State of the art

Development of a 3D mapping system for the simulation of autonomous vehicles

January 17, 2021

Project members :

Emilie Cuminal	cuminal@etud.insa-toulouse.fr
Julie Fié	fie@etud.insa-toulouse.fr
Clément Gehin	cgehin@etud.insa-toulouse.fr
Fanny Michel	fmichel@etud.insa-toulouse.fr
Claire Pouny	pouny@etud.insa-toulouse.fr

Tutors :

Elodie Chanthery Pierre-Emmanuel Hladik

Keywords : Automotive Mobility, Photogrammetry, Map, LiDAR, Data acquisition

Abstract

This paper provides an overview of the innovative project **Development of a 3D mapping system for the simulation of autonomous vehicles**. A state of the art about 3D mapping techniques will be provided in the first time. Then, the organisation of our work and project management techniques used, such as the Agile method and the Gantt diagram, as well as the concrete solution we provided to the project problematic will be presented.

Contents

Introduction

This paper provides an overview of our 3D mapping project. Our first work was to make the state of the art of the different techniques to create 3D maps. Next, we organised our work to meet the deadlines. The final part of this report deals with the achievements we made on this project.

First, we decided to create a user interface. The goal is to select the directory containing the images taken by the LiDAR or the camera and then to create a 3D map. This will allow the user to select the AutoDesk Recap workspace directory.

In order to stock the data of the LiDAR and the camera, we need a computer to make the link between the storage space and the materials. We tested several solutions, that are explained in this document. Beforehand, we configured the LiDAR thanks to the ROS environment, installed on a computer on the INSA Toulouse campus. Thanks to the developed software and the material, we acquired data to be imported in the AutoDesk Recap software to design a 3D map.

1 Context and project problematic

This project is part of the intelligent mobility chair, launched by INSA Toulouse foundation with Actia. This chair aims to associate variate industrialists to experiment innovative technologies in a constant evolution of technological and societal challenges. With the emergence of autonomous vehicles came the need to have accurate environment map, and the Actia shuttle needs a 3D simulation map.

The final objective of the project is to deliver an accurate 3D map of the INSA campus to simulate the autonomous shuttle. The need of 3D map can be explained by the fact that it offers a precision of several centimeters, more that navigation apps (like Google Maps) that only offer a precision of several meters. For a precise simulation map, the precise localisation of every details of the road environment has to be clear : traffic signs, ground markings, etc.

This project aims to design and implement a system capable of producing a 3D digital model of an environment based on video measurements and LIDAR data. This model is then intended to be used for simulation tools for the intelligent mobility chair's autonomous shuttle.

2 Main technologies

2.1 Photogrammetry

The photogrammetry is one of the most known technology to create three dimensional images.

By photogrammetry, we understand the science of making reliable measurements based on photographs. It determines qualitative and quantitative characteristics of objects from images recorded. The input of photogrammetry is photographs, and the output can either be a map, a drawing, a measurement, or a 3D model.

The principle is that the photographs are analysed and the computer may use reference points to create a 3D model. The fundamental principle used is triangulation. A stereo vision can be constructed when the same objects are observed from at least two different viewing angles, just like the perception of depth from human's eyes. A line of sight is established from the center of the camera lens to the object, and the computer reconstructs a three dimensional representation of an object with the intersection of these lines from multiple views .

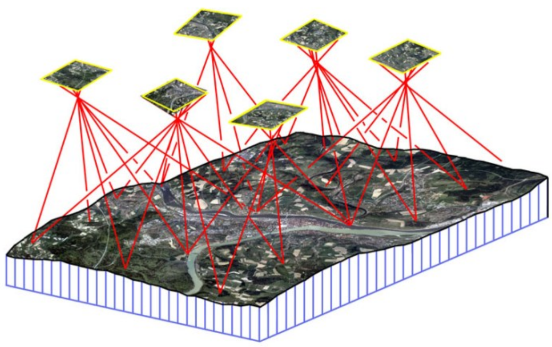

Figure 1: Photogrammetry method

2.2 LiDAR camera

The LIDAR, or light detection and ranging technology can be described as a technique for acquiring three-dimensional images with laser pulses.

The idea is very similar to microwave radar, except it works with much shorter wavelength. Indeed, the distance is measured with a laser light, which has shorter wavelengths than radio waves. This means that it will have much better angular resolution than a radar. The waves use can be ultraviolet, visible or near infrared light, depending on the application.

The principle is simple : the LIDAR camera sends electromagnetic waves and calculate the time they take to return to the source. Knowing the velocity wave, we can easily calculate the distance to an object. Moreover, the process of measuring is incredibly fast because of the light speed.

