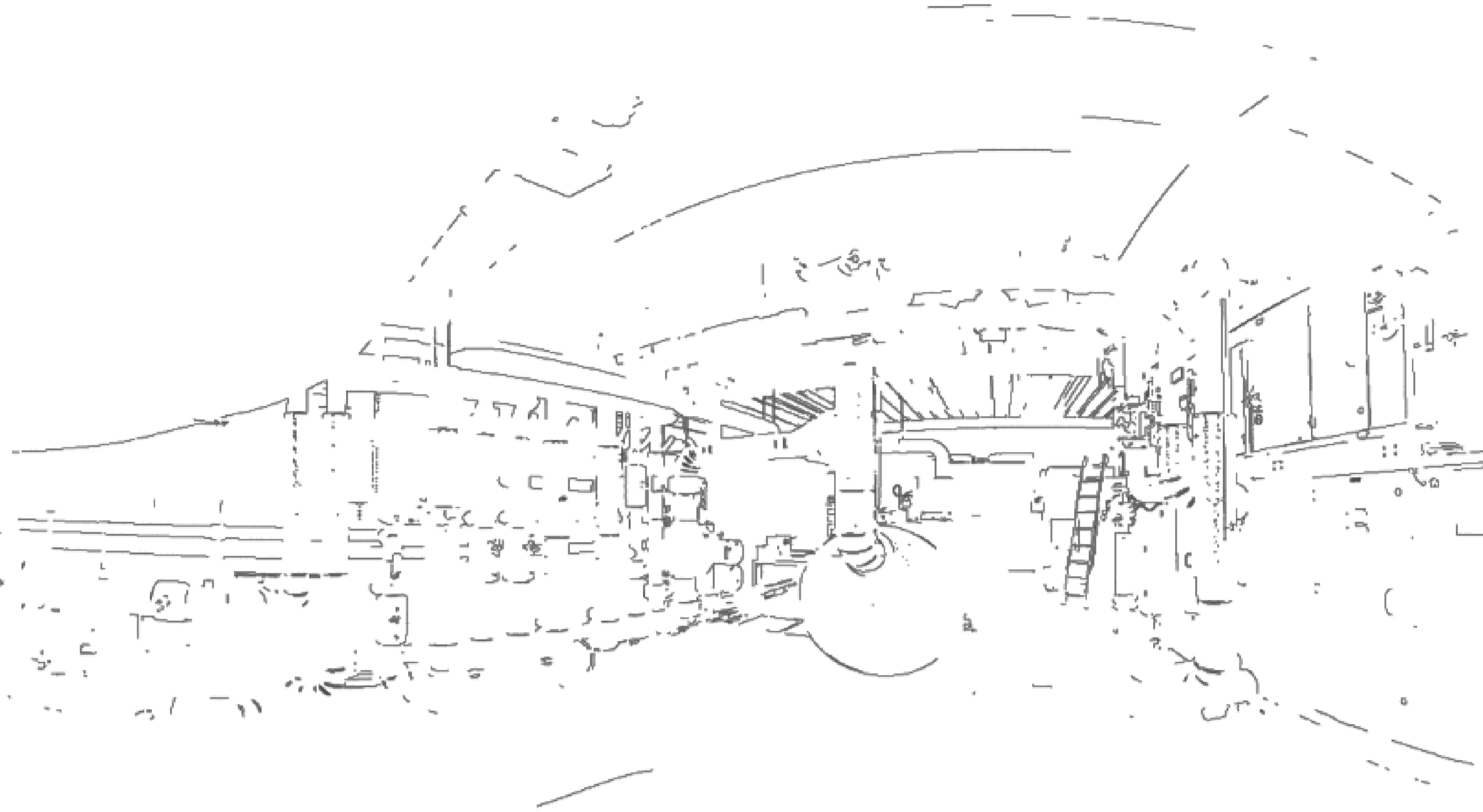
MSc thesis in Geomatics

The Smart Point Cloud framework to detect pipelines using raw point cloud generated from panoramic images

Lydia Kotoula

July 2018



THE SMART POINT CLOUD FRAMEWORK TO DETECT PIPELINES USING RAW POINT CLOUD GENERATED FROM PANORAMIC IMAGES

A thesis submitted to the Delft University of Technology in partial fulfillment of the requirements for the degree of

Master of Science in Geomatics for the Built Environment

by

Lydia Kotoula

July 2018

Lydia Kotoula: *The Smart Point Cloud framework to detect pipelines using raw point cloud generated from panoramic images* (2018)

ISBN 999–99–9999–999–9

The work in this thesis was made in the:

Geo-Database Management Centre
Department OTB
Faculty of Architecture & the Built Environment
Delft University of Technology

Fugro
Regional Innovation Centre Europe
Leidschendam, The Netherlands

Supervisors:	Ass. Prof. Ir. Edward Verbree
	Prof. Dr. Peter van Oosterom
Supervisor from Fugro	Ir. Niels Treffers
Co-reader:	Dr. Ir Ravi Peters

ABSTRACT

Spatial data acquisition and processing is rapidly developing, making point clouds available and easily accessible for many applications and to many end-users. Nowadays, point clouds are the main surveying product and have more added value than the derived products, as they keep details and they are not interpolated. Raw point clouds do not contain information that relates the points to the semantic meaning of the real word objects that are represented. Moreover, through different procedures (classification and segmentation techniques) important semantic information could be derived, creating an intelligent environment and structure, a Smart Point Cloud (SPC).

In this research, a SPC framework will be created combining different techniques and methods in order to detect the pipes in an industrial environment. Close-range photogrammetry will be used to generate a point cloud, for which panoramic images are the main source data. The features and the attributes from both data (panoramic images and point cloud) will be combined to obtain characteristics in order to select, analyze and identify the pipes, as one object. The characteristics of Three Dimensional (3D) point clouds and Two Dimensional (2D) digital panoramic images are thought to be complementary, so the interpretation of the world objects combined with point clouds and images seems a promising approach. However, the most challenging part is to relate the geometric features between the 3D point cloud and 2D image data, while a precondition for this lies in the extraction of the corresponding features.

Our approach aims to make use of the advantages from the analysis of point clouds and image processing to represent the characteristics in 3D with increased accuracy. For the image processing part, 2D image boundaries extraction is applied by using boundary detection algorithm. Then, a methodology is proposed for corresponding 2D image pixels with the 3D point cloud. Getting the boundaries in 3D, is estimated, if close to them a cylinder could fit by testing first, the accuracy of a circle-fit. By using the parameters of the best fit circle, the cylinder could be reconstructed. The points that are part of a cylinder are stored in database as one pipe with parameters: the center of the cylinder, the radius, the direction and the length. Some spatial criteria are tested to combine the segments that are part of the same pipe. The final product is to detect the pipes, as one object that will be stored and visualized from points. Our thesis work has a global vision, through point cloud processing (segmentation, structuration, registration) to attach knowledge onto points.

ACKNOWLEDGEMENTS

In this section I would like to express my gratitude to the people who involved to this thesis directly or indirectly.

First of all, I want to thank my main Supervisor, Edward Verbree for his helpful advises and inspiring thoughts through the weekly meetings which steered me in the right the direction. My second supervisor, Peter Van Oosterom for his valuable feedback and guidance. My supervisor from Fugro company, Niels Treffers who was always available and able to help me during my 9 months thesis. Although not actively involved in my thesis, other people from the company, have their share of help as well. The friendly but professional environment in the company was very motivating to work in. Finally, I would like to thank Ravi Peter for his valuable feedback as the external reader of the thesis.

A great thanks to all my friends for their phychological support. Finally, I must express my gratitude to my family for providing me with continuous support and encouragement throughout my years of study. This accomplishment would not have been possible without them.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACRONYMS

3D	Three Dimensional	iii
2D	Two Dimensional	iii
UAV	Unmanned Aerial Vehicles	22
LiDAR	Light Detection And Ranging	1
TLS	Terrestrial Laser Scanner	1
SPC	Smart Point Cloud	iii
PC	Point Cloud	
DBMS	DataBase Management System	2
PCDBMS	Point Cloud Data Managements Systems	23
VR	Virtual Reality	3
AR	Augmented Reality	3
BIM	Building Information Modeling	22
MISP	Multi-Image Spherical Photogrammetry	ix
SFM	Structure From Motion	ix
RANSAC	Random Sample Consensus	vii
HT	Hough Transform	ix
IFC	Industry Foundation Classes	23
LOD	Level of Detail	ix
PCA	Principal Component Analysis	30
CRP	Close-range Photogrammetry	11
PSP	Panoramic Spherical Photogrammetry	18
PCVE	Point Cloud Virtual Environment	24
UML	Unified Modeling Language	x
LOF	Local Outlier Factor	ix
FDN	Fixed-Distance Neighbors	31
kNN	K-Nearst Neighbors	31
RMSE	Root Mean Square Error	62
SQL	Structured Query Language	x

1 INTRODUCTION

The extremely and rapid way that technology is developing these days, introduced a growing number of capturing devices for creating point clouds. Light Detection And Ranging (LiDAR), Terrestrial Laser Scanner (TLS) are some of these and also, passive sensors such as thermal, RGB cameras are used for point cloud generation making it a popular method for gaining spatial data. There are different systems like LiDAR and Photogrammetric systems that create point clouds, the mechanisms and software that are used, are not comparable. On the one hand, the LiDAR systems measure ranges or distances from the object while the accuracy depends on the item and the distance from the target point [Carter et al., 2012]. On the other hand, the Photogrammetric systems that are based on Structure From Motion SFM Photogrammetry, provide an automatic process to find and match a limited number of common features between images. Then, these common features are used to establish both interior and exterior orientation parameters. After this iterative procedure, a dense coloured point cloud is extracted to represent the object. For this reason, the acquisition of imagery with the right characteristics is critical, while the accuracy is dependent on the size of overlapping areas of the photos and the resolution of the images [Schenk, 2005].
In this research, the main application is about detecting the pipelines, as an object from the point cloud in an industrial environment. The initial step, in order to reach the objective of this research, is to acquire the point cloud from image based technique. In this case panoramic images. Then, this point cloud will be used as an authentic source, following a smart point cloud framework for pipelines detection, in an industrial environment. The advantage of using panoramic images, instead of the classical photogrammetric images, is that it consists of greater and a more complete information of the environment due to 360°field of view, computing a general representation of the whole environment [Peleg et al., 2001]. A single panoramic image will never be enough. You want to replace a large number of separate images by fewer panoramic images. As a result, a couple of panoramas can substitute many photogrammetric models. In addition, cameras are simple, fast and easy to acquire photos, making them a very useful and powerful tool for documentation and survey [Fangi, 2007]. However, the depth of the points cannot be acquired from one Panoramic image. Multiple overlapping panoramas must be acquired in order to construct the 3D point cloud. In this research, a potential way will be demonstrated for creating the 3D point cloud by panoramic images [Lee and Tsai, 2015].

The acquisition of the data is increasing while the sensor technology is developing so fast, creates an urgent need for decision making in real time systems. The point cloud is the intermediate step for the final derivatives like mapping and 3D modeling. It becomes an obstacle in structure limitations (e.g. mesh creation) [Poux et al., 2016]. The modeling step is a semi-automatic and time-consuming procedure that creates structural problems which many times lead to the loss of important information. In this research, point clouds will be the authentic source of data. The original point cloud itself does not contain any grouping, decomposition or any other information that relates the points to the semantic meanings of the real world object that is represented. Moreover, through different procedures, like classification and segmentation techniques, important semantic information could be derived, creating an intelligent environment and structure: a SPC [Poux et al., 2016]. The direct use of the point cloud allows for easier and more complete information extraction. However, the fact that point cloud can derive from different sources and that they do not contain semantics lead us to follow the SPC framework, [Poux et al., 2016].

3D points of a given point clouds contain geometric and appearance information. The need for the standardization of the real world by creating a semantically rich storage model, lead to the need of creating an automatic procedure for object recognition and building schema models. This schema could semantically represent the elements by detailed geometrical point cloud data [Krijnen and Beetz, 2017]. According to the figure 1.1, a global approach with a combination of different sources of knowledge is possible in order to structure this model schema. The combination of these three aspects: the domain, the geometric and the device expertise is the main objective for a SPC framework. The device's expertise depends on the sensor which is used for the data acquisition or the combination of sensors that may be used for the application. From this information, important point attributes are derived like the position, the density, the color, the intersection and the accuracy of the point. At this stage, after the point cloud acquisition, a filtering procedure should be followed for noise reduction and minimizing the outliers [Poux et al., 2016]. Concerning the geometric expertise, it refers to the geometric characteristics of a group of points. There are several tasks that are common at this point cloud procedure (segmentation, classification, feature detection). Through these common techniques, essential knowledge could be attached to the current points. The method that will be followed depends on the type of the application and the output product. The combination of spatial and semantic knowledge into a schema for object recognition is the meaning of the domain knowledge and demands a specific schema related to the real world. Thus, the structure of the domain expertise is represented in schema of database which stores the point attributes and the segmented features in order to combine them through queries [van Oosterom et al., 2015]. The structure and the organization of the data will help to inject semantics into the data, which will be stored in the DataBase Management System (DBMS), retaining the attributes from the other procedures, as it is shown in Figure 1.2 [Poux et al., 2016].

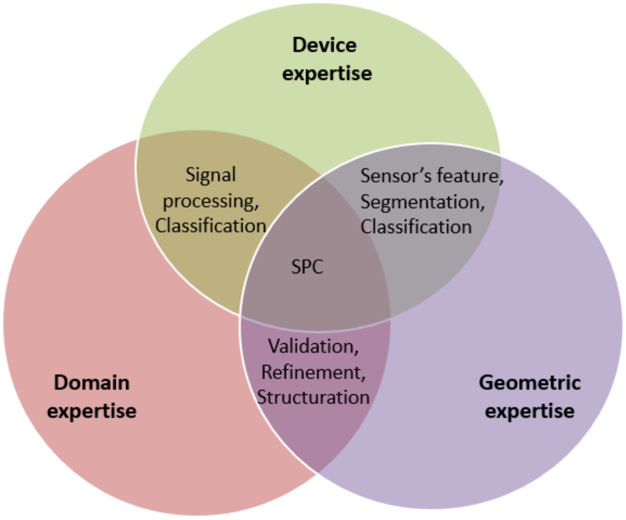

Device expertise
Signal processing, Classification
Sensor's feature, Segmentation, Classification
SPC
Validation, Refinement, Structuration
Domain expertise
Geometric expertise

Figure 1.1: SPC-Constitution [Poux et al., 2016].

While the SPC approach can be extended to all types of point cloud, it is illustrated on a colored point cloud, generated from imaged based photogrammetric technique. Color is an attribute which is mostly related with the image based techniques. The color is attached to points and it has essential value that definitely contributes in the processing for better results. Color segmentation and classification are methods

that take into account the color and could provide semantic characteristics to our point cloud. While feature detection and segmentation that focus on techniques both from a geometrical and algorithmic point of view will increase automation, performance and completeness of our methods regarding the context and domain adaptation (transferring knowledge from the source to target) [Poux et al., 2016].
Pipes are a common type of object in industrial environments. It is hard to automatically match pipes' features across multiple views because they often have very homogeneous surfaces. The geometry of the pipelines should be kept as simple as possible (cylinders).
Initially, the colored point cloud that is created from the panoramic images is used as input data. It is essential to combine the point cloud with raster and vector data and every 3D point should be connected through a pixel with the source image. The features and the attributes from both data (panoramic images and point cloud) will be combined to get characteristics from both sources in order to select, analyze, manipulate and identify the pipes. First, the edge detection on the panoramic images will be done to identify the boundaries of the pipes. Then, a connection between the points with the pixels of the images is computed. By having this knowledge, a segmentation procedure to identify the pipelines from the point cloud will be easier to apply. The main objective of this research is to combine these two methods and create a smart framework that has a general and standard use through specific schema, in order to apply semantics and store data for visualization and identify pipes, as one real world object. As a final product, the user could define the start and the end of the pipe, and also, some useful geometrical characteristics of the pipelines, like the center, the direction, the radius and the length, that will be helpful for the recognition of each specific pipe.

1.1 OBJECTIVES AND RESEARCH QUESTIONS

The goal of this thesis is the direct use of the point cloud following the Smart Point Cloud framework for the detection of pipelines in an industrial environment. The point cloud that will be generated from panoramic images, is used as input data. The main research question that this thesis is focused on:

What is the Smart Point Cloud way to provide a framework for object detection (pipelines), using panoramic images to create enriched point cloud?

To achieve the main question, some sub-questions are also relevant:

1. What are the differences between point clouds that are created using terrestrial (close range) photogrammetry and laser scanners?

2. To what extent do the number of panoramic images affect the generated point cloud in terms of density and accuracy?

3. What is the optimal way for object detection, using feature detection and segmentation techniques on images, as applicable to point clouds?

4. What is the best way to combine domain-geometric-device knowledge for pipelines detection?

5. Which are the possibilities of representing the point cloud in Virtual Reality (VR)/ Augmented Reality (AR)?

1.2 RESEARCH SCOPE

- There are several software products for generating point cloud through images. Some of these software will be tested using panoramic images and the result will be evaluated. Moreover, a further analysis will be followed comparing, if the number of the images and in what extend affects the final point cloud density.
- Two methods will be combined for object detection. The one is based on 2D image detection techniques and the other one on point cloud segmentation and shape detection. Both techniques will be combined and tested, providing an efficient environment for point cloud interaction.
- A general framework for smart data structure will be addressed combining domain, geometric and device expertise in an efficient DBMS structure.
- The reconstruction of the object after the detection to eliminate the modelling step is out of the scope of this research. This way, the time processing and the difficulties that derive from the modelling could be avoided. The main idea is to group the points that are part of the object and to store them as one object.

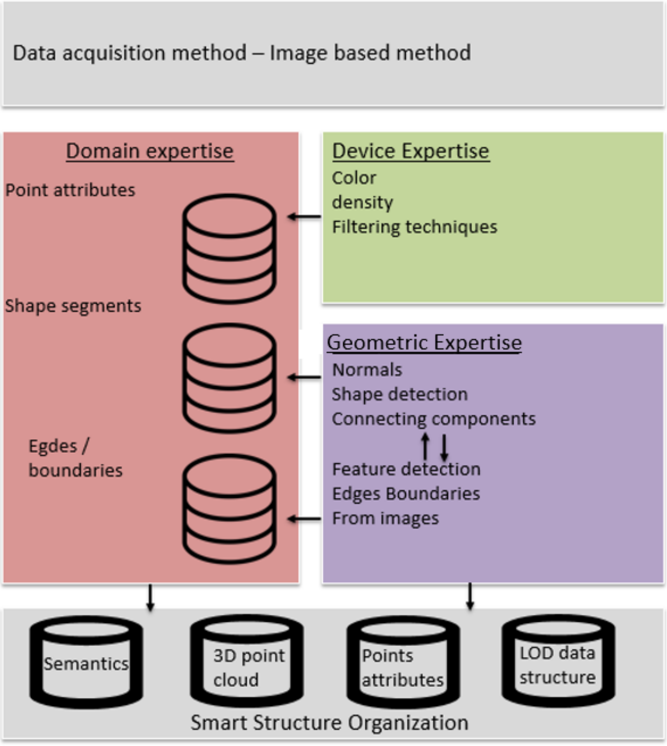


Figure 1.2: Workflow for smart point cloud structure, [Poux et al., 2016].

1.3 SCIENTIFIC RELEVANCE

It is of high importance to understand the huge speed of data acquisition and the necessity of making decisions in real time while considering different and difficult situations society tackles every day. It is in real high demand to create applications that will solve global problems providing a general framework making it applicable to a wide field. It is also important also to mention the use of the point cloud as an authentic source and not as an intermediate step. It is proposed to combine point cloud properties with human knowledge to extract information providing user-centered data. This research is lying on a framework that will provide an object recognition combining different 2D and 3D methods. A lot of research has already been done directly using the point cloud for segmentation and object detection.

Our aspiration is to provide a smart framework that could combine all the initial data and the advantages for quick and better results. Panoramas have the advantage, due to 360°field of view, of a complete boundary object detection that would be useful for the pipe detection. Eventually, this knowledge directly attached to the 3D points, makes the whole procedure easier. Hence, a general framework is proposed at Figure 1.2 for a smart point cloud creation. Another key challenge is the data structure, in which a specific DBMS is created that will contain device, geometric and domain expertise in a way that semantic queries are implemented [Poux et al., 2016]. In addition, one of the goals of this research is to provide a low cost image based technique for point cloud creation, which will be quick and accurate.

1.4 THESIS OUTLINE

The thesis document is organized as followed:

- Chapter 2 provides the theory and the fundamental ideas in order to introduce the relevant topics of the proposed methodology that this research is based on.
- In Chapter 3, several approaches are mentioned, describing all the research that has been done related to the topic of the thesis.
- Chapter 4 describes the prototype system design that is proposed including all the steps that are implemented in order to get the final result.
- Then, Chapter 5 continues with the implementation and experiments. This Chapter describes the different approaches that are tested following with their results in detail. Finally, for the validation of our method, the results are compared with a general object detection method that is applied directly to the point cloud.
- Chapter 6 gives a summary of the most significant conclusions and future work. The problems and the answers to the research question and subquestions are explained in detail.

In addition to the previously described chapters, this thesis document includes the following Appendices:

- A gives an overview of the tables and the relation with the UML shcema that the prototype is implemented and stored. The SQL code that is generated from the implemented Python scripts is displayed for the creation and the storage of the data.
- B provides the panoramas after the Hough line transform detection in a bigger size for better visualization.

Figure 1.3 illustrates the chapters and their roles to the overall research approach. It is a clear overview of how the whole thesis is structured, giving the hierarchical steps that were followed in each chapter and their relation with the subquestions.

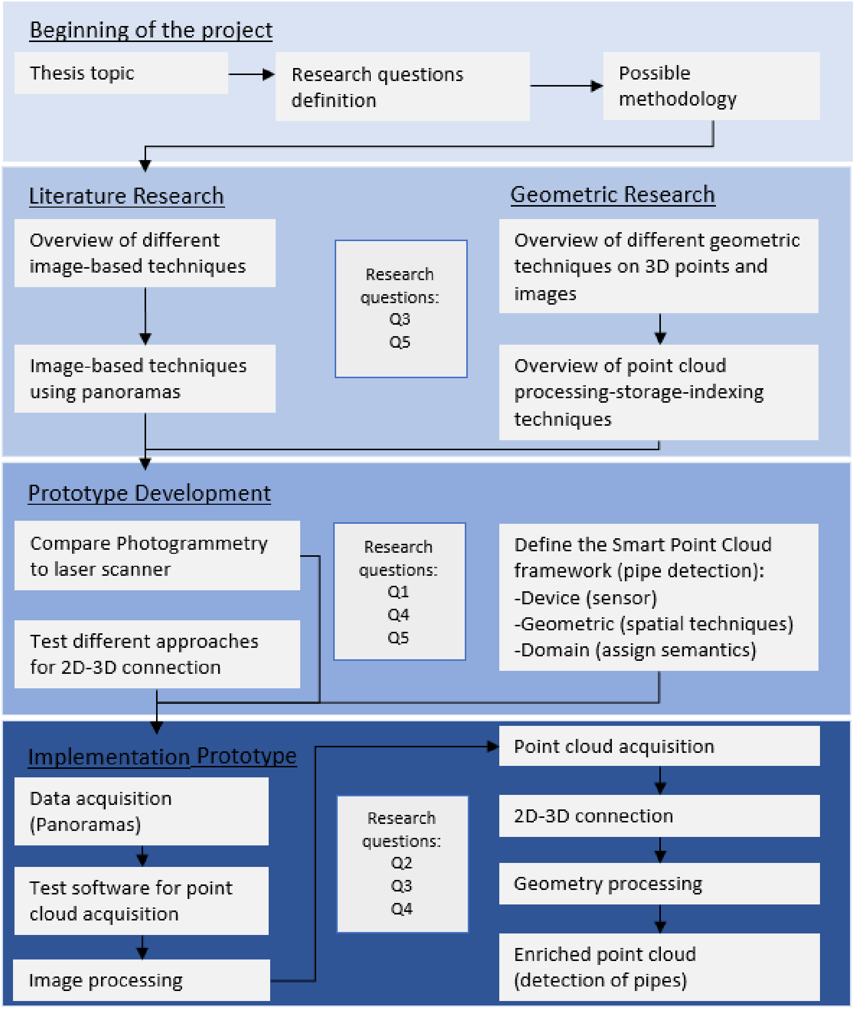

Beginning of the project
Thesis topic
Research questions definition
Possible methodology
Literature Research
Overview of different image-based techniques
Image-based techniques using panoramas
Research questions: Q3 Q5
Geometric Research
Overview of different geometric techniques on 3D points and images
Overview of point cloud processing-storage-indexing techniques
Prototype Development
Compare Photogrammetry to laser scanner
Test different approaches for 2D-3D connection
Research questions: Q1 Q4 Q5
Define the Smart Point Cloud framework (pipe detection):
-Device (sensor)
-Geometric (spatial techniques)
-Domain (assign semantics)
Implementation Prototype
Data acquisition (Panoramas)
Test software for point cloud acquisition
Image processing
Research questions: Q2 Q3 Q4
Point cloud acquisition
2D-3D connection
Geometry processing
Enriched point cloud (detection of pipes)

Figure 1.3: Illustration of the chapters and their roles to the overall research approach.

2 THEORETICAL BACKGROUND

The theoretical background chapter aims to provide the relevant theoretical knowledge to the reader, in order to follow the rest of the subjects that will be described in the thesis document. More specifically, Section 2.1, is all about point clouds, introduces the need of interpreting the real world, definitions and the concept of the SPC framework. In Section 2.2, the concepts that are used for 2D and 3D feature detection are theoretically explained. Section 2.3 introduces the theory of the close-range photogrammetry and how the 3D information could be extracted from 2D images. Finally, Section 2.4 is about the panoramas and their importance in general use.

2.1 POINT CLOUD & SMART POINT CLOUD DEFINITION

Our main goal is to represent the real world in a scene. However, compared to human vision and due to the limited field of view, it is impossible to describe the real world in one scene. For different types of sensors (digital cameras, laser scanners) the data results are usually in form of 2D imagery and 3D point cloud data [Weinmann, 2016].

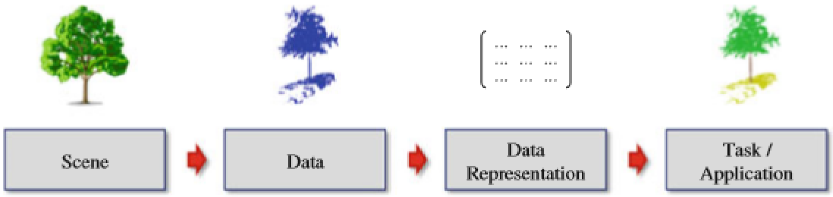


Figure 2.1: Relation between data scene, data representation and data application, in an example using a tree, Weinmann [2016] .

In Figure 2.1, the example shows the relationship between the data source and the final product. But first, the definition of the point and point cloud should be clear for the reader at this section, in order to understand and follow the rest of the methodology. The term point is used to specify a unique location in space. The space could be in 2D, from images or in 3D, from scanned points or reconstructed points. A point in general has no dimensions (length, area or volume) [Weinmann, 2016]. The term point cloud is used to describe a set of data points in a given space. A point cloud, as a data structure, represents a collection of 3D points [Weinmann, 2016]. Our research is mainly based on 3D points that are acquired from 2D images, representing the generated part of surfaces in a scene. This project concentrates on 3D point clouds, as a collection of 3D points which are characterized by spatial X,Y,Z coordinates. Additional attributes are assigned, as it is depicted in Figure 2.2.

A new data structure, SPC structure, is followed by a new concept that combines the attributes of the points with human knowledge. The main idea is to interact with the raw data in a smart way, in order to extract the information that is required based on the user's needs, into an intelligent environment. The challenges which are addressed relating to this idea are many and mainly concern the three fields that are analyzed, data acquisition, object detection and assign semantics. The user-center structure combines different approaches (device, geometric, domain) in a more powerful and robust workflow, Figure 1.1. The device expertise includes the data acquisition and data fusion process. The combination of the different sensors could provide relevant information and a more complimentary scene of interest. At this stage, some specific attributes are attached to points depending on the data acquisition method. Techniques like filtering and noise reduction, could improve the quality, in order to continue with feature extraction. The geometric expertise has to do with the object detection and shape recognition when points have similarities. This stage includes all the techniques that are based on automatic detection, like segmentation and classification. The result is that the points are grouped and could be analyzed easier compared to the unstructured data. The domain expertise is basically the idea going from the feature to the labelled data. This introduces the concept of data association for data mining. A database system is a common and useful way to record the data and ontologies for further analysis. The point cloud needs to be structured retaining spatial and relation information deducted from the previous procedures. For the visualization part, it is vital to work with a well organized structure (spatial clustering/ indexing), to be able to handle the massive data and make queries for validation. This whole workflow describes an intelligent environment creation, where the end-user will directly have access to relevant information based on the proposed methodology process.

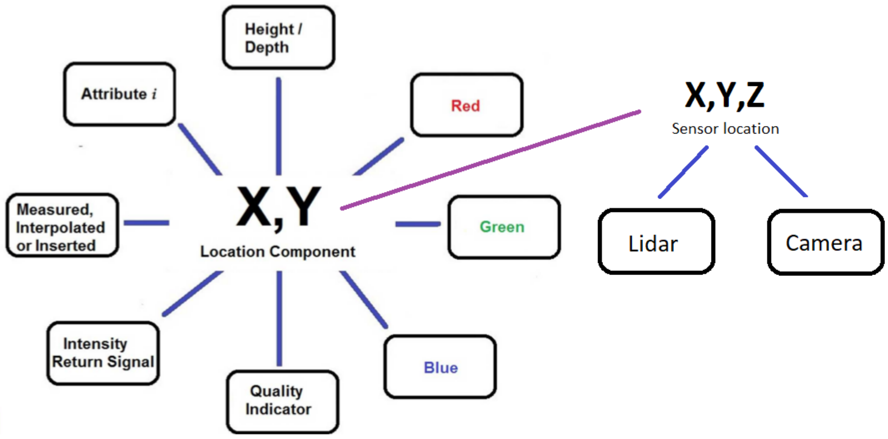


Figure 2.2: Nucleus of point cloud.

