



**HAL**  
open science

# Vers de nouvelles perspectives lasergrammétriques : optimisation et automatisation de la chaîne de production de modèles 3D

Florent Poux

► **To cite this version:**

Florent Poux. Vers de nouvelles perspectives lasergrammétriques : optimisation et automatisation de la chaîne de production de modèles 3D. Sciences de l'ingénieur [physics]. 2013. dumas-00941990

**HAL Id: dumas-00941990**

**<https://dumas.ccsd.cnrs.fr/dumas-00941990>**

Submitted on 4 Feb 2014

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

**CONSERVATOIRE NATIONAL DES ARTS ET METIERS  
ÉCOLE SUPÉRIEURE DES GÉOMÈTRES ET TOPOGRAPHES**

---

**MÉMOIRE**

présenté en vue d'obtenir

le **DIPLÔME D'INGÉNIEUR CNAM**

**Spécialité : Géomètre et Topographe**

par

**Florent POUX**

---

Vers de nouvelles perspectives lasergrammétriques : Optimisation et  
automatisation de la chaîne de production de modèles 3D

**Soutenu le 10 Juillet 2013**

---

**JURY**

**PRÉSIDENT :** M. Christophe PROUDHOM

**MEMBRES :** M. Jean-Michel FOLLIN Professeur référent  
M. Patrick BEZARD-FALGAS  
M. Anthony GUEROUT  
M. Pascal RAYMOND  
M. Ghyslain FERRE

**Maître de stage :** M. Roland BILLEN

## REMERCIEMENTS

*En préambule de ce mémoire qui est l'aboutissement d'une formation complète et internationale d'ingénieur, je voudrais exprimer toute ma gratitude vis-à-vis des personnes qui ont contribué à son élaboration.*

*Mes chaleureux et sincères remerciements vont tout d'abord à mon maître de stage **Roland BILLEN** pour son attention, son encadrement, sa disponibilité, ses conseils avisés mais aussi pour le projet qu'il m'a permis de réaliser au sein de son équipe et le goût pour la recherche qu'il m'a communiqué.*

*Je remercie en amont mon professeur référent Jean-Michel FOLLIN pour le temps qu'il m'a accordé, son accompagnement tout au long de ces derniers mois, sa rapidité son efficacité et son aide précieuse.*

*Ensuite, j'exprime ma gratitude à l'ensemble de l'Unité Géomatique de l'Université de Liège : leur accueil chaleureux et leur aide précieuse ont participé à l'élaboration de ce travail.*

*J'adresse en particulier un grand merci aux assistants **Pierre HALLOT** et **Benoît JONLET** pour leurs remarques pertinentes et constructives ainsi que pour leur bonne humeur m'ayant permis d'évoluer dans une atmosphère agréable.*

*Je suis également reconnaissant des conseils et de l'accompagnement de **Cyril Carré**, ancien ingénieur ESGT de l'équipe.*

*Je tiens également à remercier ceux sans qui je n'aurai pas été là aujourd'hui, c'est-à-dire mes frères, mes sœurs, mes parents, mes grands-parents, ... : ma famille. Un immense merci.*

*Je veux adresser une pensée particulière pour mes amis et ma Freundin dont la présence et les encouragements furent une réelle motivation.*

# SOMMAIRE

<b>REMERCIEMENTS</b>	<b>2</b>
<b>SOMMAIRE</b>	<b>3</b>
<b>INTRODUCTION</b>	<b>5</b>
<b>1 CONTEXTE DE L'ETUDE</b>	<b>7</b>
1.1 L'Unité Géomatique de l'ULG	7
1.1.1 Présentation	7
1.1.2 L'environnement de travail	7
1.2 La cathédrale Saint-Paul de Liège	8
1.2.1 Son histoire	8
1.2.2 Structure et organisation	8
1.2.3 Un intérêt archéologique et patrimonial	9
<b>2 ÉTAT DE L'ART</b>	<b>10</b>
2.1 Acquisition des données par scanner laser	10
2.1.1 Qu'est-ce qu'un scanner laser terrestre ?	10
2.1.2 Principes physiques et théoriques	10
2.2 Traitement des données	12
2.2.1 La consolidation	12
2.2.2 Le nettoyage du nuage	13
2.2.3 La segmentation	14
2.2.4 La modélisation	16
2.2.5 La texturisation	18
2.3 Diffusion, publication et applications	19
2.3.1 Différentes applications pour différentes utilisations	19
2.3.2 Différents supports/formats de diffusion	20
<b>3 PRETRAITEMENT &amp; CONSOLIDATION</b>	<b>22</b>
3.1 L'obtention des données de la cathédrale	22
3.1.1 Organisation structurelle de la cathédrale	22
3.1.2 Le levé terrain de la cathédrale : méthodologie et solutions	23
3.2 Les suites logicielles de traitement de nuages de points	25
3.3 Le recalage de la cathédrale	26
<b>4 SEGMENTATION ALGORITHMIQUE</b>	<b>28</b>
4.1 Évolution et cheminement conceptuel	28
4.2 Principe théorique	29
4.2.1 Définition et focalisation	29
4.2.2 Le calcul des normales et l'estimation de courbure	30
4.2.3 La segmentation	31
4.3 Insertion dans l'environnement logiciel	32
4.3.1 Les besoins de l'algorithme et les solutions logicielles	32
4.3.2 La création de l'algorithme et explications	32
4.4 Les résultats	36
4.4.1 La colonne	36
4.4.2 Le château de Colonster	37

4.4.3	3 Colonnes de la cathédrale Saint-Paul et chapiteaux	37
4.4.4	Bilan & validation	38
<b>5</b>	<b>MODELISATION, TEXTURISATION ET NIVEAUX DE DETAILS</b>	<b>40</b>
<b>5.1</b>	<b>Utilisation des différentes modélisations</b>	<b>40</b>
<b>5.2</b>	<b>Modélisation de l'objet dans sa totalité</b>	<b>40</b>
5.2.1	Création du maillage brut : MeshLab	40
5.2.2	Simplifications et corrections	42
5.2.3	Texturisation	42
<b>5.3</b>	<b>Modélisation hybride post-segmentation</b>	<b>43</b>
<b>5.4</b>	<b>Étude comparative</b>	<b>43</b>
<b>6</b>	<b>VERS DIFFERENTS SUPPORTS POUR DIFFERENTES UTILISATIONS</b>	<b>45</b>
<b>6.1</b>	<b>Une application archéologique de précision</b>	<b>45</b>
6.1.1	Le format U3D et la création PDF 3D	45
6.1.2	Des possibilités d'exploitation étendues	46
<b>6.2</b>	<b>Visite virtuelle WebGL</b>	<b>46</b>
6.2.1	Le contexte	46
6.2.2	La création de l'environnement WebGL	47
6.2.3	La visite sur serveur	48
	<b>CONCLUSION</b>	<b>49</b>
	<b>REFERENCES BIBLIOGRAPHIQUES</b>	<b>50</b>
	<b>SITES INTERNET</b>	<b>51</b>
	<b>ANNEXES</b>	<b>52</b>
<b>Annexe 1</b>	<b>Caractéristiques du matériel FARO</b>	<b>53</b>
<b>Annexe 2</b>	<b>Comparaison des suites logicielles de traitement de nuages de points</b>	<b>54</b>
<b>Annexe 3</b>	<b>Tutoriel CloudCompare</b>	<b>58</b>
<b>Annexe 4</b>	<b>Rapport Technique de Levé et recalage de la Cathédrale Saint-Paul de Liège</b>	<b>60</b>
<b>Annexe 5</b>	<b>Algorithme MATLAB de recalage</b>	<b>72</b>
<b>Annexe 6</b>	<b>Algorithme de segmentation créé</b>	<b>74</b>
<b>Annexe 7</b>	<b>Méthodes de maillage et paramétrisation</b>	<b>77</b>
<b>Annexe 8</b>	<b>Principe du format .obj texturé</b>	<b>80</b>
	<b>LISTE DES FIGURES</b>	<b>82</b>
	<b>RÉSUMÉ</b>	<b>84</b>



## INTRODUCTION

Depuis maintenant plusieurs années, l'utilisation de scanners lasers 3D est de plus en plus répandue. Des professionnels de tous secteurs (géomatique, architecture, contrôle qualité, police criminelle, ...) qui travaillaient habituellement avec des plans et des représentations schématiques bidimensionnelles ont aujourd'hui la possibilité de donner une nouvelle dimension à leurs outils. L'apport volumique de la 3D offre une analyse approfondie, rendant compte de la spatialité qui permet à l'utilisateur de développer une vision fidèle de l'espace environnant.

En outre, la technique lasergrammétrique possède des atouts indiscutables pour le relevé d'éléments architecturaux et de larges scènes. En quelques minutes, il est possible d'obtenir des millions de points 3D avec une densité élevée et une précision de l'ordre de quelques millimètres, quand auparavant cela demandait des semaines par tachéométrie. Cependant, ces avantages certains peuvent être ternis par une phase de post-traitement exigeante, due au traitement d'énormes quantités de données brutes non structurées.

Il apparaît clairement la nécessité de trouver des solutions pour optimiser cette phase afin de la rendre plus simple, plus rapide et automatique, pour arriver un jour à une interaction utilisateur minimale.

C'est au sein de l'Unité de Géomatique de l'Université de Liège, en Belgique, que ce travail d'optimisation et d'automatisation de la chaîne de traitement des données 3D a été mené. Il s'inscrit dans un projet attaché à la modélisation de la Cathédrale Saint-Paul de Liège pour un usage d'archéologie du bâti médiéval. Ce projet vise à retranscrire - grâce à un nuage de plus de 2 milliards de points décrivant intégralement l'intérieur et partiellement l'extérieur de cette œuvre religieuse - la réalité archéologique d'un monument aux détails très fins : ornements, sculptures, vitraux, gabarit des pierres, peintures murales, mortiers, ...

Cependant, la transmission et l'intégration de ces données brutes dans des bases de données ou leur diffusion sont très lourdes compte tenu de la grande quantité de points. L'exploitation future de ce nuage exige alors un traitement précis et pertinent pour créer, structurer, simplifier, analyser et modéliser les données acquises. Les exigences de retranscription formelle de la réalité conditionnent grandement la qualité des traitements.

La pertinence et la précision de différentes reconstructions ainsi que le niveau d'analyse de la scène dépendent d'une étape majeure : la segmentation. Son but est d'extraire d'un large volume de données 3D différents groupes qui partagent un lien logique, afin de les considérer judicieusement comme une même entité. La segmentation manuelle d'un nuage de point est extrêmement chronophage et imprécise pour de grands volumes. C'est pourquoi il est essentiel de développer une méthode automatisée et efficace pour élever les perspectives de production à un autre niveau.

La modélisation post-segmentation permet le passage d'un ensemble de points groupés à un modèle surfacique composé d'un nombre réduit d'entités géométriques, liées entre elles par des relations topologiques. Plus léger et manipulable, un modèle 3D de haute qualité est un élément de choix pour être intégré à des systèmes d'information ou comme support d'archivage pour des données archéologiques.

Ainsi, la segmentation et la modélisation font partie du processus de production aboutissant à la création de différents modèles pour différents usages. Le passage à la 3D permet de nouvelles avancées et ouvre la discipline sur plusieurs applications possibles. La production d'un modèle unique ne suffira pas à combler les différentes attentes.

Alors qu'un archéologue utilisera ces données pour archivages, pour de la mesure d'élévations, de volumes, de couleurs; la création d'une visite virtuelle pour valoriser le patrimoine bâti impliquera une gestion complexe de différents niveaux de détails pour une utilisation fluide sur tous supports.

Pour répondre à l'objectif fixé par le sujet du mémoire, c'est-à-dire optimiser et automatiser la chaîne de production de modèles 3D lasergrammétriques, deux questions se posent : quelle méthodologie de traitement des données employer pour obtenir un support archéologique 3D simple, précis, fidèle à la réalité et aisément diffusable ? Comment automatiser un tel processus pour des perspectives de production améliorées ?



Ces problématiques internes au projet de l'Unité permettent d'affiner les objectifs du TFE, qui sont résumés dans les 5 points suivants :

- Définir les opérations de prétraitements et de consolidation indispensables à l'obtention de données cohérentes au bruit minime;
- Développer une approche de segmentation semi-automatique et versatile adaptée à des ouvrages d'arts aux détails fins;
- Tester différentes méthodes de modélisation pour différents éléments et développer une technique d'assemblage des modèles 3D créés;
- Proposer une chaîne de production optimale de conception de modèles 3D orientée vers des usages archéologiques et facilement reproductibles dans d'autres contextes d'utilisation;
- Créer une visite virtuelle de la cathédrale Saint-Paul par une approche « nuages de points » pour une diffusion sur internet.

Ces différents objectifs impliquent la convergence de connaissances de domaines variés : lasergrammétrie, topographie, infographie, programmation et développement. Cette synergie décuple la puissance de la recherche et permet des avancées conséquentes. Ainsi, afin de développer de nouvelles méthodes et approches novatrices, une phase de recherche documentaire exhaustive pour déterminer le savoir actuel dans les différents domaines servira de base aux travaux réalisés.

Je vais alors traiter dans ce mémoire les différentes phases du levé de la cathédrale Saint-Paul jusqu'à la reconstruction finale, en détaillant les points importants et en apportant les connaissances nécessaires à la bonne compréhension des développements effectués. Je traiterai toutes les phases de la chaîne de production d'un support numérique traduisant la réalité physique, en détaillant les étapes 4, 5 et 6 du processus de production :

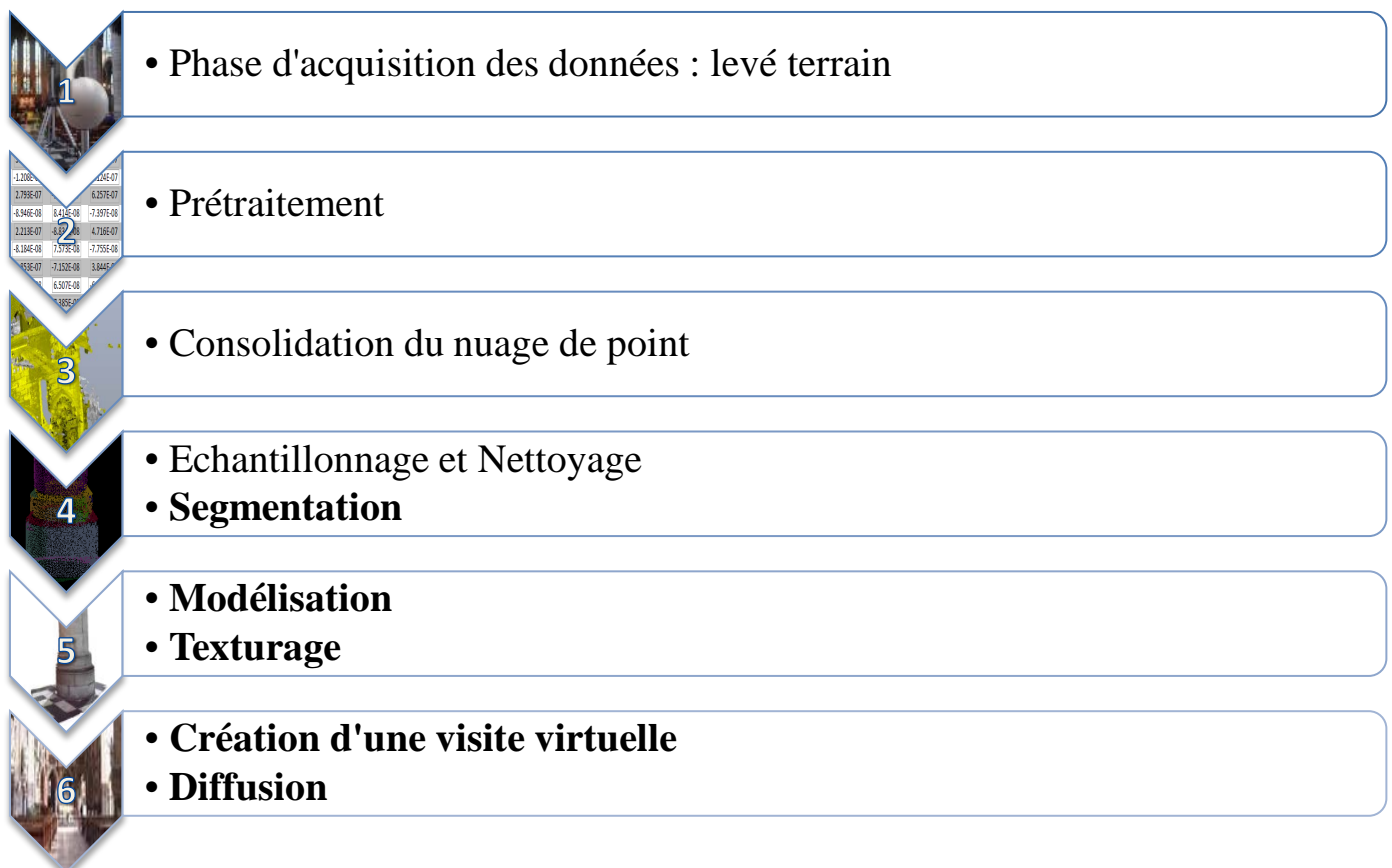


Figure 1 Chaîne de traitement d'un nuage de point obtenu par Laser Scanner

# 1 CONTEXTE DE L'ÉTUDE

## 1.1 L'Unité Géomatique de l'ULG

### 1.1.1 Présentation

L'Unité de Géomatique de l'Université de Liège a une histoire qui remonte à 1986 avec la création du laboratoire SURFACES (« Service Universitaire de Recherches Fondamentales et Appliquées en Cartographie et Études Spatiales»). Depuis, la création d'un cursus Géométrie en 1992 et le rassemblement des différentes chaires de cartographie & SIG, Topographie & Géométrie, Photogrammétrie & Télédétection en 1997 en font un lieu de rencontre pluridisciplinaire de professionnels, chercheurs & professeurs de différents horizons.

A la suite de nombreuses publications et de travaux de recherche aboutis, l'unité est aujourd'hui reconnue pour son expertise dans les modèles 3D urbains et dans la modélisation spatio-temporelle.

L'équipe est composée de 20 personnes qui partagent des compétences en technique d'acquisition (Levés, Lasergrammétrie, GNSS-GPS, Photogrammétrie, Télédétection,...) et qui travaillent en étroite collaboration. Mon stage s'est déroulé au sein de la section Topographie & Géométrie qui comprend :

- Roland Billen : chef de l'équipe et maître de stage
- Pierre Hallot : premier assistant et docteur en sciences
- Benoît Jonlet : assistant mi-temps et géomètre
- Cyril Carré : ancien élève de l'ESGT maintenant ingénieur de recherche et doctorant
- Vincent Delfosse et Michelle Pfeiffer : Chercheurs
- Loubna El Mansouri & Rafika Hajji : boursières FNRS

### 1.1.2 L'environnement de travail

Je fus pour la réalisation de ce TFE basé dans un bureau commun avec Pierre HALLOT et Benoit JONLET, dans les complexes de l'unité de géomatique située sur le domaine du Sart-Tilman, surplombant la ville de Liège.