Figure 2: Robosense LiDAR

2.3 Reality Capture

Reality Capture is a photogrammetry technique aiming to estimate a 3D structure from its 2D image sequences and laser scans. This is done during three phases : [1]

- **Capture:** a site is scanned using laser scanners such as **LIDAR** and/or camera.
- **Compute:** using a reality capture software such as **AutoDesk Recap**, scan data is automatically registered, integrated into a coordinate system, cleaned up, and analyzed. The final product is a point cloud or mesh.
- **Create:** design with the previous model point cloud or mesh.

This technique will be used in this project using AutoDesk software solution that will be explained later.

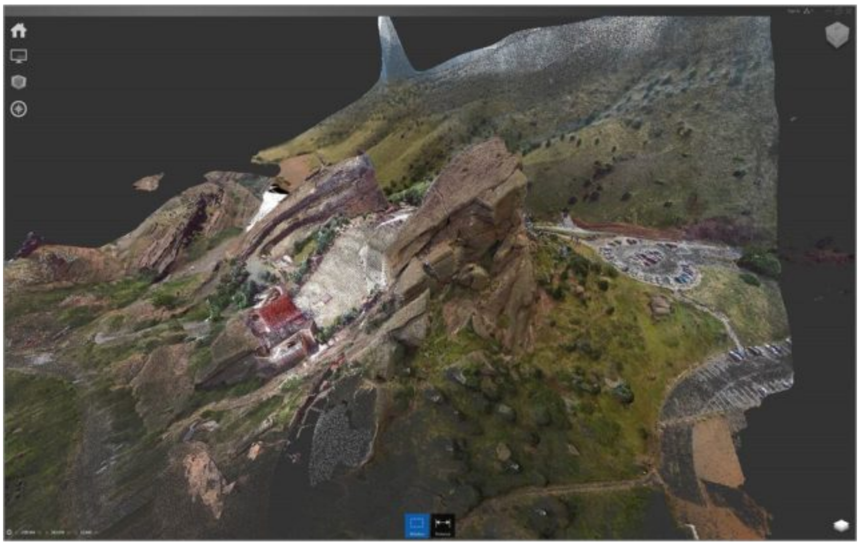

Figure 3: An example of a Reality Capture

3 State of the art : existing techniques for 3D mapping

3.1 Stereoimages

A first solution consists in taking stereoimages from an aircraft or an helicopter. This means taking photos from two sensors : one with the left view and one with the right. The two cameras take the same scene but with a slight different point of view (like our two eyes). A relief is created after formatting.

3.2 2D to 3D stereoscopic conversion

For the stereoscopic conversion, the first step consists in segmenting images into homogeneous regions. Then, a statistic map is created with the representation of the spatial distribution. This map allows solving detection as well as problem classification and noise deletion. A depth map can then be estimated. Finally, a stereoscopic image is generated by calculating the parallax values of each region using the generated depth-map and the input image.

3.3 UAV method and associated software

Creating 3D maps by using Unmanned Aerial Vehicle (UAV) like helicopters, aircraft or drones to take photos from the air is the first step in order to acquire images and geodata. This solution is relatively expensive because it requires special material and components. Software is available to provide analysis tools of flight data in order to operate an UAV.
Then the goal is to generate 3D point clouds. The first step is to select the appropriate images and the one that can be usable. Then a software and a web service allow to generate 3D point clouds that have automatic features such as adjustment calibration. We can mention Microsoft Photosynth, ARC3D (Automatic Reconstruction Conduit), Bundler, CMVS/PMVS2 or AgiSoft PhotoScan.[2]

3.4 3D maps creation from 2D images

The goal of this method is to apply plural parallax processing to 2D input images. There are different steps to follow : [3]

- Convert an input image having pixel values into a brightness image having brightness values
- Generate a depth map having depth information from the brightness image
- Generate a left eye image, a right eye image and a reproduction image by parallax-processing the input image using the generated depth map. The right eye image is delayed in the right direction for only the background area to apply positive parallax to the first pixel. The same is done for the left image but in the opposite direction.

3.5 Autonomous cars and their 3D mapping technology

3.5.1 Waymo, by Google

Waymo is a self-driving car project delivered by Google. As it drives, Waymo uses a LIDAR to build up a detailed snapshot of the environment, a radar to detect the distance of objects and their speed, and a high resolution camera to detect visual information, like the color of traffic lights for example. Then, it combines all of the acquired data. Waymo's LIDAR is called Laser Bear Honeycomb and can offer up to a °95 vertical FOV and up to a °360 horizontal FOV.