Our proposed methodology first, focuses on data acquisition methods which according to our framework, is an important part to attach the needed attributes to the points. Figure 2.2 For Point cloud acquisition, there are two methods that are generally used and they are categorized between active and passive techniques. Passive techniques, for point cloud generation rely on radiometric information represented in the form of 2D images and the accuracy depends on the resolution and quality of the image. This technique is based on information arising from simple intensity measurements per pixel. In contrast to the passive techniques, active techniques use a scanning device, which emits a signal and records the respective angle between the scanner and the object. This thesis project will use the passive techniques in order to generate the 3D point cloud [Weinmann, 2016]. Photogrammetry, as a passive technique, characterized by massive manual data processing, necessary high technical skills and long processing time. Over the past 10 years, improvements in hardware and software have bettered image-based tools and algorithms to the point. Nowadays, image-based techniques and laser scanners can deliver comparable geometrical 3D results. Figure 2.3 depicts the fast improvement of passive techniques, which seem promising for the future.

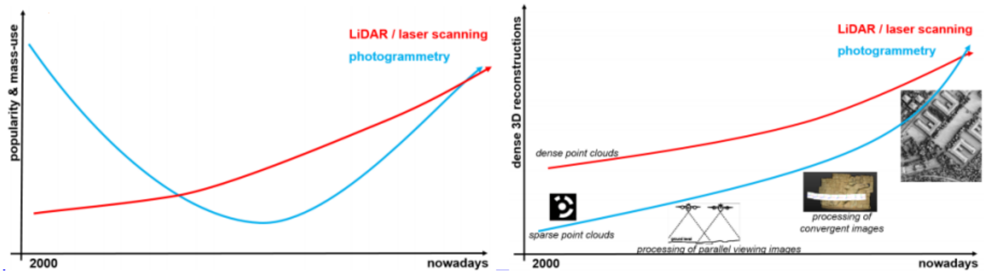


Figure 2.3: How the improvements depict in popularity and final result in 3D reconstruction [Remondino, 2018].

2.2 FEATURE DETECTION

In this Section, the primitive detection methods are mentioned. These are usually used as components for feature extraction in 2D and 3D space. These concepts provide knowledge from the 2D data to 3D points. After these procedures points do not only have (x,y,z) value, but a meaning in 3D environment as well. The main feature detection techniques that are part of our methodology, are explained in this Section. These techniques are associated with the geometric expertise which is mentioned in Section 2.1.

2.2.1 Edge detection on images

There are different filters that are applied to 2D images for detecting edges. Edges are the differences and the changes between the pixels' intensities which are the boundaries of the objects. The purpose of edge detection in general is to significantly reduce the amount of data in an image, while preserving the structural properties to be used for further image processing. The edge detection is the first step for identifying an object on an image. There are three main operators in image processing, the Gradient (Robert, Sobel, Prewitt), Laplacian and Canny operator [Shrivakshan, 2012].
The Canny edge detection algorithm has better performance compared to the other filters with regards to the following criteria:

1. Detection: The probability of detecting real edge points should be maximized while the probability of detecting non-edge points should be minimized.

2. Localization: The detected edges should be as close as possible to the real edges.

3. Number of responses: One real edge should not result in more than one detected edge.

The algorithm is implemented in the following steps:

1. Smoothing: Blurring of the image to remove noise.

2. Finding gradients: The edges are marked where the gradients of the image has large magnitudes.

3. Non-maximum suppression: Only local maxima should be marked as edges.

4. Setting thresholds: Potential edges are determined by thresholding.

5. Hysteresis: Final edges are determined by suppressing all edges that are not connected to a strong edge.

Each step is described with formulas, in Section 4.2.4. The performance relies on the changing parameters which are the smoothing Gaussian filter and the threshold values. The user can change these parameters making this algorithm applicable to different environments. The main advantage is that it performs well, even when the images are noisy. Nevertheless, the calculation of the gradient can cause angle suppression and the whole procedure is time consuming, making people avoid it [Shrivakshan, 2012]. The implementation of this method is briefly explained in Chapter 4, Section 4.2.4 and the results are analyzed in Chapter 5.

2.2.2 Hough Line Transform 2D & 3D

The Hough transform is a technique for feature extraction used in image processing. This technique collects imperfect instances of objects within a certain class of shapes by a voting procedure in cells. This voting scheme is carried out in a parameter space, where the candidate objects are obtained as local maximas in the accumulator space. The Hough transform was developed for the identification of lines in the image, but later works extended the Hough transform to identify positions of different shapes, for example circles. In this thesis, Hough transform is used to detect lines in 2D images and lines in 3D cloud points, obtained by image-based technique using panoramic images. The Hough line transform is a technique that is used in our methodology on 2D panoramic images and directly to 3D points. The concept of this technique is explained here, as this method is used in our framework for feature line extraction, explained in Chapter 4, PROTOTYPE DESIGN.
The Hough Line Transform is a voting technique in parameter space. The main idea is to record all possible lines where each edge point lies and to search for lines that get many votes. At the beginning, the hough space bins are initialized to zero. Then, for each pixel (p) find all the Hough space bins that represent primitives containing p. In the end, find all the bins with the most support. Connection between image (x,y) or space (x,y,z) and Hough (m,b) spaces means that a line corresponds to a point in Hough space, Figure 2.4. For example, to go from image space to Hough space for a given set of points (x,y) it should find all (m,b) that y=mx+b. It was later extended to also work on 3D point clouds. The idea of the Hough Transform in 3D is to discretize the space, giving all the possible direction of lines and to let each point determine to which line it belongs. The direction of the line with the most points means that a line could fit. The theory is the same as it is a voting method but when a third dimension is added the algorithm becomes more complicated because the model parameters that are required are more. Anything that can be parameterized (in a small number of dimensions) could be detected using Hough Transform. The implementation of this technique is presented in Chapter 4, Section 4.2.4.

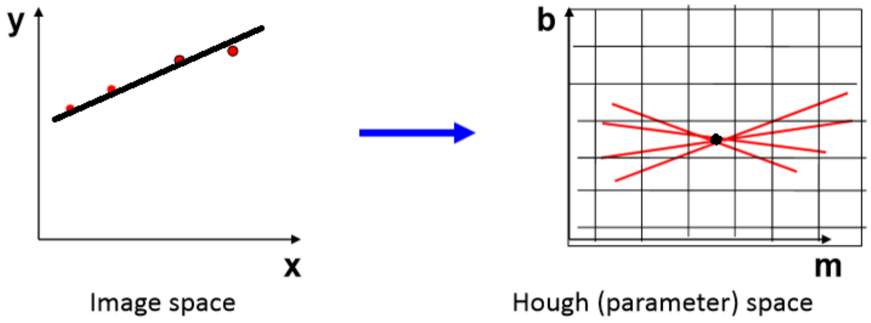


Figure 2.4: Connection between image(x,y) and Hough(m,b) space.

2.2.3 Random Sample Consensus

RANSAC method is another broad technique for feature extraction in 2D and 3D environment. In this thesis, RANSAC technique is applied directly to 3D point cloud for cylinder extraction to compare its performance with the results from our method.
RANSAC technique provides solutions by randomly using the minimum number of points that are required to estimate a specific model. As pointed out by Fischler and Bolles [1981], unlike conventional sampling techniques that use as much data as possible to obtain an initial solution, RANSAC uses the smallest subset from the data that is sufficient to estimate the model's parameters. The input data must not contain outliers and all the data that are part of the solution must agree with the model. The RANSAC algorithm is sufficiently large to estimate the model parameters and which contains no outliers. Then, all inliers will agree with this model. This is done by randomly drawing data subsets, estimating corresponding models and assessing how many data elements agree with each model. The maximal set that is obtained is used as input for a least squares model estimate. Optionally, a minimal threshold on the size of the largest consensus set can be used to decrease the probability of accepting an erroneous model. Thus, to use the RANSAC algorithm requires:

1. A parametric model given the minimal number of data.
2. A method for least squares parameter estimation.
3. A threshold that checks if a given data element agrees with the model.

This technique is technically explained in Chapter 4, Subsection 4.2.9. Its results and the comparison with our results are provided in Chapter 5, Subsection 5.4.6.

2.3 CLOSE-RANGE PHOTOGRAMMETRY & STRUCTURE FROM MOTION TECHNIQUE

Photogrammetry is the science of using 2D images to generate accurate measurements in 3D. To do that, the 3D information that is missing when the image is captured needs to be recovered [Schenk, 2005].
In our project, close-range photogrammetry is used. The type of the method that is used depends on the application and the accuracy that is required for the final product. Figure 2.5 shows the relationship between the accuracy and the result of the product for different methods.

By definition, Close-range Photogrammetry (CRP) is simply photogrammetric data collection and processing where the subject is less than 300 meters away. Several factors have made this a powerful tool suitable for many applications. The strongest of these is the recent advent of 3D photogrammetry, often referred to as "multi-ray photogrammetry": a technique that takes overlapping, stereo-paired photographs and turns them into 3D point clouds [Schenk, 2005].
Photogrammetry is a 3D measurement technique which uses central projection imaging as its fundamental mathematical model. The shape and position of an object are determined by reconstructing bundles of rays in which, for each camera, each image point P', together with the corresponding perspective centre O', defines the spatial direction of the ray to the corresponding object point P, Figure 2.6. Providing the image geometry within the camera and the location of the image system in object space are known then, every image ray can be defined in 3D object space. This is called internal orientation which defines the relative position of the images.

From the intersection of at least two corresponding, named homologous, in separated image rays, an object point can be located in three dimensions. In stereophotogrammetry two images are used to achieve this but in multi-image photogramme-

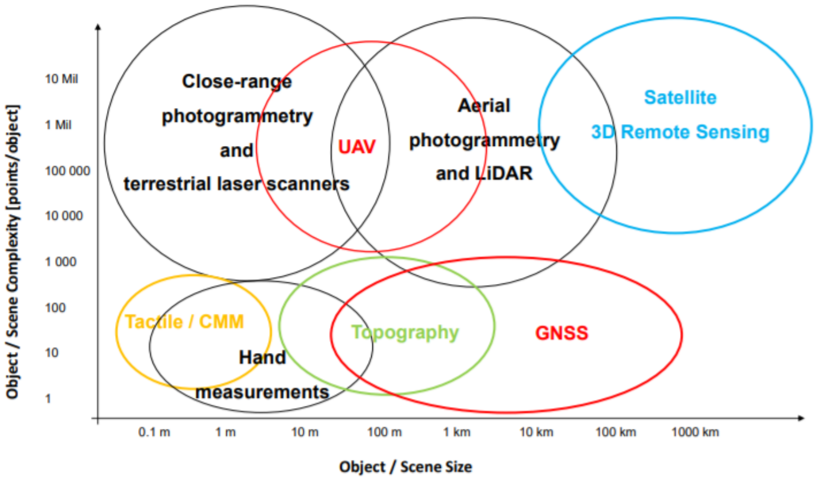


Figure 2.5: Relationship between object size and accuracy for different measurement methods [Remondino, 2018].

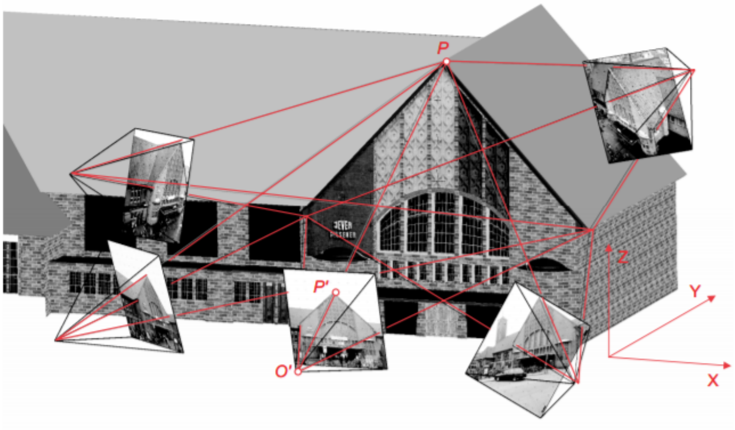


Figure 2.6: Principles of photogrammetry.

try, the number of images that you can use is unlimited. The more images the better and more accurate the result. There are four main operations in photogrammetry:

- Internal orientation (camera calibration)
- Relative orientation
- External orientation
- Absolute orientation

The operations of internal orientation and relative orientation can be done in a fully automatic way (eg. SFM). Internal orientation is the process by which, a three dimensional coordinate system is associated to an image. It implies the estimation of the distortion parameters. For the internal orientation, the camera parameters are needed. This involves the calibration parameters which are: the focal length, the coordinates of the principal point, the coordinates of the judicial marks and the radial distortion.

To continue with the relative orientation, a successful interior orientation is first required. Relative orientation is the determination of the relative position and orientation between cameras.

The absolute orientation requires a successful interior and relative orientation. External orientation corresponds to the correct position and orientation of a camera with respect to a 'world' coordinate system. Absolute orientation corresponds to the correct position, orientation and scale of a three dimensional model with regard to a real coordinate system. Both operations require control data. The absolute orientation involves the relation between the model and the actual 3D environment, known as georeference.

Today, the available technology increases the automatic procedure of the photogrammetry software that is available. The SFM technique implementation is based on automatically solving the image orientation and 3D structure of the scene using connected images. In Figure 2.7, the structure of SFM algorithm is explained. First, the set of feature points are identified across the images in order to estimate the related position of the images based on scale, orientation and affine transformations [McCann, 2015]. Next, after the correspondence between the features, the internal (focal length, distortion) and the external (rotation, translation) parameters are estimated. Before starting to estimate the 3D location of each feature, it is important to mention that a first pair of images with lot of matches and a large baseline is suggested, for better reconstruction (step 3 in Figure 2.7). Then, starting from this pair of images a bundle adjustment is computed and next, all the images are used until, each feature point is triangulated providing a robust position estimation [Golparvar-Fard et al., 2011].

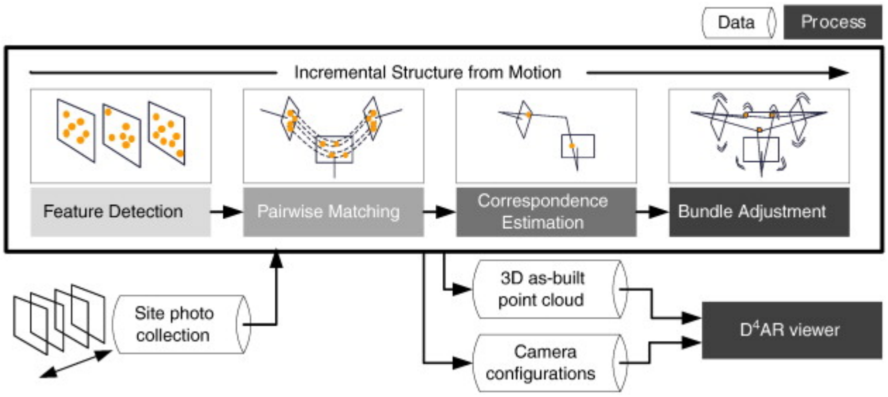


Figure 2.7: Structure from motion (SFM) steps -[Golparvar-Fard et al., 2011].

2.4 IMPORTANCE OF PANORAMAS

Panoramas are images that provide a large amount of data comparison to conventional images with more restricted fields of view [Huang et al., 2008]. As the computers become faster, more applications and studies using panoramas became widely accessible. Some aspects that demonstrate the importance of the panoramas are:

Immersion. The large field of view, providing a more complete and continuous is the importance of the panoramic images which possibly enhanced by 3D stereo viewing.

Realism. A photo realistic quality of a 'virtual word' can be achieved by a dense set of panoramas. With VR systems, stereo visualization can also be implemented with free navigation in virtual panoramic world. A panorama is possible implementation of a VR system. To contrast this with other VR systems, it is called image-based VR [Huang et al., 2008].

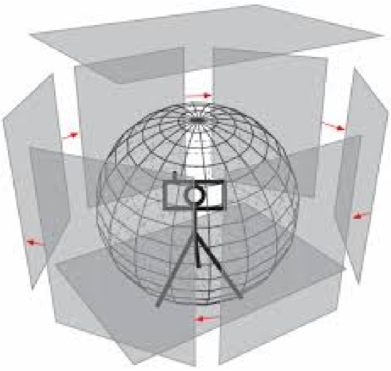


Figure 2.8: Stitching panoramas.

Simplification. Stereo 3D reconstruction can be based on 360°panoramic images comparing the complicated process of merging multiple depth images.

Localization. Equipment with panoramic vision can recognize location by comparing an incoming panorama with memoring maps. This strategy improves robustness because of the further information, provided by a larger field of view.

Compression. A panorama composed from image sequence of a video camera.

A panoramic image is a wide field of view image with a full view of 360°. Panoramas can be created on an extended planar image surface, on a cylinder, or on a sphere. Traditional panoramic images have a single viewpoint, also called the 'center of projection'. Panoramic images can be captured by simple digital cameras using fish-eye lens or simple stitching methods for generating panorama images.

A fish-eye lens are wide-angle lens that produces distortion intended to create panoramic image. Fish-eye lenses achieve extremely wide angles of view. Instead of producing images with straight lines of perspective (rectilinear images), fish-eye lenses use a special mapping (like sphere) which gives images a characteristic convex non-rectilinear appearance, Figure 2.9. The major drawbacks of these images is that have low resolution close to center, severe distortions, inefficient usage of images, These drawbacks makes this type of panoramas not suitable applications where high accuracy and resolution is required.

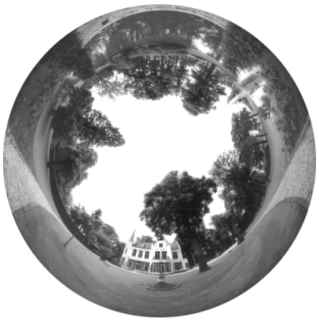


Figure 2.9: Spherical image using fisheye len [Luhmann, 2004]

High resolution and image accuracy is of major importance for the photogrammetric use of digital panorama cameras. Both objectives can be reached if the complete imaging geometry is modeled and calibrated appropriately [Luhmann, 2004]. The simplest way of generating a digital panorama is by 'stitching' together a set of overlapping digital images [Huang et al., 2008]. Today, stitching is a very common technique for commercial and professional panoramas 2.8. Image stitching is the process of combining multiple photographic images with overlapping fields of view to produce a high resolution panorama. In the methodology, we are working with digital panorama camera. The panoramas are recorded with fish-eye lens and the camera is mounted on a rotating panoramic head and rotates around the projection center.

3 RELATED WORK

The related work chapter aims to provide the available methods currently done in the three different parts (device, geometric and domain) related to our methodology. Therefore, the Chapter is organized as followed: Section 3.1 provides a short overview of existing point cloud data acquisition methods using spherical images. Then, Section 3.2 focuses on all the applications that have used the SPC framework. Sections 3.2.1 and 3.2.2 are concentrated more at our prototype and provides all the different techniques that are implementing about pipe detection direct on point clouds. Also, this Chapter provides an overview about point cloud management, indexing, Sections 3.2.3 and 3.2.4 and the possibilities representing the point cloud in VR, Section 3.2.5.

3.1 IMAGE BASED TECHNIQUES FOR POINT CLOUD GENERATION

Generally, a lot of research has been done and many applications have been created, using image based techniques for dense matching, documentation and survey reasons. Image based techniques are low-cost, quick, with a wide variety of field applications, especially referring to cultural heritage. Due to the urgency of a detailed documentation and representation, it is becoming a popular method. The utility of the panoramas for enhancing the quality of the documentation is not yet broadly applied. Moreover, there are some researches that have been done which use panoramic images. There are two methods that are used for 3D reconstruction: the Spherical Photogrammetry [Fangi, 2012] and Structure from Motion method, which is now a wide spread technique able to acquire low-cost feature detection and modeling representation [Snavely et al., 2006].
Multi-Image Spherical Photogrammetry (MISP) approach uses the spherical photos for a quick geometric method, data acquisition for documentation and reconstruction, [Fangi, 2012]. There are many advantages that lead to using this technique such as, the speed, the low-cost and the completeness of the documentation [Fangi, 2011]. This technique is also, named emergency photogrammetry, as it is mentioned in d'Annibale et al. [2011]. Because due to all of these advantages can be regarded as one of the best methods to deal effectively, quickly and inexpensively the emergence of the protection and preservation of our architectural heritage. The main disadvantage of using panoramas is the lack of stereoscopic. It is difficult to identify many points in a pair of panoramas, but an attempt to tackle this problem has been carried out in d'Annibale et al. [2010]. First, by creating the point cloud by open source online services and then, to project back the oriented panoramas to allow edit and to correct the model to the best coincidence, following the principles of photogrammetry.
Three-dimensional dense matching construction using SFM is one of the most spread image based methods. The SFM implementation is based on automatically solving the image orientation and 3D structure of the scene using connected images. The necessity of using multiple photos and the need of optimizing the results in accuracy has made large progress [Snavely et al., 2006]. Wu [2013] introduced a new strategy to improve the speed and the accuracy by re-triangulating the feature matches. An interesting approach has also been proposed by [Cefalu et al., 2017], using hierarchi-

cal SFM in combination with global image orientation techniques. Using panoramic images is possible to improve SFM results with a priori acquisition of high resolution and geometry of panoramic images [d'Annibale, 2011]. A first approach using 720°panoramas to create 3D-point cloud is presented in [Lee and Tsai, 2015]. Within a panorama, the panoramic images cover the zenith and nadir is usually called a 720°panorama. The result is based on Visual SFM and CMVS toolkit for a close range photogrammetry at a cultural heritage site at Kinmen.
An integration of the two methods seems to have been attempted in [d'Annibalea and Malinverni, 2013]. Their proposed framework involves the combination of different techniques for high accuracy in documentation and provision of new solutions for VR navigation, Figure 3.1. Furthermore, d'Annibale [2011] proposed to combine the advantages from the two methods and to compare the orientation results deriving from the two mentioned technologies. The objective of this research is to speed up the 3D reconstruction.

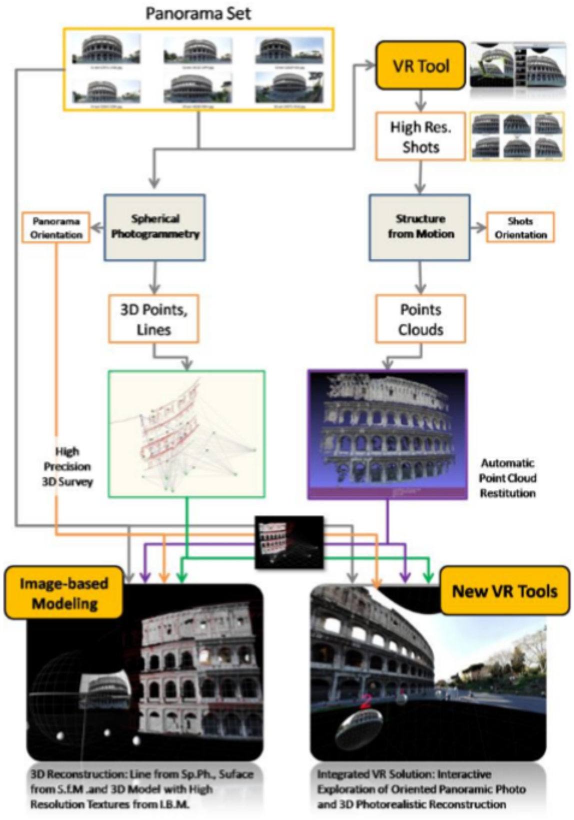

Panorama Set
VR Tool
High Res. Shots
Panorama Orientation
Spherical Photogrammetry
Structure from Motion
Shots Orientation
3D Points, Lines
Points Clouds
High Precision 3D Survey
Automatic Point Cloud Restitution
Image-based Modeling
New VR Tools
3D Reconstruction: Line from Sp.Ph., Suface from S.f.M .and 3D Model with High Resolution Textures from I.B.M.
Integrated VR Solution: Interactive Exploration of Oriented Panoramic Photo and 3D Photorealistic Reconstruction

Figure 3.1: Two different 3D techniques(MISP and SFM) into an Image-based Modeling solution.

There are model cameras with a wider field-of-view by using a non-linear mapping function. For example, to model omnidirectional cameras, Kannala and Brandt [2006] parametrize the point by its spherical angle coordinates and apply a radially symmetric function based on a polynomial of the angles. The distortion model must also, be adjusted to suit this generic model. In the case of an omnidirectional camera, it has been recommended to model the distortion as a function of the spherical angle parameters [Castro, 2015].
Panoramic Spherical Photogrammetry (PSP) makes use of the spherical panoramas. They are taken from the same point and stitched together on a sphere [Szeliski and Shum, 1997] and then are mapped on a plane using the type of cartographic projection (Snyder projection). The main advantages of the spherical panorama concern

the field of view. That can be 360°x180°and in high resolution. However, its the weakest point is the orientation procedure that is difficult to perform, this is the main goal that they are tangling with, in Fangi [2015].
Some projects had already been implemented using panoramic images for 3D point cloud creation, using commercial photogrammetry software. In Kwiatek and Tokarczyk [2015] and Kwiatek and Tokarczyk [2014] the panoramas are taken through an panoramic video. They extract the panoramas from the immersive video and they use them for dense point cloud creation. The result seems promising and the comparison with the point cloud which is generated from laser scanner is analyzed.
The most common applications that are using photogrammetry with panoramas as input images, for survey documentation, are those that are related with monuments and buildings. This makes sense as the color and the details are very valuable for survey documentation, like in Pisa et al. [2010].

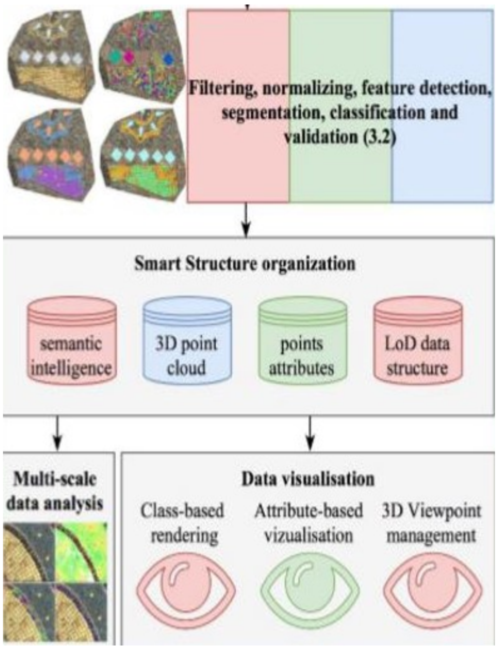

Filtering, normalizing, feature detection, segmentation, classification and validation (3.2)
Smart Structure organization
semantic intelligence
3D point cloud
points attributes
LoD data structure
Multi-scale data analysis
Data visualisation
Class-based rendering
Attribute-based vizualisation
3D Viewpoint management

Figure 3.2: Smart Point Cloud workflow to assign Archaeological semantic injection [Poux et al., 2017b]

3.2 SMART POINT CLOUD

The framework for this research application follows the structure of a SPC. The first contribution is developed from a PhD research by [Poux et al., 2017b]. The objective of this research is to examine the problems and the challenges that the new framework will face, providing a review of the applications and the approaches that are related to every expertise that the framework is based on the first implementation according to this framework is structured by [Poux et al., 2017b]. First, it tries to extract information from the point cloud after the geometric processing of a Tesserae (segmentation, feature extraction) and then, to classify the segmented points according to their characteristics, in order to assign semantics for gaining Archaeological knowledge. The structure of the smart point cloud framework is presented in Figure 3.2, where the relations between the three domains are clearly depicted. The project remains the same but its focus now turned to the connection between the domain knowledge and the classification structure. By doing this, a semantically rich point cloud data structure is provided by a real time Web application [Poux et al., 2017c]. Another important contribution is made by [Rusu et al., 2008], where

this article provides a structure about creating a 3D house plan where a kitchen is turned into a real world representation.
The number of applications and the field of this structure could become too wide so, the related work is focused on our research application. The objective of our research application is to apply the SPC framework for object detection, in our case the main object are the pipes. Segmentation through different approaches is provided, using point cloud, as raw data in combination with image-featuring extraction from the panoramic images. Then, an intelligent environment where the user could identify pipes, as one object in an industrial environment is described. The main sections that the application is focused on are described below.

3.2.1 Feature detection on point cloud

The automation point cloud segmentation is one of the most important processes. Acquiring the geometry from the actual point cloud, many automation algorithms already exist, RANSAC [Fischler and Bolles, 1981], HT [Hough, 1962] and Region Growing, are highly used for shape extraction. Particularly, many implementations have been done in order to extract cylinders into industrial areas combining or using these methods. An interesting implementation is presented by [Chaperon and Goulette, 2001] using RANSAC and the Gaussian image for cylinder extraction, Figure 3.3. Another algorithm for pipelines extraction is presented by [Rabbani and Van Den Heuvel, 2005], where HT for automatic pipelines detection is presented using the Gaussian sphere as well. The Gaussian sphere is introduced by [MPd, 1976]. This sphere provides the great circle from the intersection of the sphere and the plane, passing through the origin of the cylinder, the normal of this plane is the axis of the cylinder. Rabbani et al. [2006] developed a region growing algorithm for pipeline segmentation in an industrial area. This algorithm is based on the smoothness constrain, computing the normals and the connectivity using the K-nearest or fixed distance algorithm. A highly relevant work is provided by [Su et al., 2016], using octree-structure for segmentation the point cloud data in industrial applications. In this framework, the point cloud is voxelized based on octree and then, a segmentation for industrial primitives (pipes, vessels) is computed based on orientation, curvature and connectivity criteria. There are many cylinder fitting algorithms that use the point cloud data without outliers. Nurunnabi et al. [2017] concentrates on incomplete 3D point cloud data with the presence of outliers. A robust cylinder fitting algorithm is proposed in this article using robust statistical approaches (robust PCA and robust regression) and an existing circle fitting method, Figure 3.4. These articles tackle with the problems of automatic detection and shape fitting which are essential methods and procedures in industrial design and engineering

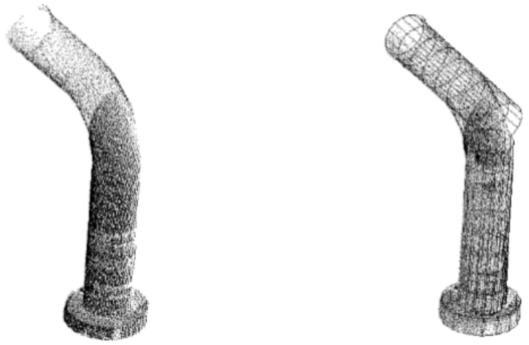


Figure 3.3: The results using a random sampling method and the Gaussian image, [Chaperon and Goulette, 2001]

requirements.

