

EPITA–École Pour l’Informatique et les Techniques Avancées  
CSI–Calcul Scientifique et Image  
LRDE–Laboratoire de Recherche et Développement de l’EPITA

**Various contributions to *ContextCapture<sup>TM</sup>* :  
Scan Finder, Point Cloud Visibility and Point Cloud  
Compression**

**Alexandre Gbaguidi Aïsse**

Supervised by Cyril Novel

Submitted in part fulfilment of the requirements for the degree of  
Software Engineer of EPITA, Paris, August 2018.



## Abstract

Text of the Abstract.



## Acknowledgements

I would like to express (whatever feelings I have) to:

- My supervisor
- My second supervisor
- Other researchers
- My family and friends



## Dedication

Dedication here.





# Contents

<b>Abstract</b>	<b>i</b>
<b>Acknowledgements</b>	<b>iii</b>
<b>1 Introduction</b>	<b>1</b>
<b>2 The company</b>	<b>4</b>
2.1 Bentley Systems . . . . .	4
2.2 Acute 3D . . . . .	4
2.3 ContextCapture . . . . .	4
<b>3 Basic Concepts</b>	<b>5</b>
3.1 FIXME . . . . .	5
<b>4 Scan Finder</b>	<b>6</b>
4.1 Specifications . . . . .	7
4.1.1 Problem being addressed . . . . .	7
4.1.2 Objective . . . . .	7
4.1.3 Scope . . . . .	7
4.2 Related Work . . . . .	7
4.3 The faulty grid-pattern method . . . . .	7
4.3.1 Overview . . . . .	7
4.3.2 Grid-pattern matching . . . . .	7
4.3.3 Equation to solve . . . . .	7
4.3.4 Results and discussions . . . . .	7
4.4 The working elliptic method . . . . .	7

4.4.1	Overview . . . . .	7
4.4.2	Clustering high-density area . . . . .	7
4.4.3	Fitting ellipse . . . . .	7
4.4.4	Equation to solve . . . . .	7
4.4.5	Results and discussions . . . . .	7
<b>5</b>	<b>Point Cloud Visibility</b>	<b>8</b>
5.1	Specifications . . . . .	9
5.1.1	Problem being addressed . . . . .	9
5.1.2	Objective . . . . .	9
5.1.3	Scope . . . . .	9
5.2	Related work . . . . .	9
5.3	Direct Visibility of Point Sets . . . . .	9
5.3.1	Overview . . . . .	9
5.3.2	Implementation . . . . .	9
5.3.3	Results and discussions . . . . .	9
5.4	Visibility of Noisy Point Cloud Data . . . . .	9
5.4.1	Overview . . . . .	9
5.4.2	Implementation . . . . .	9
5.4.3	Results and discussions . . . . .	9
5.5	A custom disk-based approach . . . . .	9
5.5.1	Overview . . . . .	9
5.5.2	Implementation . . . . .	9
5.5.3	Results and discussions . . . . .	9
<b>6</b>	<b>Point Cloud Compression</b>	<b>10</b>
6.1	Specifications . . . . .	10
6.1.1	Problem being addressed . . . . .	10
6.1.2	Objective . . . . .	10
6.1.3	Scope . . . . .	10
6.2	Related work . . . . .	10
6.3	A custom arithmetic approach . . . . .	10

6.3.1	Overview . . . . .	10
6.3.2	Implementation . . . . .	10
6.3.3	Comparison with Brotli, 7Z and Zip . . . . .	10
6.4	Integration . . . . .	10
<b>7</b>	<b>Conclusion</b>	<b>11</b>
7.1	Summary of Internship Achievements . . . . .	11
7.2	Applications . . . . .	11
7.3	Future Work . . . . .	11
	<b>Bibliography</b>	<b>11</b>



# List of Tables



# List of Figures





# Chapter 1

## Introduction

The need of 3D realistic models is increasingly present in several fields such as architecture, digital simulation or civil and structural engineering. One way to obtain a 3D model might be to build it by hand using specialized modeling softwares. In this case, the realistic aspect of the model could be doubtful. A more reliable way is to use *photogrammetry* or *surface reconstruction* or both at the same time. *ContextCapture<sup>TM</sup>* is a reality modeling software that can produce highly detailed 3D models. It creates models of all types or scales from simple photographs of a scene to point clouds or both of them thanks to its hybrid processing. The usage of point clouds gained wide popularity because of the emergence of devices such as optical laser-based range scanners, structured light scanners, LiDAR scanners, Microsoft Kinect, etc. It is only in the past two years that *ContextCapture<sup>TM</sup>* has started to support point clouds.

A scanner, whether static or mobile, outputs a point cloud. Given a point cloud  $P$  assuming to lie near an unknown shape  $S$ , the general problem being solved is to construct a digital representation  $D$  approximating  $S$ . A critical step during the construction of  $D$  is usually the estimation and orientation of the normal of each point  $p \in P$ . In order to orient normals and reconstruct point clouds acquired from static LiDAR scanners, possibly multi-scan<sup>1</sup>, *ContextCapture<sup>TM</sup>* needs to know the position of each scanner in the scene, on the one hand, and the attribution of each point to a scanner on the other. The scanners locations information are not always present in the metadata of some file formats, for instance LAS file format does not even provide it. Detecting the positions of multiple scanners exclusively from a point cloud is a subject neither tackled by academics nor the industry. The scanner position information is lost only due to an export with no metadata. Thus, academic research has not

---

<sup>1</sup>Point cloud made of several laser scans

identified this subject as interesting. It's only recently that scanner position information is relevant, for a few applications like *ContextCapture<sup>TM</sup>*. The industry did not need an algorithm to retrieve the scanner position up to now and this is the main contribution presented in this internship report: a method able to detect automatically multiple scanners in a single point cloud without prior knowledge of the scene.