Mon outil de travail principal, un ordinateur muni d'un processeur i7, d'une carte graphique Nvidia GTX 260 et de 8 Go de RAM conditionna en partie la gestion du nuage de point qui posait des problèmes dès lors que trop de points étaient affichés simultanément. L'utilisation des logiciels de traitement des données SCENE et 3DReshaper étaient conditionnée à une utilisation limitée à 30 jours, contrairement aux logiciels Trimble Realworks, Covadis, Autocad Civil 3D, Map 3D, 3DSmax et aux logiciels gratuits.

D'un point de vue pratique, tout le matériel de topographie était mis à ma disposition et il m'a été possible d'utiliser stations totales (Leica TPS1200, Trimble VX) GPS (Smart Antenna Leica 1200) et laser scanners (Trimble GX, FARO FOCUS<sup>3D</sup>). Je fus également sollicité pour aider à des travaux extérieurs au TFE.



Figure 2 Matériel mis à disposition et utilisé



## 1.2 La cathédrale Saint-Paul de Liège

### 1.2.1 Son histoire

Dans le cœur historique de la ville wallonne Liège se dresse la cathédrale Saint-Paul, classée au patrimoine monumental de la Belgique. Elle a la particularité d'avoir été façonnée et modifiée au fil des âges, pour aujourd'hui être le fruit d'un mélange de styles et d'époques qui en font un ouvrage fascinant.



Figure 3 Cathédrale Saint-Paul de Liège : Façade Sud et Grande Nef

Édifiée à la fin du X<sup>ème</sup> siècle par Éracle et achevée par Notger, elle a d'abord été d'origine romane pour être agrandie et rénovée au XIII<sup>ème</sup> siècle. Les plans dessinés en 1240 illustrent un style gothique imposant, qui prit forme 200 ans plus tard pour devenir la réalité que nous admirons aujourd'hui.

Mais son histoire ne s'arrête pas là. Après la destruction de la cathédrale Saint-Lambert de Liège durant la tourmente révolutionnaire<sup>1</sup>, la collégiale Saint-Paul hérite du titre de cathédrale en 1804. Une telle promotion donna alors l'occasion de l'embellir par l'agrandissement de la tour et l'élévation d'une flèche avec des matériaux de l'ancienne cathédrale.

C'est ainsi qu'aujourd'hui, en plus de cacher des secrets riches d'une histoire mouvementée sous une image gothique, la cathédrale conserve dans son trésor plusieurs œuvres d'art et de renom : la châsse et le buste vermeil de Saint Lambert, le reliquaire de Charles le Téméraire, le Christ gisant de Jean Del Cour et bien d'autres tableaux de maître ou ornements liturgiques de valeur.

### 1.2.2 Structure et organisation

La cathédrale Saint-Paul est organisée selon un schéma mettant en avant la forme symbolique de croix latine. La grande nef comporte sept travées et se prolonge dans le bas de la tour. Elle s'appuie sur des colonnes cylindriques, surmontées d'un chapiteau aux ornements très sobre, d'un tailloir<sup>2</sup> d'où s'élèvent des colonnettes qui se déploient dans les voûtes et d'un triforium (galerie ajourée de l'intérieur au 1<sup>er</sup> niveau) fait de colonnettes et d'écoinçons (embrasure d'une baie) sculptés. Le chœur<sup>3</sup> polygonal est percé de baies à remplage (armature en pierre taillée) gothique. Le soubassement des murs est orné de deux rangées d'arcatures<sup>4</sup> aveugles très dépouillées (la première est cachée par les stalles) Les colonnes sont engagées et ornées de très petits chapiteaux servant de départ aux voûtes.

<sup>1</sup> crise économique et sociale post-révolutionnaire (1789)

<sup>2</sup> Partie supérieure du chapiteau des colonnes

<sup>3</sup> partie du plan d'une église prévue pour les clercs et les chantres

<sup>4</sup> Série d'arcades de petite dimension



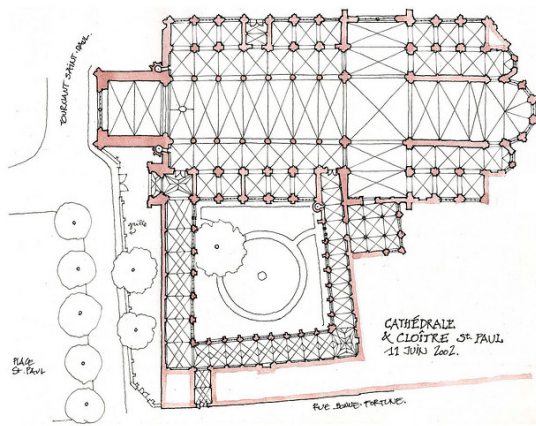


Figure 4 Plan de la cathédrale Saint-Paul

Caractéristiques techniques et dimensions :

- Longueur totale : 82.75 m
- Longueur du transept<sup>5</sup> : 33.50 m
- Largeur de la nef et du chœur : 11 m
- Hauteur maximale (sous clocher) : 24 m

Les détails fins de la cathédrale sont de l'ordre du centimètre et contrastent avec l'envergure de l'édifice mettant en avant une dimension multi-échelle.

### 1.2.3 Un intérêt archéologique et patrimonial

La Belgique est un pays ayant un goût prononcé pour la culture et sa diffusion. Grâce à sa politique passée de neutralité, elle ne fut pas ou peu sujette à des dommages et autres destructions dues aux différentes guerres ayant frappé l'Europe. Ainsi, l'héritage légué par les générations passées fait l'objet d'un soin particulier afin de transmettre de manière intact ou augmenté le patrimoine aux générations futures.

La diffusion culturelle et la conservation du patrimoine sont ainsi la cible du développement d'un support virtuel de la cathédrale Saint Paul de Liège.

Il réside également un intérêt en archéologie du bâti, discipline qui étudie les élévations de toute construction. L'excellente conservation de la cathédrale Saint-Paul, favorise la lecture des informations concernant la mise en œuvre du bâtiment. Une équipe d'archéologues est chargée de mener des missions d'étude du bâti médiéval : relevé « pierre à pierre » des façades, analyses métriques des éléments architecturaux et sculptés, analyse des traces d'assemblages, des techniques de taille, etc. Ainsi, l'analyse des matériaux pierreux, des décors et de leur composition pourra mettre en évidence l'histoire de la construction de l'édifice et d'en préciser la datation. . Dans le même ordre d'idée, la superposition des couches d'enduit (stratigraphie verticale) permet de repérer les différents états de décor d'un édifice et de fournir des indications essentielles sur les différents remaniements du bâtiment (percement ou bouchage d'ouverture par exemple).

L'utilisation conjointe des données historiques et archéologiques s'avère essentielle afin de restituer d'une manière détaillée l'histoire de la construction des édifices médiévaux (l'ouvrage de (Gerrard, 2003) raconte pertinemment l'histoire du développement de l'archéologie médiévale). On peut à l'aide d'un compte de construction, compléter les parties inconnues d'un édifice (parties hautes incomplètes, pans entiers disparus) ou préciser certains détails (présence de vitres et volets aux fenêtres, enduits peints...). À l'inverse, l'étude des vestiges peut pallier les imprécisions des textes, identifier les outils et techniques employés ou encore montrer l'évolution dans le temps d'un édifice que les textes ne décrivent qu'une fois terminé.

Cependant, en considérant l'envergure de la cathédrale Saint-Paul, cela définit un travail conséquent qui consistera à réaliser des levés pierre à pierre, des études métriques sur les hauteurs des voûtes, les maçonneries, ou encore les détails de chaque colonnade, décor et écoinçon sculpté dans un ordre de précision centimétrique. Seules les zones accessibles de l'édifice peuvent être analysées en l'absence d'échafaudages ou d'élévateur.

La possibilité d'effectuer tout ce travail grâce à un nuage de points d'une résolution millimétrique est un atout indéniable, en termes de temps, de précision et de possibilités d'exploitation. Les apports d'un tel support répondent aux exigences de précision du cahier des charges, et permettent d'exploiter facilement des zones inaccessibles avec un niveau de détail élevé. Les archéologues travaillant essentiellement sur des plans et des coupes en 2D, l'apport de la 3D permettra également de considérer la volumétrie des matériaux tout en donnant des moyens d'analyse supplémentaire.

<sup>5</sup> nef transversale qui coupe à angle droit la nef principale

## 2 ÉTAT DE L'ART

### 2.1 Acquisition des données par scanner laser

#### 2.1.1 Qu'est-ce qu'un scanner laser terrestre ?

La lasergrammétrie est une technique en plein essor qui exploite la technologie des scanners lasers terrestres, systèmes actifs utilisant la lumière laser comme support d'information (voir (Reshetyuk, 2009) pour plus d'informations sur l'utilisation du laser et son fonctionnement)

Depuis une position définie où l'appareil est stationné, l'acquisition se fait par un balayage laser terrestre, qui permet l'obtention rapide, sans contact et automatique de coordonnées 3D de points surfaciques visibles.

Le faisceau laser est émis à des fréquences élevées permettant l'acquisition de plusieurs milliers de points par seconde. Le mouvement du dispositif pour enregistrer la scène dans son ensemble est mesuré avec une grande précision permettant de déduire la position dans l'espace de tous les points d'impact du laser. L'ensemble tramé de points 3D représentant la surface relevée par l'instrument est appelé nuage de point.

Chaque scanner laser 3D possède plusieurs caractéristiques techniques qu'il est utile de comprendre afin d'utiliser un instrument adapté aux précisions attendues. Je vais donc au moyen d'une explication théorique détailler les différences entre les propositions du marché, pour ainsi déterminer quel sera ici l'outil adapté.

#### 2.1.2 Principes physiques et théoriques

Les scanners lasers actuels s'appuient sur plusieurs méthodes de mesure de distance, dont les principales sont :

- **La triangulation :**

La distance est calculée grâce à une résolution de triangle. Le laser permet de pointer l'objet avec le faisceau, le capteur permet lui de positionner ce pointé. Le point laser, le capteur CCD<sup>6</sup> (ou PSD) et l'émetteur laser forment un triangle dont on connaît un côté de distance  $d$  (entre le laser et le capteur) et deux angles adjacents à la base. Les angles connus se trouvent au niveau du laser émetteur et au niveau du capteur. L'angle à l'émetteur peut varier car le faisceau est souvent dirigé par un dispositif de déviation (un miroir). L'angle au capteur peut par conséquent varier aussi. Il est déterminé grâce à la position du faisceau d'arrivée sur la matrice CCD.

Ces scanners de grande précision (de 0,3 à 0,6 mm) ont l'inconvénient d'une portée qui dépend de la dimension de la base inhérente à l'appareil (appareil courte distance : distance de 0,8 à 10 m, appareil longue distance : de 2,5 à 25 m).

- **Le temps de vol (pulse method)**

La méthode consiste à mesurer le temps  $\Delta t$  que met le faisceau laser pour faire l'aller-retour entre la station et la surface visée. L'indice du milieu dans lequel évolue le faisceau laser et la vitesse de la lumière dans le vide étant connus, on peut calculer la distance station-surface visée grâce à l'équation :  $2 * D = \frac{c_0}{n} * \Delta t$

Avec  $D$  la distance scanner/objet,  $C_0$  la vitesse de la lumière dans le vide,  $n$  l'indice du milieu,  $\Delta t$  l'intervalle de temps entre émission du signal et sa réception (cf. Figure 6).

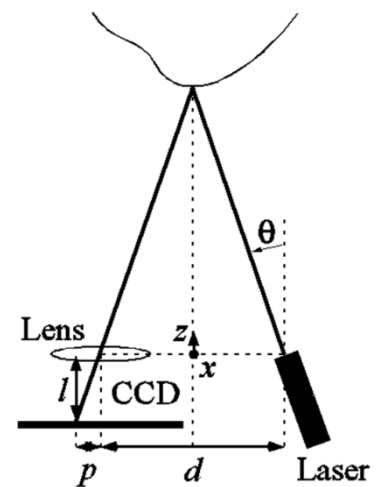


Figure 5 Principe triangulation

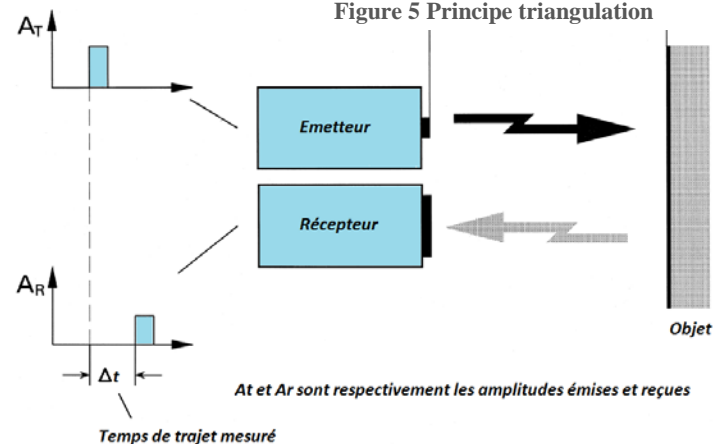


Figure 6 Méthode temps de vol inspiré de (Wehr & Lohr, 1999)

<sup>6</sup> Charge Couple Device : Capteur photographique

En général, un miroir tournant guide le faisceau de mesure. Lorsque le faisceau retour stoppe le système de mesure de temps, on peut calculer les coordonnées tridimensionnelles en prenant en compte l'angle du miroir, la vitesse du faisceau dans le vide, l'indice du milieu environnant et le temps mis par le faisceau pour réaliser l'aller-retour. L'avantage de ces scanners est de répondre à des cahiers des charges très diversifiés. Grâce à leurs précisions et leurs portées, ils peuvent être utilisés dans beaucoup de domaines. Leur seul désavantage peut résider dans la précision qui n'est, pour certains modèles, pas suffisante pour des applications bien précises comme la métrologie ou l'auscultation d'ouvrage.

• **La différence de phase (phase shift)**

Cette méthode est basée sur une mesure de déphasage. Le laser émet de façon continue une onde sinusoïdale de fréquence fixe donnée. Le récepteur du scanner reçoit le signal rétrodiffusé de même fréquence mais déphasé (cf. Figure 7).

La valeur du déphasage  $\Delta\phi$  permet d'obtenir la distance parcourue par le signal grâce à l'équation :

$$\Delta\phi = 2\pi * f * \Delta t$$

$$\Rightarrow \Delta\phi = 2\pi * f * \left(\frac{2d}{c}\right)$$

Soit

$$d = \left(\frac{\Delta\phi * c}{4\pi * f}\right)$$

$\Delta\phi$  : Différence de phase (décalage temporel) (radian)

$f$  : fréquence du signal sinusoïdale (HZ)

$\Delta t$  : Temps décalage temporel entre le signal envoyé et le signal reçu (s)

$d$  : distance parcourue (m)

$c$  : vitesse de la lumière ( $\cong 3 * 10^8$  m.s<sup>-1</sup>)

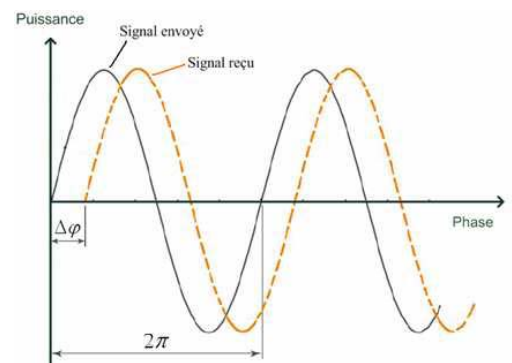


Figure 7 Principe du décalage de phase

La méthode de calcul du décalage de phase consiste à combiner les faisceaux envoyés et reçus comme indiqués dans l'équation :

$$\Delta\phi = \arccos\left(\frac{2}{T} * \int_0^T S_{\text{émis}} * S_{\text{reçu}} * dt\right)^7$$

$\Delta\phi$  : décalage de phase (radian)

$S_{\text{émis}}$  : signal envoyé ( $\cos(\omega t)$ )

$S_{\text{reçu}}$  : signal reçu ( $\cos(\omega t + \Delta\phi)$ )

$T=1/f$  : période (s)

$\omega$  : Pulsation (rad/s)

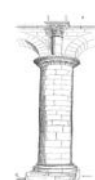
La méthode d'acquisition est très rapide, généralement plus rapide que les autres (quelques centaines de kHz) et la précision de mesure est meilleure que celle obtenue par la méthode du temps de vol.

**Conclusion**

En conclusion générale, Les méthodes de télémétrie laser ont chacune leurs spécificités et domaines d'application. Le scanner par triangulation laser bien qu'extrêmement précis n'est pas adapté à des mesures longues distances et rapides. La méthode par temps de vol a la meilleure portée mais sa précision est relativement faible. La méthode par différence de phase a une vitesse et une précision convenable mais sa portée peut être relativement limitée. Selon l'utilisation de chacun, il est utile de se référer au tableau de comparaison des méthodes de télémétrie laser sur trois critères : portée, précision et vitesse d'acquisition.

	Portée	Précision	Vitesse d'acquisition
Triangulation	+	+++	+
Temps de vol	+++	+	++
Phase différentielle	++	++	+++

<sup>7</sup>  $\int_0^T S_{\text{émis}} * S_{\text{reçu}} * dt = \frac{1}{2} * \int_0^T (\cos(2\omega t + \Delta\phi) + \cos(\Delta\phi)) * dt = \frac{T}{2} * \cos(\Delta\phi)$



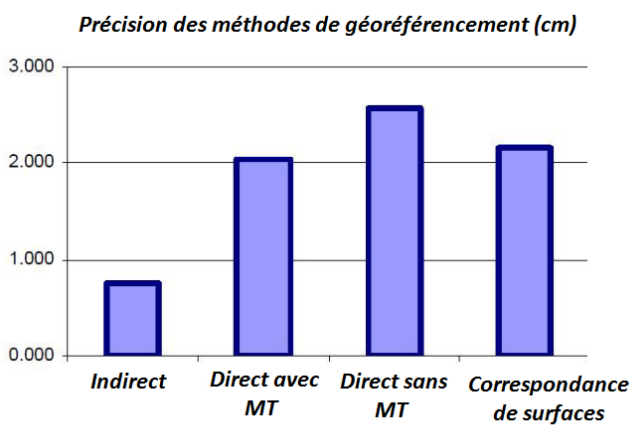
## 2.2 Traitement des données

### 2.2.1 La consolidation

« Le recalage 3D de nuages laser constitue la première étape d'un processus de traitement des données qui précède toute opération de modélisation architecturale afin de parfaire la géométrie de l'objet scanné. » (Landes & Grussenmeyer, 2011; Landes et al., 2011)

L'assemblage de tous les nuages de points acquis depuis les différentes stations sera donc la base de tout traitement postérieur. En effet pour le relevé de larges scènes (ou d'un objet 3D dans sa globalité) une seule station ne suffit généralement pas à obtenir l'ensemble du volume 3D. C'est pourquoi la problématique de recalage relatif entre en jeu.

Pour résoudre ce problème, il est possible d'effectuer un géoréférencement direct (afin d'être référencé dans un unique système de coordonnées géographique), par techniques topographiques classiques (mise en place d'un réseau de points d'appuis, centrage du scanner sur point connu, orientation, balayage des cibles) ou un géoréférencement indirect (visées sur références connues puis calcul de la position relative), qui est mieux approprié dans la majorité des cas car il donne de meilleurs résultats (Reshetyuk, 2009).