Indeed, much research has been done about cylinder fitting in 3D data. As it is mentioned above, two algorithms are mainly working directly to raw data without pre-processing so, every part is well defined by an approximate shape. RANSAC and HT are two algorithms, well known voting techniques that are used for shape detection and fitting in raw 2D & 3D data [Schnabel et al., 2007]. These two algorithms due to their complexity especially in 3D environment, discourage people to use them [Nurunnabi et al., 2017]. The main differences between these two techniques are that the HT determines the point of the parameter subspace of the model and increases the votes in the Hough space for this model. On the other hand, RANSAC, for the given model a minimal number of the points are randomly selected from which the model can be computed. According to research between these two techniques and to the articles for cylinder detection, table 3.1 presents the most important advantages and disadvantages of each method. It further demonstrates how they are each suitable depending on the type of application. In the case, of cylinder extraction, considering the fact that cylinder needs five degrees of freedom, it makes HT technique impractical. This is due to the fact that this method has difficulties fitting objects with more than three parameters. Alternatively, RANSAC technique can perform well with more parameters, assuming there is a small number of cylinders in the scene. Thus, the presence of different cylinders with different radius along the same orientation can cause difficulties in the final detection. In Section 5.4.6, RANSAC method is applied direct to the point cloud to compare the results with our prototype results.

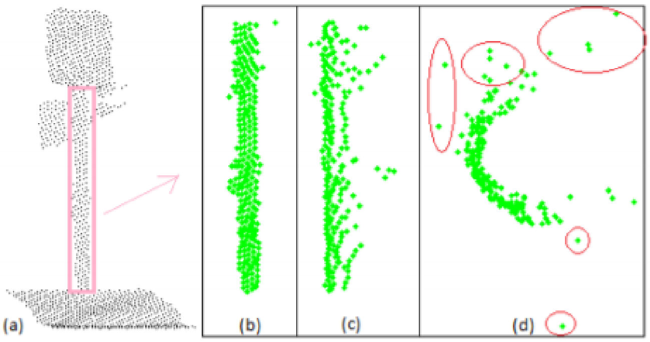


Figure 3.4: Cylinder fitting using (a) laser scanned incomplete point cloud data, (b), (c) a cylindrical segment of the cylindrical data (d) cylindrical data with outliers, [Nurunnabi et al., 2017].

Table 3.1: Pros and cons of each method.

Method	Pro's	Con's
HT	Robustness to noise	Time consuming
	Detect multiple instances	Space complexity
		Impractical with more than 3 parameters model
RANSAC	Simple	Time consuming
	Robust to outliers	Not applicable to complex dataset
	Applicable to large number of parameters	

3.2.2 Feature detection from imagery

HT and RANSAC techniques first, were implemented for shape recognition and feature extraction in 2D, images. An example of this procedure is proposed by [Pavlidis and Liow, 1990], in which an integration of the two algorithms, the Region Growing and edge detection, is presented. Including images for better segmentation results is interesting, because they provide a better source of information (clear edges, distinct boundaries of the object) [Rabbani, 2006]. A combination of these two methods is used by Vetrivel et al. [2015], where an initial detection of buildings in the 3D point cloud is combined with segmentation of the image space acquired from Unmanned Aerial Vehicles (UAV). The image segmentation could be carried out by also using other characteristics, like the color and the spatially connected pixels for feature connectivity. Wang et al. [2014] combines the 2D edge extraction from images and the 3D edge detection, in order to merge these methods and make use of their advantages for 3D edge extraction. In our methodology, Hough line transform is applied for line detection to the panoramas to extract the boundaries and then, to combine this 2D information with the 3D points.

3.2.3 Domain Adaptation

The domain knowledge refers to the way of adapting knowledge to a group of points according to spatial characteristics. Enabling semantic injection to point cloud is the first part of creating an intelligent environment, a smart point cloud. Poux et al. [2017a] is based on the work of Poux et al. [2017b] and proposes a data model construction to link knowledge to segmented or classified points, Figure 3.6. This model consists of three meta-models according to LOD, providing a structure point cloud data that is able to manage and store the points in a database system while assign semantics. LOD is the level of the information that is provided in different scale, Figure 3.5. Organizing point cloud data in LOD's/importance levels is an approach to manage large data sets in order to have a better manipulation and structure and in the end to assign semantics [van Oosterom et al., 2017]. These LOD for point clouds are proposed for handling massive multidimensional point cloud datasets for data ingestion, data management, data analytics and visualization. The proposed LOD model could be discrete (multi-scale) or continuous (vario-scale) [van Oosterom et al., 2017].

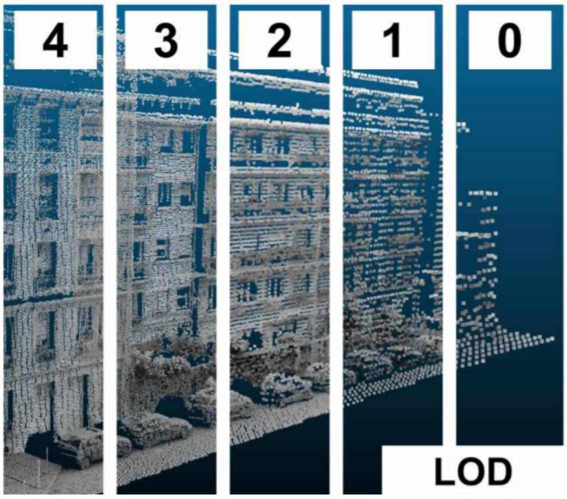


Figure 3.5: Illustration of different LOD on point cloud [Cura et al., 2016]

An important contribution is made by [Rusu et al., 2008] who provides a structure about creating a 3D object map in indoors applying data mining to a kitchen. In Weinmann et al. [2015], semantics are applied to a point cloud according to the optimal neighbors, through classification and segmentation procedures. There are some specific data models, like CityGML, IndoorGML, Building Information Modeling (BIM) which use standard structure for representing the real world, by deriving information from a direct source. For example, Xiong et al. [2013] makes an important contribution by automatically converting the raw 3D point data into BIM. The algorithm automatically, identifies the indoor features (windows, ceilings, floors).

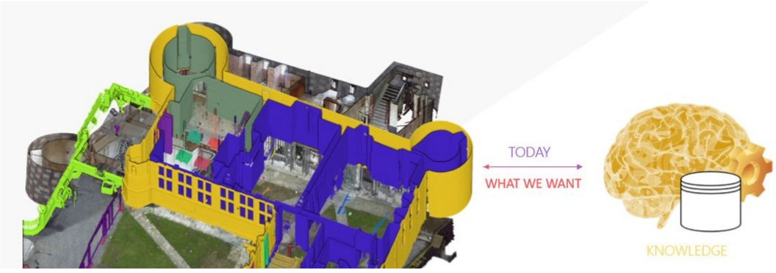


Figure 3.6: Smart Point Cloud workflow to assign semantics [Poux et al., 2017a]

Brilakis et al. [2010] also, introduces an automated generation of BIM models based on video and laser scanner data. This provides a general approach whose goal is automatic build modeling. Krijnen and Beetz [2017] suggest an extension to the Industry Foundation Classes Industry Foundation Classes (IFC) model to integrate point cloud dataset. It is explained how the semantic relationships have been favored over compression using binary serialization to the raw source data. The binary serialization is able to handle millions of points, while spatial and semantic subsets can be extracted due to the hierarchical storage model and to the association with the building elements. Another interesting article is Park et al. [2013], where a conceptual framework is presented and tries to integrate the ontology and the AR with the BIM. These articles are excellent examples explaining why metadata should be linked with the 3D data. This is not an easy task especially when a general rule is thought to be applied to all the models. Each application requires its own semantic and geometric information. According to our application, the first attempt to define a common schema of the pipelines has been done by Westdijk [2015], where a general standard data model is defined in order to structure and manage the pipeline network. Also, some international standards were mentioned in Westdijk [2015].

3.2.4 Data Base Structure & Indexing

Massive point cloud necessitate a suitable way to store, organize and manipulate the points. Making a demand of creating a DBMS that could efficiently manage the data model. van Oosterom et al. [2015] provides a general review with all the options for point cloud management, comparing also, different data management solutions (Oracle, PostgreSQL, Rapidlasso LAStools), providing improvements and optimal ways for indexing point clouds. Furthermore, point cloud is mentioned as a third type of spatial representation of data with vector and raster data, first and second type, respectively. There are already existing Point Cloud Data Managements Systems (PCDBMS) techniques that allow to store and manage the data. Nevertheless, there are still some limitations providing an efficient way to present and provide them through semantic queries. In order to store the data and provide analysis to ontologies, point cloud need to retain the information from the classification and segmentation techniques. Cura et al. [2015] provides a complete and efficient point cloud management system based on a database server, that works on groups of points rather than on individual points using relational DBMS. This framework is designed to solve all the needs of point cloud processing (loading, compressed storage, filtering, easy data access and exporting, integrated processing). Zlatanova [2006] elaborates on current possibilities of DBMS to maintain 3D data. Spatial data types and spatial operations reflect only on 2D features, but are part of 3D space. According to this paper, they are managing to store and visualize the 3D features. For data visualization and for other types of spatial data processing, it is essential to work with a structure that is able to handle billions of points and queries so,

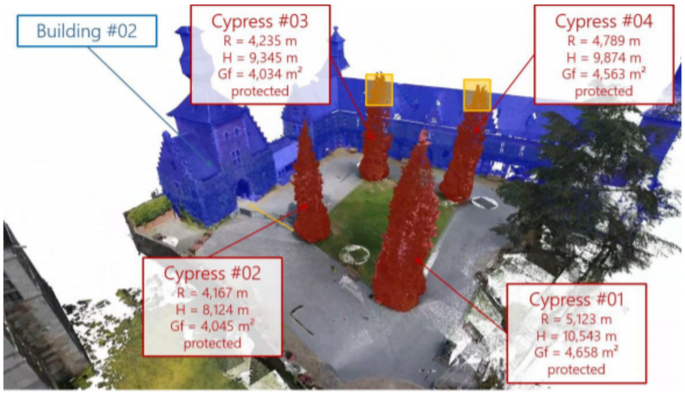


Figure 3.7: Example of a point cloud spatio-semantic query,[Poux et al., 2017a]

efficient indexing techniques and LOD structures are required. The indexation of the 3D point cloud concerns the spatial subdivision through the space through different approaches. Octree structure is the 3D of the quad-tree that is proposed by Yang and Huang [2014], providing interesting changes regarding point cloud resolution, density and distribution. The paper Richter and Döllner [2010] presents a structure to manage massive point cloud including database integration, interactive rendering and visualization. The advantages of using point clouds are many, the most important ones however are that they are LOD adaptive, give fast update and extract information.

3.2.5 Virtual and Augmented reality

VR is the immersion that the 3D environment is fully artificially generated by the computer. VR is usually made possible by wearing a VR helmet or google glasses and the 3D model. VR can entirely teleport you somewhere else. It immerses the user by stimulating their vision and hearing and by making them believe that they are in another not existing environment.

AR overlays virtual objects of the real-world environment. Users see and interact with the real world while digital content is added to it. Pokemon Go is an example of augmented reality where millions of people all over the world have been rushing with their smart phones in search for small virtual creatures.

Both VR and AR will soon become mainstream. The fact that point clouds offer a 3D environment trying to represent the real world, it leads to explore the usage of 3D point cloud in VR and AR settings. The aim of this article Mancera-Taboada et al. [2011] is to build a 3D model of San Martin's Church, Segovia from point clouds generated from terrestrial laser scanner. The mobel will be the basis for the development of a platform of virtual and augmented reality, giving easier access to disabled people. In Bruder et al. [2014] work, they try to combine VR visualization and 3D point clouds. Such Point Cloud Virtual Environment (PCVE) represents the external environment that is usually acquired by 3D scanners. They present an application scenario, in which a mobile robot captures 3D scans of a terrestrial environment and is automatically registered to a coherent PCVE. This virtual 3D reconstruction is displayed in an immersive virtual environment.

Liu and Boehm [2014] introduces an interactive segmentation method for segmenting point cloud through an algorithm based on interaction to assign a background and highlight wanted objects via user drawing. Stets et al. [2017] demonstrates an application for visualizing and labeling point clouds in VR. In the application, it is possible to label large sets of point clouds in a quick and prototype way. These interactions provide a direct immersion in a coherent structure, bridging VR with 3D data.

4 PROTOTYPE DESIGN

This Chapter contains the proposed prototype for pipe detection, following the SPC framework. Section 4.1 deals with the data acquisition and the quality of our initial generated point cloud. The second part, Section 4.2 focuses more on spatial techniques including the algorithms and the parameters that are used and how the actual result is produced applying the SPC framework on different datasets. The first Section is strongly related to the second, as it is the initial step for our SPC framework.

4.1 IMAGE BASED METHODOLOGY & DATA ACQUISITION

The procedure of point cloud acquisition has to firstly be defined in order to proceed to the main application. This is the first part of our main SPC framework, which includes the domain expertise. According to the type of the acquisition procedure, the attributes of the points are defined. The main reason that image-based technique is used, is because some valuable attributes from the points that are created from the images, try to be related with the pixels. This is the main idea, testing whether this technique could enhance the object recognition and to assign from the beginning some knowledge to the points.
Figure 4.2 displays a flowchart in which the steps for point cloud acquisition are described. The point cloud creation is based on panoramas, which is not a common way for image-based techniques. It is important to focus at this stage, as it will define our final result/product. Before testing the existing software, stitching the panoramas and project them to equirectangular projection is required. The panoramas are recorded with fish-eye lens and the camera is mounted on a rotating panoramic head and rotates around the projection center. Furthermore, high resolution of panoramas is needed, as it is mentioned in d'Annibale [2011], for better accuracy. According to the Figure 4.2, after the image procedure some software are going to be tested, like Agisoft Photoscan, Context Capture, Pix4d, as in Nikolov and Madsen [2016]. These software are not open sources, producing some limitations for the data generation (licence expires, no data extraction, limitations on input data, unknown algorithms/settings). Micmac is the only open source service for 3D reconstruction, using the SFM algorithm. Many applications have used it, like in Shults et al. [2017] and Agarwal et al. [2011]. Generating dense 3D reconstructions using the Multi View Stereo algorithms appears that the majority of the available packages are based on the Bundler package, a SFM system for unordered image collections developed by [Snavely et al., 2006].
There are many software/packages that can create dense point cloud from images. However, not so many that could use panoramic images, as spherical images. The fact that the panoramas are used for point cloud generation and not the initial images, is because it is tested how the whole procedure could be faster or minimize the time and the amount of images. Another advantage is that one panorama offers 360°field of view that could be used for object and boundaries extraction. From one panorama you can observe and full detect the objects more easily than from one plain image that covers part of the object. This advantage is one of the main reasons that our concept is based and it will be investigated, Figure 4.1.

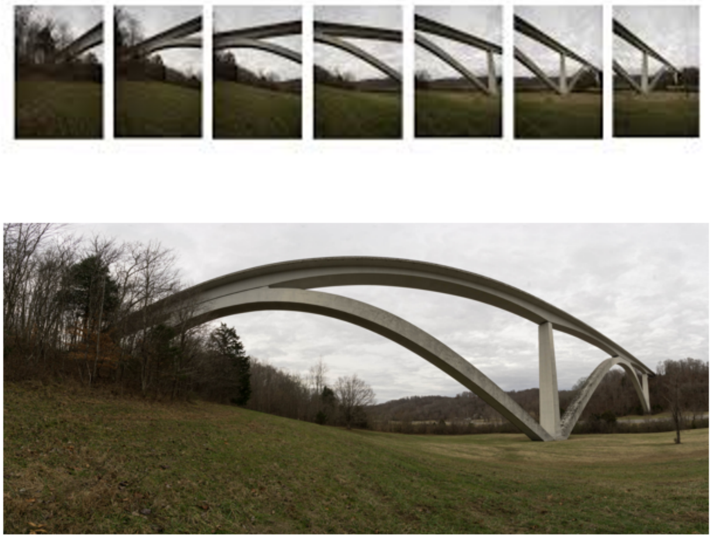
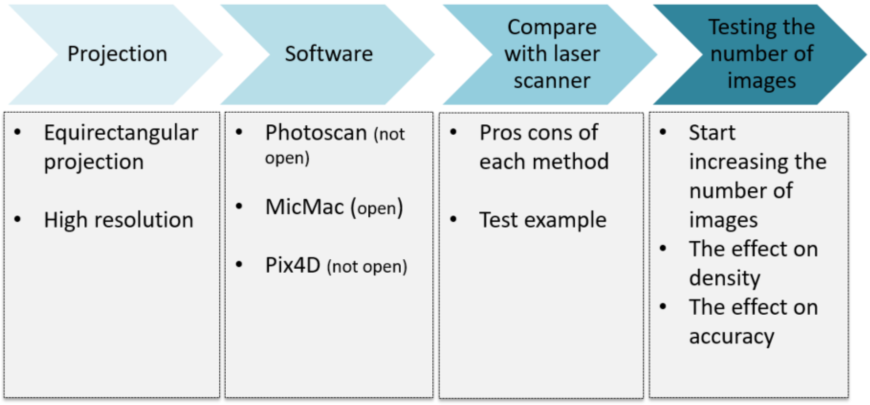


Figure 4.1: Object detection.

The comparison between the point cloud from the available software and the point cloud from laser scanner could be done comparing the density between the one for the image-based technique and the one from the laser scanner. It is known that the density of the laser scanner varies from the characteristics of the laser scanner that the company offers and that scanner has its own characteristics. The density of the point cloud generated from the images differs. There are many parameters that can influence the final point cloud. The evaluation of the result depends on many factors that are analyzed in Section 4.1.1. A theoretical comparison will take place at this chapter and more statistical values will be available in Chapter 5, including an example, comparing the two methods.



Projection	Software	Compare with laser scanner	Testing the number of images
• Equirectangular projection • High resolution	• Photoscan (not open) • MicMac (open) • Pix4D (not open)	• Pros cons of each method • Test example	• Start increasing the number of images • The effect on density • The effect on accuracy

Figure 4.2: Image Based Methodology.

4.1.1 Imaging Vs Ranging

There is a lot of research that have already been done comparing the two methods. Each method has its own advantages and disadvantages. The method that is more suitable always depends on the application and the accuracy that you want to achieve. Also, the equipment and the budget play important role in order to find the best combination to achieve the best result. Many times the combination of the

two methods is the preferable choice.
Generally, laser scanning is used, if you need a high level of accuracy over a large space. Photogrammetry is a better tool if you are documenting smaller spaces and are looking for less accuracy but more visual photo realism. This is why photogrammetry is mainly used for the creation of environments for games and movies. It is also very common in archaeological community, for cultural heritage 3D documentation, because it allows more realistic textures for a lower price than laser scanning. On the other hand, laser scanning is more applicable to industries because of its level of accuracy over a large distance. Engineers are not really interested in textures but more into accuracy errors over distances. Laser scanner manufacturers publish a number of accuracy specifications for their products, including angular accuracy and beam divergence. In laser scanners, some of the factors that affect accuracy are largely independent of distance like the range accuracy, while others are mainly proportional to distance, like the size of the beam footprint and the angular accuracy. The main factors that influence the accuracy of the final result are:

- The accuracy of the distance measurement derived from the laser beam.
- The divergence of the laser beam as it travels from the scanner to the surface.
- The accuracy of the angular measurements of the scanner.
- The accuracy of the registration of more than one scans and reference the 3D data in the appropriate co-ordinate system.

The main factors that determine the accuracy of the final point cloud using digital photogrammetry are:

- The area that covered by one pixel in the image, called pixel size. The size of the pixel on the ground is determined by the focal length of the lens used and the distance from the camera to the object. The pixel size and the lens of the camera are the main factors that define the accuracy of the final result.
- The relationship between the distance between the cameras and the distance from the cameras to the object.
- The accuracy of the georeference means the registration with the actual coordinate system.

The question which technique is better than the other is a controversial one and can not be answered in general. It will be a mix of many factors such as the type of the object, the applicability, economical support, client requirements, etc. Table 4.1 is summarizes some of the advantages and disadvantages of each method. In many cases, a combination of both techniques is usually recommended.

Table 4.1: Comparison of features: Imagery vs Ranging.

Point cloud from laser scanner	Point cloud from images
No color information	Good color information
High cost technique	Low cost technique
Automation in data capture	Lower automation in data capture
Light is not required	Light is required
No easy for data capture	Easy for data capture
Time consuming on field	Less time consuming on field

By knowing the characteristics of the laser scanner that the company uses, the density and the accuracy of the point cloud can be easily estimated on a specific area when taking into account some initial constrains. In order to cover the object that is required sufficiently, scanning the object more than one time is needed from different angles. This leads to definitely getting the final point cloud more dense and noisy. The outliers of the point cloud can be caused by different sources. Outliers are points that are not part of surfaces or do not belonging to local neighbors. The main part where that noise could exist is close to boundaries. This occurs because of boundaries occlusion. The laser beam has a shape of an ellipsoid and when it hits on the boundary of an object it splits into two, creating random points between the object that is front and behind. Other important sources of noise are the reflectanse of the surface and the multipath reflection [Sotoodeh, 2006].

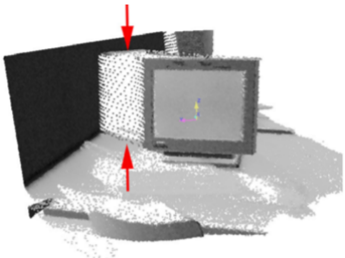


Figure 4.3: Outliers at the boundary of an occlusion between the edge and the back wall are the result of the noise problem. [Sotoodeh, 2006].

In our case, image based technique is used for our application using panoramic images for object detection. Throughout our methodology, it is investigated, if the connection between image information with the point cloud could improve the object detection results. Images offer a clear boundary detection and through specific filter it could be easily extracted. The density and the accuracy of the point cloud that is generated from laser scanner differ, especially when the angle of the beam and the normal of the plane is changed. As a result, the accuracy close to boundaries is not always the desirable. Combining a better boundary visualization with a different type of images, panoramas and the fact that the whole object could be clearly detected, leads to choose our method to be image-based.

4.1.2 The influence of the number of panoramas in point cloud generation

In this part of our research is investigated, how the number of the images affect the final product. The number of the images will be decreased and the differences are tested between the point cloud, estimates the density and the accuracy. In terms of density, the density value will always be estimated depending on the number of the nearest neighbors defining a certain radius. This is named the local point density and for each point is computed, considering only the number of the points within a sphere. This sphere is defined from the beginning by setting a radius to a default number to include k-neighbours close to the point, Figure 4.4.

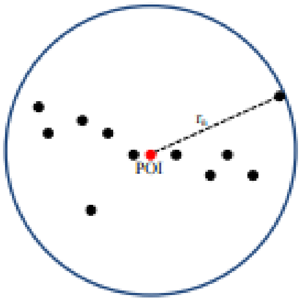


Figure 4.4: Neighbors of point at specific radius [Lari and Habib, 2012].

The function to calculate the density (D) of each point is calculated from:

$$D = (n+1)/\pi r^2 \quad (4.1)$$

Where the πr^2 is the 2D density that is defined from the begging to search the points that are within this area calculating the euclidean distance (d) in 3D.

Change detection will be done, taking into account the distances from the previous point cloud to the next one using more images, specifying also, a distance threshold. The final result will be evaluated for the alignment of the images and for the density. The density of the points changes according the decreasing of the images is tested using less images. The results are displayed in Chapter 5 and the differences in density are estimated using Cloudcompare software using the nearest neighbors, according to the local density method that is explained above.

4.2 SMART POINT CLOUD APPLICATION FRAMEWORK

The proposed general framework from [Poux et al., 2016] is followed for this application:
1. Acquisition of the raw point cloud data.
2. Filtering and Normals estimation.
3. Geometry recognition including segmentation techniques in 2D and 3D environment.
4. Smart point cloud structure.
5. Semantics integration.

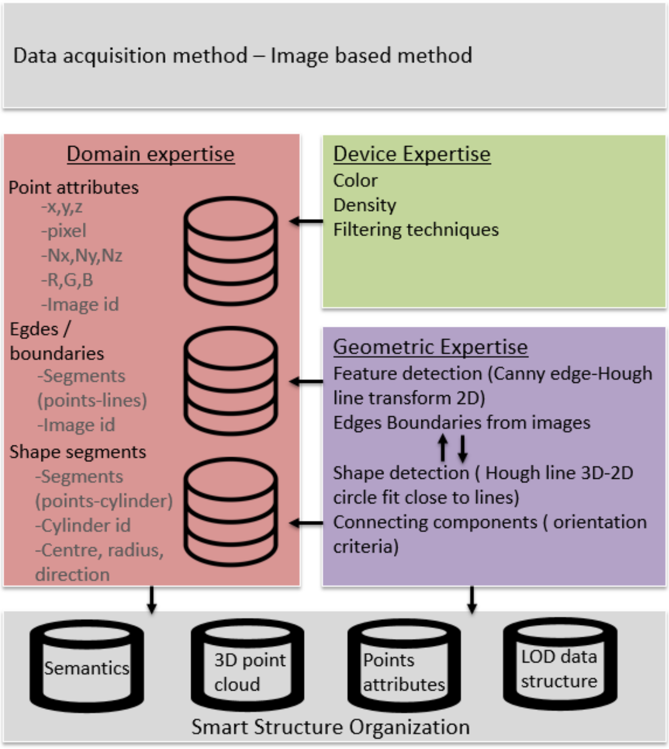

Data acquisition method – Image based method
Domain expertise
Point attributes
-x,y,z
-pixel
-Nx,Ny,Nz
-R,G,B
-Image id
Egdes / boundaries
-Segments (points-lines)
-Image id
Shape segments
-Segments (points-cylinder)
-Cylinder id
-Centre, radius, direction
Device Expertise
Color
Density
Filtering techniques
Geometric Expertise
Feature detection (Canny edge-Hough line transform 2D)
Edges Boundaries from images
Shape detection (Hough line 3D-2D circle fit close to lines)
Connecting components (orientation criteria)
Semantics
3D point cloud
Points attributes
LOD data structure
Smart Structure Organization

Figure 4.5: SPC framework.

This structure still contains the three domains, as shown in Figure 5.20 (device, geometric, domain). First, the concept of the main application and the capture object should be defined. Then, the acquisition of the data defines the device expertise.

In our case, the point cloud will be acquired from imaged based technique using the panoramic images in an industrial environment. Using panoramic images for point cloud generation could assign some extra attributes to the points: the RGB color, the density, the normals and the registration with the images are some of them. At this step, the raw point cloud should pass through filtering for noise reduction, Subsection 4.2.2.

4.2.1 Acquisition of the raw point cloud data

The acquisition of the raw point cloud, as it is mentioned in the previous subsection, is generated from panoramic images from three different datasets. The photos are either from an online source or by a panoramic camera and had taken by cooperation with Fugro company where one and a half hour was spend in total. For this purpose a Nikon 5200 Dslr with a fish-eye lens mounted and a panoramic tripod head, which allows 360°rotation of the camera (nodal point). The places that were chosen in order to take the panoramas were carefully planed, in order to have a full view of the room and to take more images than the required, to test the density of the final result. It is really important to plan the data capture procedure. In photogrammerty, photos (quality, light conditions) and the overlapping area are the most important factors that someone should take into account for good quality of the point cloud. Following an automatic procedure by the software, the dense point cloud is generated. The fact that is generated from image based technique, the final point cloud is attached with some attributes, like the RGB color and normals, which are estimated by the software form Principal Component Analysis (PCA) algorithm and are important for our further implementation. More details about the type of data, the equipment and some statistics about the number of the images and the time that is needed for the point cloud generation, is represented in the next Chapter 5 IMPLEMENTATION & RESULTS PROTOTYPE, Section 5.2 about Data Acquisition & Software.

4.2.2 Filtering and Normals estimation.

A downside of image-based methods is that they are prone to producing outliers and noise in the point cloud due to matching ambiguities or image imperfections (lens distortion, sensor noise etc). The resulting point clouds are often noisy, [Wolff et al., 2016]. For that reason, a noise reduction algorithm is used in order to reduce the noise of the raw point cloud and make the whole procedure easier and more reliable by removing and the wrong points and . This section describes a density based algorithm that deals with the noise reduction, regardless to the varying density of the point cloud.

It relies on the Local Outlier Factor (LOF) of each object, which depends on the local density of its neighborhood. The neighborhood is defined by the distance to the MinPts nearest neighbor. The MinPts is a predefined value, which corresponds to the minimum number of points in calculation of density. The LOF algorithm is a noise reduction method which computes the local density of a given point with respect to its neighbors. It is considered as outlier samples that have a lower density than their neighbors. It is obvious that the density around an outlier object is different from the density around its neighbors. The general concept is that a local outlier has the local density low comparing to the local densities of its Nearest Neighbors, see figure 4.6.

The number of neighbors (MinPts) is one of the parameters that you have to choose with the below characteristics:

- Greater than the minimum number of objects a cluster has to contain, so that other objects can be local outliers relative to this cluster, and

- Smaller than the maximum number of close objects that can potentially be local outliers.

In practice, it is difficult to find such information, taking MinPts neighbors 10-20 appears to work well in general according to Sotoodeh [2006], considering a dense point cloud.

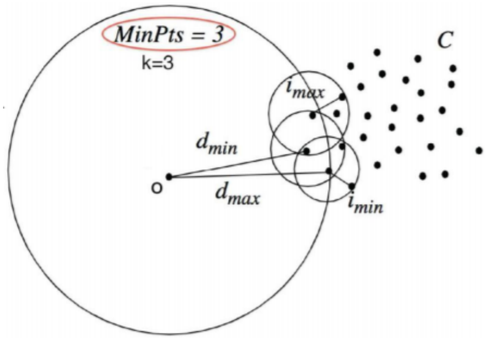


Figure 4.6: Explanation of the LOF.

The local outlier factor is based on the local reachability density of a point,

$$lrd_{MinPts}(o) = 1/\left(\frac{\sum maxdist(o,p)}{|N_{MinPts}(o)|}\right)$$

,where p $\in MinPts(o)$.

