setting myself milestones, to always have something at the end, but especially to realize as soon as you are on the wrong track. As for qualifying conditions, nothing particular to say. Aside from my computer, which I used for the rest, I had always access to the equipment I needed. The lack of coaching, is sometimes difficult, because we start from the scratch, and must complete a work by launching into the unknown. But it helps to gain more autonomy.
Visiting professors from uB (Professor Alamin MANSOURI and Jean Batiste THOMAS) was good, although late. Indeed their advice and opinions of the project, in the absence of coach were comforting, but at the same time, they arrived at the end of the course, and the cards were already played. Luckily, I played the right cards (or at least, less bad).
The lack of compensation, and scholarship for the stay are difficult to assume.
Live several months in one of the richest and most expensive countries in the world, is not something easy. Back in France, you feel richer, in several senses, but especially rich in experience and knowledge.
As part of a thesis or research laboratories are very well equipped and the course of my internship in such an environment was something good.
There are regular conferences to present new findings ... The language barrier can be overcome in most cases (the majority of people speak English, and over time you begin to understand Norwegian). So I'm not disappointed with my choice. This is a project that I chose to do because I am already working on this type of achievement. I intend to continue it in private or in a professional setting. It's to do. By anyway, it was an opportunity for me to combine several points, firstly life and discover Scandinavian country, secondly validation of a university degree, but above all, the opportunity to work on a project that was close to my heart in a renowned professional setting.

Project by
Karim MOULAY – karim@moulay.me

In collaboration with
Alaric COTTIN – alaric_cottin@etu.u-bourgogne.fr

Supervised by
Jon Yngve HARDEBERG - jon.hardeberg@ntnu.no
&
Sony GEORGE - sony.george@ntnu.no

Hosting institution
ColorLab NTNU Gjøvik

References :
wikipedia.org
developer.google.com
docs.unity3d.com
Image de couverture : daum.net

Please remember to check also the images attached