3.5.2 Lvl5

Lvl5 was founded by ex-Tesla engineers Andrew Kouri and Erik Reed, that previously worked on Tesla's Autopilot team, and George Tall, a computer vision engineer from iRobot.
Lvl5 creates high resolution and precision 3D maps for self-driving cars and robots, using computer vision. The method consists in the capture of video by dashboard cameras of self-driving cars that are then turned into high-definition and high precision 3D maps, constantly being updated in order to always accurately reflect the road condition and every information needed by the self-driving cars to detect and plan their routes safely.
This system does not use LIDARs because its creators think that the LIDAR technology is not ready to be used for safely working autonomous cars.

4 Work organisation and project management

4.1 Management organisation

We decided to follow an agile method as our project management policy.
This method allows us to plan tasks with short and precise targets that each member of the project can realise in autonomy. Short meetings are planned in order to see the advance and the done tasks since the last one. In our case, we have decided to meet with our tutors once every two weeks to discuss about the work done, as well as the encountered problems and our main questions.

For this project, we have short deadlines to meet : all the deliveries are due to January 2021. That is why we have created a Gantt diagram to organise our work.

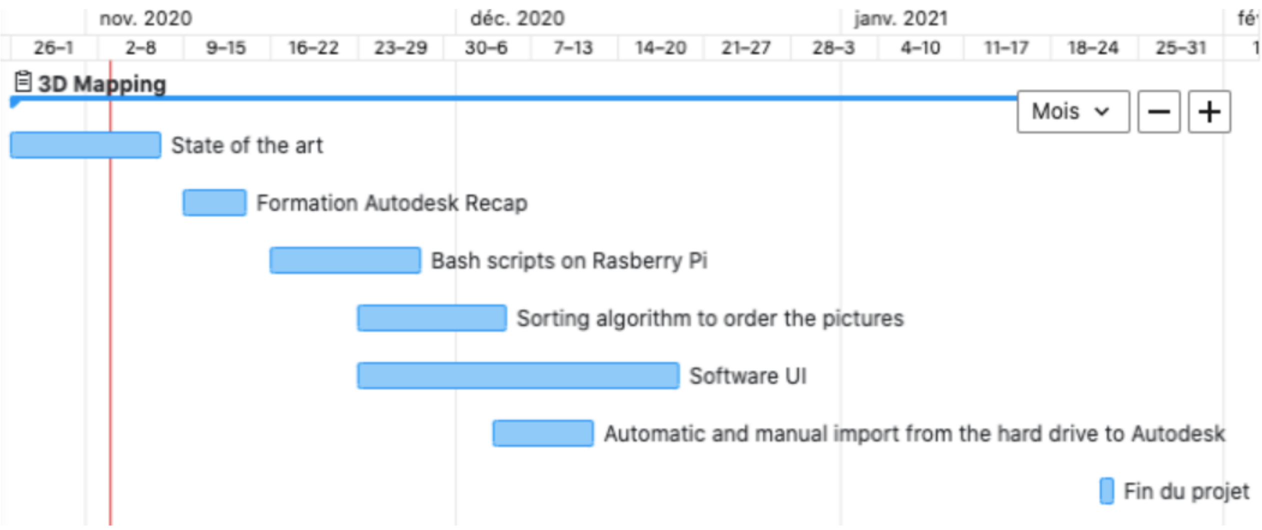

Figure 4: Gantt diagram

First, we focused our work on the redaction of the state of the art and the training on the AutoDesk Recap software that will be used during our project. It is important to first do some research to be informed about what has already been done and the existing solutions. That is why all the team worked on the state of the art.

In a second time, we did the development part consisting in the redaction of scripts to automatise the transfer of pictures from the camera to the AutoDesk Recap software, which creates 3D maps. We also designed a user interface.

We decided to adopt a horizontal management where every member of the project is at the same level as the other. We created an organization table on *Trello* where we kept trace of all the work left to be done, with deadlines, in the form of different *To Do lists*. This way, every member can choose which tasks they do.

4.2 Project risks

Before starting the concrete realisation of the project, we made an analysis about existing risks and determined their estimated time of appearance.
The first one is the ethical issue, which can be due to the non-respect of private life. It can appear at the release of the project, when it will go public and be re-used in other circumstances than just

the simulation of INSA shuttle.

The second problem can be the non-access to the material which lead us to redefine the project's framework due to the impossibility to see the final realization. This problem happened during lockdown due to the sanitary crisis.
Then, a risk that often happen is the human risk, more specially bad communication between the members, bad distribution of work or poor involvement in the project. These issues can have a bad impact at the beginning if the tasks to be completed are not distributed well and if the team is not motivated.
Furthermore, it is important to handle marketing risks such as competition if a new product appears on the market with similar functionalities. The perception of the project is a key point to be sure of the well understanding of the project.
Unexpected life event can happen : in our case, the Covid-19 crisis which can completely reshape the project and force to adapt quickly and on a large scale.
Not delivering all the deliverable on time can be a major problem since we only have 3 months to complete the project, in addition to other projects and courses. Unfortunately, this risk is present all project long and will be even more as the deadline approaches.
Finally, we have to manage the project well in order to avoid unattainable goals or an exceeded budget.