The local reachability density is the inverse of the average max distance based on the MinPts nearest neighbors of a point, Figure 4.6. The LOF is finally, defined as:

$$LOF_{MinPts}(o) = \frac{\sum \frac{lrd_{MinPts}(p)}{lrd_{MinPts}(o)}}{|N_{MinPts}(o)|} \tag{4.2}$$

The estimation of the normals is computed using the PCA algorithm. PCA is a method for analysing data distribution using the eigenvectors and eigenvalues of covariance matrices. Since it is a time consuming method, it is only applied during the geometry processing for the points that the normal is needed to estimate the direction of the normal. The general algorithm is applied and the normals are estimated based on the closest neighbors of the point. The idea is to form neighborhoods based on distance. There are two general methods that are usually used. The one is Fixed-Distance Neighbors (FDN) based on the number of the neighbors that are predefined, Figure 4.7. The other method named, K-Nearst Neighbors (kNN) is based on the distance for the existing neighbors 4.8. Apart from normal vector estimation, it can also be useful for the removal of outliers, as it is mentioned before.

The idea is that the underlying surface, therefore a 2D plane should fit. Each points is a vector of three variables (x,y,z). The local variation of the point is computed using the co-variance matrices. These matrices shown how the neighbors of the point distribute from the center. The smallest eigenvectors of theses matrices indicate the normals vectors of the best fitting plane. Each eigenvector indicates the direction and is associated with the smallest eigenvalue corresponds the direction of the least variation. This is the normal of the plane.

4.2.3 Image(2D)-Points(3D) connection

Following the SPC framework for object detection the three main expertises are defined. Each one with its own characteristics, the geometric expertise contains all

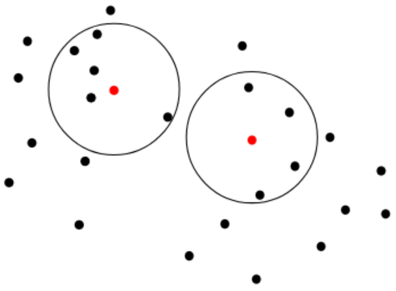


Figure 4.7: Fixed-Distance Neighbors

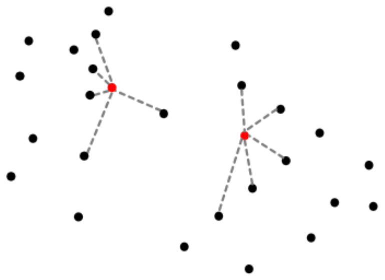


Figure 4.8: k-Nearest Neighbors.

the methods that are related with the geometry data processing (segmentation, classification). The main goal is to combine the information from both sources for better results. The geometric knowledge is based on the geometry detection of the pipelines through different methods (see Figure 4.9). The features and the attributes from both the datasets (panoramic images and point cloud) will be combined to get characteristics from both sources in order to select, analyze, manipulate and identify the pipes, as one object.

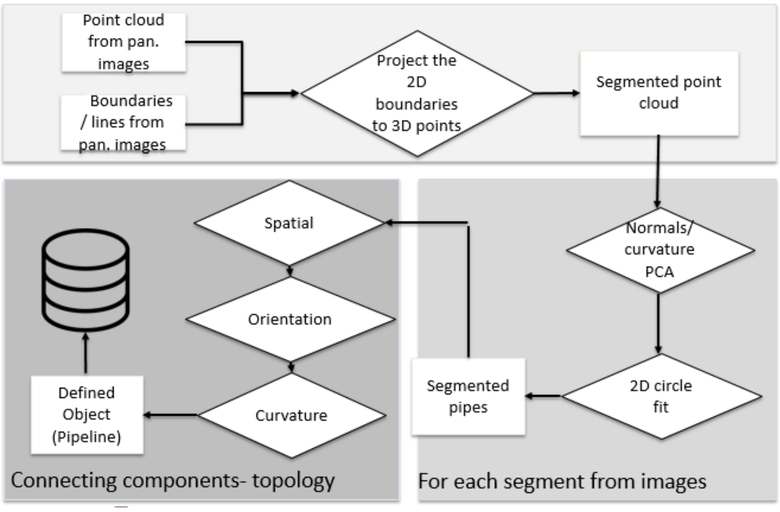

Point cloud from pan. images
Boundaries / lines from pan. images
Project the 2D boundaries to 3D points
Segmented point cloud
Normals/ curvature PCA
2D circle fit
Segmented pipes
Spatial
Orientation
Curvature
Defined Object (Pipeline)
Connecting components- topology
For each segment from images

Figure 4.9: Geometric Methodology.

The characteristics of 3D point clouds and 2D digital panoramic images are thought to be complementary, so the combined interpretation of objects with point clouds and image data is a promising approach for better object recognition. However, the most challenging part is the geometric referencing of the point cloud to the image data. These methods will result in an efficient, fast and accurate way to extract the pipes.

Our first step and contribution is to form the relation between the 3D points and the pixels of each image. In order to achieve this, two types of transformations must be implemented. First, the local coordinates of the system where the point cloud is referred to, are converted to spherical coordinates (ϕ, θ) and then, the spherical coordinates are projected back to image coordinates (pixels), see Figure 4.10. In general, the coordinates of a 3D point P=(Px,Py,Pz) (either in world or local coordinates) are given by spherical coordinates.

$$\phi = arctan(\frac{P_x}{P_y}) \tag{4.3}$$

with $\phi \in [0,2\pi]$, and either

$$\theta = arctan(\frac{P_y}{\sqrt{P_x^2 + P_y^2}}) \tag{4.4}$$

with $\phi \in [0,2\pi]$. D is simply the distance from P to the origin of the coordinate system. For a spherical capturing surface, D is constant and (θ, ϕ) defines a 2D coordinate system for this non-Euclidean surface.
It is considered as discrete coordinates u,v defined by

$$u = w \times 0.5 + f \times \tan^{-1}(\frac{X}{Y}) \tag{4.5}$$

and

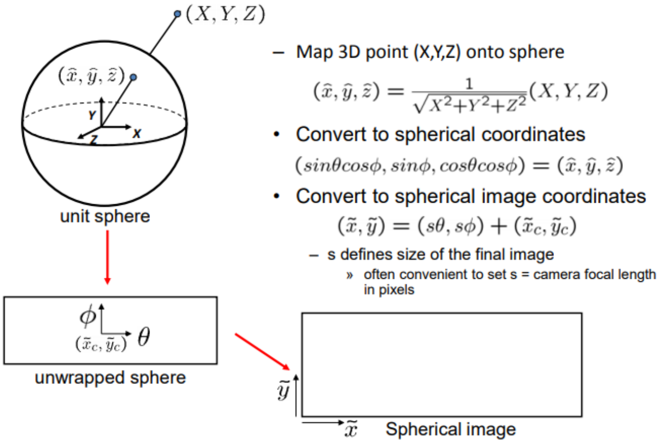
$$v = h \times 0.5 + f \times \tan^{-1}(\frac{Y}{\sqrt{X^2 + Y^2}}) \tag{4.6}$$

Where:

$$f = \frac{w}{2 \times \pi} \tag{4.7}$$

and w, h width and height of the image in pixels.

These formulas could also be used in the opposite way in order to connect the pixels that are part of edges to the 3D points. The above formulas are used in order to project each 3D point to one panorama taking into account the affine transformation matrix that is provided by the software and the position of the panorama that is projected. This projection is preferable than the other way because by this way, it is indicated that 3D point can always be projected on the panoramic image. A pixel in the image is not guaranteed to be a point in 3D.



– Map 3D point (X,Y,Z) onto sphere

$$(\hat{x},\hat{y},\hat{z}) = \frac{1}{\sqrt{X^2+Y^2+Z^2}}(X,Y,Z)$$

- Convert to spherical coordinates

$$(sin\theta cos\phi, sin\phi, cos\theta cos\phi) = (\hat{x},\hat{y},\hat{z})$$

- Convert to spherical image coordinates

$$(\tilde{x},\tilde{y}) = (s\theta, s\phi) + (\tilde{x}_c,\tilde{y}_c)$$

  – s defines size of the final image
    » often convenient to set s = camera focal length in pixels

Figure 4.10: 3D Points-Pixels projection.

4.2.4 Image processing

Image processing is a really important procedure in order to detect the edges of the panoramas and to connect them with the point cloud. For this application, two forms of the image processing have been applied to get the boundaries. First, the canny edge detection is applied to the panoramas and then, depending the generated image, the Hough transform detection is applied to get the straight lines

that are the possible boundaries of the pipes. The algorithm of the implementation is analyzed here and the results and the problems are displayed in Chapter 5. The algorithm and an example of each technique are displayed, in order to understand how these techniques are working.
Canny edge detection is used, in order to find the edges on the images, as it is the most accurate and promising method for edge detection Canny [1987]. The edge in 2D image could be the boundaries of the objects, the surface normal discontinues, the reflectance of the material and the light discontinuities. To implement the Canny edge detector a standard approach is proposed, in Figure 4.11, the result from each step of the procedure is represented.

1. Gaussian convolution: The Gaussian filter is used for each image to smooth image from the noise.

2. Image gradient: To compute the image gradient. There is the magnitude of the gradient and the direction of the gradient.

   Gradient equation:

   $$\nabla f = [\frac{\partial f}{\partial x}, \frac{\partial f}{\partial}] \tag{4.8}$$

   Gradient direction:

   $$\theta = \tan^{-1}(\frac{\partial f}{\partial x} / \frac{\partial f}{\partial}) \tag{4.9}$$

   The edge strength is given by the gradient magnitude:

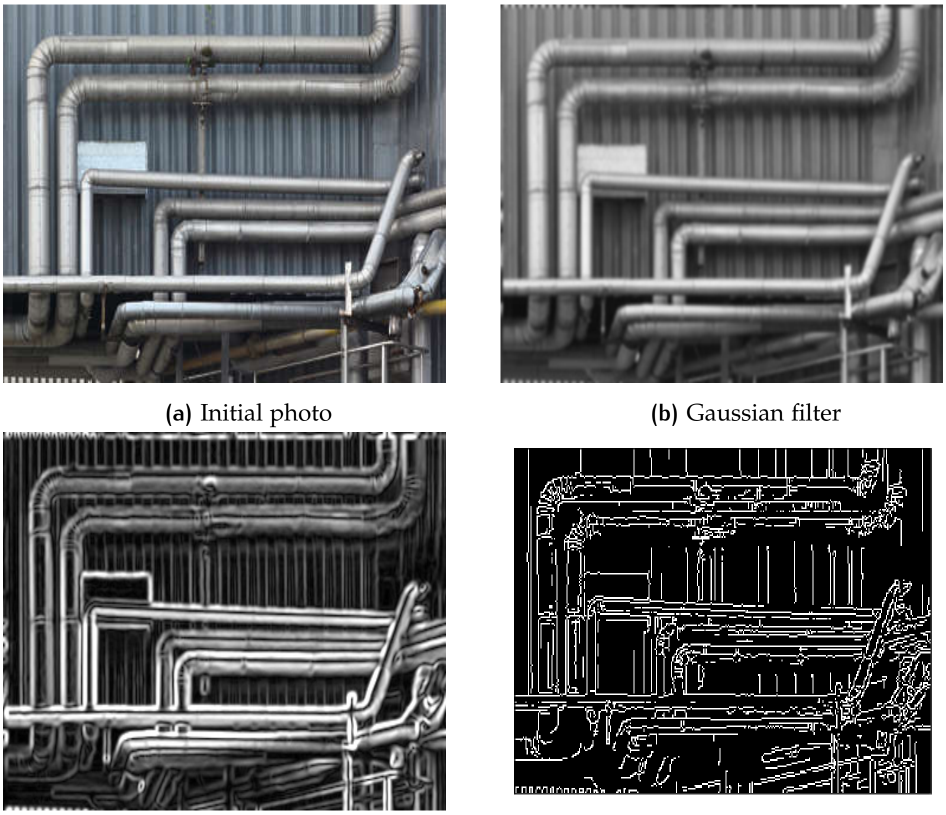
   $$\|\nabla f\| = \sqrt{(\frac{\partial f}{\partial x})^2 + (\frac{\partial f}{\partial y})^2} \tag{4.10}$$

3. Setting Canny thresholds: To apply the Canny method two thresholds should be defined (low and the high). Both thresholds are represented in percentage of the norms of the gradient.

4. Non-maximum suppression: For every point in the image the gradient direction is used to compare the value of the gradient norm with the norm of its neighbors, if it is above the high threshold, the pixel is considered a candidate edge pixel. However, if the value is lower than the high threshold, it will be considered in the subsequent hysteresis step.

5. Hysteresis implementation: The Hysteresis implementation uses the low and the high Canny Thresholds. The procedure starts by using the initial collection of edge points that are obtained after the high threshold and after the non-maximum suppression. After that, a recursive algorithm is used to get the new edge using the low threshold. A point where the norm of the gradient is between the lower and the higher thresholds is actually an edge pixel and then, the neighbor pixels are recursively explored.

The results really depend on the thresholds that are defined and also, on the initial quality of the images. It will be difficult to detect the clear boundaries of the objects, if the image is blur or has some poor light conditions.
After the canny edge detection the implementation continues, applying the Hough line transform to the panoramas. This technique is for the detection lines from the panoramas, supposing that the boundaries from the pipes are represented as straight lines on panoramas. The steps of the algorithm are described below:



(a) Initial photo

(b) Gaussian filter

(c) Compute Gradient

(d) Final Edges

Figure 4.11: Different steps for Canny edge detection.

1. Built a parameter space, for line slope m and intercept c.
2. Create an accumulator array A(m,c).
3. Set $A(m,c) = 0 \forall (m,c)$
4. Get the image edges using Canny edge detector
5. For each pixel on the image apply the equation $(x_i, y_j) \forall (m_k, c_l)$ apply the equation that represents a line.

   $$c_l = -x_i m_k + y_j$$

   Then, increment:

   $$A(m_k, c_l) = A(m_k, c_l) + 1$$

6. Find the local maxima in A(m,c) that indicates a line in parameter space.

The information on the images is contained mainly on the edges, as the contrast is usually much better. For that reason the image processing will help us identify the boundaries. Moreover, in industrial environments the curved objects are universally present, and for these objects just edge-localized information is not enough for automatic detection and fitting, hence this becomes a major limitation. In Chapter 5, three approaches are tested in order to connect the 2D information from images (pixels) with the 3D actual points using the test data. For the SPC framework a combination of the second (2D with 3D connection) and the third approach (Direct connection pixels with 3D edges) is combined in order to get the boundaries of the pipes, applying both canny detection and HT to the panoramas. More details about the implementation of the approaches are described in Chapter 5, Section 5.3, Projections on test data.

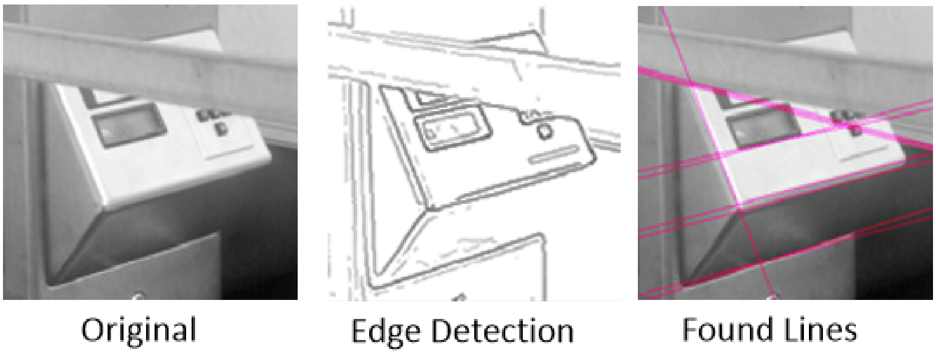


Figure 4.12: Example using HT.

4.2.5 Geometry recognition

The geometric methodology that is followed for the point cloud object detection is represented in Figure 4.9. After the image processing, the connection between the boundaries of the panoramas with the boundaries in point cloud is achieved. The final goal is to segment the initial point cloud to points that are part of a pipe. By this way, the segmented points will get knowledge and meaning, connected with the real environment. Segmentation is the process of dividing the given point cloud or image into a number of disjoint subsets, each of which is spatially coherent. To continue with the geometric methodology, image edge detection must already be applied to detect the points-lines that are boundaries of the cylinders. Outlines the borders of different regions, is followed by grouping the points close to detected boundaries. In that way, it is easier to detect, if these points are part of cylinder following our methodology, Figure 4.9. In many cases, the edges do not form closed boundary curves and it can be difficult to make correct grouping decisions resulting in over or under-segmentation. In our case, the pipes have no actual edges in 3D but in 2D, you can easily extract the boundaries. HT is widely used for line detection on images [Wang et al., 2014]. It is considered that the boundaries of the pipes are straight lines. Images provide a complementary source of information, as they capture the object boundaries where point clouds are noisy. Following the methodology which is described in subsection 4.2.3, points that are part of the boundaries will be merged with the raw point cloud. From each line-segment in points, a line fit is done using the HT in 3D.
The implementation of the algorithm HT in 3D follows the same theory as in 2D. Due to the additional dimension, the number of the parameters are larger in 3D space than in 2D, making the same procedure more difficult for the same output [Watson, 2017]. In [Dalitz et al., 2017], the 3D Hough line transform is applied to point clouds. The main objective is to fit lines close to the 3D points-boundaries. The idea of the Hough transform in 3D is to discretize the space giving all the possible direction of lines. Each point determines to which line it belongs and the direction with the most points, means that a line could fit. For the discretization of the line direction, the method that is used, is based on tessellation of Platonic solids. The solid with the highest number of vertices is the icosahedron, Figure 4.13. The theory is the same as in 2D, described in the previous section.

After getting the number and the parameters of each line, it is investigated, if a cylinder could fit close to these lines. Initially, the points that are close to the line will be gathered, with close distance line-point in 3D smaller than 0.05m (4.14).
The line in 3D is specified with the vector and a point that lies on the line. The parameter function of the line:

$$x = a + \vec{b}\,t \tag{4.11}$$

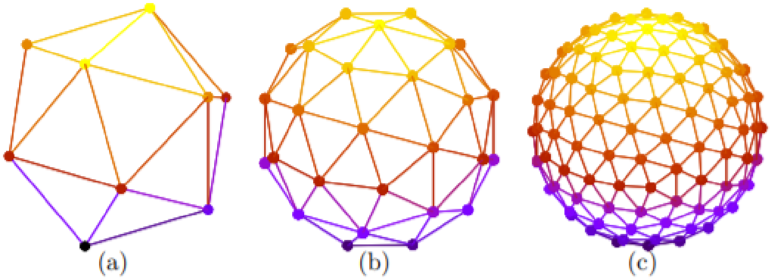


Figure 4.13: Icosahedron after 0, 1, 2 tessellation steps [Dalitz et al., 2017].

where $a = (x, y, z)$ and $\overrightarrow{b} = (b_1, b_2, b_3)$
The squared distance between a point on the line with direction t and a point $(x_0, y_0, z_0$ is computed:

$$d^2 = ((x - x_0) + b_1 t)^2 + ((y - y_0) + b_2 t)^2 + ((z - y_0) + b_3 t)^2 \tag{4.12}$$

In order to get the minimum distance set:

$$d(d^2)/dt = 0 \tag{4.13}$$

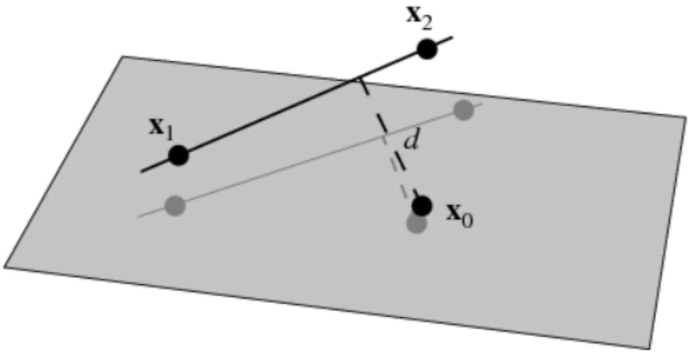


Figure 4.14: Smaller distance line-point in 3D.

After getting the points that are close to the line, it is examined, if a 2D circle will fit. The plane that the points are projected on is defined as the perpendicular plane to the line. Lalonde et al. [2006] worked in a similar way and developed two algorithms based on 2D projection and 3D fitting for tree trunk segmentation. They showed that results from the PCA based on 2D projection technique work with better results than the results from 3D fitting.
A circle that fits as close as possible to the set of points is represented in 3D space by this parametric equation:

$$P(t) = rcos(t)u + rsin(t)(n \times u) + c, 0 \leq t \leq 2\pi \tag{4.14}$$

with radius r , center C and normal unit vector n . Vector u is any unit vector perpendicular to n . If we specify orientation of the circle in space by zenith angle ϕ and azimuth θ:

$$\mathbf{n} = \begin{pmatrix} \cos(\phi)\sin(\theta) \\ \sin(\phi)\sin(\theta) \\ \cos(\theta) \end{pmatrix}, \quad \mathbf{u} = \begin{pmatrix} -\sin(\phi) \\ \cos(\phi) \\ 0 \end{pmatrix}. \tag{4.15}$$

The circle fitting method can be broken down into the following steps:

- Project the mean-centered points onto the fitting plane in new 2D coordinates.
- Using the method of least-squares fit a circle in the 2D coordinates and get circle center and radius.
- Transform the circle center back to 3D coordinates. The fitting circle is specified by its center, radius and normal vector.

Projecting points onto the fitting 2D plane the Rodrigues rotation (3D rotation) formula is used. The axis of rotation k is the cross product between the direction of the plane and the directions of the new X-Y new coordinates.

$$\mathbf{k} = \mathbf{n} \times (0,0,1)^T \tag{4.16}$$

$$\mathbf{P_{rot}} = \mathbf{P}\cos(\theta) + (\mathbf{k} \times \mathbf{P})\sin(\theta) + \mathbf{k}\langle \mathbf{k}, \mathbf{P} \rangle (1 - \cos(\theta)) \tag{4.17}$$

To fit a circle to the cluster of points might sound like an easy task, but in 3D space it gets a bit more complicated. The fact that the projected plane has already be known make the procedure easier. First, we project the 3D points onto the plane and get new planar coordinations for them. Finally, we could use the method of least-squares to fit a 2D circle into the planar points and then project the 2D fitting circle back to the 3D coordinates.
To fit a circle to set of n points the equation in 2D for a circle with radius r and center (xc,yc) will be like this:

$$\begin{aligned} (x - x_c)^2 + (y - y_c)^2 &= r^2 \\ (2x_c)x + (2y_c)y + (r^2 - x_c^2 - y_c^2) &= x^2 + y^2 \\ c_0 x + c_1 y + c_2 &= x^2 + y^2 \end{aligned}$$

So, the parameters must minimize the function, applying least square adjustment. Circle fitting is performed in the projected 2D space. If the root mean square error (RMSE) of the circle fitting is less than one, the segment is likely to be a cylinder. Knowing the parameters of the circle, its possible to construct the cylinder surface. The center of the circle and the direction could give us the line that passes from the center of the cylinder and with the radius, the surface could be reconstructed. Then, with these characteristics the points that are close to the line and with distance similar to radius are part of the cylinder.

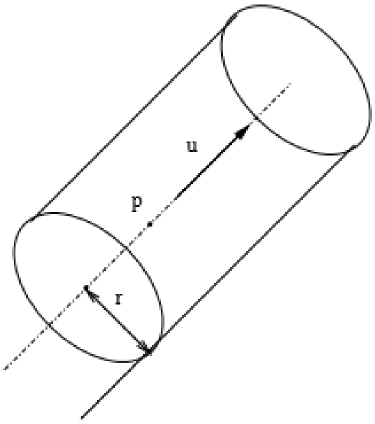


Figure 4.15: Geometry of cylinder.

The segmented points after some criteria for the connectivity and orientation [Su et al., 2016], could be detected and connected, as one object. Merging the segments in one object is really important, as it offers many advantages in visualization and also, allows useful information to be associated with the merged segments. Specifically, for the spatial connectivity, each segment will be tested whether they have the same axis orientation and if they are close enough or have common points, in order to be considered as part of the same pipe. Then, if these criteria are full-filled, these parts will be merged and stored in a database, as one object. The structure of the data should combine and implement an efficient store of the data retaining the device, the analytic and the domain attributes. The smart linking of the data conclude to semantics injection.

4.2.6 Fit second polynomial to segmented points

Since, the main object of our interest is the cylinder, the segmented points could be represented as a second polynomial surface. Except from the 2D circle fit on the projected 2D points, the second polynomial fit is tested, if it could be fitted to the segmented points. The points in each segment are fitted to a quadratic equation:

$$f(x,y) = ax^2 + bxy + cy^2 + d \tag{4.18}$$

So, the four parameters must minimize the function, applying least square adjustment. Given the fitted polynomial, the principal curvatures k1 and k2 of the surface f(x,y) can be computed from the polynomial coefficients as follows:

$$k_1 = (a+c) + \sqrt{(c-a)^2 + b^2} \tag{4.19}$$

$$k_2 = (a+c) - \sqrt{(c-a)^2 + b^2} \tag{4.20}$$

If the segmented points are part of a cylinder, one of the principal curvatures are zero. According to the article [Su et al., 2016], because the data are noisy and the point cloud is not a perfect cylinder, the principal curvature cannot be zero. In that case, it is defined that a segment could be cylindrical, if $k1 > 5, k2 < 1$. These threshold values are defined from [Su et al., 2016], where they were chosen using a histogram based analysis on a subset that manually labeled the cylindrical and planar segments. The principal curvature of a cylinder is related with the radius, where:

$$r = \frac{1}{k} \tag{4.21}$$

This is an alternative way to investigate, whether the points are part of a cylinder. The results of this method are displayed in the next Chapter, Section 5.4.5. The comparison of these methods to conclude which is the best option for cylinder fitting and the analysis is explained in the next Chapter.

4.2.7 Smart point cloud structure

The SPC UML model (see Figure 4.16) is formed by the main components needed for the data structure. It starts with the primitive geometry of a point which is defined by three coordinates (x,y,z). Each point has specific attributes that derive from: the device knowledge (density, color), the geometric knowledge (normals) and the domain knowledge [Poux et al., 2017a]. At each stage, the attributes are stored in a database using relational model for fast extraction and manipulate the points individually or as segments. A large number of points with similar attributes are the point segments which in the end are characterized as pipes. The UML model shows the relationship between the components, with the purpose of a fast integration and efficient and easy way to apply semantics [Rabbani, 2006].

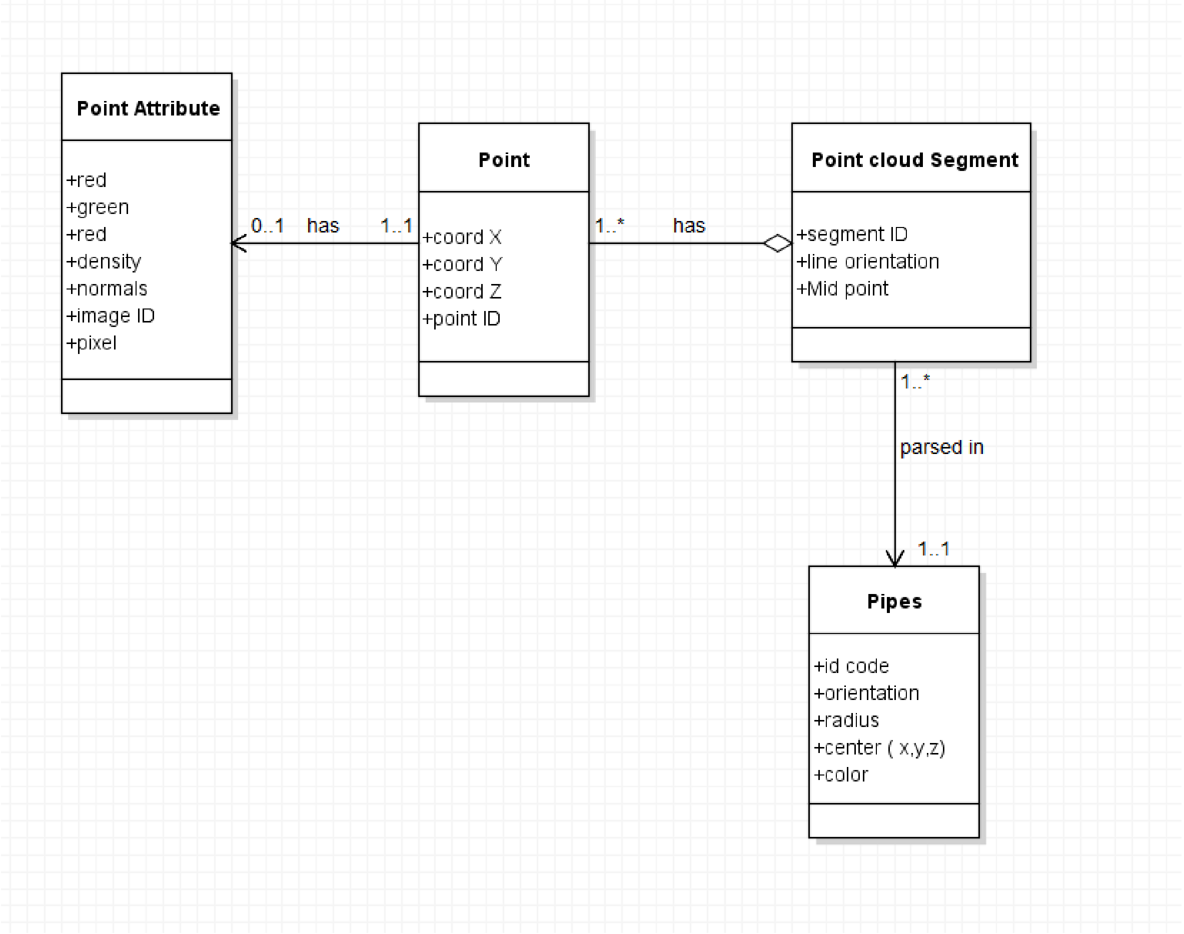


Figure 4.16: UML for data structure.