Knowing scanners position is one thing, orient normals is another. To do so, *ContextCapture<sup>TM</sup>* needs to know for each point which scanner best sees it, there is no need to know exactly which scanner generated it. This enters into the realm of visibility of point clouds. One way to retrieve visibility of point clouds is, firstly, to reconstruct the surface and then use the underlying mesh to compute visibility. But to reconstruct the surface we need to orient the normals, there is a bit of a chicken-and-egg problem. In *Direct visibility of point sets* [KTB07], a simple algorithm that computes visibility without reconstructing the surface is introduced: the Hidden Point Removal (HPR) operator. But this operator is not robust to noise which happens quite often in LiDAR point clouds. Some improvements have been proposed in [MTSM10] but are not good enough to handle LiDAR point clouds noise. Actually, both papers show results obtained on point clouds describing small closed surfaces such as the statue of David [dav] or the Stanford bunny [bun]. Moreover, our purpose here is not to find which points are visible from a precise viewpoint. Different scanners can see the same points and we want to know which scanner best sees it in order to have an accurate normal orientation. We introduce in this internship report a custom point cloud visibility method that serves our purpose and works well with LiDAR point clouds, regardless of the sampling density.

Finally, the last subject covered during this internship is point cloud compression. Reconstructing a surface requires effective machines. The more the scene to reconstruct is bigger, the more the reconstruction can last longer. *ContextCapture<sup>TM</sup>* provides a cloud service which gives the opportunity for people not having any clusters or high-performance machine to do the job. The problem is that, point clouds can be very huge, up to one hundred (100) gibabyte and more. And if something happens while uploading, the upload restarts from scratch. Point cloud compression can be addressed in two ways: geometric compression [GKIS05, SK06] or pure arithmetic compression regardless of the kind of file being compressed. We present... FIXME ...

This report is organised as follows. Chapter 2 present Bentley Systems, Acute3D, *ContextCapture<sup>TM</sup>* and how the achieved work is positioned in the company's business line. After introducing in Chapter 3 some useful definitions for a better understanding of the report, we describe the achieved work of Scan

Finder, Point Cloud Visibility and Point Cloud Compression respectively in Chapter 4, Chapter 5 and Chapter 6. Note that in each chapter, we recall the context, the issue addressed and the expected result before going into details. Finally Chapter 7 summarizes all the work, evaluates my contributions to *ContextCapture<sup>TM</sup>* and assess what this experience has brought to me.

## Chapter 2

# The company

### 2.1 Bentley Systems

### 2.2 Acute 3D

### 2.3 ContextCapture

## Chapter 3

# Basic Concepts

### 3.1 FIXME

Text of the Background.



## Chapter 4

# Scan Finder

### 4.1 Specifications

#### 4.1.1 Problem being addressed

#### 4.1.2 Objective

#### 4.1.3 Scope

### 4.2 Related Work

### 4.3 The faulty grid-pattern method

#### 4.3.1 Overview

#### 4.3.2 Grid-pattern matching

#### 4.3.3 Equation to solve

#### 4.3.4 Results and discussions

### 4.4 The working elliptic method

#### 4.4.1 Overview

#### 4.4.2 Clustering high-density area





## Chapter 5

# Point Cloud Visibility

### 5.1 Specifications

#### 5.1.1 Problem being addressed

#### 5.1.2 Objective

#### 5.1.3 Scope

### 5.2 Related work

### 5.3 Direct Visibility of Point Sets

#### 5.3.1 Overview

#### 5.3.2 Implementation

#### 5.3.3 Results and discussions

### 5.4 Visibility of Noisy Point Cloud Data

#### 5.4.1 Overview

#### 5.4.2 Implementation

#### 5.4.3 Results and discussions

## Chapter 6

# Point Cloud Compression

### 6.1 Specifications

#### 6.1.1 Problem being addressed

#### 6.1.2 Objective

#### 6.1.3 Scope

### 6.2 Related work

### 6.3 A custom arithmetic approach

#### 6.3.1 Overview

#### 6.3.2 Implementation

#### 6.3.3 Comparison with Brotli, 7Z and Zip

### 6.4 Integration

## Chapter 7

# Conclusion

### 7.1 Summary of Internship Achievements

Summary.

### 7.2 Applications

Applications.

### 7.3 Future Work

Future Work.

# Bibliography

- [bun] Stanford bunny. [https://en.wikipedia.org/wiki/Stanford\\_bunny](https://en.wikipedia.org/wiki/Stanford_bunny). Accessed: 2018-07-05.
- [dav] David (michelangelo). [https://en.wikipedia.org/wiki/David\\_\(Michelangelo\)](https://en.wikipedia.org/wiki/David_(Michelangelo)). Accessed: 2018-07-05.
- [GKIS05] Stefan Gumhold, Zachi Kami, Martin Isenburg, and Hans-Peter Seidel. Predictive point-cloud compression. In *ACM SIGGRAPH 2005 Sketches*, page 137. ACM, 2005.
- [KTB07] Sagi Katz, Ayellet Tal, and Ronen Basri. Direct visibility of point sets. In *ACM Transactions on Graphics (TOG)*, volume 26, page 24. ACM, 2007.
- [MTSM10] Ravish Mehra, Pushkar Tripathi, Alla Sheffer, and Niloy J Mitra. Visibility of noisy point cloud data. *Computers & Graphics*, 34(3):219–230, 2010.
- [SK06] Ruwen Schnabel and Reinhard Klein. Octree-based point-cloud compression. *Spbg*, 6:111–120, 2006.