MT : Mesure tachéométrique

Figure 8 Précision des méthodes de géoréférencement inspiré de (Reshetyuk, 2009)

Cependant, ceci implique de passer par une étape dite de « consolidation » pour placer les différents relevés issus des différentes stations réalisées (dans un système lié à chaque station de scan) dans un système unique de coordonnées. Cela suppose qu'il existe au moins 3 paires de points homologues entre différents nuages : il peut s'agir de cibles artificielles (sphères ou surfaces réfléchissantes) qui seront reconnus par le logiciel d'acquisition adéquat ou d'éléments naturels intangibles (plans, surfaces, coins, ...) qui utilisent la méthode de correspondances de surfaces moins précise (cf. Figure 8) mais utile en cas d'un nombre de cibles insuffisant. Lors de la mise en place de points homologues il faut veiller à ne pas les placer sur un même axe (pour éviter une consolidation peu fiable dû à une géométrie inadaptée), et de multiplier les points homologues pour avoir une redondance des observations (permettant d'affiner la transformation).

Une telle opération se base sur une transformation Helmert 3D à 6 ou 7 paramètres (le facteur d'échelle est souvent négligeable grâce au compensateur interne<sup>8</sup> du scanner 3D), permettant de déterminer les différents paramètres de transformations (translations  $T_x$ ,  $T_y$ ,  $T_z$  ; rotations  $\theta_x$ ,  $\theta_y$ ,  $\theta_z$  ; facteur d'échelle  $\mu$ ) :

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix}_2 = \begin{bmatrix} T_x \\ T_y \\ T_z \end{bmatrix} + \mu * R * \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}_1$$

Où

$$R(\theta)_x = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\theta) & -\sin(\theta) \\ 0 & \sin(\theta) & \cos(\theta) \end{bmatrix}, R(\theta)_y = \begin{bmatrix} \cos(\theta) & 0 & \sin(\theta) \\ 0 & 1 & 0 \\ -\sin(\theta) & 0 & \cos(\theta) \end{bmatrix}, R(\theta)_z = \begin{bmatrix} \cos(\theta) & -\sin(\theta) & 0 \\ \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Pour cela, il faut transformer dans un premier temps cette équation en équation linéaire. Dans la majorité des cas, les angles sont faibles et permettent d'approximer la matrice  $R$  ( $\cos(\theta) = 1$  et  $\sin(\theta) = \theta$ ). Sinon, il faut effectuer des premières rotations approchées pour se placer dans le cas d'approximation. On résout ensuite un système des moindres carrés  $Ax=B$  et l'on obtient les paramètres de transformation. Ces paramètres peuvent être appliqués aux différents repères concernés pour se placer dans un système unique de coordonnées.

<sup>8</sup> Le compensateur de niveau intégré à la majorité des scanners lasers permet de contrer tout mouvement du scanner pendant son fonctionnement.

Pour obtenir un modèle affiné automatiquement, il est également possible d'effectuer une consolidation basée sur deux approches différentes décrites dans (Al Shawa, 2006) :

- Dépendant des entités géométriques : elle a pour objectif de retrouver des entités géométriques homologues (cylindres, plans, sphères, ...) dans les deux nuages de points. Le principal défaut consiste en une grande dépendance aux détails locaux saillants, mais également un temps de calcul assez important. L'avantage principal est d'accentuer les propriétés géométriques au lieu de dépendre seulement des points du scan.
- Indépendant des entités géométriques : la distinction au sein de ce groupe se fait selon une correspondance sous-jacente (méthode DARCES basée sur RANSAC, Transformation NDT, ICP) ou par techniques de vote (Hachage par la transformée de Hough, Hachage géométrique).

De toutes les méthodes qui sont présentées habilement par L. Alshawa, la plus couramment utilisée est la méthode ICP. Elle consiste à calculer itérativement la matrice de transformation recalant le mieux plusieurs ensembles de données 3D (courbes, surfaces, plans,...). Le principe de l'algorithme est d'itérer les deux étapes du recalage : la mise en correspondance des données (association des points par critères du plus proche voisin) et l'estimation de la transformation des paramètres de transformation (rotation et translation). Ces étapes sont répétées jusqu'à convergence de l'algorithme lorsque l'erreur résiduelle de distance entre les points appariés est inférieure à un certain seuil. Plusieurs méthodes existantes basées sur une méthode ICP répondent à ce problème décrite dans (Douadi *et al.*, 2005) qui s'accordent à dire qu'elle varie principalement sur la sélection des points à appairer (échantillonnage des données, sélection des point de contrôle) ; la technique d'appariement utilisée ; la pondération des paires de points appariés ; le rejet des mauvais appariements ; le type de critère à minimiser et l'algorithme de minimisation utilisés pour l'estimation de la transformation.

Le problème de cette méthode, est le risque de convergence vers un minimum local et non pas global, accentué par la présence de bruit. De plus, malgré les différentes variantes qui essaient de palier aux problèmes rencontrés, l'algorithme reste semi-automatique et demande une action utilisateur pour fonctionner correctement.

### 2.2.2 Le nettoyage du nuage

Une fois la consolidation effectuée et validée, il faut nettoyer le nuage afin de supprimer les points faux et aberrants et limiter l'influence du bruit. En effet, lorsque la réflectance d'un point est trop faible, l'erreur de mesure est augmentée (la qualité du signal retour est faible), et le point est alors incertain. De plus, le rayon laser ayant un certain diamètre, il est possible qu'il touche deux surfaces distinctes renvoyant un signal faux. Ce bruit inhérent aux mesures peut ajouter une épaisseur là où il n'y en a pas qu'il faut donc éliminer au moyen de différents filtres (Le bruit dû au passage de voiture, de personnes doit être écarté manuellement).

En outre, la densité de points étant généralement non constante suivant la position des différentes stations de scans, il existe des solutions d'échantillonnage. Suivant la méthode utilisée, il est possible d'échantillonner le nuage de point pour garder une densité de point constante, ou suivant les discontinuités, etc., permettant d'alléger le nuage de point sans perte d'information utile (la Figure 9 montre la possibilité d'une réduction de 65% du volume des données – 5 millions/14 millions – tout en conservant un niveau de détail sub-millimétrique).



Figure 9 Exemple de différents échantillons de différentes densités appliquées à un vitrail de la cathédrale

### 2.2.3 La segmentation

Le processus de segmentation de nuage de point est une étape essentielle dans la chaîne de traitement. Elle conditionne la pertinence et la précision d'une reconstruction de la scène ciblée. En effet, c'est elle qui sera la base d'une analyse du contenu total en différents sous-groupe aux propriétés différentes, apportant des informations en amont de la globalité considérée. C'est donc cette étape qui permet de dégager les différents éléments constitutifs d'un nuage de point pour être traité, analysé, visualisé de manière indépendante.

En amont des travaux de (Douillard & Underwood, 2011), il apparaît que la segmentation de données denses, non structurées et bruitées est un problème majeur. La segmentation manuelle de nuages de plusieurs milliards de point étant extrêmement chronophage et imprécise, la recherche d'une automatisation d'un tel processus est au cœur des recherches lasergrammétriques, et par extension au centre de ce mémoire.

Il existe différentes approches de segmentation automatique ou semi-automatique ((Castillo, 2013; Chaperon, 2002; Lerma & Biosca, 2005; Schnabel et al., 2008, 2007, 2006) qui diffèrent selon l'approche théorique sous-jacente, le critère de similarité retenu au sein d'un groupe de point, les besoins logiciels et algorithmiques, mais surtout de par les résultats et les possibilités d'applications. Ces méthodes existantes peuvent être regroupées en 4 catégories selon leurs spécificités

#### **Segmentation par détection de coin/contour :**

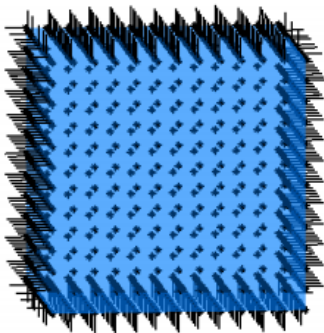


Figure 10 Estimation des normales d'un cube – source (Castillo, 2013)

Les algorithmes de segmentation basés sur les arrêtes sont constitués de deux étapes : la détection d'arrêtes qui définit les bords des différentes régions, suivies par le regroupement des points à l'intérieur des limites donnant le segment final. Les arrêtes, suivant l'approche, sont définies par un changement dans les propriétés de surface locales (normales, gradients, courbure ou autre dérivées d'ordre supérieur). Les travaux intéressants menés par (Castillo, 2013) et (Rabbani, 2006) sont basés sur l'estimation de normales locales où la détection des arrêtes se fait par un changement brutal du critère d'uniformité choisi. Certains autres critères comme la connectivité locale, le gradient type sont également pris en compte.

#### **Segmentation par reconnaissance de formes :**

La segmentation du nuage de points est basée sur les formes géométriques reconnues et extraites. Les algorithmes les plus connus et utilisés sont :



Figure 11 Algorithme RANSAC appliqué à plusieurs colonnes via CloudComapre

**RANSAC** : Ce paradigme largement étudié est bien détaillé et résumé dans les travaux de (Schnabel *et al.*, 2007). Il permet d'extraire des formes en sélectionnant aléatoirement des groupes de points et en construisant la primitive géométrique correspondante. Le candidat résultant est testé avec tous les points traités pour connaître la précision de l'approximation de la primitive. Après plusieurs essais, la forme qui donne les meilleurs résultats d'approximation est extraite et l'algorithme continu sur le reste des données. Cet algorithme, conceptuellement simple, robuste face au bruit et facilement extensible pose des problèmes de temps de traitement élevé, de sur/sous-segmentation, et de difficulté d'isolation des détails fins.

**HOUGH** : La transformée de Hough est une méthode d'extraction des primitives basé sur le changement d'un espace objet à un espace des paramètres. (Borrmann *et al.*, 2011) a utilisé la transformée de Hough pour détecter les plans de toiture tandis que (Rabbani & Heuvel, 2005) soulignent la dépendance du temps d'implémentation de l'algorithme par rapport à la résolution du nuage de points, ce qui inhibe sa perfection.

### Segmentation basées sur les surfaces :

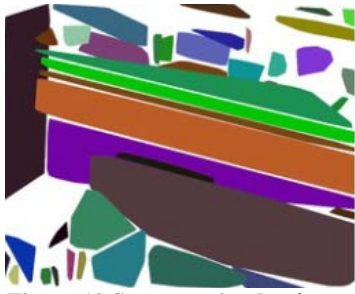


Figure 12 Segmentation basée sur les surfaces. Source (Holz, 2013)

La segmentation basée sur les surfaces utilise les propriétés locales des surfaces pour mesurer la similarité et regroupe les points qui sont spatialement proches et qui ont des propriétés surfaciques similaires. Ces méthodes sont relativement moins sensibles au bruit, et sont plus rapides que les méthodes de détection d'arrête. Deux approches différentes ont été étudiées et proposées dans (Douillard & Underwood, 2011), qui sont : Bottom-up et Top-Down<sup>9</sup>. La méthode Bottom-up est initialisée par la sélection d'un pixel source, et va faire grandir la région selon un critère de similarité (utilisée par la majorité des méthodes existantes). Top-down assigne un pixel à un groupe et fait correspondre une surface à ce dernier. Certains travaux comme ceux de (Holz & Behnke, 2013)

utilisent ce principe appliqué à un modèle maillé et non de nuage de points.

### Autre approches

Certaines approches sont des « hybrides » des méthodes précitées. Un bon exemple est la méthode développée par (Rusu & Blodow, 2009; Rusu, 2010), qui détermine des représentations d'objets géométriques consistant en un mélange forme/surface. Très efficace car combinant les forces d'une représentation géométrique et surfacique, cette approche a tout de même le désavantage de considérer des objets fortement réguliers (ce qui est très rarement le cas dans l'étude d'archéologie du bâti).

Il existe également des segmentations basées sur l'étude colorimétrique, mais ceci est rarement utilisé car les points d'une même région ne partagent pas réellement les mêmes caractéristiques colorimétriques/radiométriques.

### Synthèse

Toutes ces différentes approches ne sont pas adaptées à des ouvrages aux détails fins, ou sont trop limitées à un usage spécifique. J'ai ainsi relevé les principaux problèmes posés par toutes les méthodes de segmentations étudiées et testées :

- Elles sont trop limitées à des applications sur des surfaces planes, ou des scènes industrielles
- La proportion de bruit et l'irrégularité des formes produit souvent le phénomène de sur-segmentation, particulièrement concernant les objets sphériques ou cylindriques, avec un haut niveau de courbure.
- La sous-segmentation ne permet pas de relever les détails, et approxime souvent plusieurs formes de paramètres différents en une seule, provoquant une large perte de précision et de pertinence.
- Certaines méthodes de segmentation demandent en entrée un grand nombre de paramètres, parfois abstrait ou qui ne représentent pas grand-chose et ne permettent pas de réaliser une segmentation maîtrisée.
- L'usage de la 3D n'est pas complet. Il existe plusieurs méthodes qui travaillent réellement en 2D, puis qui sont portées vers la 3D par usage de coupes multiples etc, mais cela entraîne une non prise en compte de beaucoup de données couplé à un traitement lourd et non optimisé.
- De très complètes approches permette une bonne détection de plans, et surfaces planes (Hakim Boulaassal, 2010), mais malheureusement cela limite essentiellement l'usage à des scènes industrielles ou simples, et n'est pas adapté dans notre cas.

C'est dans cette optique et en ayant en tête les faiblesses de ces méthodes que je m'appliquerai à développer une solution permettant une segmentation adaptée, naturelle, efficace et précise.

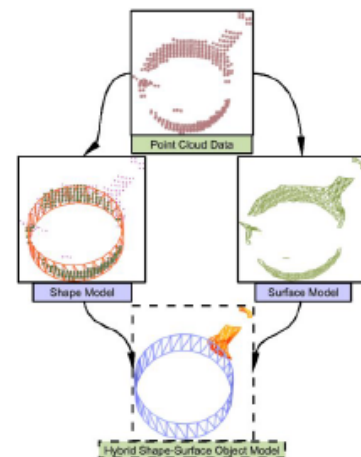
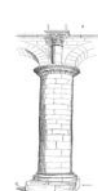


Figure 13 Approche de segmentation hybride. Source (Rusu & Blodow, 2009)

<sup>9</sup> Une traduction appropriée serait de bas en haut, et du grossier vers le détail





## 2.2.4 La modélisation

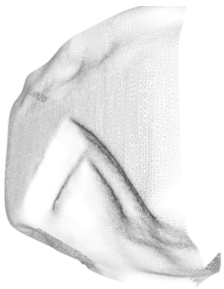


Figure 14

Modélisation d'une arcade

Afin d'implémenter et d'utiliser les données scanners obtenues et traitées, il faut transformer les nuages de points en modèles surfaciques et « imperméable ». Un nuage de points n'affiche aucun lien palpable : tous les points sont indépendants physiquement de leurs voisins. Pour une représentation visuelle et une utilisation virtuelle, la plupart des applications utilise une représentation surfacique. Il est nécessaire de créer ces liens, analogues à des fils de fers liant les points entre eux, pour obtenir un modèle maillé représentant topologiquement l'organisation du nuage de point : c'est la modélisation.

La production d'un modèle reste soumise à une succession d'étapes interdépendantes conditionnées par la nature de l'existant et le type de représentation à atteindre

- Le modèle 3D restitution: En archéologie, le modèle 3D est souvent utilisé pour la représentation de reconstitutions de vestiges à partir de fragments. Dans ce cas un modèle élaboré à partir des relevés est complété par les connaissances et l'expertise de spécialistes du domaine (archéologues, architectes, etc...). Ils valident toutes les hypothèses émises lors de la modélisation pour aboutir à la complétude du modèle : une reconstitution.
- Le modèle 3D « tel que saisi » : C'est la modélisation la plus fidèle à l'objet mesuré, dont la pertinence relève uniquement de la qualité des mesures et du processus d'élaboration du modèle. Il représente la construction d'un objet tel qu'il existe. Les informations nécessaires à sa construction sont uniquement issues du relevé de l'objet lui-même.
- Le modèle 3D « tel que construit »: Il se réalise sur la base des mesures effectuées sur l'objet sans aucune extrapolation. En archéologie, cette modélisation est utilisée pour masquer l'effet de l'érosion de la pierre et les défauts mineurs dus au temps. Utilisée dans un cadre industriel, ce type de modélisation permet également le redéploiement d'installations (revamping) en reconstituant l'état réel du bâtiment. La modélisation construite sur la base des mesures jugées nécessaires représente une réalité géométriquement idéale.

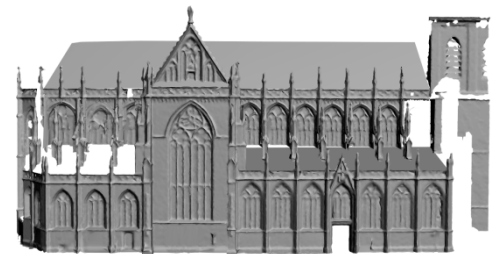


Figure 15 Modélisation de la façade Sud réalisée avec 3DReshaper

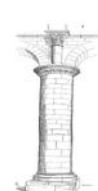
Avec les avancées technologiques fulgurantes, et l'évolution constante du matériel informatique, les problèmes rencontrés 5 ans auparavant en matière de gestion graphique ne sont plus forcément d'actualité. Cependant, la volonté de développer un produit flexible pour une utilisation variée (public, professionnel, archéologique) demande la prise en compte de l'hétérogénéité des technologies existantes. L'optimisation et la création de différents niveaux de détail est donc essentielle à la réalisation d'un tel travail. En parallèle, la précision étant une qualité essentielle pour le travail lasergrammétrique, les différentes représentations et simplification ne doivent pas altérer ce paramètre.

Il existe plusieurs méthodes de modélisation, chacune ayant ses spécificités, ses atouts et faiblesses.

### Modélisation paramétrique

Le principe d'une modélisation paramétrique est de décrire une forme géométrique grâce à un ensemble de paramètres ou d'équations mathématiques. Les travaux de (Beder & Förstner, 2006) présentent une méthode de calcul alternative de cylindres à partir d'un échantillon de points, et leur représentation paramétrique. Des cylindres sont détectés par usage de formes a priori (largement utilisé pour la détection dans les scènes industrielles, voir (Bey, 2011)) et décrits par l'équation :  $(X - s)^2 + (Y - t)^2 - r^2 = 0$ , dont les paramètres sont déterminés classiquement par la substitution  $u = s^2 + t^2 - r^2$  ( $s, t$  sont les coordonnées du centre de la base du cylindre,  $r$  est le rayon du cylindre).

Ainsi, plutôt que de traiter des milliers de lignes, contenant des points ou des triangles liant ces points pour obtenir un modèle surfacique, une (ou plusieurs) équation permettra cette représentation.



Il s'agit d'un formidable gain d'espace, de temps de traitement et de complexité d'affichage. Les problèmes de sommets et de triangulation sont réduits à leur plus simple expression (voir travaux de (Liu *et al.*, 2006))

Cependant, ce genre de modélisation, bien que très adapté aux formes géométriques simples et parfaites, sera limité pour la représentation et la retranscription de la réalité.