4.3 Budget estimation

We estimate a total budget of 4832€ to buy all the material required and the AutoDesk Recap software licence for a year.

Material	**€**
Autodesk Recap licence	426€ per year, free for students
Raspberry pi	77€
Hard drive	32€ (320 Go)
Lidar	3500€
Camera LI-USB30-AR023ZWDRB	698€
Cable 5M-USB3-TRIG	109€
Total	**4832€**

Figure 5: Project budget estimation

5 Provided solution

5.1 Concept

After the study of existing solutions and the specifications of our use cases, we have elaborated a solution for the realisation of the project.

The Raspberry Pi will sent instructions to the camera and to the LIDAR. Then, bash scripts that we implemented will launch the data acquisition, asking the user to enter a name for the folder containing the acquired data. This data will be sent to a hard disk. The Raspberry Pi will coordinate the launch of the LIDAR and the camera and will be the intermediary between the data acquisition equipment and the data storage folder.

Finally, we created an user interface in C#, fetching the stored data on the hard disk and sending them to the AutoDesk Recap software. The software is able to generate 3D maps thanks to all the acquired images taken by the camera and the LIDAR.

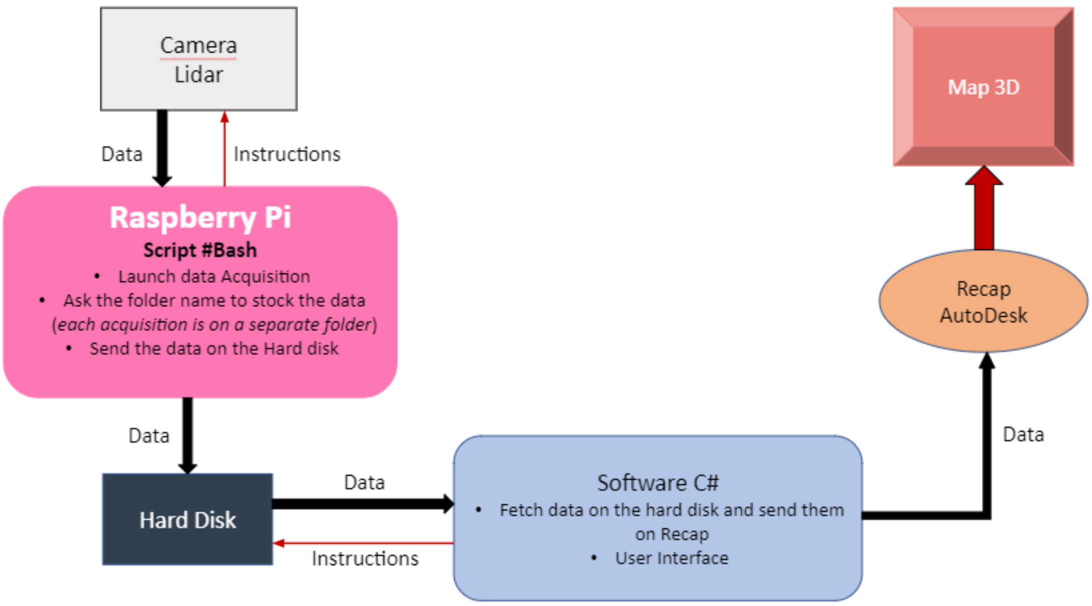

Figure 6: Solution architecture

5.2 The ROS environment

To use the LIDAR, we need to install ROS environment, LIDAR drivers and RSView software. ROS, which stands for Robot Operating System, is a middleware for robotics and a framework that facilitates the use of packages to control a robot. It is an open source software. ROS provides libraries and tools to help software developers to implement robotics software and more specifically frameworks for robot software development and standard application.
Programming and implementation are possible thanks to the different packages. It also allows communication with sensors as well as between the robot hardware and the robot program and handle complex robotics systems. It is not a traditional operating system, but it is essentially for low-level device control.
ROS has a graph architecture because processing takes place in nodes. It is not a real time operating system : it allows low latency in robot control.

5.3 Installation of ROS and the LIDAR drivers on the Raspberry Pi

We wanted to use the LIDAR with a Raspberry Pi to have a mobile system, allowing us to take it around the campus for data acquisition. The aim was to move the system on the campus to map it. However, we had difficulties with the installation of ROS and the LIDAR packages.