4.2.8 Semantics integration

In general, the goal is to obtain spatial and semantic information from the available data (images, point cloud) of an industrial environment in order to identify the pipes. Retrieving this semantic information is essential for further analysis and interpretation. This step gives a meaning to the collected data, giving the opportunity to also, interpret them in AR or VR environment. The AR and VR could append virtual information to the real world and be utilized as a useful tool for field work inspection Park et al. [2013]. All this information must be retained and structured to achieve interoperability. If standardization has international recognition, more and more people will use them and as a result, the semantic interoperability will be achieved [Westdijk, 2015]. According to the UML three tables are creating. One with the points and their attributes, second the segments where the cylinder is detected and final, the pipe with the parameters and the segments that are part of it.

4.2.9 RANSAC implementation algorithm direct to point cloud

After the implementation of our methodology, the cylinders are detected from a sample of our point cloud, which are generated from panoramic images. Compared our method with the generic methods (RANSAC, HT) that are directly applied to the point cloud, their performance is analyzing, applying the RANSAC algorithm.
The RANSAC approach will be applied where the shapes are detected via a random sample consensus. The basic RANSAC approach repeats the following steps:

1. Randomly select samples from the input points.

2. Fit a cylinder shape model to the selected samples.

   A cylinder is specified by an axis and a point lying on the axis C. A vector specifies the direction of the axis W and the radius of the cylinder. Any point X may be written uniquely as:

   $$X = C + y_0 U + y_1 V + y_2 W = C + RY$$

   Where R is the rotation matrix of U, V, W and Y a column vector (y_0, y_1, y_2).

3. Count the number of inliers to the shape. Inliers are the points that are part of the cylinder's shape under a user-specified error tolerance.

These steps are repeated for a prescribed number of iterations and the shape with the highest number of inliers, is detected.
The error between a point and a shape is defined by its distance and normal deviation to the shape. A random subset corresponds to the minimal number of points required to uniquely define the cylinder. According to the parameter model, the minimum number of points that are required for cylinder detection.
For very large point sets the basic RANSAC method is not practical. The main idea behind the efficient RANSAC method consists in testing a shape against subsets of the input data. Shape candidate is constructed until the probability to miss the largest candidate is lower than a user-specified parameter. The largest shape is extracted until no more shapes could cover the minimum number of points. The documentation that is followed, it is provided by CGAL Documentation. In our test, the same part of point cloud generated from panoramas that is used in our methodology, is also used for the RANSAC implementation.

5 IMPLEMENTATION & RESULTS PROTOTYPE

This chapter describes the implementation details of the SPC framework that is described in Chapter 4. It is more specific about what has been implemented and which techniques and parameters have been tested and used. Section 5.1 introduces the tools and the libraries that are used and Section 5.2 focuses on the different data and the software that are tested. Then, the results of the implementation, Section 5.4 will be presented according to the theory in Chapter 4, PROTOTYPE DESIGN. Finally, a comparison of the results applying RANSAC algorithm is provided in Section 5.4.6.

5.1 TOOLS & LIBRARIES

The programming language Python and database management system PostgreSQL were used for the implementation. Libraries and packages that were needed in order to implement the algorithms that were introduced in the methodology, are listed below:

- NumPy: Package for scientific computing with Python.
- Psycopg: Package for connection Python with database system PostgreSQL.
- Matplotlib: Package for plots and visualization the results.
- Scikit: Python library for geometry implementation.
- Pillow: Library for image processing
- Opencv: It is an Open Source Computer Vision Library, which was designed for computational efficiency and with a strong focus on real-time applications.

Furthermore, software CloudCompare is used mainly, for visualizing the point cloud and for processing. Software Autopano is used for stitching the panoramas provided by Fugro.

5.2 DATA ACQUISITION & SOFTWARE

At the beginning of the implementation, some test images were used in order to test the existing software. Since the images that should have been provided from the company, were not available from the beginning, some test data were initially used for testing the software and start the implementation for point cloud acquisition. Eight panoramic images were downloaded from: `https://support.pix4d.com/hc/en-us/articles/360000235126-Example-projectslabel7`
from the Church of Santiago", located in Betanzos (La Coruña, Spain) taken by an NCTech iSTAR spherical camera. These images are more detailed with texture and more information. In addition, other panoramic images are used for the point cloud generation, provided by the company Fugro, using camera Nikon 5200 Dslr. These photos (10 panoramas) represent an empty room in a building and were provided until the images with the pipes (23 panoramas) were available. Moreover, it was a nice opportunity to test the point cloud that is generated from panoramic images, representing totally different environments. Furthermore, these data were used for

the first steps of our implementation, Section 5.3. It is a nice example to illustrate that our methodology can be implemented for different datasets and prove that edges are extracted from 2D panoramic images and linked with the boundaries of objects which are represented in 3D.

For each panorama represents the empty room, eight photos are taken seven horizontal and one vertical, using camera Nikon 5200 Dslr with a fish-eye lens mounted and a panoramic tripod head, which allows 360°rotation of the camera (nodal point). The stitching of the separate photos is done on a specific program named, Autopano to produce one equirectangular panorama, Figure 5.1.

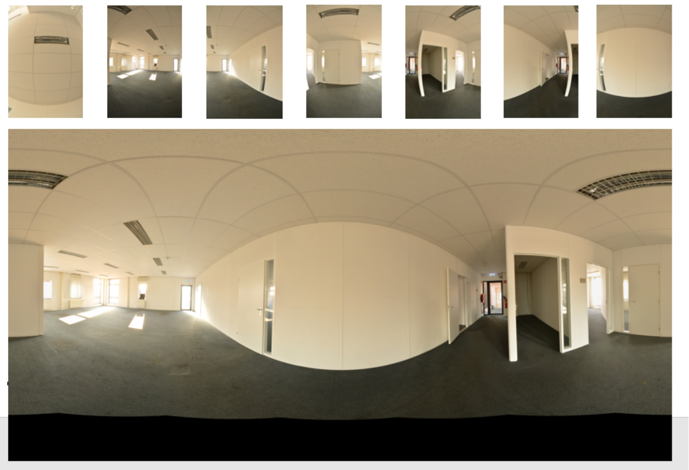


Figure 5.1: Stitchnig the panoramas

The projection obtained by a sphere and un-wrapped on a flat rectangular plane surface is known as equirectangular projection of the sphere. It is the direct mapping of the sphere latitude and longitude on the horizontal and vertical coordinate system respectively. It is called as non-projection or rectangular projection as no scaling or transformation is applied. The resulting frame obtained by equirectangular projection appears to be distorted. The objects at the center are spatially compressed and are stretched towards the top and bottom. As seen in the Figure 5.2, top and bottom regions represent zenith and nadir respectively. Equirectangular frame covers 360°horizontally and 180°vertically.

The algorithms that are working with close-range photogrammetry are based on feature detection so, the whole dense point cloud generation needs more time while rendering the photos from the empty room compared to the time that is needed for the images that depict the church. It is important to mention that the software needs 'help' with the empty room. This means that some markers were put for the orientation of the images. The panoramic images from the different datasets are presented in Figure 5.2 and 5.3.

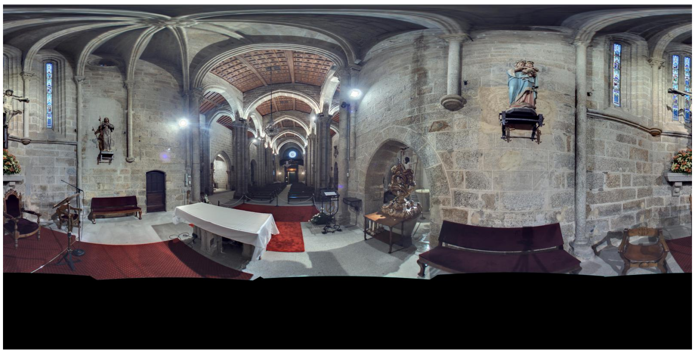


Figure 5.2: Panoramic image church

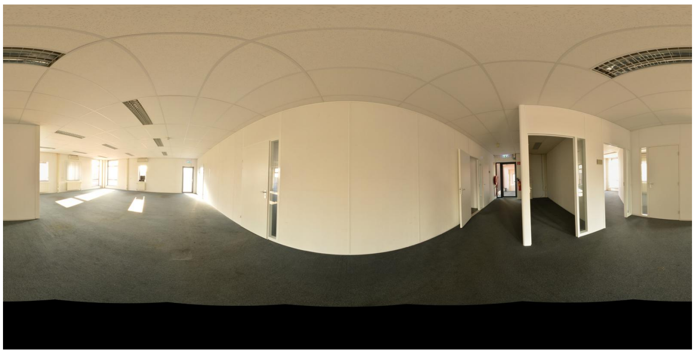


Figure 5.3: Panoramic image from an empty room

The images that were used at the final implementation to detect the pipes, were taken in a room with pipes at Fugro. More details about the type of the images are provided in section 5.4. These data are used in the end to create the SPC framework.

5.2.1 Software comparison and statistics

Existing software, as it is mentioned in Chapter 4, PROTOTYPE DESIGN, were tested and examined, whether they work with panoramic images. At Table 5.1, the characteristics of each software are displayed. The type of the images, the output data and some limitations that must be taken into account before using them. The entering of data that the available software need, vary depending on the software. Others need the pre-calibration of the camera and others have the parameters of the calibration in database depending the model of the camera that is imported. That makes the whole process of testing and comparing the available software more difficult.
In recent years, efforts have been made to generate models from spherical panoramas automatically. These processes were applied in Agisoft Photoscan and Pix4d. From January 2013, Agisoft Photoscan enabled spherical panoramas to be imported. From April 2014, this program enabled point clouds with spherical images to be generated automatically. Agisoft Photoscan uses SFM, as the algorithm to determine the orientation of the cameras and to position object points. Photos are treated as spherical panoramas and no calibration parameters of spherical cameras can be imported or edited. Pix4d also, supports the Equirectangular format of spherical cameras for processing. Images of this format are automatically detected as spherical images. Other software have some other limitations: working with fisheye lens or with common image frames.

Table 5.1: Software comparison.

Software	License	Image Type	Results	Constrains
Agisoft Photoscan	30 days trial	Spherical	Point cloud/mesh	
Context Capture	20 days trial	Fisheye lens	mesh	
Pix4d	15 days trial	Spherical	Point cloud/mesh	No more than 55 MB
Micmac	Open source	Fisheye lens	Point cloud/mesh	No interface/ command
nFrames/Sure	20 days trial	Fisheye lens	Point cloud	Pre-alignment info

The goal of this thesis is to generate the point cloud using panoramic images. Due to their 360°field of view and the fact that they could replace many pair of images, it is investigated, if they could minimize time and storage. Working with one image instead of many images has many advantages. To begin with, the connection between points and pixels could be done at once and could enhance the image processing making the whole procedure easier and more valuable. These software are generating the point cloud from the common features (pixels) between the images. It was really surprising the fact that the software do not provide this information.

Actually, the first step for point cloud generation is to find the relative position between the images, providing tie points. Only for these points, the software provide the correlation between the pixel of the image and the 3D point that is generated. Based on these tie points the software is able to continue creating the dense point cloud.
The panoramas are created from 7 separate images (6 horizontal and 1 vertical). Autopano is a suitable software for stitching the panoramas as the generated panoramas are corrected from distortion.
The software that are tested for generating dense point cloud are displayed in Table 5.1. General information is provided about the time, the generated points, the procedure and the functionality of the software. As it has been mentioned before, from the tested software only two of them provide the functionality to use panoramas as spherical images for point cloud generation, Table 5.2 displays some characteristics about these software. In Figure 5.4, an example using Agisoft Photoscan with spherical test images is presented 5.2.

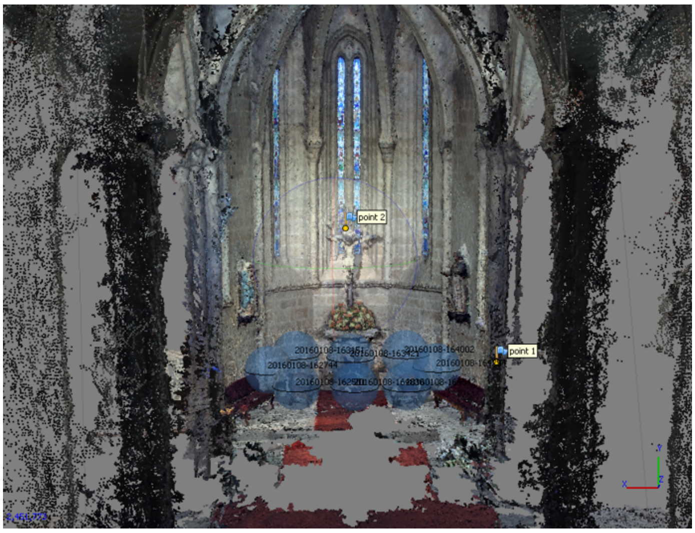


Figure 5.4: Point cloud generated from panoramic images using Agisoft Photoscan.

Table 5.2: Software details.

Software	Images	Tie points	Dense cloud	Time
Agisoft Photoscan	8	7 616	2 4551 773	3 hours
Pix4d	8	10 859	1 098 077	3 hours

Both software/packages are using the panoramas as spherical images, with the center of the image, as an ideal sphere. The point clouds that are generated from the two software, are compared in CloudCompare. The results are displayed in Figure 5.5. A diagram with statistics is provided taking into account the approximate distances between the closest points. In the diagram, it is shown that the differences between the two point clouds are less than 3cm. This is not such a big difference taking into account that the registration error is close to 1cm.

5.2.2 Comparing point clouds to Image-Based with Laser scanner technique

Both techniques have their own advantages and disadvantages. Depending on the application and the accuracy that you want, you can choose the method that is more applicable. For our implementation the point cloud is generated from 23 panoramic images (with pipes) using camera Nikon 5200 Dslr. The software generates a total of

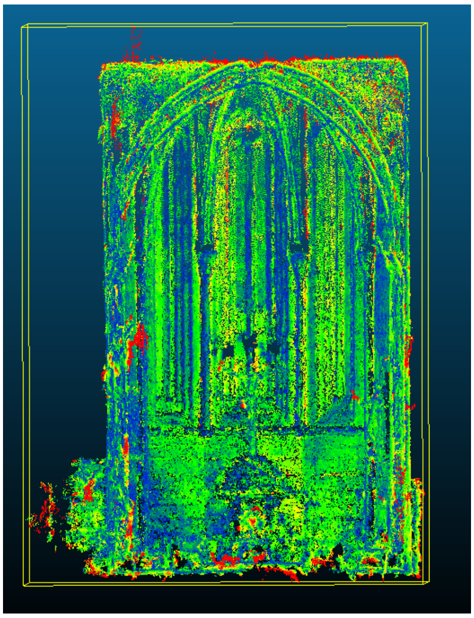


Figure 5.5: Compare point cloud from the two software.

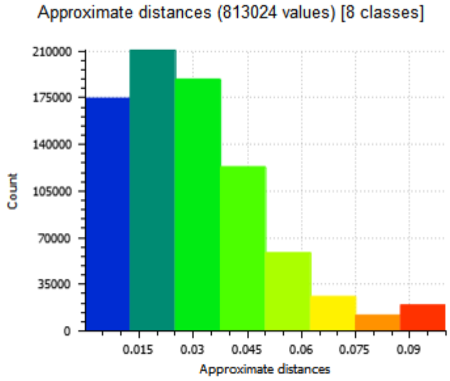


Figure 5.6: Approximate distances between two different point clouds.

40 011 485 points and the density is depicted in Figure 5.9c. In order to compare the two methods in a more quantitative aspect, an example will be tested to estimate in average, the density of each method. From our point cloud a part $1.5m \times 1.5m$ of a wall is taken and then, the density is computed, Figure 5.7 with radius 0.02cm. The amount of points that are part of this wall are 141 505.

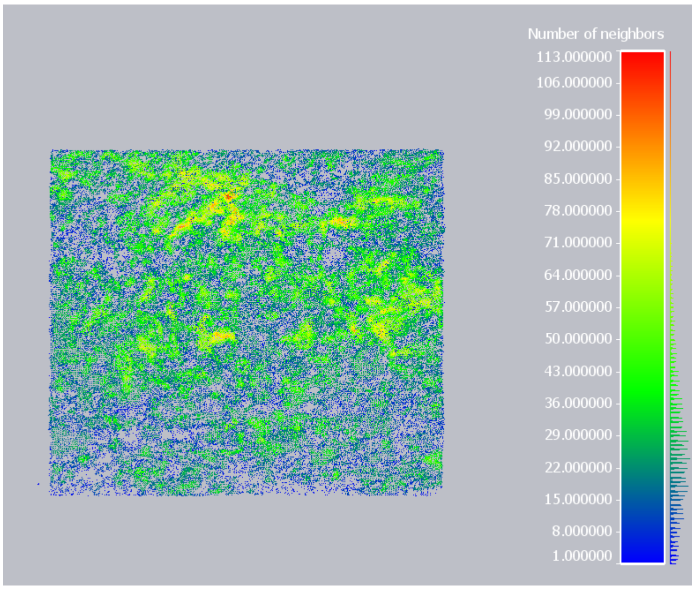


Figure 5.7: Estimated density using part of the wall.

The laser scanner that Fugro uses is Zoller + Frohlich imager 5010C the characteristics are displayed in Table 5.3.

Taking this example of a wall, the density of the wall segment from our generating point cloud (141 505 points) is compared to the one that will theoretically be created with the laser scanner. It is supposed that the laser scanner will be located close to a wall about 2m-3m distance and then, taking into account that there are no obstacles and knowing the characteristics of the laser scanner, it is possible to compute the number of the points. The angular step is 0.004 deg, so for the distance 2m of

Table 5.3: Laser scanner characteristics.

Laser scanner	Z&F Imager 5010C
Beam divergence	< 0.3 mrad
Range	187.3m
Min distance	0.3m
Resolution range	0.1mm
Data acquisition rate	1.016 million pixel/sec
Linearity error	$<= 1mm$
Vertical field of view	320°
Horizontal field of view	320°
Vertical resolution	0.004°
Horizontal resolution	0.002°
Vertical accuracy	0.0007°rms
Horizontal accuracy	0.0007°rms

the wall the step is 0.0007m. In a area of 1.5×1.5 the total points are 306 122 448 points. This number is far bigger than the point cloud that is generated from the images. This is only an example to see the differences between the two methods. Generally, the amount of the points that could be captured from a laser scanner, depends on many parameters: distance, inclination, obstacles etc. Because this is only an example and not an actual implementation, it is not possible to estimate the differences in accuracy between these two techniques.

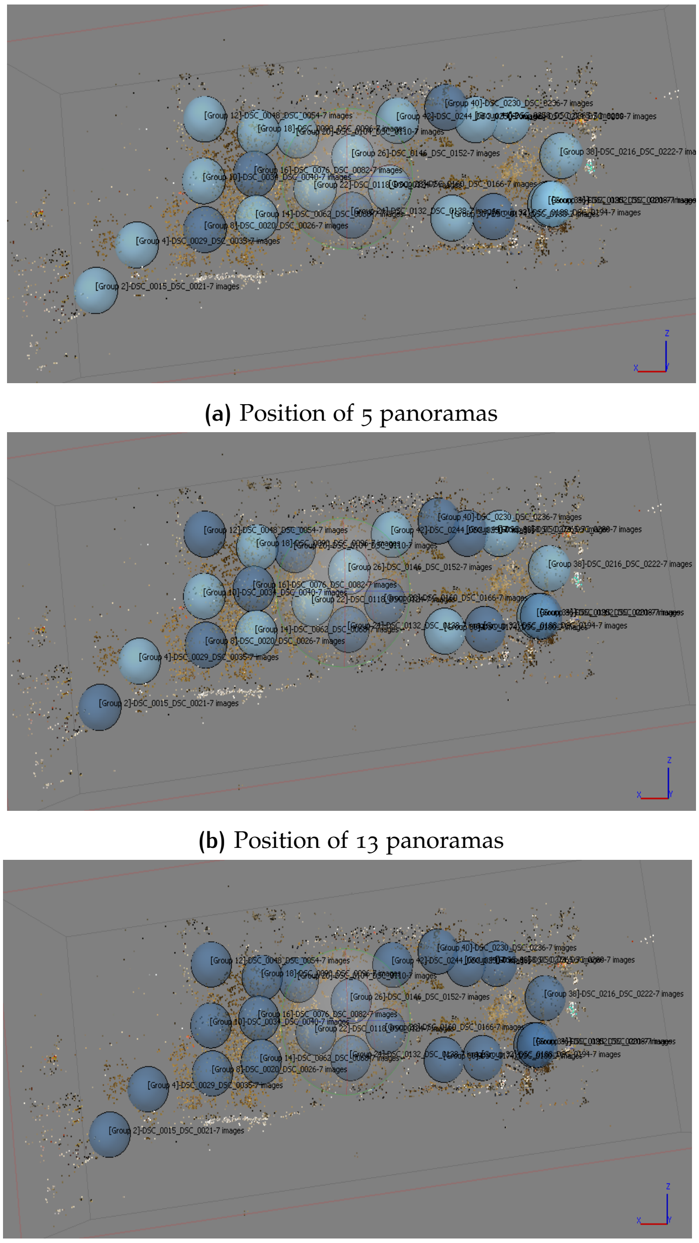
5.2.3 The impact of the Panorama Numbers

The next aspect that was tested is the optimal number of images that are necessary for a point cloud similar to the one that is acquired from the laser scanner. In the test, three versions of the network were used. The first one uses all the (23) panoramas, the second one, half of the panoramas (10) and the last one, a quarter of the panoramas (5). Table 5.4 demonstrates that, as the number of the panoramas grows, the number of the points increases.

Table 5.4: Agisoft Photoscan details while decreasing the number of the panoramas.

Images	Quality	Dense cloud	Time
23	High	40 011 520	6:30 hours
10	High	25 619 925	3:30 hours
5	High	6 133 907	1:00 hours

It is important to notice the number of points that are measured by the software. Within almost 6 and a half hours, the program generated 40 011 520 points in high quality from 23 images (with pipes), but only 3 hours were needed for generating 25 619 925 points from 10 panoramas. The Figure 5.8 shows the location of the panoramas, after the alignment of the images that is needed before continuing with the dense point cloud generation. The blue spheres show the position of the images. When the sphere is dark blue, it means that the software is taking into account the image in order to create the point cloud. If the image has light blue color, it means that it is unable and the software is not taking into account this image for the generation of the point cloud. The accuracy of the different point clouds that are generating from the different number of the panoramas, it is evaluated comparing the point clouds. Because the alignment of the position of the panoramas do not changed also, the differences between the points are less than 1cm. The comparison has been done in CloudCompare.


[Group 12]-DSC_0048_DSC_0054-7 images
[Group 40]-DSC_0230_DSC_0236-7 images
[Group 26]-DSC_0146_DSC_0152-7 images
[Group 38]-DSC_0216_DSC_0222-7 images
[Group 16]-DSC_0076_DSC_0082-7 images
[Group 14]-DSC_0062_DSC
[Group 8]-DSC_0020_DSC_0026-7 images
[Group 4]-DSC_0029_DSC_0035-7 images
[Group 2]-DSC_0015_DSC_0021-7 images
z
x
y

(a) Position of 5 panoramas

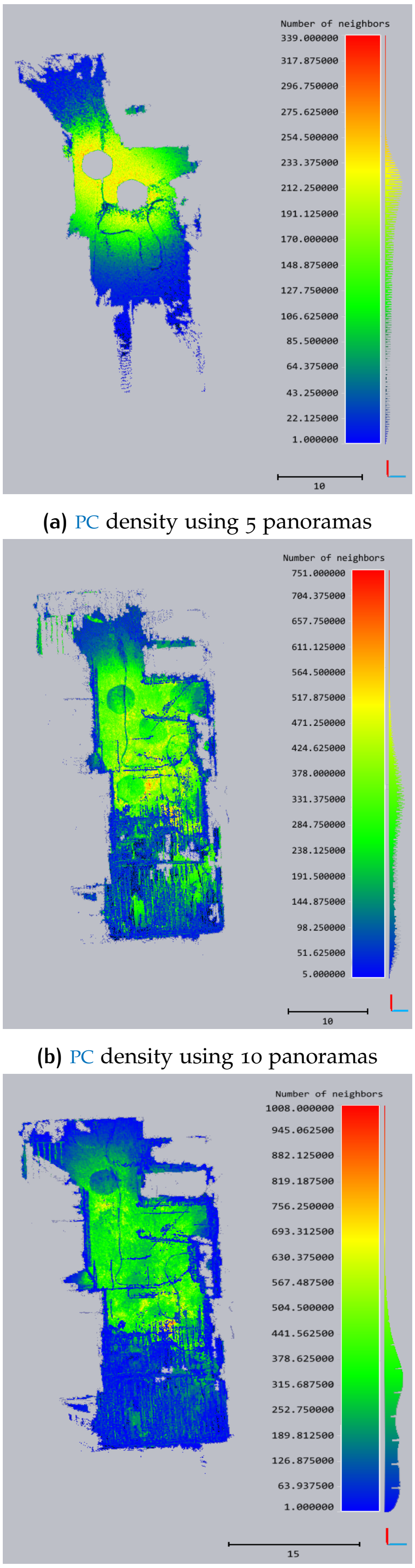
(b) Position of 13 panoramas

(c) Position of 23 panoramas

Figure 5.8: The alignment of the panoramas using different number.

The density of each point cloud is computed to show the differences while the number of the panoramas is decreased, 5.9.

The density is estimated by counting the number of neighbors N (inside a sphere of radius R) for each point. In our case the number for the density will be estimated in a radius of 0.05m. When comparing the results that are presented in Figure 5.9, it is noticeable that in general the part with the highest density of the whole point cloud is the floor. This is because the floor is a flat surface and could easily be reconstructed. It is important to mention that according to the Figure 5.9c which is the one using the 23 panoramas, the highest number of neighbors is 1008 in a radius of 0.05m. In contrast to the point cloud that is generated by 10 panoramas, the highest value reaches 751 neighbors. Furthermore, according to the diagram



(a) PC density using 5 panoramas

(b) PC density using 10 panoramas

(c) PC density using 23 panoramas

Figure 5.9: The density of the point cloud according to the different number of the panoramas, estimated with a sphere of 0.05 radius.

which is placed at the right of each figure, the average value of the neighbors that are in 0.05m radius, is almost the same, which is close to 330 neighbors.

5.3 PROJECTIONS ON TEST DATA

Following our geometric methodology, in Figure 4.9 the main goal is to find an efficient way to combine different methods for object detection. The methodology can be applied to different set of datasets. In our case, the pipes are the main object. Every object has its own characteristics and could be described by its own boundaries. Boundaries are important information and could definitely be useful for object detection. Images offer better boundary detection and through image processing, the edges can be extracted. The correlation between the 2D boundaries with the 3D points is valuable information. In this section, different approaches are implemented to ultimately find the most efficient way to achieve the connection between the panoramas and the point could that has been generated from them.
Agisoft Photoscan software provides a Python script that allows the user to execute all the procedure, providing access to the internal parameters. The projection of the 3D points back to the panoramas, taking into account the internal and the orientation parameters, it is possible for each image. As a result, each 3D point is projected, getting the pixel that is related to the specific panoramic image. Because of the 360°field of view, it is difficult to have points out of the image field. Moreover, this does not mean that every 3D point in the whole point cloud could be projected to one panorama (occlusion). After research, through these different kind of software and also, contacting with the developers, we could not find a software that could provide information about the relation between the 3D point and the image that is strongly related to. Through the Python script the projection error is estimated from each 3D point to the related image. This way, an estimation about the relation with the image is provided. The following approaches are computed to test whether the edges of the images could be related with the actual edges in the 3D point cloud.

5.3.1 First approach-Projection from 2D to 3D

First, image processing to the initial panoramas is applied to find the edges on the panoramic images. Canny edge detection is used in order to find the edges on the images, as it is the most accurate and promising method for edge detection Canny [1987]. In Chapter 4, the theory of the algorithm is briefly explained. The detected edges are stored with their positions and their orientations, Figure 5.10 depicts the canny edge results. Different thresholds are tested to reduce the noise in the final image. Lighting conditions and the reflectance of the material could cause problems at the final result.

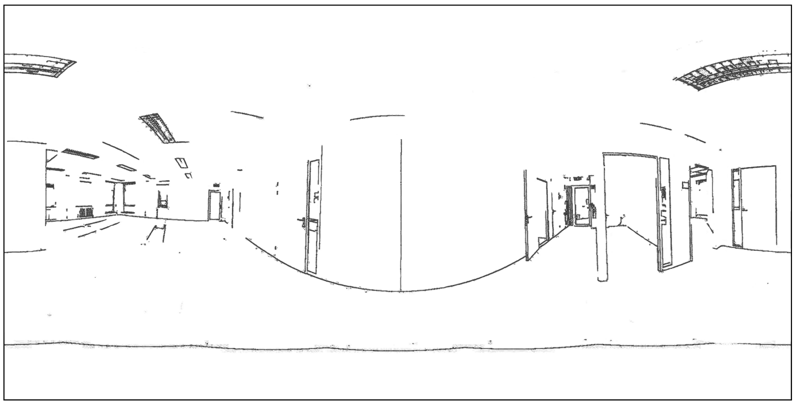


Figure 5.10: Canny edge detection.

In the first implementation, the boundaries (pixels) were used from each panorama in order, to project them, as 3D points and to check from this way, if we could get the 3D edges, Figure 5.11. This projection could be done using the python script that integrated to Agisoft Photoscan, using the internal parameters of each panorama. The result is not as it is expected. The result is even worse, if the edges are combined from the projection from different panoramas, Figure 5.12. The edges from the images are not necessarily responding to the actual 3D points and as a result, the projected edges do not have the real depth. It's worth mentioning that some pixel-boundaries of the panoramas are not connected with the real point cloud because the software could not reconstruct them.
Many details have been detected from the image edge detection which as points do not exist, as a result a lot of noise is produced, Figure 5.12.

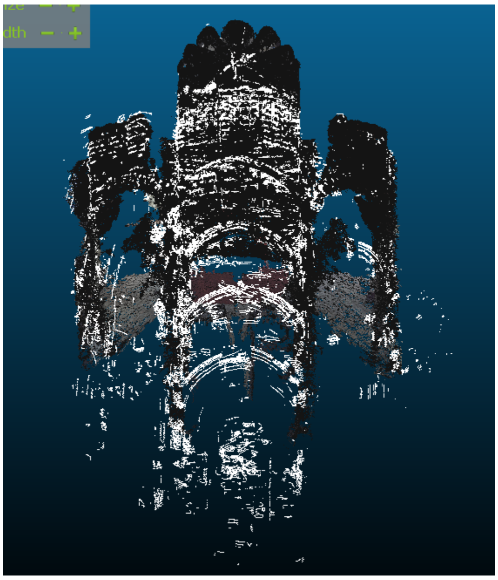


Figure 5.11: First approach: Edges in white color direct projected in 3D space.