Pour une représentation fidèle il est nécessaire de coupler plusieurs types de modélisations, et d'obtenir un point d'équilibre entre les différents usages. De plus, comme il sera traité plus loin, ce genre de représentation pose des problèmes pour l'application de textures.

### Modélisation géométrique

En fil de fer, surfacique ou volumique, elle a pour principe la mise en correspondance de formes géométriques à un objet ou une scène pour le modéliser parfaitement. Les algorithmes existants sont efficaces pour la visualisation, les transformations géométriques ou la réalisation d'opérations booléennes<sup>10</sup>.

La modélisation géométrique définie dans (Hakim Boulaassal, 2010) utilise les nuages de points comme support pour la réalisation de modèles tels que construits. Elle se fait selon différentes méthodes, souvent complémentaires :

- La modélisation à base de primitives géométriques qui s'appuie sur le travail de l'opérateur et une série de primitives géométriques (point, segment, courbe, boîte, cylindre, etc.).
- La modélisation par reconnaissance automatique et reconstitution, basée sur des outils de géométrie différentielle qui permettent la segmentation automatique de nuages de points. Cette méthode est utilisée dans l'ingénierie inverse<sup>11</sup>, notamment dans les travaux de (Bey, 2011). Quoique rapide du fait de son automatisation, la méthode est inadaptée aux objets architecturaux en général.
- la modélisation à partir de bibliothèques « métiers » paramétriques, dont une approche par connaissance a priori de l'objet est développé dans (Liu *et al.*, 2006). cette logique de développement peut se révéler intéressante, cependant les bibliothèques existantes ne possèdent pas à l'heure actuelle de primitives ou objets architecturaux nécessaires à la modélisation architecturale complexe.



Figure 16 Modélisation géométrique d'une colonne de la cathédrale

### Modélisation polygonale (par techniques de maillage)

Un modèle maillé (polygonal) est un ensemble de sommets (vertices), d'arrêtes (edges) et de faces qui définissent la forme de l'objet que l'on souhaite reconstituer. Les faces sont généralement des triangles, des quadrilatères ou des polygones convexes simples. Contrairement à la modélisation géométrique volumique, la représentation est ici implicite (seules les faces sont explicites).

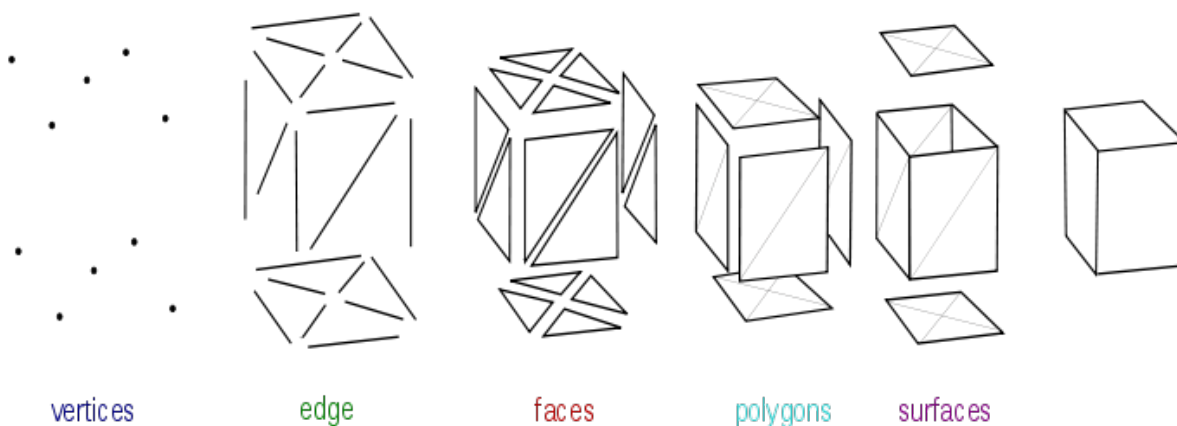
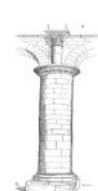


Figure 17 Éléments d'une modélisation polygonale. Source (Pauly, 2003)

<sup>10</sup> Opérations sur des solides utilisant les opérateurs définis par Boole (union, intersection, soustraction)

<sup>11</sup> Etude d'un objet pour en déterminer le fonctionnement interne ou la méthode de fabrication



Il s'agit de la méthode majoritairement utilisée car elle présente les avantages d'être simple à mettre en place et d'être fidèle au nuage de point à modéliser. En contrepartie, selon la définition du niveau de détail, un modèle maillé par méthode polygonale sera bien plus lourd et plus compliqué à traiter qu'une modélisation géométrique.

Cependant, cette méthode largement étudiée dispose d'une quantité de possibilités de simplifications, traitée dans (Thomas Lewiner, 2005) et (Tishchenko, 2010), qui expliquent les contraintes associées à de tels objets (manifold<sup>12</sup>, watertight<sup>13</sup>, ...), les possibilités de subdivision, les différentes approches utilisées pour obtenir le résultat le plus propre, simple et fidèle.

### 2.2.5 La texturisation

Une fois le modèle maillé, il est nécessaire d'appliquer une texture pour le rendre photo-réaliste. Similairement à l'emballage d'un objet dans du papier cadeau, la texture s'applique surfaciquement à l'objet, suivant différentes techniques au choix de l'utilisateur. Le principe est de « découper » une feuille de papier 2D représentant la texture pour qu'elle s'enroule et s'applique parfaitement à l'objet 3D : c'est le « Mapping ». La plus utilisée est l'UV Mapping, qui, analogiquement au problème des différentes projections terrestres, va projeter une texture sur l'objet 3D après avoir appliqué des transformations pour s'assurer de la prise en compte des différentes déformations nécessaires à la concordance 3D (Essayez d'emballer parfaitement une sphère dans une feuille de papier, et vous verrez que ce n'est pas possible sans transformations).

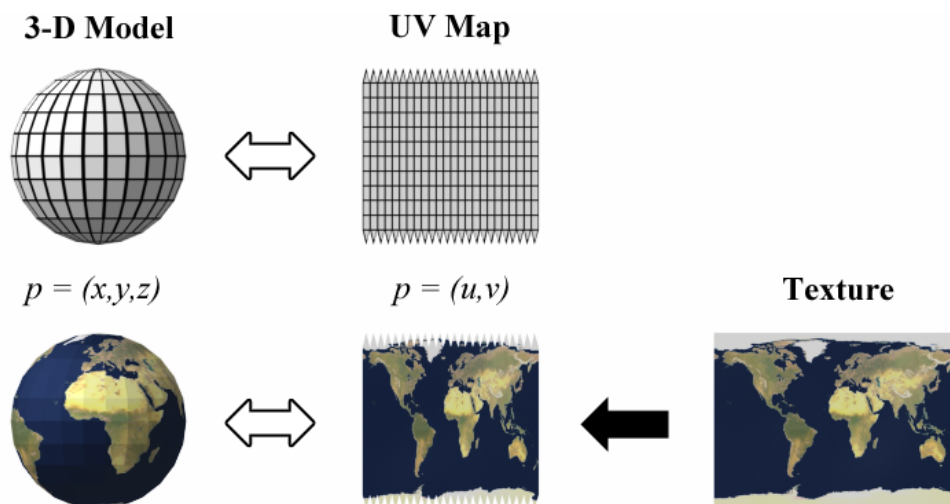


Figure 18 Représentation de l'UV Mapping par un exemple appliqué au globe terrestre. Source Tschmits

Ainsi, le but est de trouver les coordonnées  $u$  et  $v$ , telles que, en considérant le point surfacique  $p=(x,y,z)$ ,  $p=(u,v)$ .

Il existe également ce que l'on appelle la paramétrisation ou la spécification d'un modèle/objet géométrique. Il s'agit en général d'un raisonnement mathématique qui identifie un groupe complet de coordonnées ou de degrés de liberté d'un système, d'un modèle ou encore d'un processus (sans prendre en compte l'utilité pour certains designs). Pour deux surfaces ayant la même topologie il existe un mapping bijectif<sup>14</sup>, qui dans le cas de maillages triangulaires s'appelle « mesh parametrization ». Le paramètre est alors la surface du maillage qui décrit le modèle. Il s'agit d'un outil très puissant pour le traitement de maillages, dont il existe plusieurs variantes selon le domaine du paramètre ou les propriétés de la paramétrisation. Cet outil est particulièrement utile lorsque les textures sont réalisées à partir de la colorisation RGB<sup>15</sup> de chaque point.

<sup>12</sup> Un objet manifold est un objet topologiquement correct. C'est-à-dire que tous les points constitutifs correspondent à une surface topologique correcte.

<sup>13</sup> Imperméable

<sup>14</sup>  $\forall y \in F, \exists ! x \in E \mid f(x) = y$ , où  $F$  est l'ensemble de départ et  $E$  l'ensemble d'arrivée

<sup>15</sup> Red/Green/Blue : Composantes couleurs Rouges/Vert/Bleu ou RVB en français

## 2.3 Diffusion, publication et applications

Consolidation, Segmentation et Modélisation permettent d'aboutir à des données traitées qu'il faut maintenant exploiter concrètement. Afin de déterminer la pertinence d'une utilisation d'un nuage de point ou d'un modèle 3D, il est intéressant de réaliser une étude contextuelle des applications existantes et de recenser les différents moyens de transmission possibles.

### 2.3.1 Différentes applications pour différentes utilisations

#### Animation vidéo

L'animation vidéo est une technique utilisée par exemple pour la création de films ou de leurs effets spéciaux. Elle permet de créer par exemple des visites d'un modèle 3D. Elle a l'avantage d'être un support très maniable, et de ne nécessiter que peu de connaissances informatiques (outre logicielles). Ainsi, la vidéo est une référence en termes de diffusion puisqu'elle permet de communiquer un élément visuel pour présenter les lieux et introduire un travail sur des nuages de points. Il est possible de jouer sur les différents niveaux de détails pour obtenir une vidéo légère et qui conserve une qualité visuelle irréprochable (nécessité d'avoir 24 images/secondes minimum pour une vidéo fluide).

Cependant, l'animation vidéo impose des points de vue ou un cheminement d'exploration du modèle 3D/Nuage de point. Ainsi aucune interaction n'est proposé : il s'agit d'une application destinée à de la pure présentation et prise de connaissance.

#### Anaglyphe

L'anaglyphe est une image qui peut être vue en trois dimensions à l'aide de filtres de couleur (placés devant les yeux de l'utilisateur). Il se base sur le principe de la stéréoscopie qui permet de capter une perception du relief grâce à deux images homologues représentant la même scène mais vue de points légèrement décalés. L'anaglyphe a longtemps été utilisé pour créer des images ou des vidéos en trois dimensions. La plupart des logiciels d'infographie 3D permettent de créer des anaglyphes assez facilement (voir Figure 19 créée avec le logiciel Anaglyph Maker).

Le principal inconvénient de l'anaglyphe est la nécessité de posséder des lunettes possédant des filtres de la même couleur que ceux utilisés pour l'image sous peine de voir des silhouettes fantômes<sup>16</sup> pour visionner l'image ou la vidéo.

Cette principale contrainte rend l'anaglyphe instable pour une large diffusion, car suivant le dispositif de diffusion (les couleurs n'étant pas restituées de la même façon), des problèmes peuvent apparaître, liés au contraste, clarté, .... Deux contraintes viennent s'ajouter au tableau : le temps de calcul est deux fois plus long que pour une image ou une vidéo normale car il faut deux images au lieu d'une et les couleurs sont dénaturées à cause de l'utilisation de filtres.

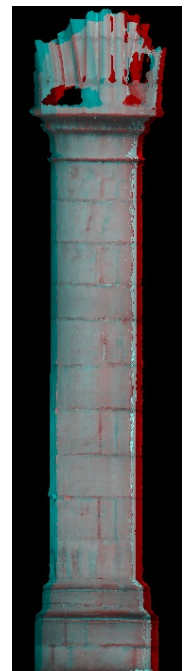


Figure 19 Anaglyphe



Figure 20 Réalité augmentée

#### Réalité augmentée

La réalité augmentée est l'intégration du virtuel dans le réel. Elle désigne les différentes méthodes qui permettent d'incruster de façon réaliste des objets virtuels dans un environnement réel. La technologie actuelle insère des images synthétisées/modèles 3D sur les images du monde réel au moyen de l'appareil photo d'un appareil mobile ou grâce à l'utilisation de lunettes spéciales. Largement utilisé par les pilotes de chasses pour l'ajout d'informations sémantiques liés à des objectifs, ou des reconnaissances de cibles, elle s'est démocratisée récemment publiquement grâce à l'usage des Smartphones ou des jeux vidéos (mais reste encore utilisée par une minorité). Elle nécessite une

<sup>16</sup> Conséquence d'un filtrage imparfait, ce sont des images parasites issues d'une séparation partielle des deux images homologues



cohérence absolue entre les deux flux réels et virtuels (notamment lors de mouvements de points de vue) pour maintenir un lien rigide entre les deux mondes : l'utilisation de capteurs/marqueurs pour assembler les modèles est généralement l'outil de prédilection (exemple Figure 20 **Erreur ! Source du renvoi introuvable.** par marqueur Augmented)

### Réalité virtuelle

Une visite virtuelle classique en temps réel est une visite interactive qui permet à l'utilisateur de se déplacer lui-même dans le modèle 3D. Il est libre d'explorer toute la scène considérée dans la limite prévue par le concepteur. Elle est dite « temps réel » car l'ordinateur effectue à chaque instant un rendu d'image sans attente. La création d'une visite de qualité nécessite des connaissances en programmation, en infographie, en design et d'autres domaines pour gérer les interactions, les effets, les déplacements et bien plus, suivant les compétences.

Son principal avantage réside dans la multiplicité des plates-formes disponibles pour la recevoir : Internet, Smartphones, PC, Consoles, tablettes, etc..

#### 2.3.2 Différents supports/formats de diffusion

L'utilisation et la diffusion des applications réalisées nécessitent l'utilisation d'un support de diffusion. Il existe différents formats de gestion de fichiers 3D (collada, ifc, ds, ...) gérés par des logiciels d'infographie et de CAO/DAO<sup>17</sup> (SketchUp Pro, Blender, Cinema 4D, Maya, ...) permettant l'échange de ces données 3D. Néanmoins, dans le cadre d'un usage archéologique basé sur un nuage de point, voici les possibilités de distribution :

#### Applications Android/IPhone/Windows/Consoles/...

Il s'agit d'applications suivant le support (.exe pour Windows, .apk pour android, ...) qui fonctionnent en accord avec le support de diffusion choisi. Ceci permet de s'affranchir des contraintes de poids des fichiers créés, et permet une utilisation en local (sans support internet). Ainsi, il est facile de créer des applications interactives qui permettent à l'utilisateur de visiter la cathédrale de Liège depuis son Smartphone, console de salon ou pc.

### Imprimantes 3D

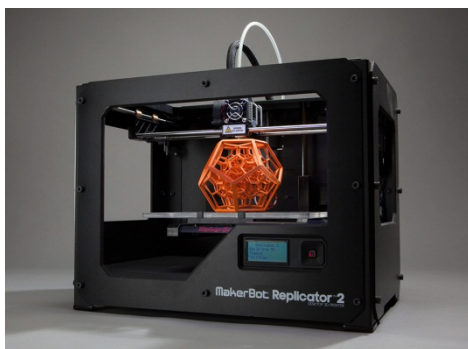


Figure 21 Imprimante 3D Markerbot

Les imprimantes 3D se multiplient sur le marché, devenant aujourd'hui vraiment accessibles. La diminution conséquente du prix d'une imprimante 3D (premiers modèles à 400€) et des matières premières utilisées (PVC, résine, ...) entraîne une démocratisation pour un usage public en explosion. Plusieurs start-up<sup>18</sup> ont senti le potentiel des imprimantes 3D et ont développé des produits intéressants pour la création d'un modèle réel à faible coût.

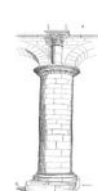
Cependant, bien qu'adaptée pour des modèles 3D de petites dimensions, la technique est assez compliquée à réaliser sur un modèle lourd. En effet, bien que le nombre important de polygones soit bien géré par l'imprimante, la nécessité d'avoir un maillage sans défaut est compliqué et chronophage. Un modèle doit :

- Ne présenter aucuns chevauchement de surface ;
- Être manifold;
- Être watertight (certainement la contrainte la plus compliquée à mettre en place pour de gros volumes);
- Être orienté (chaque face doit avoir sa normale à l'extérieur).

### PDF 3D

<sup>17</sup> Conception assistée par ordinateur/Dessin assisté par ordinateur

<sup>18</sup> Jeunes entreprises naissantes



Le format PDF (Portable Document Format) s'est imposé comme le format d'échange de référence devenant un standard international. Tous les PC sont équipés d'un logiciel capable de lire les fichiers PDF. Il est possible d'intégrer des éléments interactifs et 3D (notamment grâce au logiciel Adobe Acrobat Pro) permettant d'effectuer tout type de mesures et de manipulations sur le modèle 3D. Il est alors possible de visualiser des fichiers texturés (malheureusement les nuages de points ne sont pas pris en compte) très lourds, comprenant plusieurs millions de triangles, bénéficiant d'une compression avancée (fichier 80% fois moins lourd que le format .obj) et d'un support universel.

Il offre des perspectives de diffusion intéressantes, utile en archéologie comme dans d'autres domaines (industriel, retro-conception,...) pour la transmission aux clients de données manipulables et analysables sans connaissances préalables.

### Viewer

Les viewers sont des versions simplifiées des logiciels de traitements de nuages de point, de CAO/DAO, qui permettent une visualisation couplée à une utilisation basique (en lecture seule, sans édition possible) du logiciel de traitement complémentaire. On est dans le cadre d'une utilisation de bureau dite « desktop » par opposition aux clients web qui sont les navigateurs enrichis de plug-ins. Concernant les nuages de points, les plus courants sont Realworks Viewer, CloudCompare Viewer, Autocad Viewer, 3dReshaper Viewer, ...

### Web 3D (GoogleEarth, WebGL, Moteur plugin, ...)

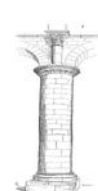
L'utilisation d'un support Web 3D diffère largement suivant les techniques employées et les méthodes mises en œuvre. Je vais différencier 4 cas :

- Google earth (également Plugin Web) : possède une des plus grandes banques de données 3D au monde. Néanmoins, un modèle créé à partir d'un scanner laser n'est pas adapté pour ce type de diffusion. Il créera un modèle trop lourd et trop complexe pour n'être seulement visible qu'en taille réduite. Les modèles implémentés sont des modèles simplifiés.
- Plugin Web : Il s'agit d'un ajout (add-on) qui est lié au navigateur web utilisé (Chrome, Firefox, Opéra, Microsoft Explorer, ...) et qui est nécessaire à l'utilisateur pour pouvoir lancer le programme créé. En plus de la contrainte logicielle, cette méthode pose le problème du poids du fichier. En effet, pour visiter le monument en temps réel, l'internaute doit charger les éléments présents dans la scène. Plus le poids du fichier est important, plus le temps de chargement est long. Il faut aussi penser que chaque internaute ne possède pas la même vitesse de connexion ni la même carte graphique. Par conséquent, cela impose une simplification à l'extrême du modèle 3D. Les plugins les plus utilisés sont : Unity 3D, ShockWave, Burster, ShiVa, ...
- JAVA et Flash : Ces deux technologies universelles sont généralement installées sur tous les ordinateurs pour gérer le flash et le Java (requiert la machine virtuelle Java qui exécute des programmes compilés). Néanmoins, ces deux formats ne peuvent gérer du 3D temps réel<sup>19</sup> seulement sur les modèles légers (3000 à 5000 polygones pour flash).
- WebGL : Ce format permet d'utiliser le standard OpenGL<sup>20</sup> depuis le code JavaScript (langage de programmation) d'une page web. C'est une technologie permettant le Web-3D tout en exploitant les accélérations matérielles 3D à l'aide des pilotes OpenGL de la carte graphique. Depuis 2009, plusieurs navigateurs utilisent nativement cette technologie (Chrome, Safari, Firefox) à très fort potentiel, mais les applications sont peu nombreuses actuellement. Elle permet un affichage en temps réel, sans temps de chargement de développements 3D. C'est la technologie utilisée pour réaliser la visite virtuelle de la cathédrale Saint-Paul sur les nuages de points : plus de 500 millions sans aucun temps de chargement et une dépendance seule de la carte graphique de l'utilisateur (grâce à l'OpenGL qui exploite les accélérations matérielles), sans impact sur la fluidité par gestion de LOD<sup>21</sup>.