First, we tried to install ROS and the drivers on a Raspberry Pi 3 model B with the Raspberry Pi Operating System (OS) Raspbian. The problem was the incompatibility of the OS and the drivers. We tried with several versions of ROS, but the problem remained the same : it consisted in errors with some packages of the drivers and we could not solve the problem despite our efforts.

To resolve this incompatibility we tried to install an Ubuntu operating system. Nevertheless, the raspberry pi had not enough memory to support an Ubuntu OS. So we tried with another type of Raspberry, a Raspberry Pi 4 model B, with more memory. The problem was that the drivers we found were compatible only with the version 18.04 and below of Ubuntu. We did not find an image of Ubuntu 18.04 that worked for the Raspberry. We think that the problem could be the RAM of only 2GB of the Pi. We assume that it might work with a card with more RAM.

Figure 7: Raspberry Pi 4 and the LIDAR

5.4 Installation of ROS and the drivers on a computer

Finally, we decided to install ROS environment on a computer. We installed the version 18.04 of Ubuntu, the melodic version of ROS and the drivers available on the Git repository[4].
The next step was to install RSView to plot the data of the LIDAR. On the figure below, we can see an image of the data acquired by the LIDAR.

5.5 Bash scripts

We wanted to write bash scripts that could launch RSView, acquire data and save them in a folder automatically. It would be very useful to coordinate the camera and the LIDAR and also to begin an acquisition at the same time.
Unfortunately, by lack of time and because we did not have the camera, this code has not been implemented. We launched RSView and chose the folder manually.

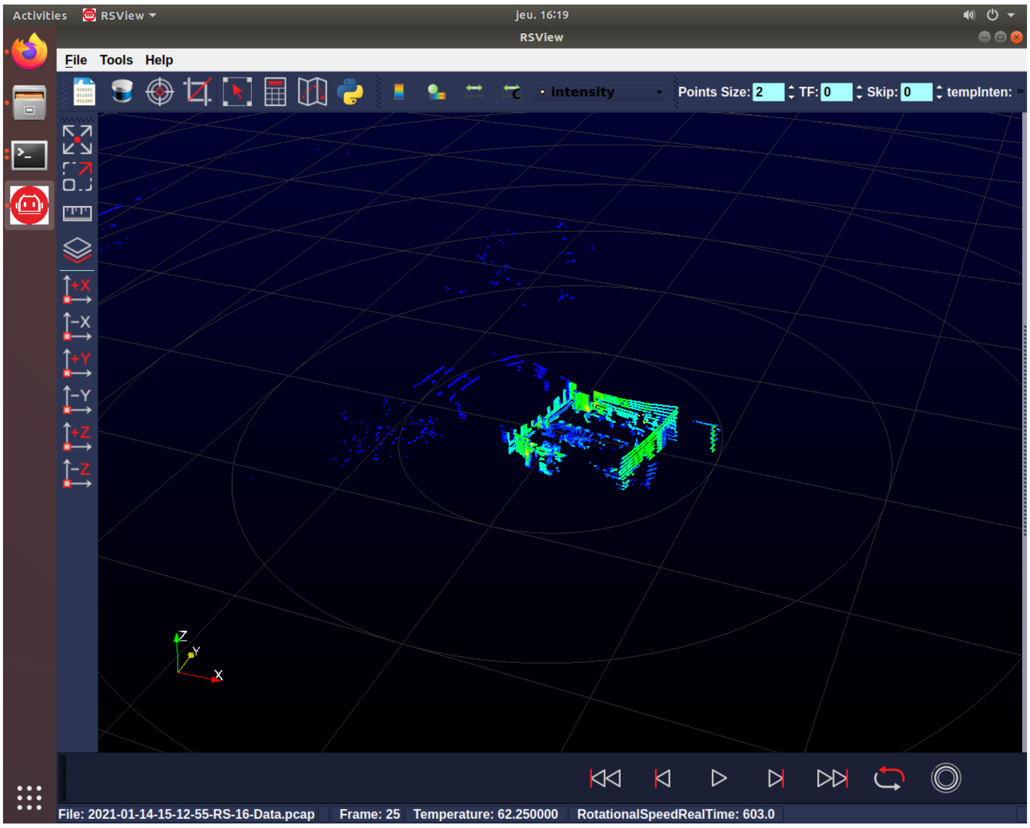

Figure 8: LIDAR data

5.6 Software Solution

5.6.1 Design

We needed to transfer the acquired data by the embedded system to a computer to create a 3D map. We decided to develop a software solution to automatise this task. The main purpose of this solution is to organise directories and convert videos into pictures to import them later into AutoDesk Recap and create a 3D map.