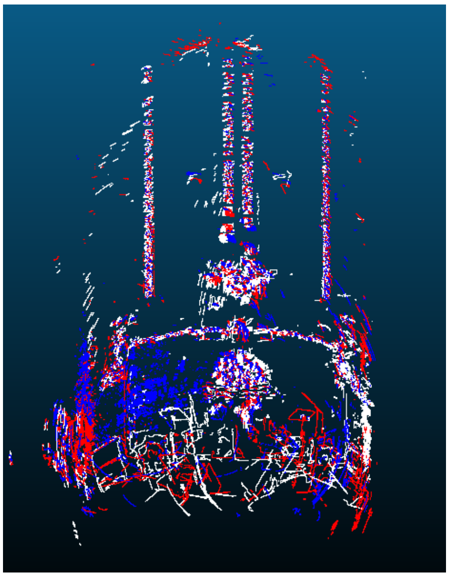


Figure 5.12: First approach: Edges projected direct to 3D space from more than one image.

5.3.2 Second approach-Projection from 3D to 2D

Taking into account some constraints, the existing 3D points were projected back to pixels for each image, explained in Chapter 4 , Section 4.2.3. This introduced some problems that were followed with some possible solutions. Due to image occlusion, it is not possible that all 3D points are part of one image. Of course, some points would not be generated from one image, so this led us to remove, the points that have a projected error from each image. Thus, a maximum distance between the points and the image position is specified. Another important constraint is to define the closest distance from the position of the 3D point to the 3D edges, which are projected from the first approach to the point cloud. Define a distance threshold, the edges could be related with the closest points. This results to the points being related to the edges from each image. Each constraint speeds the whole procedure and gets the desirable output, Figure 5.13.

5.3.3 Third approach-Direct projection through the point cloud generation

Another idea was to detect the straight lines on the panoramas and project them to the initial photos. Hough line transform is used for the line detection on images,

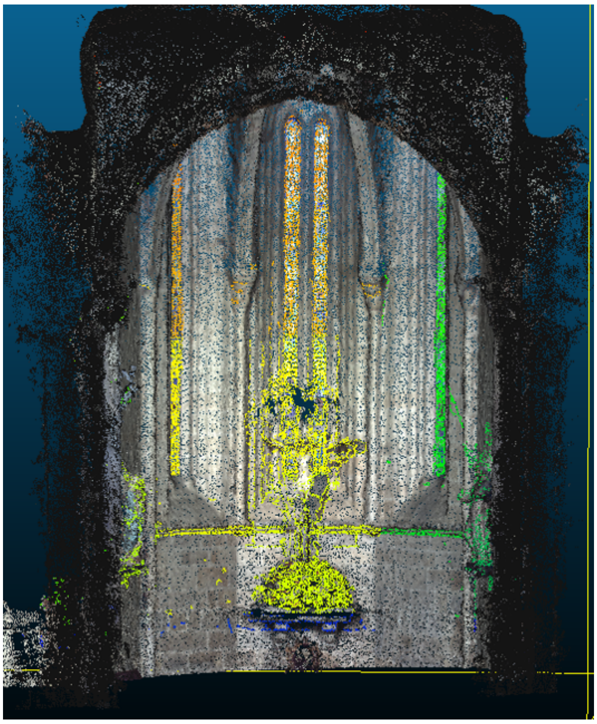


Figure 5.13: Connection existing 3D points with the pixels which are edges on images.

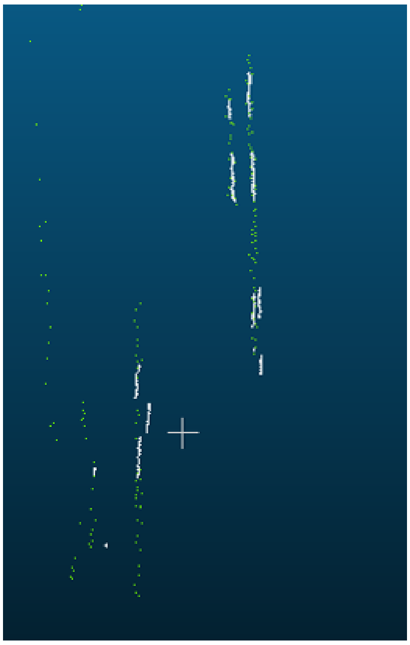


Figure 5.14: Second approach: 3D points that corresponds to pixels which are part of the edge in image

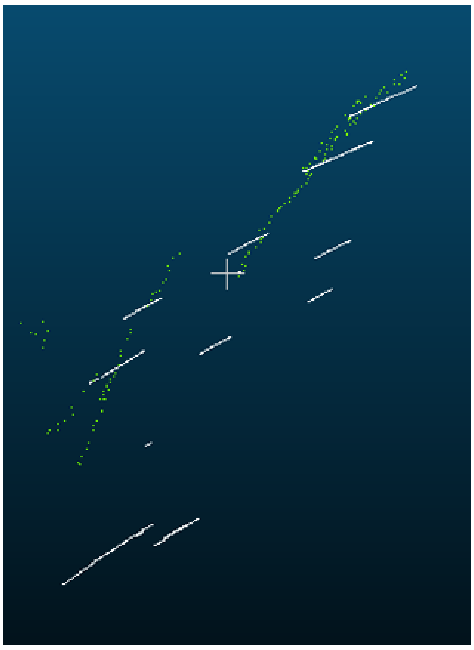


Figure 5.15: Second approach: Differences between pixels that are edges on images, direct projected in 3D with the actual 3D points.

Figure 5.16.

The software for the point cloud generation took these images as input. After applying the HT the lines detected are colored red. If the lines in the panoramas exist in the same position in more than three panoramas, it is expected to color the point cloud red at this part. The result in Figure 5.17, seems promising, as some lines are depicted as red points through the whole point cloud. This way, some of

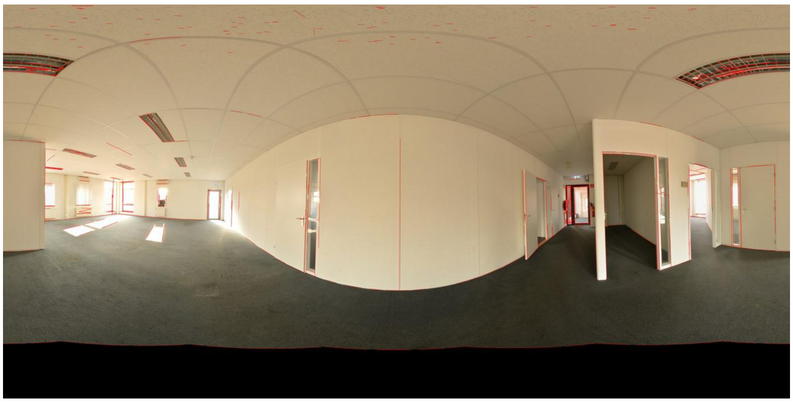


Figure 5.16: Hough line 2D transform.

the boundaries are clearly defined directly to the point cloud, Figure 5.17 and 5.18. The fact that the lines are colored in red, create some doubts for the final result because the algorithms are using the features for the image orientation.

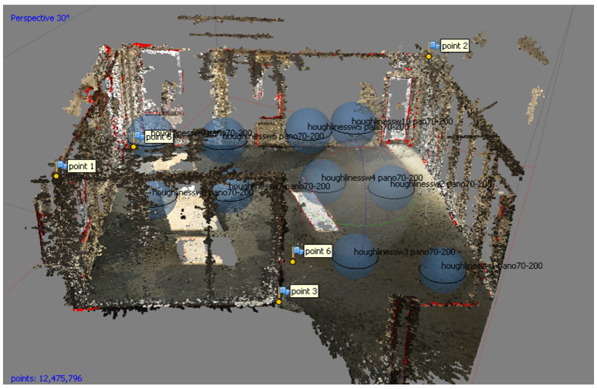


Figure 5.17: Point cloud with edges in red color.

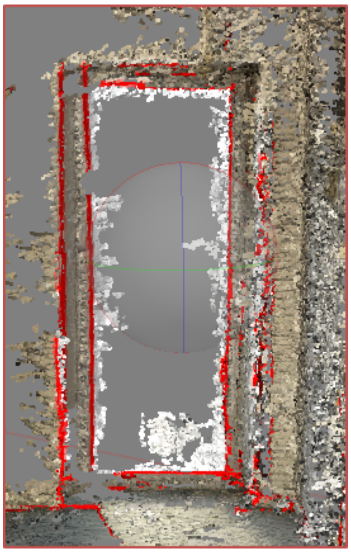


Figure 5.18: Door with edge detection.

Then, as the edges are defined according to our methodology described in Chapter 4, Figure 4.9, the next step is to fit the lines to the points that are boundaries. To achieve this, it is tried to fit Hough 3D line transform to the points that are red in the point cloud. As mentioned before, this method is used for the 2D direction to detect lines through raster images. Now, our scope is to apply this method in 3D environment. Due to the additional dimension, the number of the parameters are larger in 3D space than in 2D, making the same procedure more difficult for the same output [Watson, 2017]. The algorithm has been already explained in Chapter 4, Section Image processing. In article [Dalitz et al., 2017], Hough 3D line transform is applied on point clouds. Some parameters are predefined in order to execute the algorithm:

- The xy step width (dx): Which defines the hough space cell in the xy plane.
- The maximum number of lines: The maximum number of lines that will return.
- The minimum vote count. The lines with less than the minimum votes are not returned.

- The returned information is: The detected lines are returned with the number of the points, the anchor point and the line direction. Figure 5.19 shows the result for our test data.

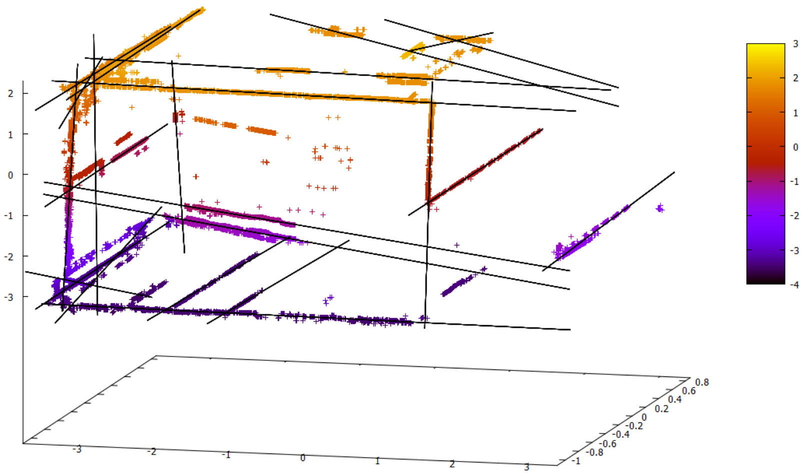


Figure 5.19: Fit lines to points applying 3D Hough line transform (Data: Empty room)

In the end, the parameters of the lines are defined in the 3D environment. These parameters are the normal vector (the direction of the line) and the one point part of the line (the mid point of the points that are part of the line). These parameters, are possible to use in order to detect, if a cylinder exists close to them. In Section 5.4, the data for our main application are tested, combining the 2D boundaries with 3D points part of pipe boundaries, for the connection points-pixels, following the methodology that is described in Chapter 4, Section 4.2.3

5.4 SMART POINT CLOUD FRAMEWORK FOR PIPE DETECTION

In this Section, the SPC framework is applied for pipe detection using the panoramas for point cloud acquisition. In Figure 5.20, the different expertises are clearly defined, while each one has its own characteristics and techniques that are applied to the point cloud for the pipe detection. Every part of the methodology is fully connected to the other, making the whole methodology strongly related and based to the data base system. Following this framework, the results from each expertise are stored in such a way that provide, an intelligent environment for analyzing the data. The techniques take full advantage from the given data 2D and 3D to combine the features in order to detect the pipes. The whole procedure of the implementation of this framework is provided in this Section.

The actual data which are the panoramic images with the pipes, the whole procedure is applied from the beginning, combining the second and the third approach. For the final point cloud 23 panoramas are used. The result is presented in Figure 5.23 and some statistics about the procedure are shown in Table 5.5
The processing procedure includes two main stages:

1. The first part is the camera alignment. At this stage, the software tries to find common points on the photographs and matches them in order to find the position of the camera and defines the camera calibration parameters. As a result, it creates a sparse point cloud.

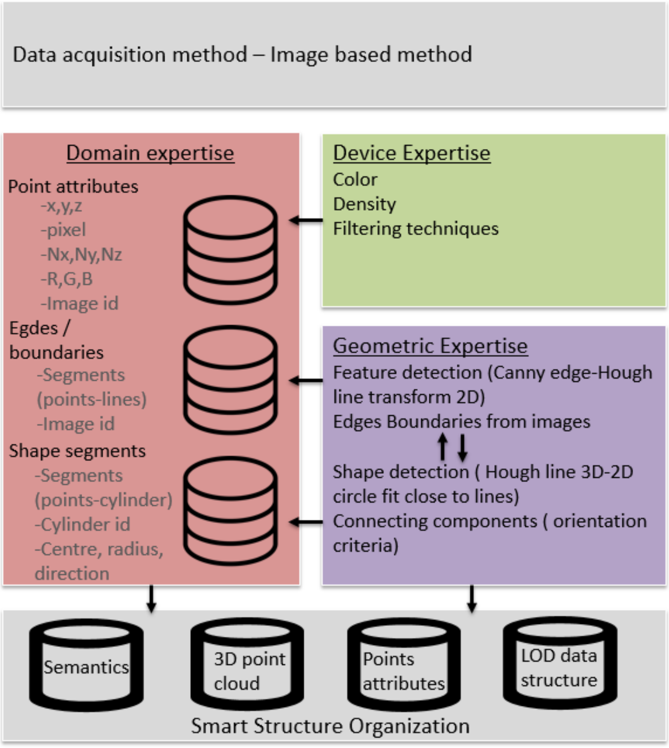

Data acquisition method – Image based method
Domain expertise
Point attributes
-x,y,z
-pixel
-Nx,Ny,Nz
-R,G,B
-Image id
Egdes / boundaries
-Segments (points-lines)
-Image id
Shape segments
-Segments (points-cylinder)
-Cylinder id
-Centre, radius, direction
Device Expertise
Color
Density
Filtering techniques
Geometric Expertise
Feature detection (Canny edge-Hough line transform 2D)
Edges Boundaries from images
Shape detection (Hough line 3D-2D circle fit close to lines)
Connecting components (orientation criteria)
Semantics
3D point cloud
Points attributes
LOD data structure
Smart Structure Organization

Figure 5.20: SPC framework.

2. The next part is about generating the dense point cloud based on the camera positions and pictures.

Table 5.5: Information about software stages on same source data (with pipes).

Stage	Images	Type of image	Quality	Result	Time
Alignment	322	Fish eye	Low quality	No result	
Alignment	46	Spherical	High	47 841 tie points	1:40 hours
Alignment	23	Spherical	High	Not enough memory	—
Dense point cloud	23	Spherical	Medium	11 543 997	1 hour
Dense point cloud	23	Spherical	High	40 011 520	6:30 hour

According to the manual of the software Agisoft Photoscan , the number of photos that can be processed, depends on the available RAM and reconstruction parameters. Assuming that a single photo resolution is 10 MPix, 4 GB RAM is sufficient to make a model based on 30 to 50 photos. 16 GB RAM will allow to process up to 300-400 photographs. In our case, the available RAM is 16 GB RAM but, the resolution of one panorama is 16882 × 8441. This means that the resolution of one image is almost 10 times higher than the single one. Therefore to what concerns processing, the computer should be really powerful.

The point cloud that is used for the main application is the one that is created from 23 panoramic images with high quality and the result is a dense point cloud with 40 011 520 points, Figure 5.21. The problem is that the point cloud is always generated from images that are more noisy than the ones that are generated form the laser scanner. In Section 4.2.2, in Chapter 4 a method in order to reduce the noise of the point cloud is presented and is applied to our point cloud.

Filtering the point cloud is an important procedure because it removes the outliers and the noise, that will probably create problems in the implementation and influence the final result. After filtering the point cloud the points reduced to 32 415 567 points. The difference is observed when looking at the two Figures 5.21 and 5.22.

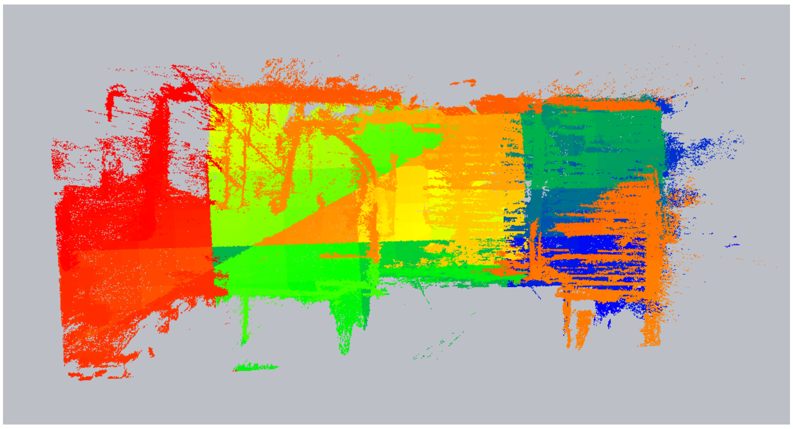


Figure 5.21: The raw point cloud.

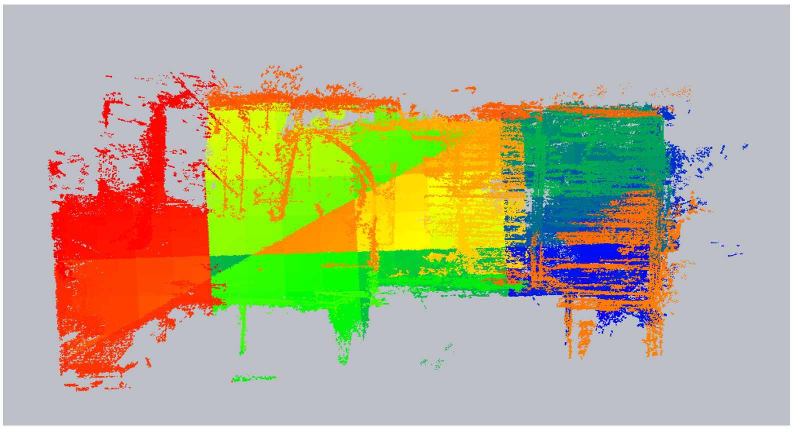


Figure 5.22: Filtered point cloud from noise and outliers.

5.4.1 Extract the boundaries in point cloud from panoramas

The combination of the two methods that are analyzed in Section 5.3.2 and 5.3.3 contributes to a better estimation of the boundaries of the cylinders in 3D. After applying the Canny detection and the Hough line detection to the panoramas did not provide a result with clear points of the boundaries from the cylinders. As it can be observed in Figure 5.23 and in Figure 5.24, only the points that are part of the boundaries are represented.

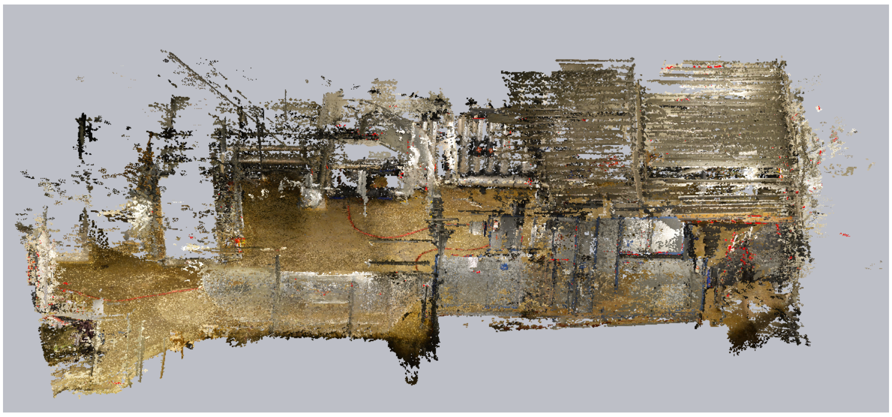


Figure 5.23: Point cloud with red boundaries from images

In photogrammetry, in order to get the actual 3D point cloud from the panoramas, it is required to see the same part of one real object at least in three images. Our

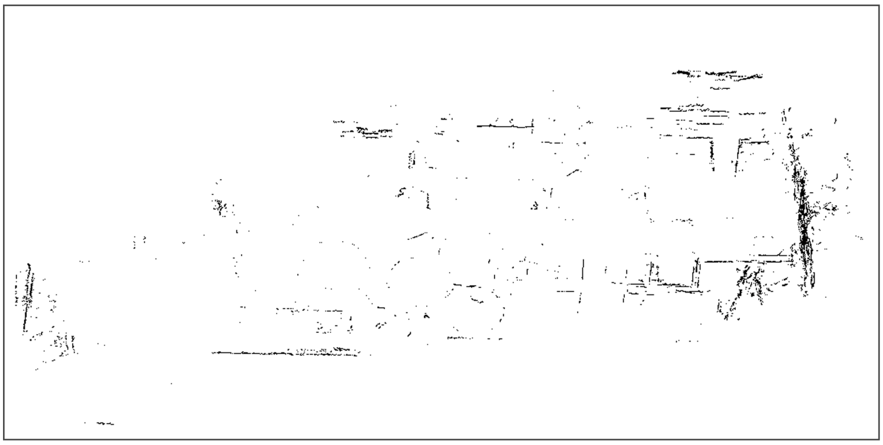


Figure 5.24: The points that are part of the boundaries from images

concept, works really well when the edges of the object are clear and can be detected in all the images. In Figure B.3, there is an example of the boundaries of the door where the 3D point boundaries are clearly visualized in our final point cloud.

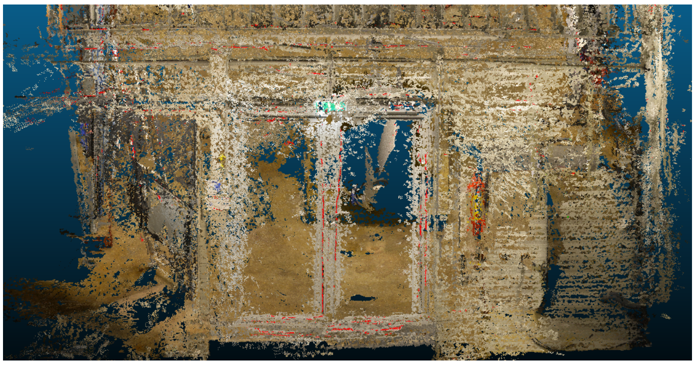
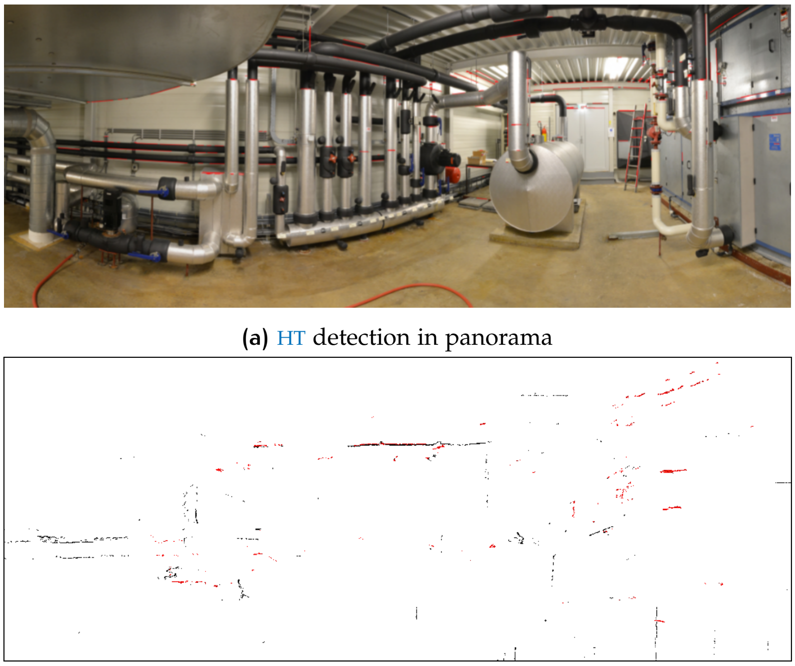


Figure 5.25: Example: Boundaries of the door.

The problem is that the cylinders have no edges and can be projected with different boundaries, always depending on the angle the photo is captured. Hence, it is difficult to have the boundaries of the same pipe in different three images, in order to depict with the same color in the 3D point cloud. Then, the second approach is used to enhance the final result. In the second approach, as described in the previous Section, the lines from the HT detection from some panoramas are re-projected from 2D in 3D. This approach contributes in providing more points that are lines in the point cloud, in order to acquire more points-boundaries that are part of cylinders. Figures in 5.26, 5.27, 5.28 are 3 different example panoramas with cylinders, demonstrates the projection of the boundaries to the point cloud. In Appendix B, panoramas are provided in bigger size, for better visualization of the Hough Line Transform results.

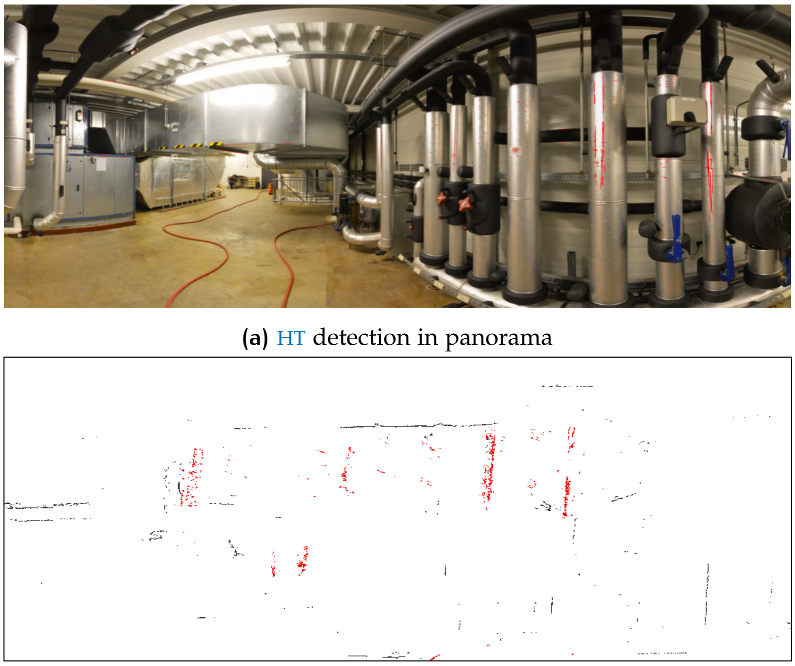
Projecting the points which are part of the lines in 3D environment has results in the final output. In the first example, Figure 5.26 more points are projected in the same line segment or are extended at the same direction of the existing points-line segment. In the third example, the result is more confusing, since the 3D points that are projected on the panorama are not part of the big cylinder. By projecting them back to the point cloud as boundaries probably confuses us, because in the end a line is not able to fit to scattered points.
While having the points that are part of lines, the 3D Hough line transform is again applied now in 3D environment, as it is described in Chapter 4, PROTOTYPE DESIGN, in Section 4.2.5.



(a) HT detection in panorama

(b) Projection in 3D the pixels that are boundaries

Figure 5.26: First example to enhance the boundaries in 3D.

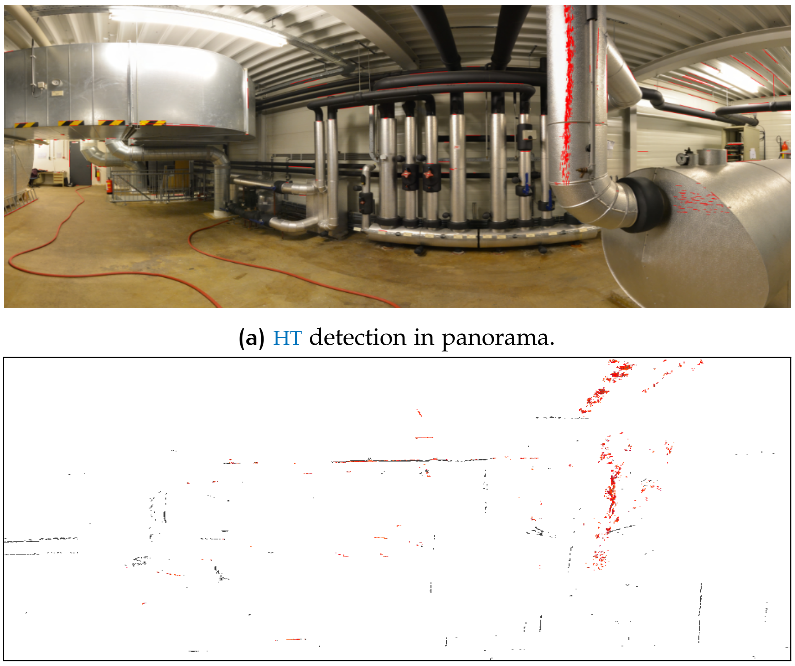


(a) HT detection in panorama

(b) Projection in 3D the pixels that are boundaries

Figure 5.27: Second example to enhance the boundaries in 3D.

For the final implementation, the methodology for the pipe detection is applied in a segmented part of the whole point cloud, Figure 5.8. The points that are boundaries of the pipes, projected on the 3D environment are also, shown in red at the same Figure. Then, the 3D HT is applied to get the parameters lines. Three lines are detected and the points that are close to these lines are tested, if they are part of the cylinder.



(a) HT detection in panorama.

(b) Projection in 3D the pixels that are boundaries.

Figure 5.28: Third example to enhance the boundaries in 3D.