<sup>19</sup> le rendu des données est immédiat en opposition à la 3D précalculée d'une animation 3D par exemple

<sup>20</sup> langage de programmation pour la conception d'application générant des images 2D et 3D.

<sup>21</sup> Level of Detail : Niveau de détail



### 3 PRÉTRAITEMENT & CONSOLIDATION

#### 3.1 L'obtention des données de la cathédrale

##### 3.1.1 Organisation structurelle de la cathédrale

Le levé lasergrammétrique de la cathédrale Saint-Paul implique une organisation liée à l'architecture, au schéma du bâtiment religieux et aux différents points d'appuis connus en coordonnées Lambert 72 (réseau dense obtenu par levé GNSS et tachéométrie de 25 points). Ainsi, les données obtenues sont découpées en 66 scans (nuages de points obtenus à partir d'une station) obtenus depuis 66 stations. Des relevés complémentaires ont été réalisés pour compléter les relevés effectués. Les données terrains se hiérarchisent de cette façon :



Figure 22 Niveau 0

**Niveau 0 :** Les trois plus importants scans au rez-de-chaussée ont été effectués au centre de la nef, à la croisée du transept et enfin au centre du chœur. Vu la distance supérieure à 20-25 mètres entre le scanner et les objets les plus éloignés, ces scans possèdent une résolution fine, et donc un plus grand nombre de points levés. La cathédrale Saint-Paul dispose de nombreuses chapelles. A l'intérieur de celles-ci, se retrouvent des œuvres et des sculptures remarquables. Pour chaque chapelle, de nombreux scans (entre 4 et 6) se sont révélés indispensables, nécessaires afin d'éviter les zones d'ombre. Les collatéraux nord et sud s'identifient également à des lieux où plusieurs scans sont indispensables.



Figure 23 Triforium

**Niveau 1 (Triforium) :** Les combles du niveau 1 ont présenté certaines difficultés pour le scannage des voûtes et de la charpente. L'étroitesse et la relative instabilité du plancher, l'accès difficile aux voûtes, la poussière ont été autant d'éléments perturbateurs pour la bonne pratique du scanner. La charpente est assez conséquente et imposante. Les bois la constituant ne sont pas droits, ce qui implique des points supplémentaires à mesurer. Cette voûte, vue depuis son sommet, est capitale car c'est elle sera l'objet d'une coupe demandée par les archéologues.



Figure 24 Niveau 2

**Niveau 2 :** Au niveau 2, le scannage n'est pas contraint par l'étroitesse des lieux. Placé sur un plancher relativement stable,

l'ensemble a pu être scanné rapidement.

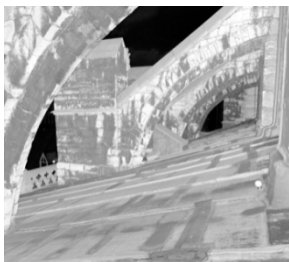


Figure 25 Le toit

**Le toit :** Le toit de la cathédrale est un autre endroit insolite où le scanner a mis en valeur sa rapidité et son adaptation à n'importe quel type de lieu. Avec la pente du toit, le trépied du scanner est réglé de façon à ce que l'instrument soit le plus horizontal possible. Les pieds en caoutchouc du trépied ont permis une très bonne adhérence au toit et ainsi une bonne stabilité. Le levé lasergrammétrique des arcs-boutants permet de les situer dans l'espace et de les modéliser afin d'en tirer un maximum d'enseignements.

**Le cloître :** Le cloître a été levé grâce à deux stations appuyées sur 3 points connus (21, 22 et 23). La façade nord intérieure du cloître était malheureusement en construction (échafaudage), ce qui empêcha la visibilité de détails intéressants les archéologues. Les deux scans ont été réalisés avec une résolution de 1/2



Figure 26 Le cloître

**La façade Sud :** La façade Sud constitue un levé complémentaire, et a demandé l'implantation de nouveaux points d'appuis (car certains points avaient été dégradés ou avaient disparus). Ainsi, nous avons créé une station libre par tachéométrie à partir de 4 points d'appuis, qui nous aura servi pour placer une sphère pour le recalage absolu des 3 scans effectués pour lever la façade. Étant un lieu public très fréquenté, nous avons dû déterminer des positions optimales pour limiter le bruit dû au passage de voitures et de passant.



Figure 27 Façade Sud

### 3.1.2 Le levé terrain de la cathédrale : méthodologie et solutions



Figure 28 FARO FOCUS<sup>3D</sup>

Au regard de la comparaison des différents types de scanners lasers et de l'application considérée (cf. II -1 -B -), le choix d'un scanner 3D à différence de phase est tout à fait judicieux : le scanner FARO FOCUS<sup>3D</sup> (Voir Annexe 1)

La cathédrale de grandes dimensions demande une portée conséquente pour un recouvrement et un levé optimal, et demande parfois des stations dans des lieux étroits et fragiles (combles, toits, ...). Avec une portée de 0,6 m à 120 m, et une compacité bluffante (240 x 200 x 100 mm pour 5 kg, trépied en fibre de carbone rétrécissable), le levé du grand hall autant que l'utilisation dans des endroits exigus et/ou instables ne pose aucuns problèmes, et permet d'apprécier grandement la polyvalence et la simplicité de mise en station du FARO FOCUS<sup>3D</sup>.

De plus, étant donné qu'il s'agit d'un lieu public et afin de limiter le bruit de mesure, nous avons besoin du système d'acquisition le plus rapide possible. Pouvant mesurer jusqu'à 976 000 points /s avec un bruit de 0.60 mm à 90% de réflectance sur donnée brutes, le passage de voitures (en extérieur), visiteurs, ou les imprécisions de mesures dû à une réflectance insuffisante produiront un bruit limité. Le FARO FOCUS<sup>3D</sup> a un champ de vision de 305° (vertical) x 360° (horizontal), et permet ainsi de relever une scène entière par une rotation de 180° autour de son axe vertical.

Bien entendu, la précision est également un critère de choix pour relever certains détails inférieurs à 5 mm pour les archéologues. Avec une précision annoncée de  $\pm 2$  mm (à 10m), et grâce à une interface intuitive qui permet de régler le pas de balayage<sup>22</sup> limité par la résolution angulaire<sup>23</sup> de 0.009°, soit une résolution spatiale<sup>24</sup> de 8 mm à 100 m. Ainsi, suivant le pas de balayage défini (1/1, 1/2, 1/3, ...) et le X-Control (Nombre de fois où chaque point sera relevé), la durée (et la précision) du scan sera modifiée, et peut varier de quelques minutes à une heure (15 minutes pour un pas de 1/2 et un X-Control de 3). Tout ceci conditionne également la densité du nuage de point obtenu, dont la trame sera de plus en plus allongée et déformée à mesure que l'angle d'incidence du faisceau laser augmente (et parfaitement alignée lorsque la surface est perpendiculaire au laser).

D'un point de vue de mise en pratique, le scanner laser FARO FOCUS<sup>3D</sup> est équipé d'un compensateur à axe double, qui avant chaque numérisation vérifie s'il se situe dans une plage de  $\pm 5^\circ$  par rapport à l'horizontale, et compense en fonction avec une précision de 0,015°. Son baromètre électronique permet de calculer sa hauteur relative par rapport à une valeur de référence et la présence d'un compas électronique permet de s'orienter par rapport aux points cardinaux.

A l'arrivée sur le terrain, nous disposons de plusieurs points géoréférencés grâce à des mesures par station totale. Le scanner est fourni avec plusieurs cibles sphériques de différentes tailles (rayon de 37 mm et 72,5 mm) qui feront office de cible pour tout recalage absolu ou relatif (cf. p.26).

Pour obtenir un levé de qualité, il est nécessaire de respecter certaines consignes concernant l'utilisation du laser scanner. Je vais donc développer une méthodologie qu'il est utile d'avoir en tête lorsque l'on souhaite réaliser un levé 3D. Elle est issue de différents tests réalisés, et de notre expérience personnelle de l'utilisation du FARO pour obtenir les nuages de points de la cathédrale Saint-Paul.

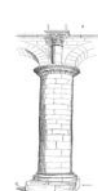
On est amené à scanner la cathédrale à partir de points de vue multiples pour plusieurs raisons :

- La limitation du champ de visibilité du scanner,
- Les contraintes de recouvrement des scans,
- Les dimensions de la scène,
- Les conditions de visibilité susceptibles de masquer des parties capitales de l'objet,
- L'intérêt de scanner les différentes parties de l'objet en adoptant une résolution qui varie en fonction de sa complexité et de ses détails.

<sup>22</sup> Espacement entre 2 points successifs choisi par l'utilisateur avant de lancer le balayage laser

<sup>23</sup> Capacité de l'appareil à mesurer de façon indépendante deux objets situés sur des lignes de visées adjacentes

<sup>24</sup> Plus petit espacement entre deux points successifs que l'on puisse mesurer à une distance objet/scanner fixe





- Il s'y ajoute les conditions de chantier et les temps nécessaires au scannage.

La méthode opérationnelle mise en place ici sera alors celle de la station libre. Les coordonnées d'au moins une station seront calculées grâce à un relèvement à partir d'un minimum de 3 points géoréférencés (une surabondance permettra un contrôle appréciable). Ceci assurera un référencement absolu dans un système de coordonnées adapté aux législations et applications proposée (Lambert 72 pour la Belgique).

Pour que tout le levé soit intégralement géoréférencé, il nous faut procéder à un recalage relatif. Pour cela, chaque station doit posséder au minimum 3 cibles communes (Sphères) avec un nuage de point obtenu d'une autre station. Plus il y aura de points homologues, plus le recalage sera robuste et moins nous serons soumis à des problèmes nous obligeant à revenir sur le terrain. Lorsque nous avons cela en tête, nous pouvons alors intelligemment placer le dispositif qui nous permettra un géoréférencement indirect de qualité : un réseau de sphère éparpillé de manière à maximiser la redondance de points homologues.

Le choix des stations doit également être judicieux. En plus de cette contrainte de visibilité des sphères, il faut placer les stations de manière à limiter les zones d'ombres<sup>25</sup>, et de façon à ce que l'angle d'incidence<sup>26</sup> soit le plus faible possible.

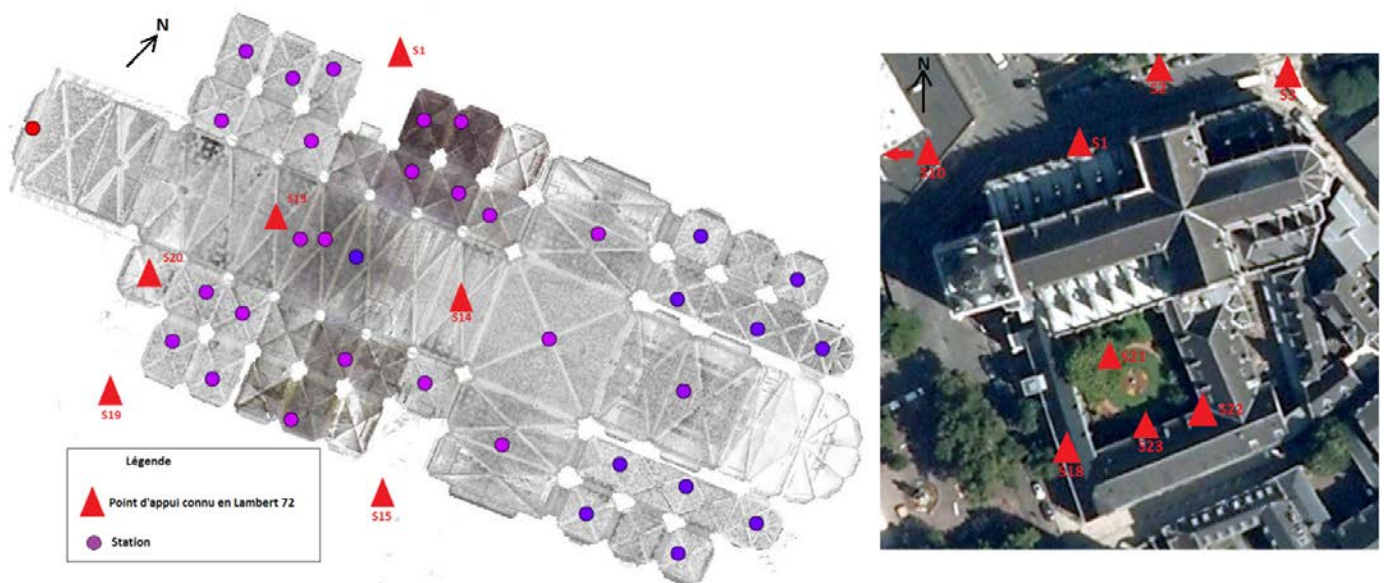


Figure 29 Localisation du réseau de points d'appuis et stations de scans réalisées à l'intérieur/extérieur de la cathédrale

Du point de vue de la mise en pratique, voici quelques recommandations pour un gain de temps considérable lors de levés par scanner laser 3D

- Établir une stratégie de mise en place du réseau selon les différentes contraintes identifiées
- Préparer un plan de situation avec les différents points de rattachement possible
- Même si mathématiquement 3 points homologues suffisent à obtenir un recalage entre nuages, il est utile et recommandé de placer un maximum de références externes (sphères/damiers), réparties autour de la station à différentes hauteurs
- Positionner l'origine du scan de façon à ce que la vue planaire soit centrée sur la partie la plus intéressante et s'aider de prévisualisation à basse résolution pour limiter le volume des données.
- Effectuer un croquis pendant le scanning identifiant tous les points placés, hauteurs respectives et remarques.
- Vérifier le formatage de la carte SD d'enregistrement en fonction du volume des données demandées par le scan.
- Choisir des périodes horaires adéquates pour limiter le bruit dû au flux de personnes.
- Adapter la résolution de scanning au niveau de détail souhaité : tenir compte de la distance objet-station, de l'angle d'inclinaison entre le faisceau incident et la surface numérisé, de la taille du plus petit objet numérisé et de la complexité du relevé

<sup>25</sup> Zones non visibles par le scanner laser qui ne seront pas relevées

<sup>26</sup> angle entre la direction de propagation de l'onde incidente (laser) et la normale à la surface considéré.

### 3.2 Les suites logicielles de traitement de nuages de points

Une fois l'ensemble des données de la cathédrale réuni, il faut définir les logiciels de traitement utilisés. La gestion des nuages de point et l'ensemble des traitements de tels volumes demandent l'utilisation de logiciels performants et complets. Malheureusement, relevant d'une technologie émergente, les différents logiciels existants ne proposent pas tous un service adapté couvrant l'ensemble de la chaîne de traitement. J'ai ainsi utilisé et réalisé une série de tests et comparatifs de différentes suites logicielles pour déterminer leur pertinence, leur efficacité, leur polyvalence, leur performance et leur simplicité.

Voici donc le comparatif résumé des qualités et faiblesses de chaque logiciel comparé (voir Annexe 2 pour description exhaustive de chaque logiciel) suivant 10 points considérés importants pour leur utilisation :

	SCENE	Realworks	3DReshaper	CloudCompare	MeshLab	Autodesk
Prix	0	0	0	2(Gratuit)	2(Gratuit)	0
Stabilité	0(Instable)	1	1	2	0(Instable)	1
Interface	0	2	2(	2	1	2
Nettoyage	1(filtres)	1	1	1	0	1
Consolidation	2(recon.)	2(recon.)	0	0	0	1
Segmentation	1	1	1	2	0	0
Modélisation	0	0	2(gros volumes)	0	2	0
Texturisation	0	1	1	0	2(coul.RGB)	0
Technicité	0	0	0	2(algorithmes)	2(algorithmes)	0
Spécialité	Consolidation	Généraliste	Modélisation	Analyse/Édition	Modélisation	Levé façade
	4	8	8	11	9	5

0 : Médiocre      1 : Bon      2 : Très bon

Pour la réalisation du TFE, j'ai dans l'ordre utilisé les logiciels SCENE, CloudCompare et MeshLab.

SCENE est un logiciel constructeur de la société FARO, et permet le traitement de nuages de points en étant spécialisé dans la consolidation des données issues du scanner FARO FOCUS<sup>3D</sup> (format .fls). Une fois consolidé, il permet d'exporter dans différents formats (.xyz, .asc, .ptc, ...) pour une utilisation tiers<sup>27</sup>.

CloudCompare (open-source et gratuit) est très efficace pour le traitement de gros nuages de points, et pour les manipulations, modifications et l'analyse de nuages de points. Technique et efficace, toutes les fonctions proposées citent l'algorithme utilisé et les outils statistiques offrent une analyse étendue. Étant donné le manque de formation sur ce logiciel, j'ai créé un tutoriel descriptif de prise en main (voir Annexe 3).

MeshLab (open source et gratuit) est un logiciel d'édition et de création de maillage proposant des outils extrêmement puissants pour la création de maillages et géométries à partir des nuages de points (format .ply, .xyz, .asc ...). Très technique et avancé, il est difficile d'accès pour les non-initiés mais les possibilités de nettoyage, échantillonnage, maillage, texturation et de mapping en font un outil de choix (malheureusement terni par la présence majeure de bugs).