This application will be used on a Windows system. This is why we decided to develop it with C# since it has libraries designed for Windows systems. Even if the software will be mostly used to import data automatically, we thought that it could be a good idea to allow user to manually import images that could be taken afterwards.

5.6.2 Directory organisation

The aim is to replace the user's work as much as possible. The software will transfer data from the embedded system to a working directory defined by the user at the installation or afterwards in the configuration. The data will be organised automatically and easy to find when using AutoDesk Recap. You can find this organisation in the following figure.

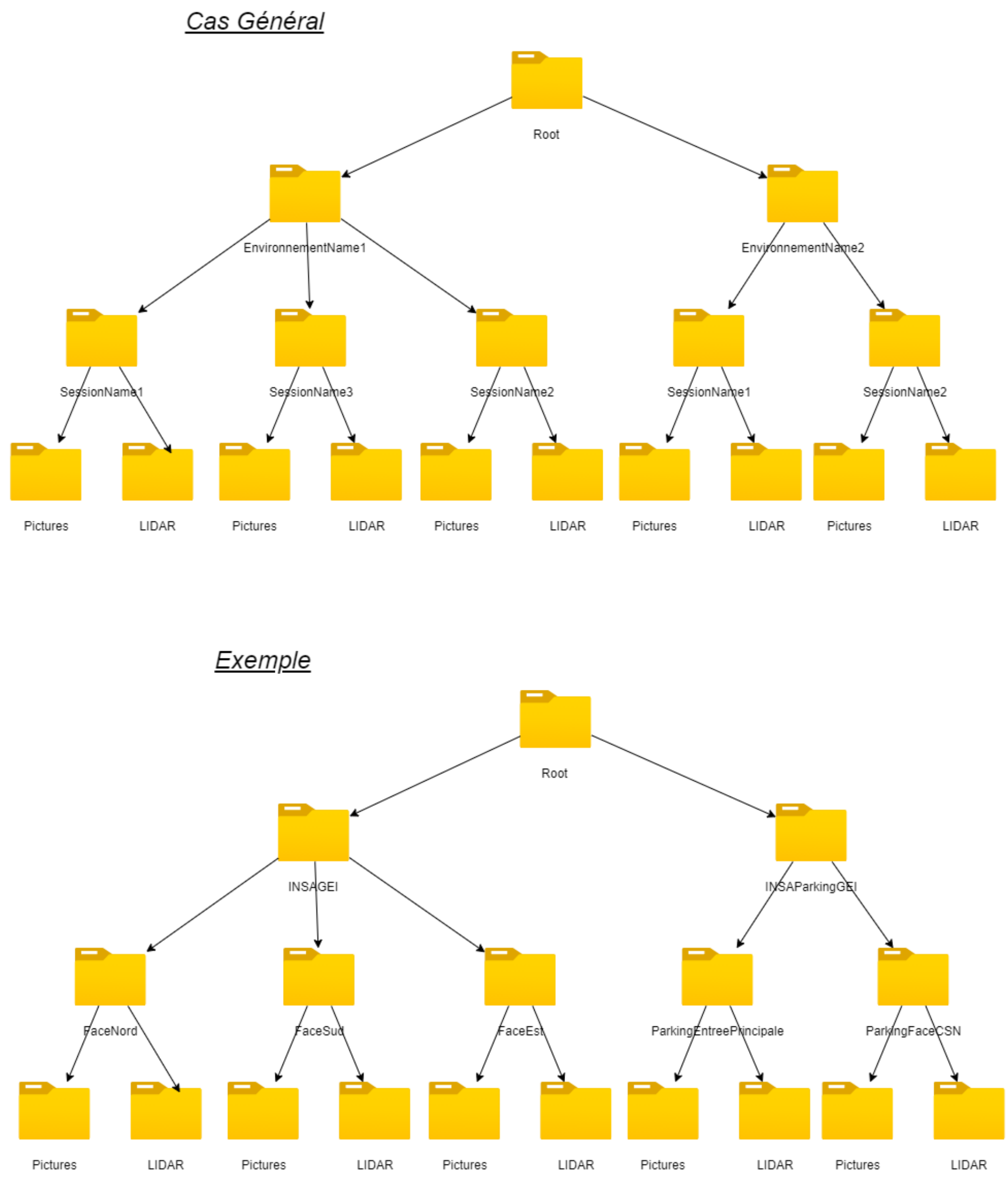

Figure 9: Directories organisation in working directory

For the data transfer, we recursively explore the directories on the embedded system and create a copy in the define working directory. We also offer the possibility to delete or keep the data on the embedded system. In this process, when we treat a video, we convert it into pictures in the working directory. A frequency is defined and configurable in the software. This frequency is used to choose which frame we keep from the video. For example, with a frequency of 30 we will keep the frames number 30, 60, 90...