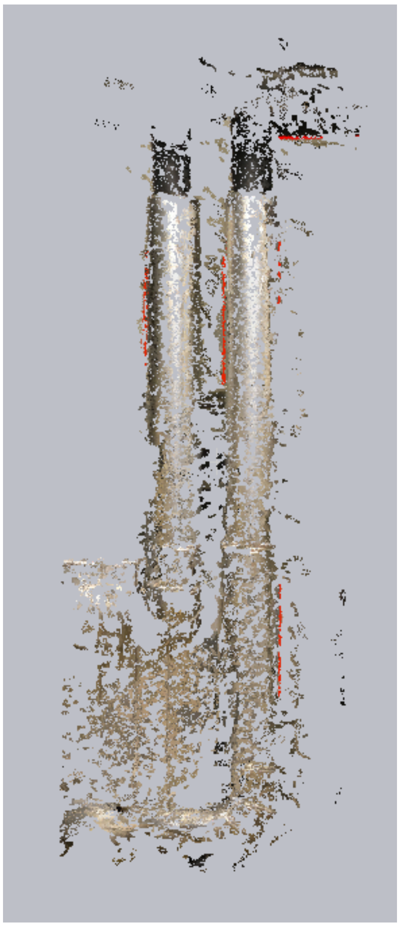


Figure 5.29: Segmented part of the point cloud for the pipe detection.

5.4.2 Fit Circle and Pipe detection

In order to test, if on the projected 2D segments, a circle could fit, the methodology is described in Chapter 4, Section 4.2.5. To do this, the perpendicular plane cut, the

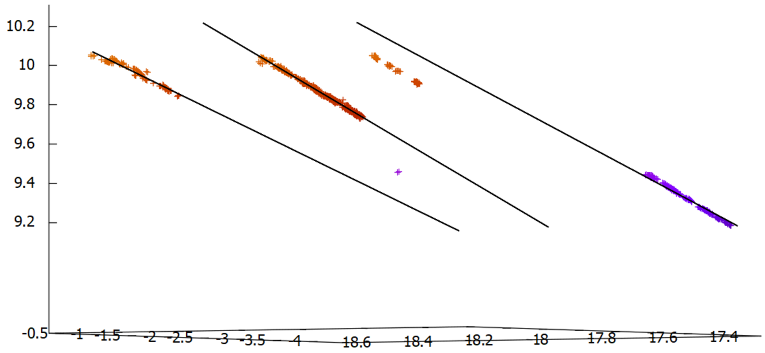


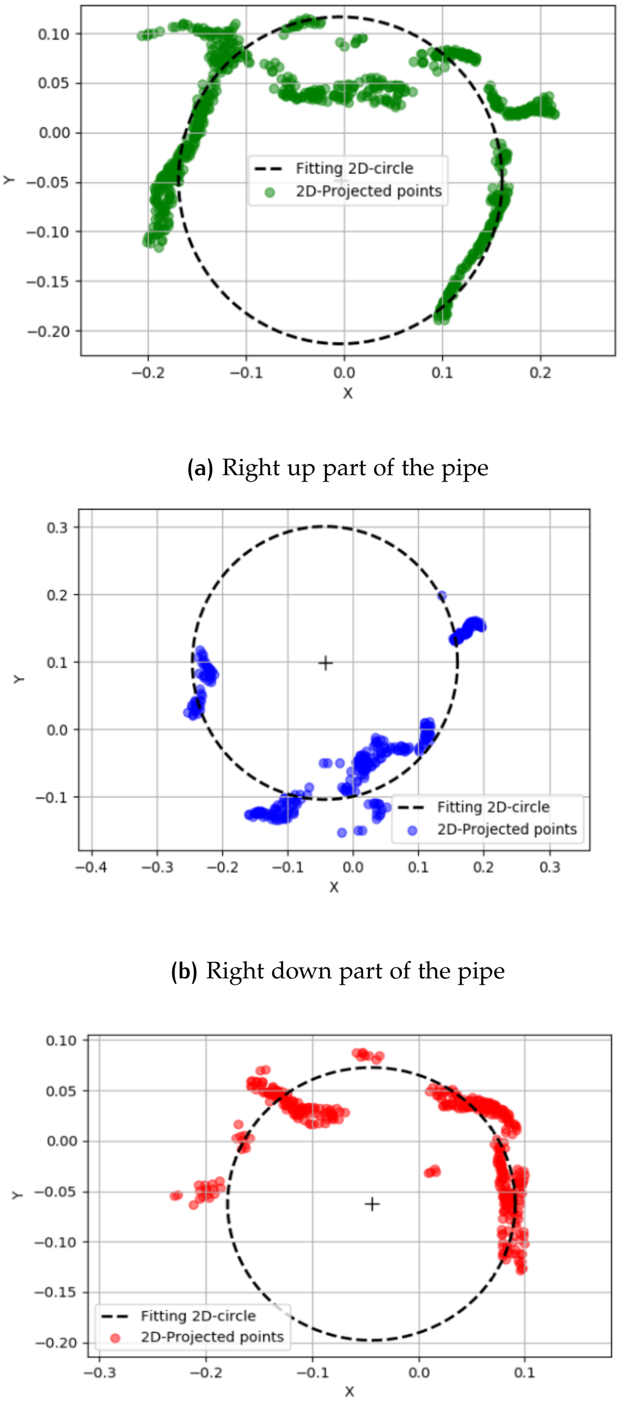
Figure 5.30: 3D HT to fit lines to segmented points.

parallel line on the mid point. Then, the points that are close to the parallel line and close to the perpendicular plane are projected on the perpendicular plane. If the points are part of a circle, then they are probably a part of a pipe too. From the 2D circle that fits on the projected points, it is able to extract the parameters in order to construct the cylinder. To specify which points are part of the cylinder, first the distance from point to line center is computed, if the distance is close to the radius then the point is part of the cylinder. The points that fulfill these conditions means that are part of the cylinder and are defined as segmented points. Table 5.8, provides the characteristics and the parameters of each cylinder. Three cylinders are detected as it is expected, Figure 5.39. In order to understand about which cylinder we are referred to, the cylinders are mentioned according their position at the segment: Right up cylinder is represented with green, the right down with blue and the left with red color. From the parameters that are displayed in Table 5.8, estimations of the accuracy of the results can be extracted. From the vector direction of the cylinders, it is shown that the cylinders are parallel to each other which is correct. Furthermore, the two segments of the cylinders are part of the same pipe. This is also proven in the connecting components that are analyzed below. It could be also guessed when observing the table that the direction and the radius have similar values.

Table 5.6: Cylinder parameters.

Cylinder	Center	Direction	Radius	RMS	Real Radius (m)
Right up	(17.9321, -0.9212, 10.089)	(-0.1314, -0.9647, -0.2281)	0.18	0.015	0.11
Right down	(17.5603 ,-3.64438 ,9.52677)	(0.12682, 0.9685, 0.2141)	0.165	0.039	0.11
Left	(18.3937 ,-1.47341 ,10.0645)	(0.0967, 0.9728, 0.2101)	0.152	0.03	0.13

According to our methodology, Chapter 4, each segment is defined and connected to the other segments fulfilled some criteria. If they fulfill the criteria then, they are considered as part of the same cylinder. It is an iterative procedure while in the end the whole pipe is identified. When this procedure is tested on the segmented part of the point cloud, three cylinders are detected. In Figure 5.31, the 2D circles are plotted applying least square adjustment, best fit of the circle. From the plots, it is obvious that the points that are close to the lines are not perfect curves and there is still a lot of noise. Then, the circle is plotted back to the 3D environment, Figure 5.40. Finally, with these parameters the surface of the fitting cylinder could be generated for visualization reasons. Finally, the cylinder is fitted to the segmented points, Figure 5.37. In our case, based on some criteria, it is examined whether the segments are part of the same cylinder. First, it is checked, if the direction of the line is the same with another line direction. Then, if the two segments have also points in common, that means that they are part of one cylinder. Finally, the Root Mean Square Error (RMSE) is computed. It is estimated how good enough the points are fit to the detected cylinders. If the RMSE is less than 1, it seems that the segment is likely to be a cylinder. For the validation of our technique, the radius of the cylinders were measured. Also, some characteristic parts were measured (door width) to be able to compute the scale of our point cloud and then, to estimate the accuracy of our results.



(a) Right up part of the pipe

(b) Right down part of the pipe

(c) Left pipe

Figure 5.31: Fit 2D circle to projected points.

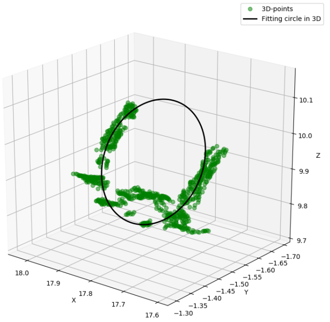


Figure 5.32: Right up part of the circle in 3D

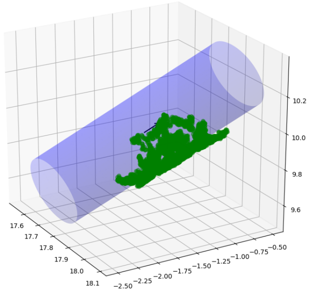


Figure 5.33: Right up part of the pipe points fit on cylinder.

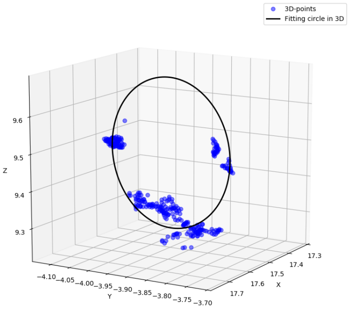


Figure 5.34: Right dowm part of the circle in 3D

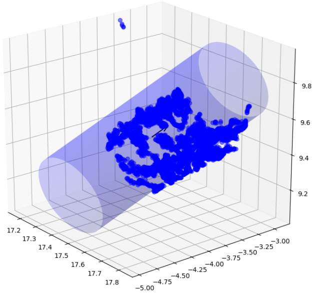


Figure 5.35: Right down part of the pipe points fit on cylinder.

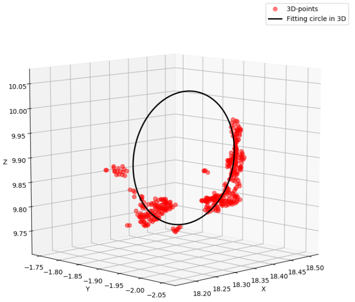


Figure 5.36: Left part of the circle in 3D

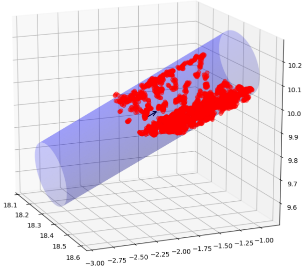


Figure 5.37: Left part of the pipe points fit on cylinder.

5.4.3 Store in Database

The final segments of the points are stored in a database. In A the SQL statements for the creation of the tables and the storage of data are displayed, following the UML schema that is defined is Chapter 4, Section 4.2.7.
The storage of the point cloud is done with PostgreSQL using PostGis extension.

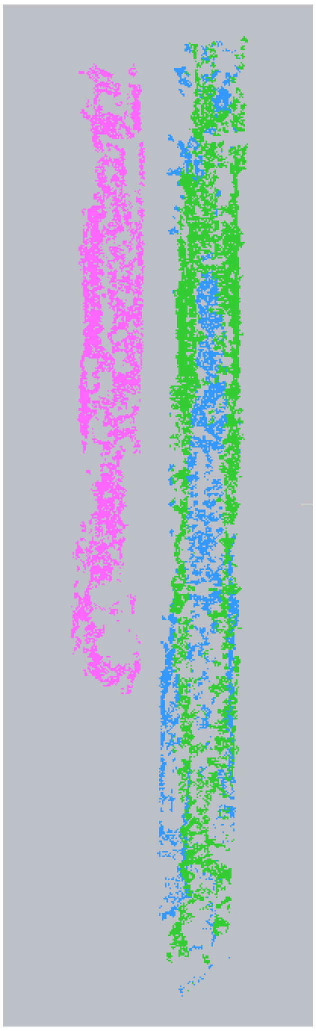


Figure 5.38: Points part of the three cylinders.

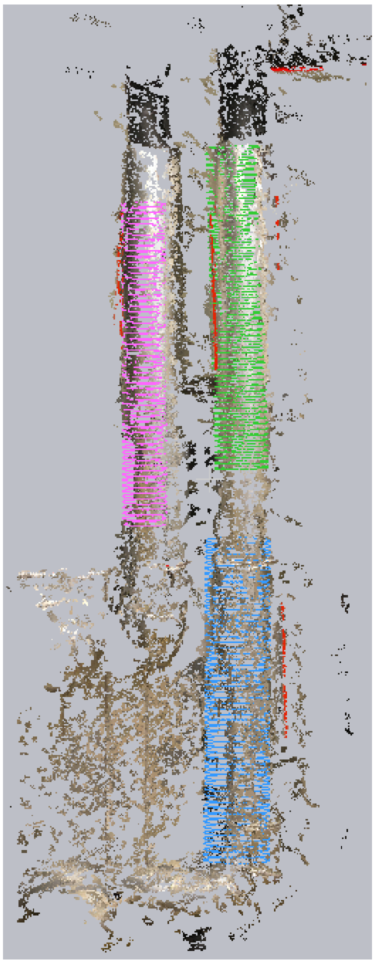


Figure 5.39: Visualization of the three cylinders on the point cloud

Three tables are created according the UML schema (points, segments, pipes). Specifically, the structure of the database storage is such that storing billions of points as individual records in a table is not an efficient use of resources. Instead, the segments are going to be stored as groups of `PcPoint`. Each pipe should hopefully contain points that are part of the same geometry. Instead of a table of billions of points, single `MultiPoint` records is a collection of point data that can be represented in the database, as a much smaller collection of points records. The geometry of the pipes is stored with the following characteristics: the Id of the pipe which is a serial number, the centre of the pipe which is stored with geometry `PointZ`, the direction of the pipes like a vector, the radius which is float and finally, the segment of points that are part of the pipe, stored as `MultiPointZ`.

5.4.4 From the point cloud to virtual and augmented reality

Interaction is one of the main factors of mixed reality. A graphic application based on Unity could be developed. This application displays the point cloud including also, the geometric information related to geometry and semantics. These elements are responsible for the selection of scene points-objects by mouse-clicking, which will provide interactivity, displaying the information that is related to the points.

It is possible to add tags which store additional attributes related to the points of interest, that will allow the user to add aspects to the point selected on the screen which is related to one pipe. In the end, when one point is clicked the whole pipe that the point is related to is colored.

5.4.5 Fit second polynomial curve

Since the main object of interest is the cylinder, which can be represented by second polynomial order, the points in each segment are fitted to a quadratic equation. According to our methodology in Section 4.2.6, the parameters of the polynomial could be estimated and for the fitted polynomial the principal curvatures of the surface can be computed. In Table 5.8, the principal curvatures and the radius, are estimated following the formulas in Section 4.2.6. The segment could be cylindrical, if $k_1 > 5$, $k_2 < 1$. Furthermore, the visualization of the estimated polynomial is represented, in Figure 5.40 ,5.41 and 5.42 for three segments. It can be observed form the Figures that the fitted polynomial surface does not look like a cylinder and only two cylinders out of the three segments are detected as cylinders. It is another way to implement our methodology and to estimate and investigate, if the segment is part of a cylinder. To compare the two methods, this method is faster than the previous because you can directly fit the second polynomia and avoid the projection in 2D and rotation techniques. Moreover, the parameters of the cylinders are not estimated directly, only the radius could be defined. The radius values that are estimated from 2D circle fit are close with the radius values that are estimated form second polynomial fit, close to 1-2cm difference.

Table 5.7: Polynomial coefficients.

Segment	k1	k2	R	Cylinder
Right up	-0.6423	-2.3285	–	No
Right down	10.4652	-3.0463	0.0955	Yes
Left	6.6012	-1.7080	0.1514	Yes

5.4.6 Compare with RANSAC results

When comparing our methodology by applying the RANSAC algorithm for pipe detection directly to the point cloud, it is vital to have a concrete idea about our methodology's performance. RANSAC algorithm is a broad technique that is general used for shape detection in 2D and 3D. The results are compared and analyzed. In Chapter 4, Section 4.2.9 RANSAC algorithm is explained using the cylinder model parameters. Our method is based on an algorithm that is provided from Oesau et al. [2018] using CGAL Library, [Hemmer, 2018]. There are five parameters that should be pre-defined and which effect the output.

- `Epsilon` and `Normal threshold`: Parameter `epsilon` defines the maximum tolerance Euclidean distance between a point and a shape. This is important value that leads to over or under segmentation, default value is 0.01. The error between a point and the shape is defined by its Euclidean distance and normal deviation. The normal deviation is computed between the normal at the point and the normal of the shape at the closest projected point on the shape. Default value is 0.95.

- `Minimum points`: The minimum points that are part of the shape that is detected. Default value is 500.

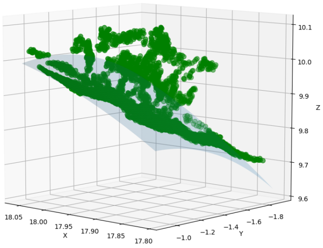


Figure 5.40: 2nd polynomial fit to right up segment.

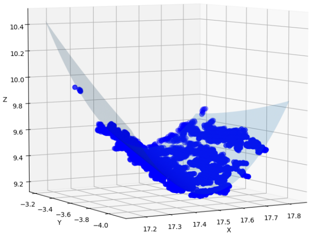


Figure 5.41: 2nd polynomial fit to the right down segment.

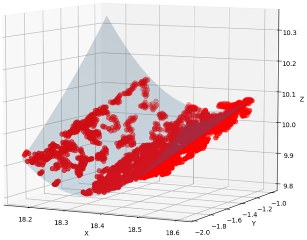


Figure 5.42: 2nd polynomial fit to the left segment.

- `k-NN`: The number of the nearest neighbors that should be taken into account for the normal estimation. Default value is 20.
- `Radius`: The `radius` threshold that determines the minimum radius where the cylinders are detected. Default value is 0.5.

After applying this algorithm to the same data, Figure 5.30, with the default values of the parameters that are analyzed above the output is that more cylinders are detected, close to 13-15 shapes. According to the results of our methodology, a prediction of the radius is provided to what concerns a decrease in the number of the detected shapes to 6. This happens due to the noisy point cloud, because the points do not have a proper geometry cylinder shape. In Figure 5.43 the detected cylinders are visualized. A good idea could be to apply the RANSAC algorithm only to the segments close to the boundaries. This way, the results could be improved. The implementation is carried out at the command prompt environment and the output are the shapes with the cylinder characteristics: center, radius, direction, points. Also, an output .ply file is provided which visualizes the point that is part of cylinders with different color. Table 5.8 shows the parameters of the cylinders that are detected from RANSAC algorithm. Six cylinders are detected with their characteristics. As it is observed only the second and the fifth cylinder are close to our results.

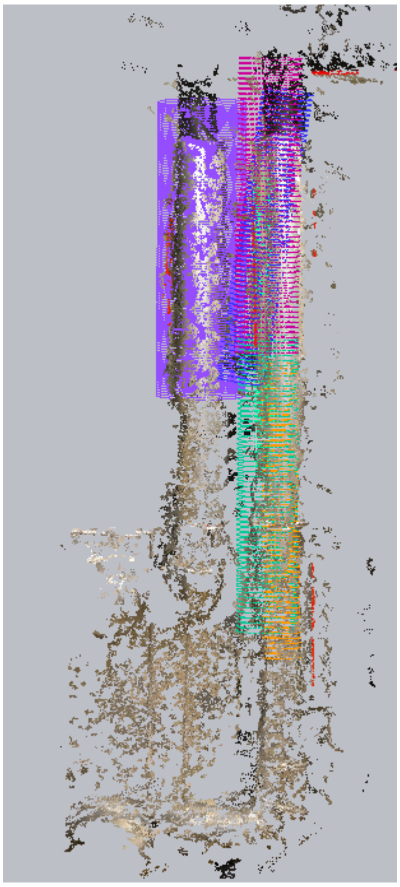


Figure 5.43: Visualization of the cylinders that are detected from RANSAC algorithm.

Table 5.8: RANSAC Cylinder parameters.

Cylinder	Center	Direction	Radius	Color
1	(17.873, -1.7268, 9.9187)	(-0.146, -0.975, -0.164)	0.065	🟩
2	(17.796, -1.814, 11.056)	(0.1185, -0.952, -0.2799)	0.126	🟧
3	(17.950, -0.9357, 10.025)	(-0.0970, -0.979, -0.176)	0.215	🟪
4	(17.724, -1.754, 9.904)	(-0.159, 0.9820, -0.1016)	0.143	🟦
5	(17.692, -2.809, 9.696)	(0.1207, 0.968, 0.215)	0.205	🟩
6	(18.293, -1.372, 10.277)	(0.120, 0.9689, 0.2158)	0.3	🟪

6 CONCLUSIONS & FUTURE WORK

This final Chapter takes into consideration all the conducted research that has been done and explained in the previous Chapters. The results and the problems that are addressed during the whole procedure lead to some final remarks and discussions. In Section 6.1 the main and the sub-questions will be briefly answered, in Section 6.2 the contribution and final remarks of our method are discussed and recommendations about the future work reviewed in 6.3.

6.1 RESEARCH QUESTIONS

This Section answers the research question and the sub-questions, which are presented in Section 1.1. In doing so, the sub-questions are being answered first. The sub-questions are essential for the whole structure of the research and also, in the end to answer the main research question. Finally, the main research question is answered.
To begin with the sub-questions:

1. What are the differences between point clouds that are created using terrestrial (close-range) photogrammetry and laser scanners?

   The differences between these two methods are defined in many aspects, see Section 4.1.1. Each one has each own advantages and disadvantages, depending on the application and the accuracy of the result that is needed, the methodology could be defined. Image-based techniques for point cloud generation provide a quick, low cost procedure that is generally used for visual requirements because the images provide a colored point cloud and is used more when the object is small and has more details. Instead, laser scanner is used more for bigger environments, producing massive point clouds with high accuracy but with difficulties in management. In many cases, a combination of both techniques is recommended. In our methodology, image based technique for point cloud acquisition, is used. The 2D features and characteristics that panoramas could offer are extracted in order to get the boundaries in 3D. Instead of applying the algorithms directly for object detection, a SPC framework is created in which, different techniques are combined in an efficient way to get the parameters of the object that are needed to be reconstructed. The choice of using images as the only input data, is not only for the data acquisition but also, contributes on the object detection.

2. To what extent do the number of panoramic images affect the generated point cloud in terms of density and accuracy?

   In Section 5.2.3, the examination whether the number of the panoramas are affecting the final result is investigated. For this reason the number of panoramas is reduced and the whole procedure is executed again. After plotting the results, it is determined that the alignment (position) of the images has not changed. Therefore, in terms of accuracy, the results should be the same too, due to the high resolution of the image. It is important to notice that the differences between the point clouds are less than 1cm. In terms of density, the number of the panoramas will definitely affect the final result. The density of the point cloud based on close-range photogrammetry is mainly determined from the percentage of the overlapping between the panoramas, the quality and the resolution of the image and the light conditions of the image. Keeping the same quality, the same resolution and changing the number of the images, changes the density. It is expected that the most dense part from the point cloud is the floor as it is flat and can easily be reconstructed. Therefore, an increase in the number of images is critical to increase the density of the point cloud. In Chapter 4, Section 5.2.3 it is shown that with a radius of 0.05m, the maximum density using 5 panoramas is 340 neighbors, using 10 panoramas is 751 neighbors and with 23 panoramas the highest density is close to 1008 neighbors.

3. What is the optimal way for object detection, using feature detection and segmentation techniques on images, as applicable to point clouds?

   The images offer better boundaries visualization. It is easier to visualize and detect boundaries on the 2D images instead, directly on the 3D point cloud. This led to the use of the imgaes, in order to connect the 2D image information with the 3D point cloud and to acquire semantic information. Furthermore, the use of the panoramas for the point cloud acquisition is a challenge. Not so many implementations have been done using spherical images for point cloud generation. The actual choice for using the panoramas due the fact that they provide 360°field of view, acquiring better results from the boundaries of one object and not part of it. As it is already mentioned, the panoramas are not only the input data for point cloud acquisition but also, part of our methodology for object detection. Combining these two main factors makes the whole procedure easier for the segmentation and object recognition. According to our methodology, knowledge is provided directly to 3D points after the point cloud acquisition. This is because the image processing for boundary detection was applied initially on the images, before these images were used for the creation of the point cloud. The boundaries are detected with red color. After the point cloud generation the red boundaries that were the same in more than three images could be directly colored red in the 3D point cloud. According to the proposed technique, the detection of the object and the object fitting are achieved faster, because the needed information is found and it can be assigned to the object providing promising results.

4. What is the best way to combine domain-geometric-device knowledge for pipelines detection?

   The SPC framework combines three different sources: device knowledge (point, density, color, intensity etc), analytic knowledge (normal, shapes, segments etc) and domain knowledge (definition, semantic attributes etc). According to our methodology, the combination of these three domains are presented in Figure 5.20. The result is to define the pipe, as one object. The device knowledge from close-range photogrammetry includes the points, the color, the normals and the density. The analytic knowledge is about the segmentation techniques that combines the 2D (pixels) with the 3D (points). Finally, the domain knowledge contains the information about the actual type and characteristics of the object with semantics. The UML model shows the relationship between the points and the segments that are part of the pipe, the attributes can directly be integrated within the database system. This method combines different characteristics from different sources by using different techniques. The method seems promising and could be applied to different datasets for object recognition in 3D.

5. What are the possibilities of representing the data in VR/ AR?

   The main idea is an application that is based on the point cloud for mixed reality (VR and AR. This application displays the enriched point cloud with information related to stored geometry, attributes and semantics. At the same time, by means of the mouse click, each point element is responsible for the whole selection of the object (pipe) that this point belongs to. After the click, the information of that pipe (geometry attributes) will be displayed. These interactions provide a direct immersion in a coherent structure, bridging virtual environment with point cloud applications.

Lastly, the main research question:

What is the Smart Point Cloud way to provide a framework for object detection (pipelines), using panoramic images for point cloud creation?

The SPC framework provides an efficient way to analyze the data by using different techniques that are suitable in each case for acquiring the results. Since the subject is always different, the methods and the techniques that are used vary, in order to provide the desirable result. The important contribution is to associate the domain knowledge with the information that you gain from the data acquisition and the geometric analysis of the point cloud with an efficient storage, showing the connectivity between the points and the real objects. The structure that is proposed in our methodology for pipe detection, is a way to combine the 2D boundary information with the 3D points. This connection benefits the geometry analysis and finally, the pipe segmentation. The fact that the panoramas are used for point cloud acquisition and for the image processing is an important contribution as it is investigated, because it can efficiently improve the final segmentation by extracting the 2D features. The main idea is to store points instead of reconstructing the object that is detected. This implementation is a concrete use case benefiting both internal relationship, geometrical and domain knowledge. A future possible implementation can be an intelligent environment, where the user will be able to interact with the points and get the information that he wants.

6.2 contribution & discussions

In this thesis, the unstructured point clouds challenges are addressed. The SPC framework addresses these challenges according to Poux et al. [2017b] and introduces the idea of providing an intelligent environment where data from different sources are combined. Applying different techniques to extract the information that is needed according to the geometric methodology, is also discussed. The concept is to eliminate the modelling step, as it is always time-consuming, error-prone and most importantly leads to a loss of crucial information that could be directly exploited. The objective is to detect the object as a group of points that belongs to an object with specific characteristics. Interpreting point clouds requires specific knowledge and spatial skills in order to extract important information for the user. This highlights the necessity to attach domain information through visual and semantic variables. All the necessary information could be retrieved from the point cloud and its attributes, without the need of additional tasks, as are needed for full reconstruction, by making a more intelligent structure.
The major contribution to our field is that a SPC model is developed, bringing intelligence to point clouds via connected 2D with 3D available information, linking the available knowledge and procedures to acquire semantics in data. Our method is implemented in Python and PostgreSQL database and allows combining semantic and spatial concepts on points. The main goal is to increase interoperability and to

acquire enriched point cloud model to be adapted by many applications through specific domain ontologies.

6.3 FUTURE WORK & RECOMMENDATIONS

This section provides some recommendations for future work. The future work has to do with areas that were not investigated due to time limitations. Possibilities for improving some parts of the methodology in order to improve the final performance, is discussed.

- Some of the limitations of the implementation that influence the results concern image processing techniques. The lighting conditions, the reflectance and the fact that the images are panoramas, bring some problems in the analysis. Moreover, noise problems influence the expected result after applying the HT technique. Regarding the lighting conditions, a better capture in combination with the thresholds applying in HT could solve the problem. The straight lines extraction from the panoramas is more difficult as the horizontal lines are presented, curved. In this case, some previous projection to rectilinear projection that keeps the lines straight could improve the final result.

- Another factor that could enhance the final result is the procedure of the point cloud acquisition. Using panoramas for point cloud acquisition is not really a common technique. An investigation on how the point cloud could be more accurate without so much noise, will enhance the performance of the object detection and the point cloud processing in general.

- It is really important to plan the data capture procedure, as in photogrammerty, photos (quality, light conditions) and their overlapping area are the most important factors that someone should take into account for good quality of the point cloud reconstruction.

- Time limitations and difficulties in planning, make it impossible to acquire the point cloud from laser scanner. So, in the end the analysis had been done in a test example, Section 5.2.2. One of the possible future work is to acquire the point cloud from the laser scanner to investigate, in which terms the point cloud that is generated from laser scanner is more complete and accurate. Furthermore, a general conclusion is that an integrated point cloud generated by merging the process of the image-based and laser scanner point clouds will improve the completeness and consistency of the results.

- Apply our methodology to the whole point cloud. While implementing the geometric methodology to the whole point cloud for pipes detection, it would be interesting to investigate the performance and to display the final results.

- Apply our SPC framework to a different dataset. Our framework is a general framework that could be applied to different datasets for object recognition. The device, geometric and domain expertise are the main parts and each of them has its own attributes and characteristics depending on the object that it is detected. In this thesis, some initial results are displayed in Section 5.3, where test data were used in the beginning and some results were provided about boundaries detection. The panoramas from an empty room are used for point cloud generation and the connection between the edges from the panoramas to the point cloud. It seems that our idea performs well when the edges are clear boundaries, Figure 5.17. A possible framework could be to apply our SPC framework to detect the doors and windows in indoor environment.

- Create points in order to provide a more complete object visualization. After detecting the points as pipes, the information about each point to which pipe is belonging is available. By this knowledge, incomplete parts of the cylinders that were not able to be captured (due to occlusion) could be generated.