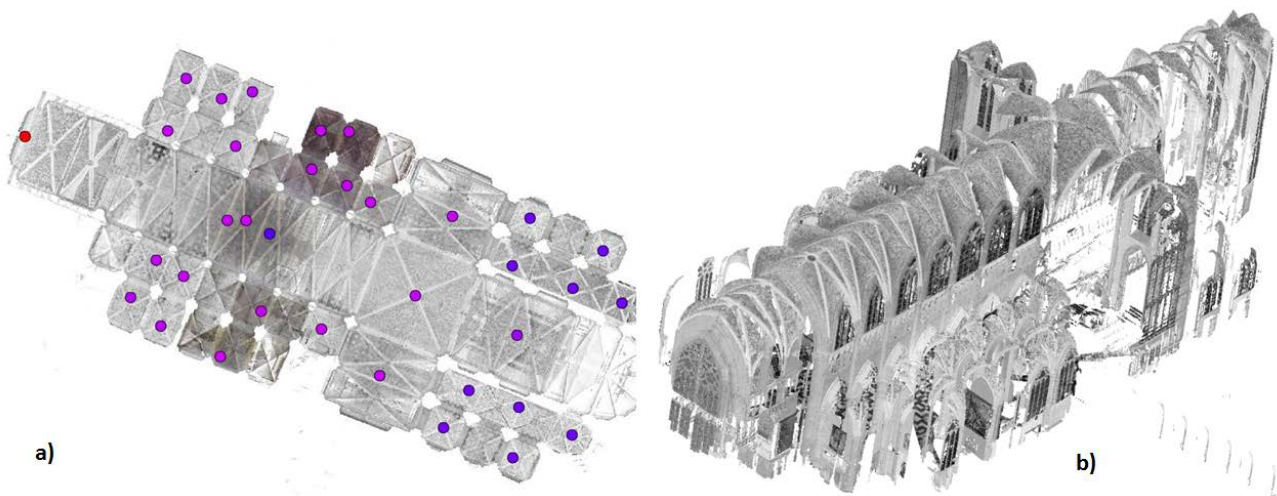


Figure 30 Nuage de point partiel de la cathédrale traité dans SCENE en a) Vue du dessus et b) Vue isométrique

<sup>27</sup> par un autre logiciel

### 3.3 Le recalage de la cathédrale

Pour pouvoir utiliser les données et recalculer l'ensemble des scans, il nous faut dans un premier temps utiliser le logiciel constructeur FARO SCENE. L'importation des données se déroule de manière automatique en créant un projet et en renseignant les scans (fichier .fls) nécessaires. Dans notre cas, les plus petits fichiers de scan avaient une taille d'un peu plus de 100 Mo et les plus importants dépassaient les 2 Go. Une fois les nuages importés, il est nécessaire de les prétraiter pour ne travailler que sur des données correctes. Pour cela, il faut corriger l'influence sur les mesures de différents facteurs :

- A partir d'un certain nombre de points, le résultat de la mesure déviara toujours de la valeur réelle. Cette déviation provient en général du bruit. Le filtre de lissage corrige en parti ce phénomène en remplaçant la valeur mesurée du point de scanning par la valeur moyenne de sa zone avoisinante
- Le point du laser du scanner n'est pas infiniment petit, mais il a un certain diamètre. Le point du laser peut donc tomber partiellement sur un support et partiellement sur le mur situé derrière ce support. Le scanner reçoit alors la lumière laser réfléctée de deux points très éloignés l'un de l'autre. Le filtre de points perdus permet de corriger ce phénomène et supprimer ces points faux.
- Parfois, les résultats de mesures ne peuvent être que dans l'intervalle d'ambiguïté. Cependant, si la distance réelle d'un objet est supérieure à cet intervalle d'ambiguïté, une valeur trop petite est affichée en tant que résultat de mesure. Les filtres de distance de base (pour retirer des points de scanning dans une plage de distance du scanner) et de points de scanning obscurs (pour éliminer les points dont la valeur de réflexion est trop faible, source de bruit) corrigent ce problème.



Figure 31 Cible Damier

La consolidation peut alors être réalisée de manière semi-automatique ou manuelle en utilisant deux types de cibles : Les cibles naturelles et les cibles dites artificielles. Les cibles naturelles sont des points et objets naturels qui font partie de l'environnement scanné (plans, angles, arrêtes, rectangles, ...). Leur utilisation est chronophage et produit des résultats imprécis ou de qualité médiocre. Leur utilisation est donc limitée au cas où le nombre de cibles artificielles n'est pas suffisant pour effectuer un recalage de qualité.

Les cibles artificielles sont soit des damiers (Besoin d'un minimum de 4 points par quadrant pour être détecté, et un angle d'incidence du faisceau laser de 45° minimum) soit des sphères (propriétés de réflectances optimisées pour SCENE, bonne homogénéité des résultats dans toutes les directions, nécessite au moins 60 points scannés pour être détectées). Elles sont utilisées en priorité pour recalculer les différents scans dans un seul système de coordonnées, car elles permettent la meilleure précision, et ouvrent sur consolidation automatisée par détection automatique.



Figure 32 Cible Sphère

Une fois les différents fichiers chargés, nous pouvons commencer la phase de consolidation. Les différentes vues proposées par SCENE permettent de s'assurer des données acquises scan par scan, et permettent de sélectionner, nettoyer ou supprimer les zones bruitées. Puis, à l'aide de la vue planaire les sphères sont identifiées manuellement par pointage, ou par attribution des coordonnées X, Y et Z de chaque sphère au moyen d'un fichier .csv. Un petit feu tricolore à côté de l'identifiant de la sphère indique la qualité du scannage de la sphère. Le feu sera au vert si la sphère a été correctement scannée et cette dernière sera alors utilisée en priorité pour le recalage ou la fusion des scans. La couleur orange signifie que la qualité n'est pas la meilleure, ainsi le recours à la sphère sera utilisé accessoirement pour l'opération de recalage. Enfin, la couleur rouge indique tout simplement la carence de points pour modéliser la sphère et calculer son centre. Elle ne sera donc pas prise en compte pour le recalage ou la fusion de différents scans. Pour les cibles en damier, l'opération est exactement identique.

Ensuite, après avoir repéré toutes les sphères dans chaque scan, le lancement de l'opération de fusion de scans (recalage relatif) s'opère. Elle consiste à faire correspondre la géométrie formée par les sphères d'un scan avec la géométrie d'un second scan. Le logiciel analyse la disposition des sphères entre elles et cherche la meilleure correspondance avec le second scan.

Cependant, cette opération de fusion des scans en analysant la géométrie décrite par les sphères ne peut parfois se réaliser (nombre de sphère insuffisant, mauvaise transformation, attribution incorrecte de sphère, ...). L'opérateur peut alors forcer ce recalage en attribuant une correspondance par nom de sphère à une même sphère dans plusieurs scans, forçant l'homologie. Disposant des mêmes coordonnées, le logiciel va pouvoir directement les associer lors de la fusion.

Lorsqu'elle est réalisée de manière automatique, cette opération est très sensible et mal aboutie. Elle produit notamment des incohérences de précision et de tensions qui ne correspondent pas aux précisions de levés. En sus, certaines sphères n'ont pas les mêmes hauteurs sur différents scans (car le levé sur 3 jours et certains vols de cibles sphériques ont entraîné un changement de hauteur des sphères appariées ensemble) qui entraîne une incohérence du recalage automatique (l'utilisation de plans ou surfaces géométrique n'améliore pas ces résultats, au contraire). C'est pourquoi il nous a fallu affiner le recalage manuel par d'autres méthodes et logiciels, pour atteindre des précisions millimétriques.

Pour le recalage absolu, il est nécessaire d'importer les points connus qui ont été stationnés par des sphères en tant que référence externe via un fichier ASCII, pour être ensuite utilisé par le logiciel pour transformer les coordonnées du système en système Lambert 72. Celui-ci contient l'identifiant du point, sa coordonnée en X, en Y et en Z, et le poids relatif de ce point. La dimension des sphères de références y est également mentionnée (dans notre situation, le poids relatif de chaque point de station est identique vu que les coordonnées avaient été ajustées par la méthode des moindres carrés).

C01 1019.8758 987.3415 126.168 1  
 C02 1009.4419 970.165 125.743 1  
 C03 1018.5553 967.4227 126.014 1  
 C04 997.9875 976.5491 126.246 1

Nom1	Nom2	Erreur
<input checked="" type="checkbox"/> TARG...	S21	0.71 mm
<input checked="" type="checkbox"/> TARG...	S22	2.01 mm
<input checked="" type="checkbox"/> TARG...	S23	2.30 mm
Erreur moyenne		1.68 mm

Figure 33 Erreur de géoréférencement sur 3 points d'appuis

Le logiciel SCENE est cependant indispensable pour coloriser les points par correspondance et projection matricielle (photos prises par l'appareil équipé d'une caméra couleur de haute résolution, compressées dans le fichier .fls). Une erreur d'un pixel non colorisé en raison d'un petit décalage entre la photo et le nuage de points peut survenir. Le logiciel possède une option qui permet d'améliorer le contraste ; ainsi, les zones sombres deviennent plus claires et permettent d'observer certains détails discrets. L'augmentation du contraste en étendant l'histogramme de l'image s'avère parfois nécessaire pour distinguer plus facilement les cibles en damiers.

Erreur d'ajustement	Distance au Scanner
0.94 mm	17964.13 mm
0.72 mm	12900.11 mm
2.39 mm	17023.83 mm
0.64 mm	10914.85 mm
0.61 mm	9494.71 mm
0.63 mm	13362.83 mm
0.75 mm	13778.54 mm
0.70 mm	12965.10 mm
0.86 mm	18864.92 mm
1.36 mm	27717.62 mm

Figure 34 Extrait du rapport d'erreur issu de Realworks

Ainsi, nous avons recalé l'ensemble de la cathédrale et regroupé les différentes parties à l'aide de SCENE dans un premier temps (voir Annexe 4 pour rapport détaillé des opérations et résultats). Les tensions (qui ne peuvent pas être considérées en unités classiques de distances) sont faibles et montre une consolidation de qualité, mais l'analyse de certains morceaux de la cathédrale dévoile des erreurs de recalage d'environ 2 cm (manque de sphère homologues, ...). Ainsi, ces parties problématiques ont été isolées et recalées à l'aide de REALWORKS et MATLAB (voir Annexe 5) pour une plus grande maîtrise de l'étape. Au final, nous obtenons un recalage relatif dont la distance entre deux sphères n'excède pas plus d'un centimètre (en moyenne 3 mm), et dont l'écart maximal entre deux nuages est d'environ 2 cm. Le recalage absolu quant à lui dévoile une erreur millimétrique.

La cathédrale est alors recalée et nous pouvons exporter le fichier au format ASCII (.xyz, .asc, ...) pour continuer les traitements.

## 4 SEGMENTATION ALGORITHMIQUE

### 4.1 Évolution et cheminement conceptuel

Le développement final de l'algorithme est le résultat d'une longue réflexion qui a évolué tout au long de ces 5 derniers mois. La définition originelle du projet est le point de départ d'une évolution progressive vers une méthode finale, riche des apports conceptuels et pratiques de ces premières pensées.

Nous voulions définir une approche de segmentation/modélisation appliquée à des colonnes pour un usage archéologique. A partir d'un nuage de point segmenté grossièrement pour écarter différentes colonnes, l'objectif était de développer une méthode simple à mettre en œuvre et efficace pour détecter les différentes variations de rayon d'une colonne pour segmenter en conséquence et modéliser chaque partie par une forme élémentaire. Les travaux de (Luo & Y. Wang, 2008) et (Chaperon & Goulette, 2001) s'attachent sur une problématique similaire en détectant les piliers d'une scène industrielle. Également, (Canciani *et al.*, 2013) proposent une méthode de modélisation géométrique complexe et manuelle des colonnes de l'église « San Carlo alle Quattro Fontane » (Rome – Italie) après une extraction automatique et dessins de profils depuis le nuage de points.

Après quelques semaines de recherche et de mise en pratique, je suis arrivé à un résultat satisfaisant basé sur une pensée conceptuelle assez simple. Le nuage de point est « scanné » par tranches (coupes d'épaisseurs définies par l'utilisateur), et tous les points inclus dans cette tranche sont projetés sur le plan moyen (voir Figure 35 Figure 35). Chaque tranche est alors comparée de proche en proche, et si le cercle des moindres carrés calculé (les projections issues des sections de colonnes analysées forment des cercles) a des propriétés similaires avec le précédent, alors la tranche correspondante est ajoutée à la région en cours. Lorsqu'une variation trop importante intervient, la région est terminée, définissant une partie segmentée, et le scan continue jusqu'à ce que tout le nuage soit segmenté. Par la suite, un algorithme de reconnaissance des formes est appliqué, reconstruisant paramétriquement chaque morceau segmenté par une forme géométrique (Cône ou cylindre).

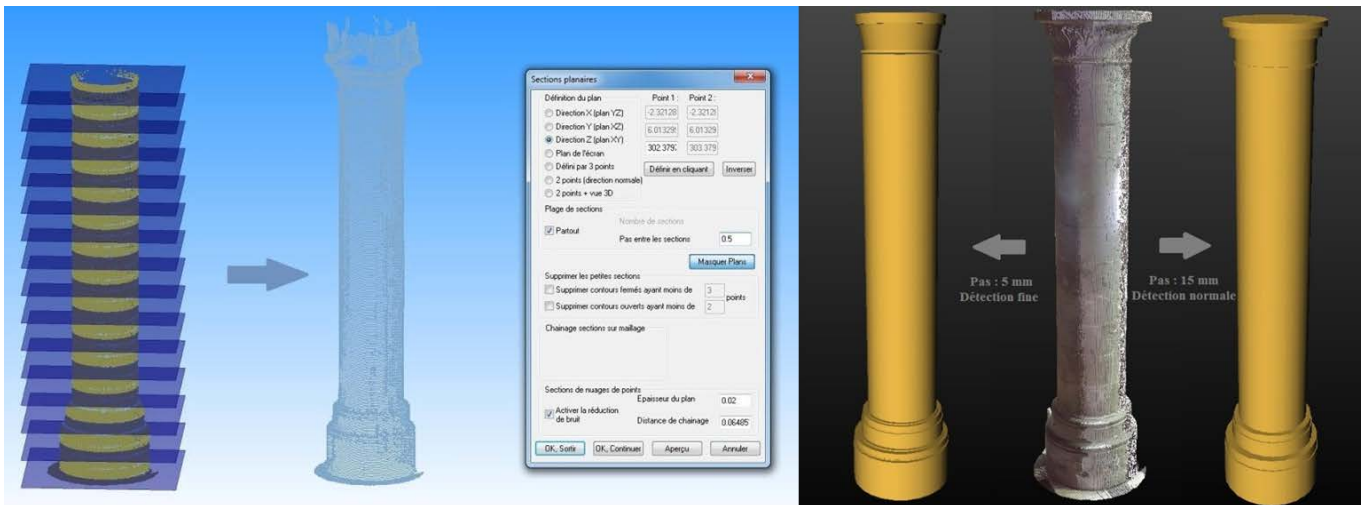


Figure 35 Réalisation des sections transversales (gauche) et modélisation paramétrique (droite) appliquée à une colonne

Bien que l'algorithme soit efficace et que l'on obtienne plusieurs niveaux de détails suivant la définition des paramètres en entrée, le principal problème de cette solution concerne l'adaptabilité. En effet, un tel algorithme suppose une connaissance à priori de l'objet à modéliser, et ne permet une modélisation que dans le cas d'objets circulaires. De plus, il suppose une segmentation préalable et ne permet de traiter que des objets isolés. De ce fait, le problème d'expansion et d'utilisation d'un tel algorithme se pose. Nous voudrions quelque chose qui soit applicable à d'autres cas, et permette de détecter et de reconstituer plusieurs formes, sans connaissances préalables.

C'est pourquoi nous avons pensé à l'utilisation des « Convex Hull » (Enveloppes convexes), qui permettraient une détection des ruptures de pente et changement de direction, selon différents niveaux de détails en fonction de l'attente de l'utilisateur. Une enveloppe convexe sélectionne l'ensemble convexe minimal qui contient le nuage de points. Pour comprendre le fonctionnement, faisons un raisonnement en 2D. Dans un premier temps le nuage de points dans sa globalité est sélectionné. L'enveloppe convexe est calculée :

$$\sum_{i=1}^n \alpha_i x_i \text{ tel que } (\forall i : \alpha_i \geq 0) \wedge \sum_{i=1}^n \alpha_i = 1$$

Elle nous donne des nouveaux points caractéristiques de rupture de pente. Par sélection itérative de chaque nouvelle enveloppe associée à chaque nouveau segment, le modèle peut être affiné pour s'approcher de la forme réelle en déterminant de nouveaux points de rupture :

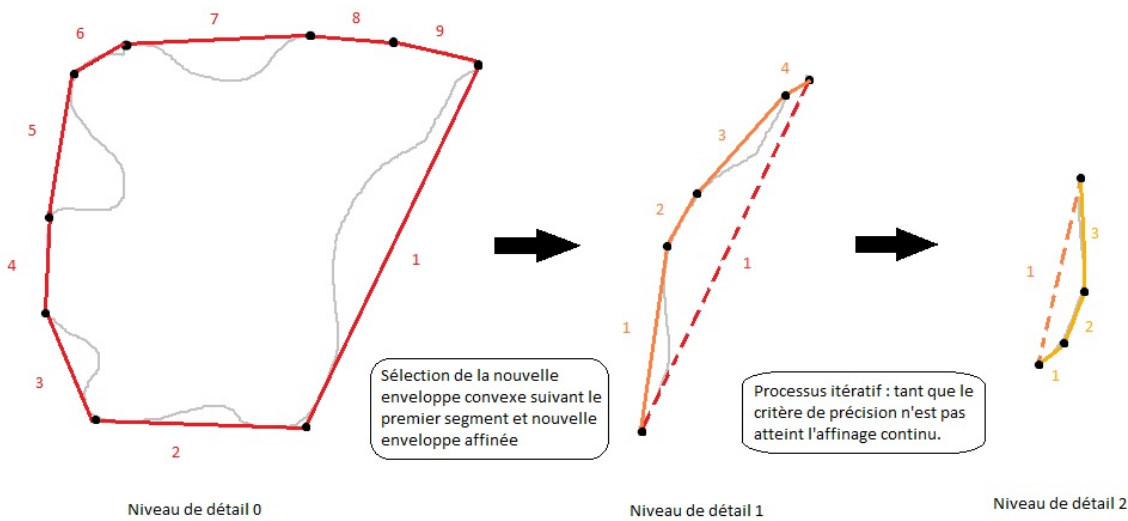


Figure 36 Principe théorique de l'approche par enveloppes convexes

Ce système nous permettrait d'obtenir rapidement tous les changements de direction et ruptures de pentes en fonction du niveau de détail désiré. La réflexion en partant du grand pour aller vers le petit offre cette possibilité de gérer des niveaux de détail par la suite. Chaque segment résultant serait constitué de tous les points compris entre deux sommets de l'enveloppe. Cependant, le passage en 3 dimensions est problématique. En effet, une telle approche est seulement adaptée pour des nuages de points parfaitement nettoyés, et ne concernant qu'un seul objet. Dans le cas contraire, nous aurons des lignes aberrantes et points incohérents, rendant la segmentation future illisible. De plus, il y a un problème concernant les formes sphériques et cylindriques. La formation de régions sera trop importante et nous aurons des zones sur-segmentées.

C'est alors que j'ai pensé à raisonner par normales. Chaque point aurait une normale déterminée grâce à un plan moyen basé sur le voisinage du point, et on regrouperait selon un critère de similarité les normales (et donc les points) entre elles. Ainsi, toutes les contraintes ou problèmes posés par les différentes méthodes disparaissent. Une scène de plusieurs objets différents pourra être traitée sans problème, où seuls quelques paramètres changeront pour être adapté à l'application voulue.

Je vais maintenant détailler la théorie développée sur laquelle l'algorithme créé a été construit.