5.6.3 Manual import

We added manual import to facilitate the correction of the model. For example, if a part of a building is missing in your map, you can take a video of the missing part with your mobile phone and add it to the project easily without using the embedded system. With this solution, we simplify even more the use of our system by allowing the user to easily correct his model.

When selected, the manual import guides the user at each step. First, the user needs to select the directory where data is stored. Then, the program will scan the working directory and propose to the user existing environments and sessions to complete. Finally, the transfer is made as if it was an automatic one.

5.6.4 User interface

We wanted the user interface to be easy to use. Since the goal is to simplify every step, a clear and light interface seemed to be the most appropriate. First, you need to log into the application. A unique user and password has been created in order to prevent a none authorised person from using it.

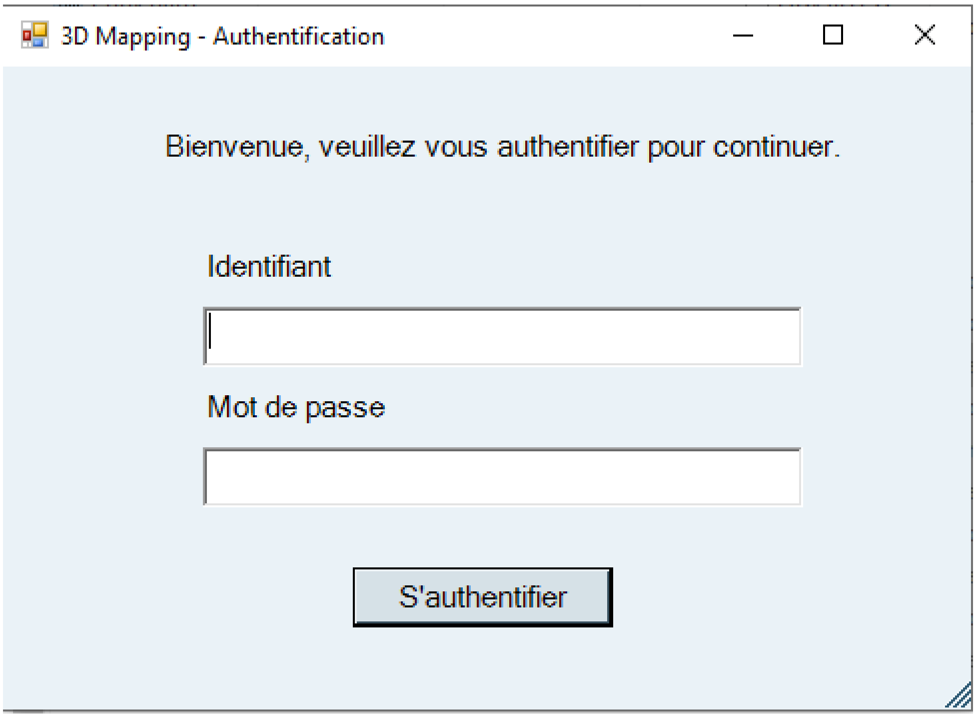

Figure 10: Sign In Window

Once the authentication is done, the interface open. As we said the interface is really simple and composed of three parts: configuration, automatic import and manual import. In the center you can launch an automatic or a manual import by clinking on the pictures. In the top menu bar you can configure the application settings such as frame frequency or working directory. Once an import is launched, you will be notified when it will be completed. If an error occurs, an explicit message will be displayed on your screen.