- Improve the automatic procedure. Due to time limitation the procedure of our methodology is not fully automatic. For example, there are different parts of the implementation that were performed separately. For instance, the Python script that is integrated in Agisoft Photoscan and the different test data that were provided, are some of the reasons that made the automation more difficult.

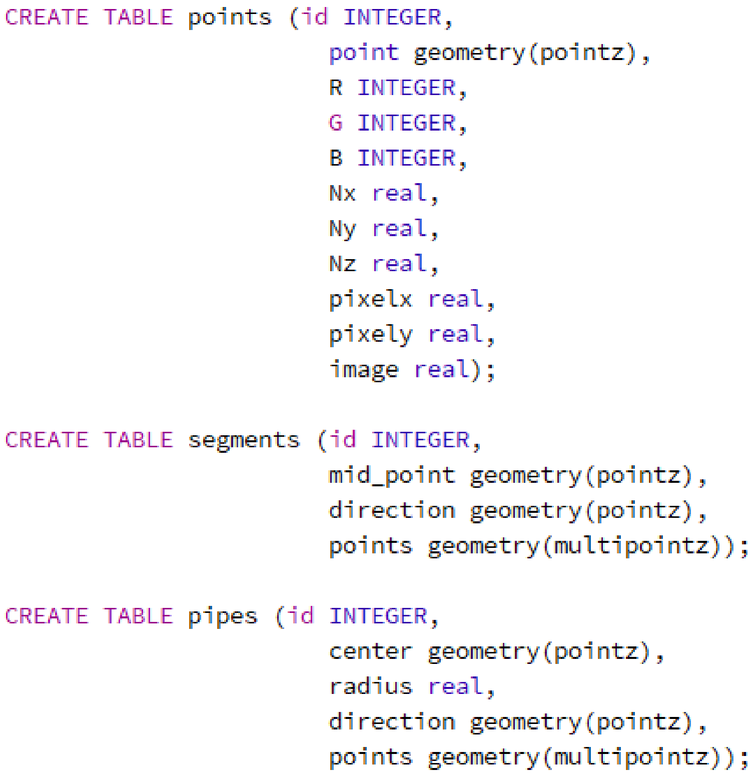
- In regard to the virtual reality representation of our point cloud. In Section 5.4.4, the whole concept, on how the point cloud could be visualized in virtual reality, is described. This is a way to visualize our results and to interact with the user. The idea is to make an interaction between the points and the user and to provide semantic information about the pipes and interaction with the industrial environment. For example, a click to a point can highlight all the points that are part of the same pipe with a unique color can be assigned. Moreover, the available information stored in the database could be displayed. Finally, possible interaction of the user with the pipes could include apart from the display of specific characteristics also, measurements from point to point.

A SQL STATEMENTS & STORAGE

This chapter contains the SQL statements that were used for the implementation of the prototype described in Chapter 4. The chapter is divided into two sections, the SQL statements concerning the creation of the tables and the loading of the sample data that were used for our methodology. The store in the database is based on the UML that is described in Chapter 4, Section 4.2.7.

A.1 CREATING TABLES

According to the UML schema three tables are stored in Database. The SQL statements are provided in Figure A.1. First, the table with the points and their attributes including also, the connection with the pixel of the image, is created. The second table is created for the segments and the final table is created for the pipes and the parameters of each.

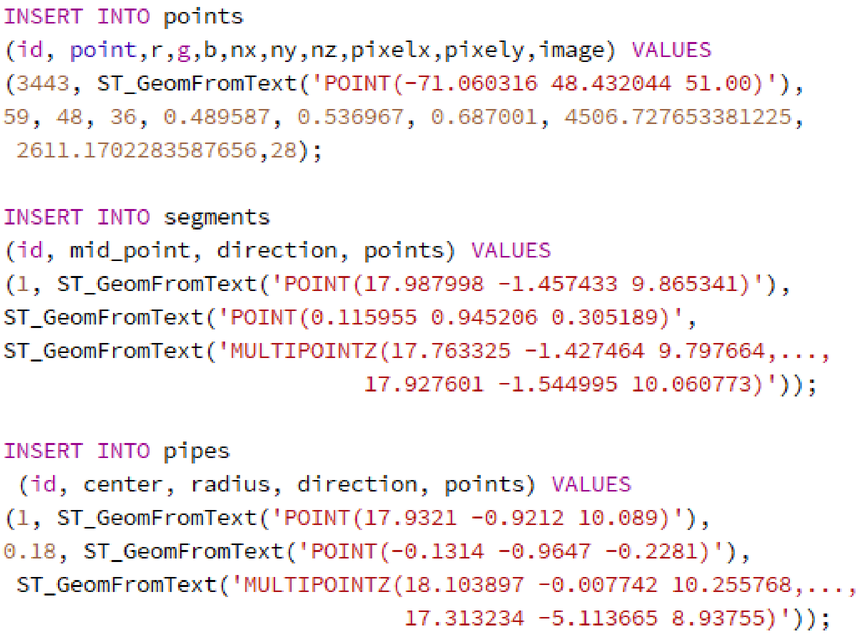


```
CREATE TABLE points (id INTEGER,
                        point geometry(pointz),
                        R INTEGER,
                        G INTEGER,
                        B INTEGER,
                        Nx real,
                        Ny real,
                        Nz real,
                        pixelx real,
                        pixely real,
                        image real);

CREATE TABLE segments (id INTEGER,
                        mid_point geometry(pointz),
                        direction geometry(pointz),
                        points geometry(multipointz));

CREATE TABLE pipes (id INTEGER,
                        center geometry(pointz),
                        radius real,
                        direction geometry(pointz),
                        points geometry(multipointz));
```

Figure A.1: SQL for the creation of the tables.

A.2 LOADING DATA

The SQL statements are described to insert the data to each table. The connection with the Database has been done in python, to insert the data in the tables is an iterative procedure. In Figure A.2, and example of the SQL statements is displayed.



```
INSERT INTO points
(id, point,r,g,b,nx,ny,nz,pixelx,pixely,image) VALUES
(3443, ST_GeomFromText('POINT(-71.060316 48.432044 51.00)'),
59, 48, 36, 0.489587, 0.536967, 0.687001, 4506.727653381225,
 2611.1702283587656,28);

INSERT INTO segments
(id, mid_point, direction, points) VALUES
(1, ST_GeomFromText('POINT(17.987998 -1.457433 9.865341)'),
ST_GeomFromText('POINT(0.115955 0.945206 0.305189)',
ST_GeomFromText('MULTIPOINTZ(17.763325 -1.427464 9.797664,...,
                           17.927601 -1.544995 10.060773)'));

INSERT INTO pipes
 (id, center, radius, direction, points) VALUES
(1, ST_GeomFromText('POINT(17.9321 -0.9212 10.089)'),
0.18, ST_GeomFromText('POINT(-0.1314 -0.9647 -0.2281)'),
 ST_GeomFromText('MULTIPOINTZ(18.103897 -0.007742 10.255768,...,
                              17.313234 -5.113665 8.93755)'));
```

Figure A.2: SQL statement loading the data.

B HOUGH LINE TRANSFORM IN PANORAMAS

In this chapter, the panoramas that are presented in Chapter 4, Section 5.4.1 are provided below, in bigger size, for better visualization of the Hough Line Transform results.

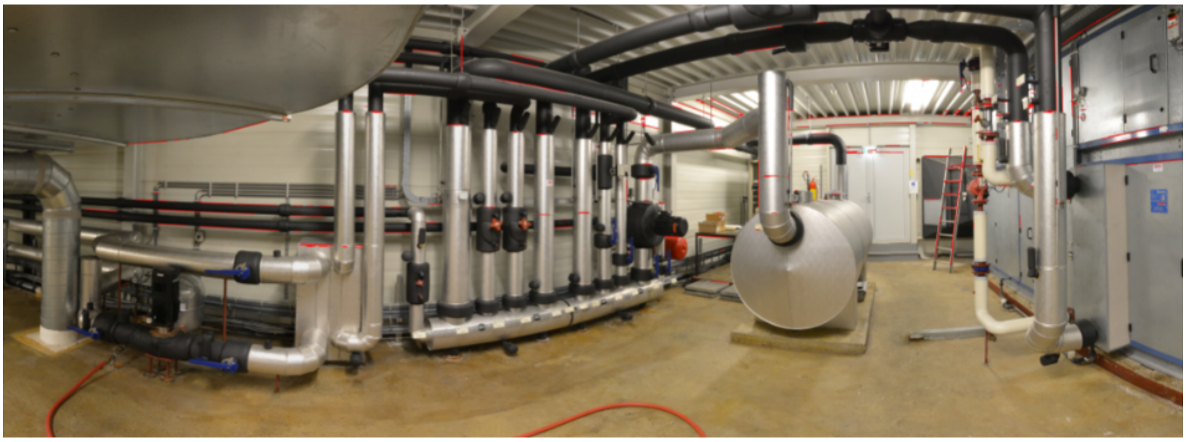


Figure B.1: Hough Line Transform results in panorama (1)

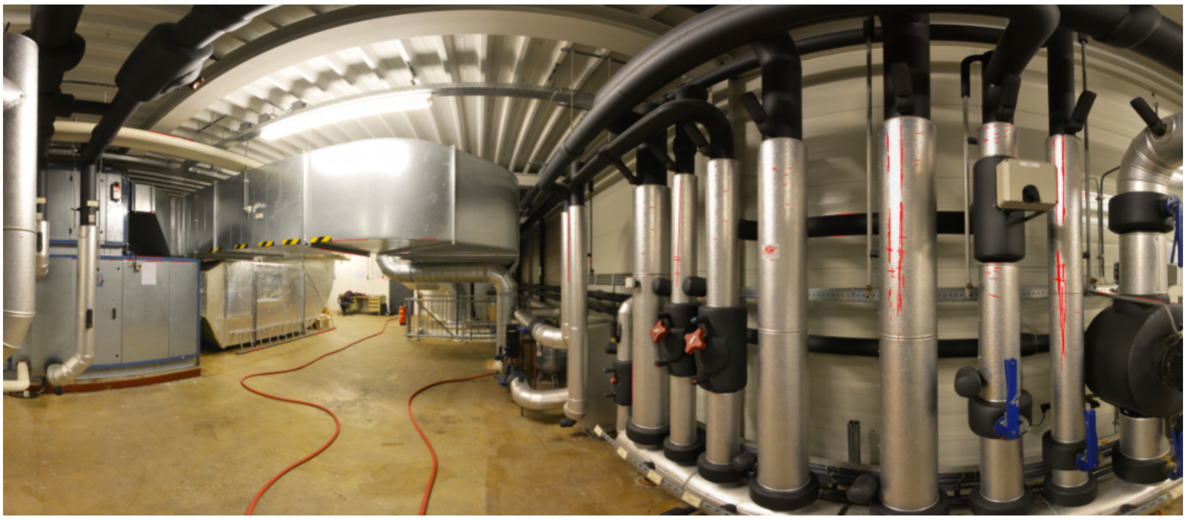


Figure B.2: Hough Line Transform results in panorama (2)

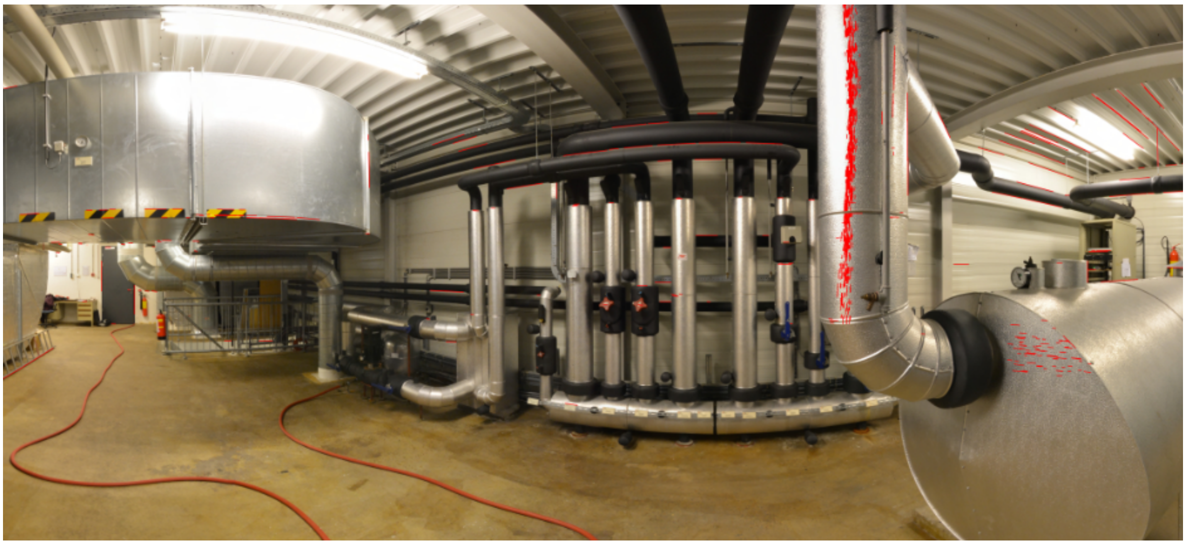


Figure B.3: Hough Line Transform results in panorama (3)

C REFLECTION

This section highlights the main contribution of our thesis and the relation with some aspects, like our master field, courses and in general, with the wider social world.

This thesis proposes a methodology for Smart Point Cloud SPC generation for detecting the pipes, as world object. The relationship between the Master Geomatics and the method applied within this thesis consists of boundary detection and segmentation techniques applied to 2D (images) and 3D (point cloud) data. Another part of the research consists the generation, manipulation and visualization of the point cloud in 3D environment. These topics are in line with the courses Sensing technologies, 3D modelling, Python programming and Geo Datasets Quality that are part of the Geomatics master. This research could not have been fulfilled without having knowledge with respect to these information, during the two years studying at this master program.

The research and proposed methodology conducted in this document, is applicable to the geomatics field. The methodology that is presented and the output can be used as input for a broad range of Geomatics-related applications. The purpose is to provide a SPC structure that is using the domain, geometric and device expertise for object detection. Identifying links and relations between the segments is essential for the connection with the surroundings to finally get a semantically rich point cloud. In our case, image processing, for boundaries extraction in 2D, is used to identify the pipes in 3D, as one world object. Providing a smart structure (2D with 3D connection) and an efficient storage way. The goal is to provide a virtual 3D environment that the user could interact with the point cloud and take the needed information. The UML schema that shows the relationships between the points and the attributes, standardize the whole procedure and make it applicable for many applications that are using the object recognition.

The work presented within this thesis is mainly useful for applications in the industrial environment but not limited to it. Point clouds have evolved to be a very important source of spatial information for many applications. The usefulness of point cloud data lies in the fact that point cloud acquisition techniques have become highly accurate and quick, allowing for better processing and optimized methods for processing techniques, in order to extract the needed information. The main object in this thesis is the recognition of cylinders (pipes), one of the geometry primitives that are applicable to industrial environment. Cylinders, as a geometrical object could be needed in other applications, for example: to detect the trunks in forests or in urban environment, to detect the traffic lights. Beyond the type of the object, the main idea of our work is to adopt a global framework for point cloud processing, involving many research fields to solve the problem of point cloud enrichment. The SPC framework using panoramas for point cloud acquisition is also applicable to other types of applications such as in 3D City modelling, 3D Indoor modelling, Cultural Heritage Documentation etc. After working on this thesis, I realized that the importance is to find solutions in order to be able to detect and extract the needed information using the most efficient techniques for manipulating and extracting the needed information from unstructured point cloud, skipping the modelling part.

BIBLIOGRAPHY

Sameer Agarwal, Yasutaka Furukawa, Noah Snavely, Ian Simon, Brian Curless, Steven M. Seitz, and Richard Szeliski. Building Rome in a day. *Communications of the ACM*, 54(10):105–112, 2011.

Ioannis Brilakis, Manolis Lourakis, Rafael Sacks, Silvio Savarese, Symeon Christodoulou, Jochen Teizer, and Atefe Makhmalbaf. Toward automated generation of parametric BIMs based on hybrid video and laser scanning data. *Advanced Engineering Informatics*, 24(4):456–465, 2010.

Gerd Bruder, Frank Steinicke, and Andreas Nüchter. Immersive point cloud virtual environments. In *Proceedings of IEEE Symposium on 3D User Interfaces 3DUI Proceedings of IEEE Symposium on 3D User Interfaces (3DUI'14)*, pages 161–162, 2014.

John Canny. A computational approach to edge detection. In *Readings in Computer Vision*, pages 184–203. Elsevier, 1987.

Jamie Carter, K. Schmid, K. Waters, L. Betzhold, B. Hadley, R. Mataosky, and J. Halleran. Lidar 101: An introduction to lidar technology, data, and applications. *National Oceanic and Atmospheric Administration (NOAA) Coastal Services Center*, pages 7–11, 2012.

Daniel Herrera Castro. From images to point clouds. 2015.

A. Cefalu, N. Haala, and D. Fritsch. Hierarchical structure from motion combining global image orientation and structureless bundle adjustment. *International Archives of the Photogrammetry, Remote Sensing & Spatial Information Sciences*, 42, 2017.

Thomas Chaperon and François Goulette. Extracting cylinders in full 3D data using a random sampling method and the Gaussian image. In *Vision Modeling and Visualization Conference 2001 (VMV-01)*, 2001.

Rémi Cura, Julien Perret, and Nicolas Paparoditis. Point Cloud Server (PCS): Point Clouds In-Base management and processing. *ISPRS Annals of Photogrammetry, Remote Sensing & Spatial Information Sciences*, 2015.

Rémi Cura, Julien Perret, and Nicolas Paparoditis. Implicit LOD using points ordering for processing and visualisation in Point Cloud Servers. *arXiv:1602.06920 [cs]*, February 2016.

Christoph Dalitz, Tilman Schramke, and Manuel Jeltsch. Iterative Hough Transform for Line Detection in 3D Point Clouds. *Image Processing On Line*, 7:184–196, 2017.

E. d'Annibale. Image Based Modeling from Spherical Photogrammetry and Structure for Motion. The Case of the Treasury, Nabatean Architecture in Petra. *Geoinformatics FCE CTU*, 6(0):62–73, December 2011. ISSN 1802-2669.

Enzo d'Annibale, Sabino Massa, and Gabriele Fangi. Photomodeling and point clouds from spherical panorama-Nabatean architecture in Petra, Jordan. In *Workshop Petra*, pages 4–8, 2010.

Enzo d'Annibale, Livia Piermattei, and Gabriele Fangi. Spherical photogrammetry as emergency photogrammetry. In *INTERNATIONAL CIPA SYMPOSIUM*, volume 23, 2011.

E. d'Annibalea and Es Malinverni. From panoramic photos to a low-cost photogrammetric workflow for cultural heritage 3d documentation. *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 5:W2, 2013.

Gabriele Fangi. The multi-image spherical panoramas as a tool for architectural survey. 2007.

Gabriele Fangi. The Multi-image spherical Panoramas as a tool for Architectural Survey. *CIPA HERITAGE DOCUMENTATION*, 21, 2011.

Gabriele Fangi. La fotogrametria sferica una nuova tecnica per il rilievo dei vicini. *Archeomatica*, 1(2), April 2012. ISSN 2037-2485.

Gabriele Fangi. Towards AN Easier Orientation for Spherical Photogrammetry. *The International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, 40(5):279, 2015.

Martin A. Fischler and Robert C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981.

Mani Golparvar-Fard, Jeffrey Bohn, Jochen Teizer, Silvio Savarese, and Feniosky Peña-Mora. Evaluation of image-based modeling and laser scanning accuracy for emerging automated performance monitoring techniques. 20:1143–1155, 05 2011.

Michael Hemmer. Algebraic foundations. In *CGAL User and Reference Manual*. CGAL Editorial Board, 4.12 edition, 2018.

Paul VC Hough. Method and means for recognizing complex patterns. Technical report, 1962.

Fay Huang, Reinhard Klette, and Karsten Scheibe. *Panoramic imaging: sensor-line cameras and laser range-finders*, volume 11. John Wiley & Sons, 2008.

Juho Kannala and Sami S. Brandt. A generic camera model and calibration method for conventional, wide-angle, and fish-eye lenses. *IEEE transactions on pattern analysis and machine intelligence*, 28(8):1335–1340, 2006.

Thomas Krijnen and Jakob Beetz. An IFC schema extension and binary serialization format to efficiently integrate point cloud data into building models. *Advanced Engineering Informatics*, 2017.

K. Kwiatek and R. Tokarczyk. Photogrammetric applications of immersive video cameras. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 2(5):211, 2014.

Karol Kwiatek and Regina Tokarczyk. Immersive photogrammetry in 3D modelling. *Geomatics and Environmental Engineering*, 9(2), 2015.

Jean-Francis Lalonde, Nicolas Vandapel, and Martial Hebert. Automatic three-dimensional point cloud processing for forest inventory. *Robotics Institute*, page 334, 2006.

Z. Lari and A. Habib. Alternative methodologies for the estimation of local point density index: Moving towards adaptive LiDAR data processing. *Int. Arch. Photogramm. Remote Sens. Spat. Inform. Sci*, 39:127–132, 2012.

I.-C. Lee and F. Tsai. Applications of panoramic images: from 720 panorama to interior 3d models of augmented reality. *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 4:189–192, may 2015. doi: 10.5194/isprsarchives-XL-4-W5-189-2015.

Kun Liu and Jan Boehm. A new framework for interactive segmentation of point clouds. *The International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, 40(5):357, 2014.

Thomas Luhmann. A historical review on panorama photogrammetry. *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 34 (5/W16):8, 2004.

Juan Mancera-Taboada, Pablo Rodríguez-Gonzálvez, Diego González-Aguilera, Javier Finat, Jesús San José, Juan J. Fernández, José Martínez, and Rubén Martínez. From the point cloud to virtual and augmented reality: digital accessibility for disabled people in San Martin's Church (Segovia) and its surroundings. In *International Conference on Computational Science and Its Applications*, pages 303–317. Springer, 2011.

Shawn McCann. 3D reconstruction from multiple images. 2015.

Carmo MPd. Differential geometry of curves and surfaces. *New Jersy: Prentice-Hall*, 1976.

Ivan Nikolov and Claus Madsen. *Benchmarking Close-range Structure from Motion 3D Reconstruction Software Under Varying Capturing Conditions*, pages 15–26. Springer International Publishing, Cham, 2016.

Abdul Nurunnabi, Yukio Sadahiro, and Roderik Lindenbergh. Robust cylinder fitting in three-dimensional point cloud data. *The International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, 42:63, 2017.

Sven Oesau, Yannick Verdie, Clément Jamin, Pierre Alliez, Florent Lafarge, and Simon Giraudot. Point set shape detection. In *CGAL User and Reference Manual*. CGAL Editorial Board, 4.12 edition, 2018.

Chan-Sik Park, Do-Yeop Lee, Oh-Seong Kwon, and Xiangyu Wang. A framework for proactive construction defect management using BIM, augmented reality and ontology-based data collection template. *Automation in Construction*, 33:61–71, 2013.

T. Pavlidis and Y. T. Liow. Integrating region growing and edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(3), March 1990. ISSN 0162-8828. doi: 10.1109/34.49050.

S. Peleg, M. Ben-Ezra, and Y. Pritch. Omnistereo: panoramic stereo imaging. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(3):279–290, March 2001. ISSN 0162-8828. doi: 10.1109/34.910880.

Cecilia Pisa, Fabiana Zeppa, and Gabriele Fangi. Spherical photogrammetry for cultural heritage: san galgano abbey, siena, italy and roman theatre, sabratha, libya. In *Proceedings of the second workshop on eHeritage and digital art preservation*, pages 3–6. ACM, 2010.

Florent Poux, Romain Neuville, Pierre Hallot, and Roland Billen. Smart Point Cloud: Definition and Remaining Challenges. 4(W1):119–127, 2016.

Florent Poux, Romain Neuville, Pierre Hallot, and Roland Billen. Model for reasoning from semantically rich point cloud data. *ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences*, 4:107–115, 2017a.

Florent Poux, Romain Neuville, Pierre Hallot, and Roland Billen. Point cloud classification of tesserae from terrestrial laser data combined with dense image matching for archaeological information extraction. *ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences*, 4:203–211, 2017b.

Florent Poux, Romain Neuville, Pierre Hallot, Line Van Wersch, Andrea Luczfalvy Jancsó, and Roland Billen. Digital investigations of an archaeological smart point cloud: A real time web-based platform to manage the visualisation of semantical queries. pages 581–588, 2017c.

Tahir Rabbani. Automatic reconstruction of industrial installations using point clouds and images. May 2006.

Tahir Rabbani and Frank Van Den Heuvel. Efficient hough transform for automatic detection of cylinders in point clouds. *Isprs Wg Iii/3, Iii/4*, 3:60–65, 2005.

Tahir Rabbani, Frank Van Den Heuvel, and George Vosselmann. Segmentation of point clouds using smoothness constraint. *International archives of photogrammetry, remote sensing and spatial information sciences*, 36(5):248–253, 2006.

Fabio Remondino. Point cloud acquisition & structuring. page 12, 2018.

Rico Richter and Jürgen Döllner. Out-of-core real-time visualization of massive 3D point clouds. In *Proceedings of the 7th International Conference on Computer Graphics, Virtual Reality, Visualisation and Interaction in Africa*, pages 121–128. ACM, 2010.

Radu Bogdan Rusu, Zoltan Csaba Marton, Nico Blodow, Mihai Dolha, and Michael Beetz. Towards 3D point cloud based object maps for household environments. *Robotics and Autonomous Systems*, 56(11):927–941, 2008.

Toni Schenk. Introduction to photogrammetry. *The Ohio State University, Columbus*, 2005.

Ruwen Schnabel, Roland Wahl, and Reinhard Klein. Efficient RANSAC for point-cloud shape detection. In *Computer graphics forum*, volume 26, pages 214–226. Wiley Online Library, 2007.

G T Shrivakshan. A Comparison of various Edge Detection Techniques used in Image Processing. 9(5):8, 2012.

Roman Shults, Petro Krelshtein, Iulia Kravchenko, Olga Rogoza, and Oleksandr Kyselov. Low-cost Photogrammetry for Culture Heritage. 2017.

Noah Snavely, Steven M. Seitz, and Richard Szeliski. Photo tourism: exploring photo collections in 3D. In *ACM transactions on graphics (TOG)*, volume 25, pages 835–846. ACM, 2006.

Soheil Sotoodeh. Outlier detection in laser scanner point clouds. *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, 36(5): 297–302, 2006.

Jonathan Dyssel Stets, Yongbin Sun, Wiley Corning, and Scott W. Greenwald. Visualization and labeling of point clouds in virtual reality. In *SIGGRAPH Asia 2017 Posters*, page 31. ACM, 2017.

Yun-Ting Su, James Bethel, and Shuowen Hu. Octree-based segmentation for terrestrial LiDAR point cloud data in industrial applications. *ISPRS Journal of Photogrammetry and Remote Sensing*, 113:59–74, 2016.

Richard Szeliski and Heung-Yeung Shum. Creating full view panoramic image mosaics and environment maps. In *Proceedings of the 24th annual conference on Computer graphics and interactive techniques*, pages 251–258. ACM Press/Addison-Wesley Publishing Co., 1997.

Peter van Oosterom, Oscar Martinez-Rubi, Milena Ivanova, Mike Horhammer, Daniel Geringer, Siva Ravada, Theo Tijssen, Martin Kodde, and Romulo Gonçalves. Massive point cloud data management: Design, implementation and execution of a point cloud benchmark. *Computers & Graphics*, 49, 2015.

Peter van Oosterom, Oscar Martinez-Rubi, Theo Tijssen, and Romulo Gonçalves. Realistic Benchmarks for Point Cloud Data Management Systems. In Alias Abdul-Rahman, editor, *Advances in 3D Geoinformation*. Springer International Publishing, Cham, 2017. ISBN 978-3-319-25689-4 978-3-319-25691-7. URL http://link.springer.com/10.1007/978-3-319-25691-7_1.

A. Vetrivel, M. Gerke, N. Kerle, and G. Vosselman. Segmentation of UAV-based images incorporating 3D point cloud information. *The International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, 40(3):261, 2015.

Ying Wang, Daniel Ewert, Daniel Schilberg, and Sabina Jeschke. Edge Extraction by Merging the 3d Point Cloud and 2D Image Data. In *Automation, Communication and Cybernetics in Science and Engineering 2013/2014*, pages 773–785. Springer, 2014.

Andrew K. Watson. *Automated creation of labeled pointcloud datasets in support of machine learning-based perception*. PhD Thesis, Monterey, California: Naval Postgraduate School, 2017.

Weinmann. Reconstruction and analysis of 3D scenes, 2016.

Martin Weinmann, Boris Jutzi, Stefan Hinz, and Clément Mallet. Semantic point cloud interpretation based on optimal neighborhoods, relevant features and efficient classifiers. *ISPRS Journal of Photogrammetry and Remote Sensing*, 105:286–304, 2015.

F. J. Westdijk. Towards risk based pipeline integrity management through integrated use of heterogeneous SDI data sources, April 2015.

K. Wolff, C. Kim, H. Zimmer, C. Schroers, M. Botsch, O. Sorkine-Hornung, and A. Sorkine-Hornung. Point Cloud Noise and Outlier Removal for Image-Based 3D Reconstruction. In *2016 Fourth International Conference on 3D Vision (3DV)*, pages 118–127, October 2016. doi: 10.1109/3DV.2016.20.

Changchang Wu. Towards linear-time incremental structure from motion. In *3DTV-Conference 2013 International Conference on*, pages 127–134. IEEE, 2013.

Xuehan Xiong, Antonio Adan, Burcu Akinci, and Daniel Huber. Automatic creation of semantically rich 3D building models from laser scanner data. *Automation in Construction*, 31:325–337, 2013.

Jiansi Yang and Xianfeng Huang. A hybrid spatial index for massive point cloud data management and visualization. *Transactions in GIS*, 18(S1):97–108, 2014.

Sisi Zlatanova. 3d geometries in spatial DBMS. In *Innovations in 3D geo information systems*, pages 1–14. Springer, 2006.

COLOPHON

This document was typeset using LaTeX. The document layout was generated using the `arsclassica` package by Lorenzo Pantieri, which is an adaption of the original `classicthesis` package from André Miede.