## 4.2 Principe théorique

### 4.2.1 Définition et focalisation

Avant de démarrer la création d'un algorithme répondant à mes objectifs, je me suis basé sur l'existant, et j'ai réalisé différents tests. Ainsi, tous les problèmes identifiés dans la partie II.C ont constitué le moteur de mes recherches. J'ai également analysé certains travaux réalisés sur la segmentation par normales de (Rabbani, 2006) et (Holz & Behnke, 2013) qui révèlent une forte variance suivant l'approche réalisée, et les résultats obtenus. Aucune des méthodes ne permet une polyvalence et une précision adaptée tant aux formes simples qu'aux formes complexes. Bien que ces méthodes soient très efficaces dans un domaine, elles sont souvent limitées dans l'utilisation et ne permettent pas un grand champ d'application.

Un débruitage préalable est souhaitable mais pas indispensable. Il permettra une estimation plus fidèle des normales. De plus un nuage de point propre et recalé avec précision permettra d'éviter des erreurs de calcul de

normales, qui peuvent dévoiler un changement de direction brusque ou des orientations aberrantes si le fichier initial n'est pas propre (ce qui induit une segmentation faussée, voir ci-dessous)

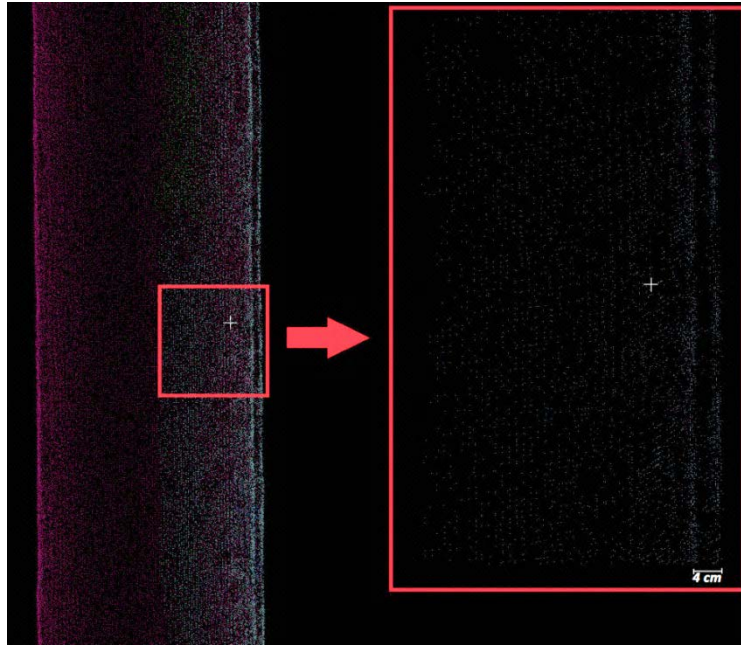


Figure 37 Influence d'un problème de recalage sur la segmentation

#### 4.2.2 Le calcul des normales et l'estimation de courbure

Bien que différentes méthodes d'estimation de normales existent (voir (Klasing & Althoff, 2009) pour des comparaisons de méthodes d'estimation de normales), je vais me concentrer sur l'une des plus simples et efficaces, définie en 1994 par (Berkmann & Caelli, 1994).

Le problème est donc de déterminer la normale à un point sur une surface définie par estimation de la normale au plan tangent à la surface, ce qui revient à résoudre un problème de moindres carrés.

A partir des points voisins  $p^k$  du point source  $p_s$ , nous pouvons estimer une forme locale qui prend en compte la géométrie de l'échantillon sous-jacent autour de  $p_s$ .

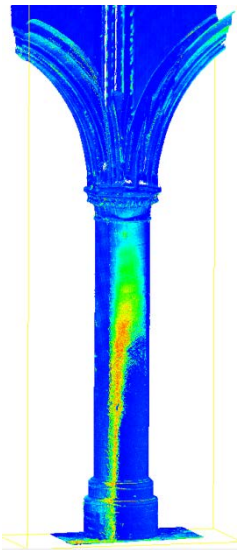


Figure 39

Classification selon la courbure

Le plan  $P^k$  est représenté par un point  $x$  et un vecteur normal  $\vec{n}$ , et la distance d'un point  $p_i \in P^k$  est donnée par :

$$d_i = (p_i - x) * \vec{n}$$

Ainsi, nous pouvons obtenir les valeurs de  $x$  et  $\vec{n}$  par moindres carrés en essayant de minimiser  $d_i$ .

Pour cela, si l'on prend le centroïde  $x$  de  $P^k$  tel que :  $x = \bar{p} = \frac{1}{k} * \sum_{i=1}^k p_i$ , la solution pour  $\vec{n}$  est trouvée en analysant les valeurs propres et vecteurs propres de la matrice de covariance  $C$  telle que :

$$C = \frac{1}{k} * \sum_{i=1}^k (p_i - \bar{p}) * (p_i - \bar{p})', C * \vec{v}_j = l_j * \vec{v}_j, j \in \{0,1,2\}$$

Avec  $k$  le nombre de points voisins considérés dans le voisinage de  $p_i$ ,  $\bar{p}$  représentant le centroïde 3D des plus proches voisins,  $l_j$  étant les valeurs propres,  $\vec{v}_j$  étant les vecteurs propres.

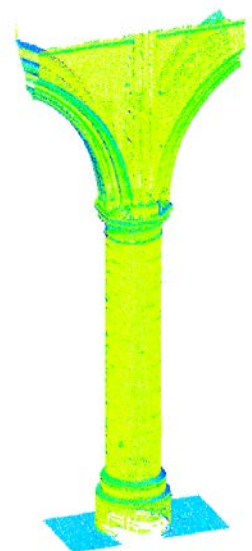


Figure 38 Classification selon les normales

L'estimation de la courbure  $H$  découle alors immédiatement de l'estimation de la normale à la surface. Elle est estimée en tant que relation entre les valeurs propres de la matrice de covariance  $C$  :

$$H = \frac{l_0}{l_0 + l_1 + l_2}$$

### 4.2.3 La segmentation

La méthode de segmentation mise en place s'appuie sur l'utilisation des normales. Elle s'articule sur une organisation en deux mouvements distincts. Dans un premier temps (segmentation 1), la méthode regroupe par un processus de croissance de région tous les points qui ont la même normale. Pour limiter l'influence du bruit ou des outliers<sup>28</sup> tout en gardant une homogénéité dans le groupement effectué, il faut définir un critère de similarité correct. En effet, la croissance de région réalisée part d'un point source  $i$ , et va tester ce critère par comparaison des normales aux voisins, choisis dans un rayon autour de ce point. Ainsi le groupement se fera si la norme de la différence entre la normale au point  $j$  et le vecteur normal moyen de la région en cours de formation est inférieur au critère de similarité, soit :

$$\sqrt{\begin{pmatrix} N_x(j) & N_{rx} \\ N_y(j) - N_{ry} \\ N_z(j) & N_{rz} \end{pmatrix}^2} \leq \sqrt{\left( \text{similarité} * \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} \right)^2}$$

Si le critère est satisfait et que le point  $j$  voisin du point  $i$  source est ajouté à la région, alors il est sélectionné comme point source, et la même opération est réalisée jusqu'à ce que la liste des points voisins soit vide (qu'il n'y ai plus de voisins n'appartenant à aucunes régions). Le résultat de la région est alors ajouté à la matrice finale  $R_f$  de la segmentation 1. Lorsque tous les points ont été traités, la segmentation s'attaque à la deuxième partie.

A partir du résultat de la segmentation 1, j'ai décidé d'étudier les relations (connectivité, angle, taille, symétrie, géométrie) entre chaque régions et ses voisines pour les regrouper en segments finaux (segmentation 2). L'Annexe 6 présente schématiquement le fonctionnement de l'algorithme ainsi que le script de la segmentation 2.

En plus d'être une méthode ouverte à des applications très diverses, j'ai construit l'algorithme de sorte qu'il utilise en entrée des paramètres qui ont un sens pour l'utilisateur. Ainsi, il s'agit d'une segmentation semi-automatique qui demande à l'utilisateur d'entrer 3 valeurs :

- Le critère de similarité : différence entre deux normales. Plus la valeur tend vers 0, plus le nuage de point sera segmenté (selon la qualité du nuage de point, choisir une valeur entre 0 et 1)
- Le rayon de recherche : Parce que la segmentation est basée sur la recherche de voisins selon une distance euclidienne, l'utilisateur est sollicité pour entrer une valeur qui constituera le rayon de recherche.
- Le détail minimum extrait : L'utilisateur rentre la valeur du détail minimum souhaité. Ce paramètre influe la seconde partie de la segmentation en adaptant une coupe pour limiter la recherche de voisins selon la symétrie, et ne dépassera pas la valeur entrée.



Figure 40 Nuage de point brut (gauche), segmentation 1 (milieu), segmentation 2 (droite)

<sup>28</sup> Points aberrants



## 4.3 Insertion dans l'environnement logiciel

### 4.3.1 Les besoins de l'algorithme et les solutions logicielles

Afin de développer la méthode de segmentation proposée, il faut choisir un outil de développement adapté. Plusieurs solutions étaient possibles, guidées par les besoins de l'algorithme et mes connaissances personnelles en programmation. Étant basé sur des calculs matriciels, j'ai décidé d'évoluer dans un environnement efficace pour faire des mathématiques, des simulations, et proposant une bibliothèque avancée pour le calcul scientifique : MATLAB.

Bien qu'étant un langage interprété assez « brut » d'un point de vue programmation, MATLAB m'a particulièrement attiré pour son approche recherche permettant de retranscrire logiquement le cheminement mathématique réfléchi, et pour sa proximité au logiciel maîtrisé SCILAB (inspiré de MATLAB). Il permet de décrire certaines opérations de manière non procédurale et permet d'obtenir rapidement des résultats à partir de courts programmes.

J'ai cependant longtemps hésité avec le développement en C++ ou en Python, qui gèrent mieux le travail sur d'importants volumes de données, et qui peuvent utiliser la bibliothèque Point Cloud Library ainsi que les bibliothèques de calcul et visualisation scientifiques pour python. Néanmoins, mes connaissances limitées dans ces langages auraient demandé que je me focalise plus sur les aspects prise en main/fonctionnement/optimisation que les aspects théorique/développement/spécialisation.

La création du programme en matlab a introduit un certain nombre de problèmes concernant la structuration des données, l'indexation logique et la complexité algorithmique.

La recherche d'efficacité en termes de productivité, de temps et de ressource est centrale d'un point de vue professionnel. Comme développé par (Goldreich, 2008), la nécessité d'une définition de la complexité (qui représente le nombre d'étapes qui seront nécessaires pour résoudre le problème pour une entrée de taille donnée) est essentielle pour définir la faisabilité d'un algorithme. Suite à cela, nous pouvons utiliser des outils pour structurer les données, et passer à une étape d'optimisation par indexation logique.

Considérant l'application à de grands volumes de donnée, il est nécessaire de trouver une solution qui puisse gérer les données séquentiellement, ou parallèlement pour diminuer le temps de traitement et la lourdeur de l'algorithme. La limitation dans l'espace de la recherche des données au moyen d'un index spatial permettrait d'accélérer l'accès à ces dernières. L'utilisation de K-D Tree, notamment traité par (Kakde, 2005), permet de structurer les données en partitionnant l'espace suivant un arbre binaire, où chaque parent possède deux uniques enfants (deux demi-espaces).

Sur un principe similaire, l'usage de l'Octree semble particulièrement adapté à notre problème. Il s'agit d'une structure hiérarchique basée sur une décomposition récursive d'un cube 3D initial, où chaque noyau de la hiérarchie représente une partie du cube initial tel que : le nœud principal d'un Octree représente tout le cube, et chaque nœud est subdivisé, générant 8 enfants (pour plus d'informations, lire (T Lewiner et al., 2010)). Ceci est vraiment adapté aux besoins de notre algorithme, et permettra par exemple une indexation spatiale diminuant considérablement le temps de traitement. L'aspect cubique permet un plus grand contrôle sur la recherche de voisins par rapport aux formes complexes des enfants de K-D Tree.

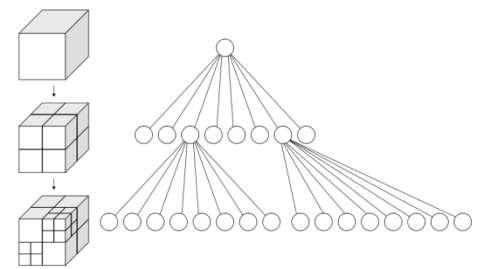


Figure 41 Arbre Octree. Source (Peixoto, 2009)

### 4.3.2 La création de l'algorithme et explications

Dans un premier temps, nous chargeons les données, qui seront traitées par l'algorithme. Nous voulons à partir d'un nuage de point, obtenir différents segments selon les critères de sélection. Le fichier d'entrée est un fichier ASCII, contenant les coordonnées d'un point par ligne : X Y Z. A ce niveau-là, nous pouvons alors commencer les traitements afin d'obtenir les normales et courbures en chaque point.

Le logiciel CloudCompare nous permet d'obtenir cela de manière simple et rapide, grâce à un plugin PCL (Point Cloud Library), utilisant un algorithme basé sur la méthode de (Berkmann & Caelli, 1994). La sélection des points supports du calcul peut être déterminée selon deux méthodes différentes :

- pour chaque point, sélection de tous les points dans un rayon défini par l'utilisateur ;
- Sélection des K-plus-proche-voisin (où K est le nombre entré par l'utilisateur) par recherche KNN<sup>29</sup>

Différents test ont été réalisés, à l'issue desquels la méthode KNN en considérant 30 voisins a été retenue (la fidélité géométrique et l'estimation des courbures et normales est optimale). Nous obtenons alors un fichier ASCII de 7 colonnes et n lignes : [X Y Z C Nx Ny Nz].

Pour le traitement dans Matlab, différents fichiers de différentes tailles ont été chargés à des fins de tests. Les fichiers ASCII sont chargés au moyen de la commande load, et attribués à une matrice qui considérera chaque colonne pour chaque coordonnée, et chaque ligne pour chaque point.

```
LA=load('nuage_colonne_complet_echantillonee_75000.asc');
```

L'algorithme créé est basé sur une croissance de région permet d'obtenir des régions connexes qui regroupent des points ayant des caractéristiques similaires.

Pour déterminer quel sera le premier point sélectionné lors de la croissance de région, j'ai décidé de classer tous les points en fonction de la courbure, pour ainsi sélectionner en premier les points dans les régions homogènes. Ceci est très important car conditionne en partie le résultat. En effet, si les points sources sélectionnés sont pris sans ordre logique, alors la segmentation sera incohérente et ne rendra pas compte de la réalité de la scène. rajouté une dernière colonne à la matrice S, qui est simplement le numéro de la ligne, donnant à chaque point un identifiant, qui sera l'outil principal de mise en correspondance au sein de l'algorithme.

```
S=[sortrows(LA,4) (1:1:size(LA,1))'];
```

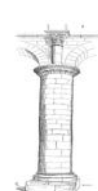
Dans un souci d'optimisation et d'efficacité, il a fallu trouver une solution pour augmenter la vitesse de traitement. En effet, la complexité initiale de l'algorithme  $O(n^3)$  impliquait des temps de traitement prohibitifs. Ceci était dû au fait d'une imbrication trop importante de boucles for/while les unes dans les autres, travaillant sur tous les points à chaque fois. Pour pallier à cela, plusieurs méthodes ont été utilisées :

- La vectorisation : Matlab est optimisé pour des opérations sur les vecteurs et matrices. Le processus consiste à essayer d'écrire du code basé sur les boucle ou traitements scalaires en utilisant des opérations matricielles et vectorielles. Ceci permet l'écriture de codes plus succincts, moins propices aux erreurs et qui s'exécutent beaucoup plus rapidement que la correspondance par boucle.
- L'indexation : Plusieurs méthodes d'indexation (logique, linéaire ou scriptée) permettent de raccourcir le temps de traitement. Le principe est d'attribuer à plusieurs éléments groupés d'une matrice (une ligne, une colonne, un groupement particulier,...) un marqueur (appelé index qui est un nombre), et qui permettra alors de réaliser les opérations sur ce groupement en travaillant simplement sur l'index, limitant le domaine de recherche et d'exécution.
- Pré allocation des espaces pour augmenter la vitesse de traitement

Souvent dans les algorithmes, nous sommes confrontés à une multitude de paramètres dont la signification est floue, et ne permet pas de se représenter clairement l'influence des paramètres. Dans un souci de clarté, d'accessibilité et d'efficacité, j'ai décidé de laisser à l'utilisateur le choix de plusieurs paramètres d'entrée qui conditionneront l'algorithme avec des valeurs adaptées pré-rentrées :

- Le pas : il s'agit du rayon de la recherche de voisins. Afin d'obtenir une zone correcte, il est bien de rentrer une valeur correspondant à la variation minimum entre deux détails de deux régions différentes. Plus le pas sera petit, plus l'algorithme sera sensible aux discontinuités de densités dans le nuage de point, et plus l'algorithme sera rapide. Plus le pas sera grand, plus l'algorithme sera lent et donnera des régions discontinues. Pour l'application à une colonne, le pas retenu est de 1.5 cm (correspondant à 2 fois la distance maximale entre 2 points.).

<sup>29</sup> KNN : K-Nearest-Neighbor ou K-plus-proche-voisin



- La similarité : Il s'agit du critère à satisfaire pour rattacher un point à une région. Si la différence entre les normales du point et de la région est inférieure au critère, alors le point est ajouté à la région. Pour un excellent rapport segmentation/fusion, un critère de 0.2 est extrêmement adapté pour la majorité des applications. Plus ce critère tendra vers 0, plus le nuage sera segmenté, plus il tendra vers l'infini, plus le nuage se rapprochera d'un seul et unique segment (Compromis pour éviter l'under et l'over segmentation)

Je vais maintenant détailler le cœur de l'algorithme :

### Description des variables utilisées

$S = [X \ Y \ Z \ C \ N_x \ N_y \ N_z \ Id]$  : matrice d'entrée comprenant les coordonnées des points, leur courbure, normale et identifiant.

Rtemp = comprend tous les éléments traités qui ont été associés à une région

Rsmall = comprends tous les éléments faisant partie de régions inférieures à 0.00001% des points du nuage

R = Région en cours de création regroupant tous les éléments de S répondant au critère de similarité

Nr = Vecteur normal moyen de la region R

N = matrice des points voisins de l'élément i sélectionné

Rf = matrice de cells<sup>30</sup> comprenant les différents segments issuent de la segmentation 1

H = [X Y Z R G B], où chaque point d'un même segment partage les mêmes composantes R G B.