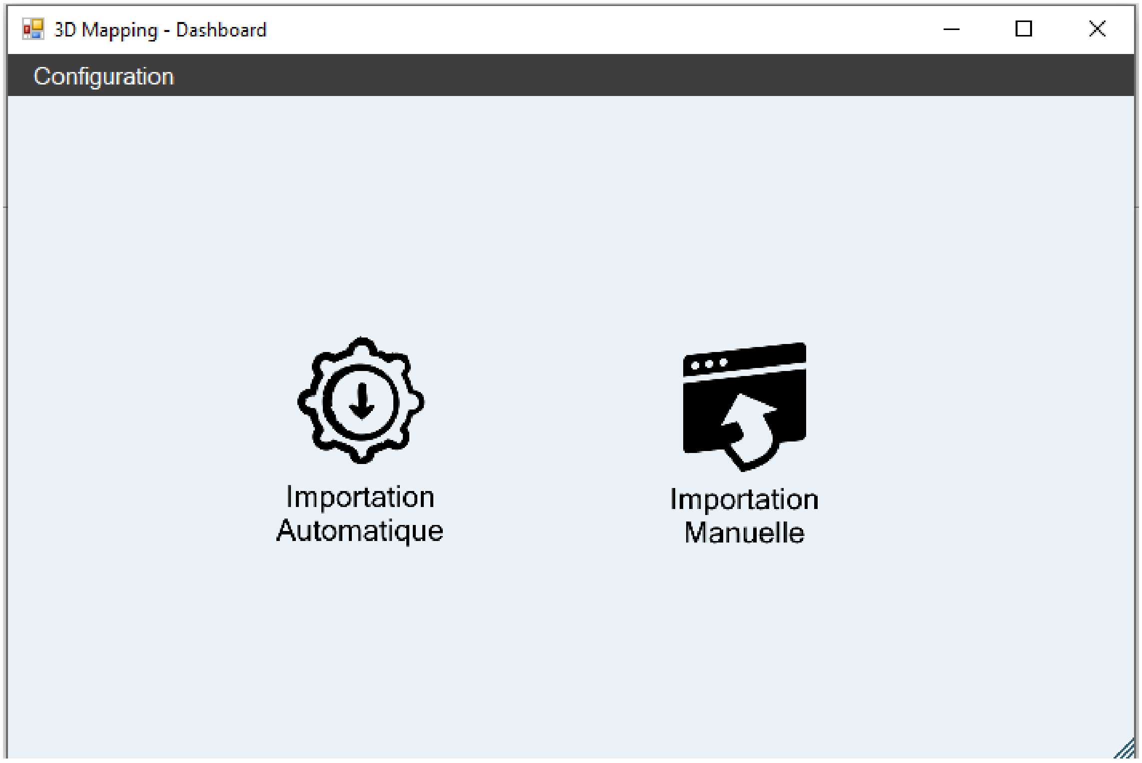

Figure 11: User interface

5.6.5 Installer

The installation of the application is really simple as well. You can retrieve the software on our GitHub on the main branch. Once the repository is cloned, you have to launch directly the application. A step by step installer will guide you. These steps are made to create and initialize the configuration files needed by the software to work. At the end, the installer will be closed, you can then relaunch the application and start using it. If a configuration file has been corrupted or deleted, the installer will start when you will launch the application to correct or recreate the missing files.

5.7 3D map creation with AutoDesk Recap Pro

AutoDesk Recap Pro is a software for the creation of 3D models from imported photographs and laser scans. Recap Photo is a service included in Recap Pro that processes photographs to create 3D representations of an area or an object. This is done by the reality capture process that allows capturing, measuring and mapping surface points to build a 3D model.
The first step is to capture data using a LIDAR and a camera. Then, Recap Pro automatically cleans up and computes the imported data to produce a point cloud or a mesh.

5.7.1 Compute data on AutoDesk Recap

With the educational license of AutoDesk we had, we were limited to the importation of 100 photographs per project. It is not enough to reconstruct a wide surface so we first tested our system to map a GEI room. We took videos of the room and imported them on the software which then framed them into photographs.
Then, we had to combine these photogrammetry data with LIDAR data. We launched a data acquisition of the same GEI room and imported it on AutoDesk.

Conclusion

This paper provides an overview of the work done this semester on the 3D Mapping project. After defining the keywords and writing a state of the art, we designed a solution to solve the initial problem. We wanted to create a system that could be moved around the campus to map it.

Unfortunately, we had some difficulties with the installation of the LIDAR on the Raspberry Pi. These problems made us lose a lot of time and we are therefore behind schedule. Moreover, the Covid-19 sanitary crisis has made team work difficult, especially when we needed equipment. For these reasons, we were not able to finish the project and to map the campus. We feel a bit disappointed not to see the final result of our work, but we hope that the provided solution could help others in the future to continue the project.

This project was a real opportunity for us to be considered as engineers more than students. We had to define a solution to a specific problem, and to take into consideration the costs and the risks of the project. The role of our tutors was to guide us without directing us, and they were always available for our interrogations. We also want to thank Paul Scanlan for his wise advice all along the semester.

References

[1] "What is reality capture?" [Online]. Available: https://www.autodesk.com/campaigns/reality-capture

[2] F. Neitzel and J. Klonowski, "Mobile 3d mapping with a low-cost uav system," *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. XXXVIII-1/C22, 2011. [Online]. Available: https://d-nb.info/114455585X/34

[3] B. suk Seo and Y.-S. Heo, "Method and apparatus for converting 2d image into 3d image," 2012. [Online]. Available: https://patentimages.storage.googleapis.com/1c/db/7f/2d3ea675a38d94/US20120087570A1.pdf

[4] [Online]. Available: https://github.com/RoboSense-LiDAR/ros_rslidar