### Segmentation 1 (S1)

Le script est appliqué à chaque élément (ligne) de la matrice S au moyen d'une boucle for. Le premier test détermine si l'élément i sélectionné est déjà présent dans la matrice Rtemp. Si ce n'est pas le cas (donc la condition est TRUE) le script continue (sinon il passe à l'élément i+1 de S). Le point i est alors ajouté à la région R, et le vecteur Nr est calculé (ligne 4). Puis, on cherche tous les points voisins de l'élément S(i) compris dans le rayon défini par le pas, et on les stocke par leur identifiant dans la matrice idx (travail sur les index pour plus de rapidité). Tous les points dont l'identifiant est dans la matrice idx (donc voisins de S(i)) sont placés dans la matrice N (ligne 6). La matrice idx est remise à zéro pour la suite, et le numéro de la ligne traitée est affiché.

```

1 Rf=[]; Rsmall=[]; Rtemp=[];
2 for i=1:size(S,1)
3     if (mean(ismember(S(i,8),Rtemp))~=1);
4         R=[R;S(i,:)]; Nr=(R(:,5:7));
5         idx=rangesearch(S(i,1:3),pas,S(:,1:3));
6         N=[N;S(idx,:)]; idx=[]; disp(i);

```

Pour chaque élément de la matrice N (tant que N n'est pas vide) la suite de l'algorithme sera effectuée. Initialement réalisé à l'aide d'une boucle for, le problème posé était celui des points traités. En effet, la boucle for n'accepte pas de réaliser des itérations sur une entité de taille non fixe, et gardera la taille de la première itération, même si de nouveaux éléments sont ajoutés. C'est pourquoi j'ai décidé d'utiliser une boucle while.

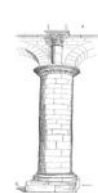
Un test (ligne 8) est effectué pour savoir si l'élément N(j) sélectionné remplit les 3 conditions conditionnant la réalisation de la suite de la boucle: il n'appartient ni à la matrice Rtemp ni à la région en cours de traitement R et répond au critère de similarité. Alors, l'élément N est ajouté à la suite de la matrice R, et le vecteur normal moyen est recalculé (ligne 10). Ses voisins sont cherchés dans la matrice S et ajoutés à la suite de la matrice R en travaillant sur la structure Octree interne à la fonction rangesearch (pour plus d'efficacité et un gain de temps considérable ligne 11). La boucle for imbriquée testera la présence dans la matrice N de chaque élément k de idx (lié par l'identifiant à S ligne 13). S'il n'est pas présent, alors il est ajouté à la matrice N (ligne 14). La boucle while est alors reprise pour  $j = j+1$  (ligne 18)

```

7     j=1; while j<=size(N,1);
8         if (mean(ismember(N(j,8),Rtemp))~=1 && mean(ismember(N(j,8),R(:,8)))~=1
9     && norm(N(j,5:7)-Nr)<=norm(similarite*[1 1 1]));
10            R=[R;N(j,:)]; Nr=mean(R(:,5:7));
11            idx=rangesearch(N(j,1:3),pas,S(:,1:3));

```

<sup>30</sup> Une cell {X} est un élément double qui dans notre cas est les dimensions du segment X. Exemple : Rf{1,1}=124579x8double



```

12         for k=1:size(idx,1);
13             if mean(ismember(S(idx(k,1),8),N(:,8)))==0;
14                 N=[N;S(idx(k,1),:)];
15             end
16         end
17     end
18     j=j+1;
19 end

```

Lorsque la boucle while a renvoyé une condition FALSE, alors le résultat de la segmentation de la région est renvoyé, et l'organisation est réalisée. La matrice Rtemp est remplie de tous les nouveaux points afin d'être utilisée pour ne pas traiter le même point plusieurs fois (travail en parallèle à l'indexation spatiale).

Le bruit est écarté et les points aberrants sont écartés grâce à Rsmall sans perte de détails fins (test sur le nombre de ligne donc d'élément de R ligne 22). La matrice N est alors réinitialisée (ligne 25) et l'algorithme reprend au début pour le point i suivant non traité de S. La boucle initiale se termine alors terminant la segmentation 1 (ligne 27). Afin d'exploiter les résultats obtenus il faut passer par une étape d'écriture des données dans un format qui peut être lu par les logiciels de traitements de nuages de points. J'ai choisi d'écrire ces données au format ASCII.

```

20     Rtemp=[Rtemp;R(:,8)];
21     if size(R,1)<=petit
22         Rsmall=[Rsmall;{R}];
23     else Rf=[Rf;{R}];
24     end
25     N=[];
26 end
27 end disp(' Done ');

```

### Segmentation 2 (S2)

La deuxième partie de la segmentation, sur un principe similaire à S1, travaille sur une matrice d'entrée (régions issues de S1) triée selon le nombre de point contenu dans chaque segment. Elle travaille sur les centroïdes des régions issues de S1, et sur les vecteurs moyens de chaque région. Chaque centroïde est sélectionné, et comparé avec son voisinage. Selon des critères de symétrie (coupe transversale où h=assimilation entrée par l'utilisateur), de géométrie locale (forme définie par l'enveloppe convexe) et de différence angulaire (étude de courbure), des coupes d'études sont réalisées limitant la recherche des voisins (bien que très faible de base car l'on travaille sur les centroïdes) aux plus susceptibles d'appartenir à une région. En effet, suite à la première segmentation, plusieurs éléments de symétries sont dégagés, et cette S2 permet de regrouper ensemble par exemple les différentes franges d'un cylindre par regroupement des centroïdes compris dans une tranche de quelques centimètres. L'autre différence avec la première segmentation réside dans l'analyse angulaire. En effet, il n'y a plus de vecteurs moyens créés, mais l'angle est comparé aux quelques voisins à chaque angle de la région en cours, pour extraire une éventuelle géométrie (Voir Annexe 6)

### Ecriture

Dans un premier temps, afin de visualiser les résultats, je travaillais avec la visualisation Matlab. Cependant, lorsque le nombre de point traité excédait 15000, la visualisation n'arrivait pas à afficher correctement et de manière fluide l'ensemble du nuage de points.

```

for i=1:size(Rf,1); E=[]; E=Rf{i};
cmp=rand(size(Rf,1),3);
grid on; hold on;
scatter3(E(:,1),E(:,2),E(:,3),9,cmp(i,:))
end

```

J'ai donc décidé d'écrire un fichier où chaque segment serait coloré d'une couleur aléatoire au format ASCII pour une visualisation dans CloudCompare (ou tout logiciel de traitement de nuages de points).

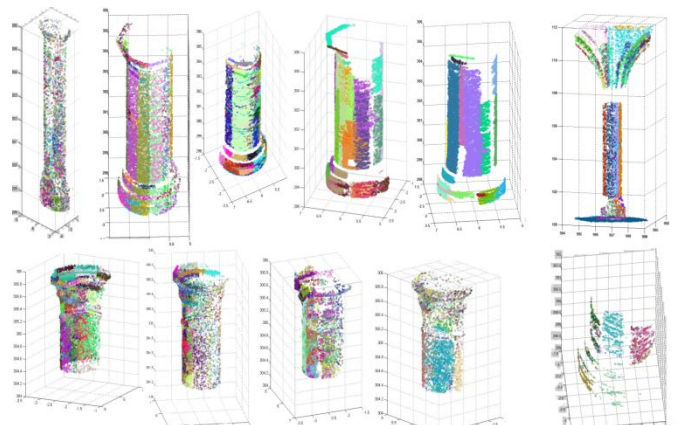


Figure 42 Résultats et différents tests visualisation Matlab

Etant donné que la matrice Rf contient tous les segments en tant que « cells », j'ai dû au moyen de boucles extraire chaque élément de ces cells et ajouter 3 colonnes R G B dont la valeur est identique pour les points d'une même cell. Je fus ici confronté principalement au problème de dimension et de calcul de matrices aux dimensions différentes, résolu par l'emploi de fonctions telles que cell2mat (transformer une cellule en matrice) ou repmat (pour répéter une matrice ou une ligne suivant la dimension voulue) et la démultiplication d'une boucle for en 2 pour introduire f.

```
E=zeros((size(Rtemp,1)-size(cell2mat(Rsmall),1)),11);
%E=[X Y Z C Nx Ny Nz n°ligne_S R G B]
cmp=rand(size(Rf,1),3);
for i=1;
    E(1:size(Rf{i},1),1:8)=Rf{i};
    E(1:size(Rf{i},1),9:11)=repmat(cmp(i,:),size(Rf{i},1),1);
    k=size(Rf{i},1);
end
for i=2:size(Rf,1);
    f=k+size(Rf{i},1); E(k+1:f,1:8)=Rf{i};
    E(k+1:f,9:11)=repmat(cmp(i,:),size(Rf{i},1),1);
    k=f;
end
H=zeros((size(Rtemp,1)-size(cell2mat(Rsmall),1)),6);
H(:,1:3)=E(:,1:3);H(:,4:6)=round(E(:,9:11)*100);
dlmwrite('1475000_colonne_segmentation1.asc', H, '\t')
```

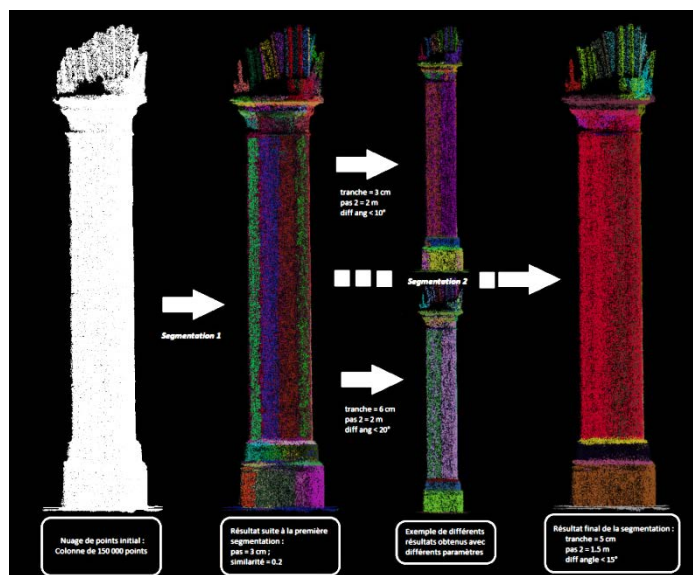
## 4.4 Les résultats

L'algorithme développé s'appuie sur plusieurs milliers de tests différents, qui ont permis d'avancer petit à petit et de corriger tous les problèmes rencontrés lors de la création du script. Ces avancées se traduisent en termes de gain de temps, de performances, mais aussi et surtout de résultats et d'adaptabilité. Ainsi, j'ai testé le script sur des environnements divers et variés, pour le confronter à un panel étendu de cas variant suivant l'échelle, les primitives géométriques constituant la scène, le nombre de points, l'homogénéité de la distribution, la densité moyenne et la qualité globale (consolidation, filtrage préliminaire, bruit). Voici 3 exemples de résultats mettant en évidence l'adaptabilité de la nouvelle méthode de segmentation développée.

### 4.4.1 La colonne

Une colonne de la cathédrale Saint-Paul a été retenue pour les tests en priorité. Il s'agit d'un très bon exemple pour prouver la détection d'enchaînement de formes cylindriques. Ainsi, les primitives géométriques constituant la scène sont principalement des cônes/cylindres ou sphères. Le résultat présenté concerne un nuage de 1 million de points, qui fut échantillonné de manière homogène pour garder une distance minimale entre deux points de 1 mm. Forte densité et bruit minimum, la segmentation de cette colonne démontre une application orientée

précision qui permet la détection des différents enchaînements cylindriques.



On distingue clairement les différences entre les 2 segmentations, la première groupant plusieurs points pour réaliser des régions correspondant à des bandes verticales. Ceci permet de conceptualiser clairement la théorie de la méthode, qui regroupe dans un premier temps les points ayant des normales égales (lorsque la similarité est réduite, les bandes sont plus fines et plus sensibles au bruit, qui peut faire apparaître des discontinuités). La deuxième segmentation arrive clairement à regrouper les différentes régions aux propriétés symétriques, de

taille et angulaires similaires, retournant un résultat très pertinent.

#### 4.4.2 Le château de Colonster

Le château de Colonster (levé avec le Scanner Laser Trimble GX) est une application à bien plus grande échelle sur un nuage aux propriétés bien différentes. En effet, on retrouve ici un nuage non homogène, constitué principalement de plans, d'une densité plus faible (720 000 points et une distance aléatoire entre deux points voisins variant de quelques millimètres à plusieurs centimètres) qui fait apparaître beaucoup de bruit sur la tour.

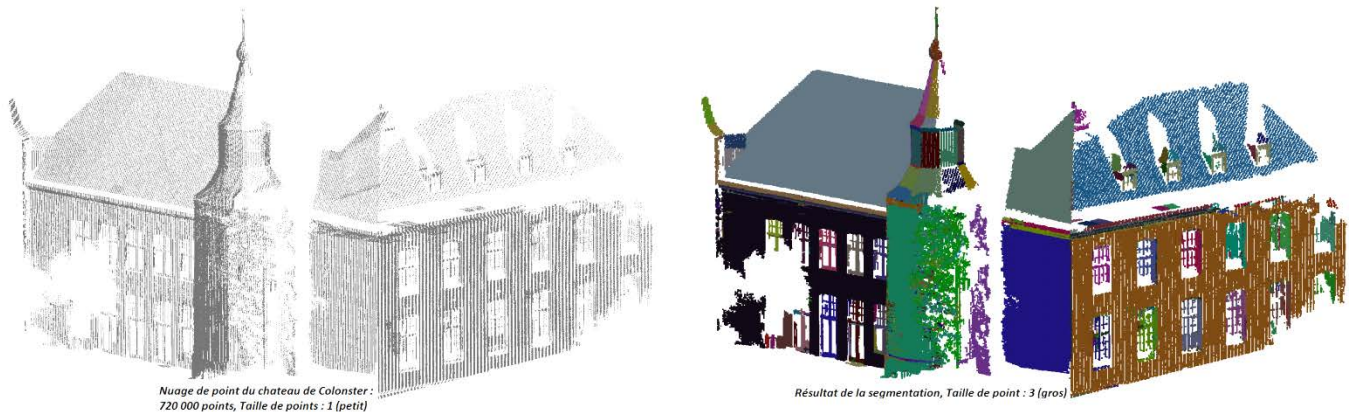


Figure 44 Segmentation appliquée au château de Colonster levé par Scanner GX

La première chose qui apparaît visible, est une excellente détection des formes planes. On comprendra aisément que le critère de normalité est très adapté pour des scènes contenant des primitives planaires. Les fenêtres sont également détectées en tant qu'ensemble séparé des façades, et permettront une analyse utile dans le cas de la plupart des bâtiments. On voit également clairement l'influence du pas (exemple sur la tour) qui attribuera une partie de la tour (violet) dans une nouvelle région car trop éloigné de ses voisins bleu cyan. La présence d'un bruit très prononcé sur la tour est immédiatement détectée grâce à la segmentation, qui permettra d'écarter ce phénomène au moyen d'une sélection de la région concernée.

Ainsi, le changement d'échelle et de densité ne pose aucuns problèmes à l'algorithme.

#### 4.4.3 3 Colonnes de la cathédrale Saint-Paul et chapiteaux

Le dernier cas présenté, est celui d'un mélange complexe à échelle intermédiaire de formes géométriques complexes et variées au sein de la cathédrale Saint-Paul de Liège. Basé sur un échantillon moyennement dense de 500 000 points, la segmentation a été confrontée à une hétérogénéité géométrique (plans, cylindres, cônes, sphères) qui de par les résultats, confirme l'adaptabilité de la méthode proposée.

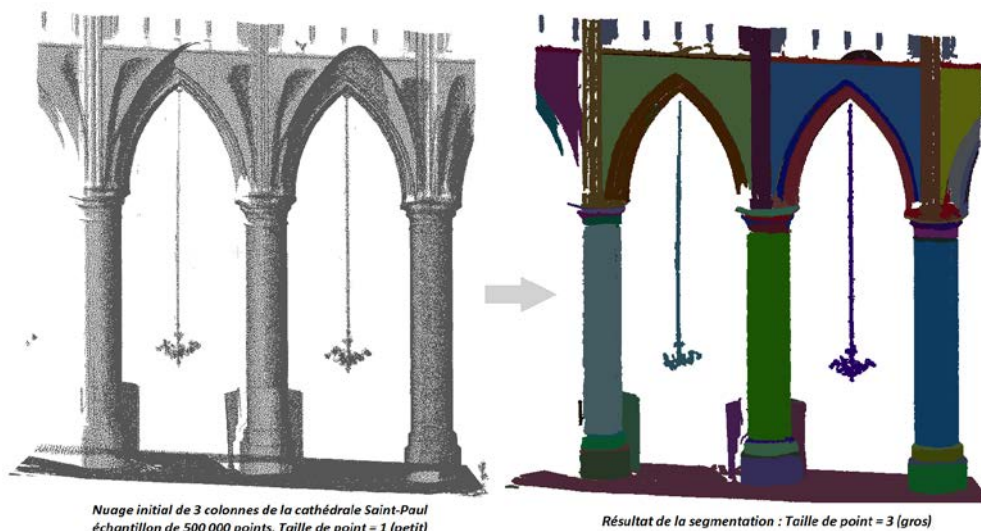


Figure 45 Segmentation d'un extrait de 3 colonnes de la cathédrale Saint-Paul

On remarque clairement la détection des 3 plans des arcades, ainsi que les différentes arcades. Ceci n'est pas au détriment d'une sous-segmentation des colonnes, qui restent segmentées en 8 segments décrivant des primitives géométriques différentes.

#### 4.4.4 Bilan & validation

Clairement, le résultat de la segmentation est indépendant de la scène étudiée (mais pas de la précision de consolidation). Lorsque les paramètres d'entrée sont adaptés, la classification de la scène révèle une organisation logique, s'affranchissant d'une étape chronophage et imprécise par segmentation manuelle. Une analyse et exploitation approfondie et de précision peut alors succéder à cette ordonnance semi-automatique.

Pour un résultat optimal, il a été nécessaire de s'affranchir des « segments » minimes (comportant un nombre de points inférieur à 1/1000 du nombre total de points) pour éviter l'effet d'une sur-segmentation. Néanmoins, ces points ne sont pas supprimés mais toutes ces micros régions sont ajoutées à la région Rsmall (colorée en rouge) pour pouvoir être éditée par l'utilisateur.

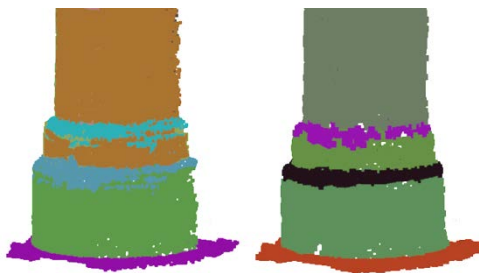


Figure 46 Influence du pas sur une colonne.

a) pas= 0.07 cm b) pas=0.02 cm

J'ai mené plusieurs études pour présenter l'influence des différents paramètres et le temps total d'exécution du script.

Dans un premier temps, il est nécessaire de noter que seul le paramètre d'entrée « pas » a une influence notable sur le temps d'exécution du script. De plus, il est réellement important dans le sens où le nombre de régions créées et leur continuité est grandement défini par ce paramètre (voir Figure 46)

Le critère de similarité (second paramètre qui n'influe que la première segmentation) est également primordial et a une grande influence sur le résultat. Ce fut une motivation supplémentaire d'implémenter un algorithme semi-automatique pour laisser l'utilisateur libre de définir des paramètres en prévision d'un résultat recherché. Par exemple, si l'on veut détecter grossièrement plusieurs parties, alors on diminuera ce critère pour atteindre  $5 \cdot 10^{-1}$ , tandis que descendre à  $1 \cdot 10^{-2}$  sera très exclusif à des régions planes (et applicable seulement dans le cas de nuages de points parfaitement nettoyés et consolidés).

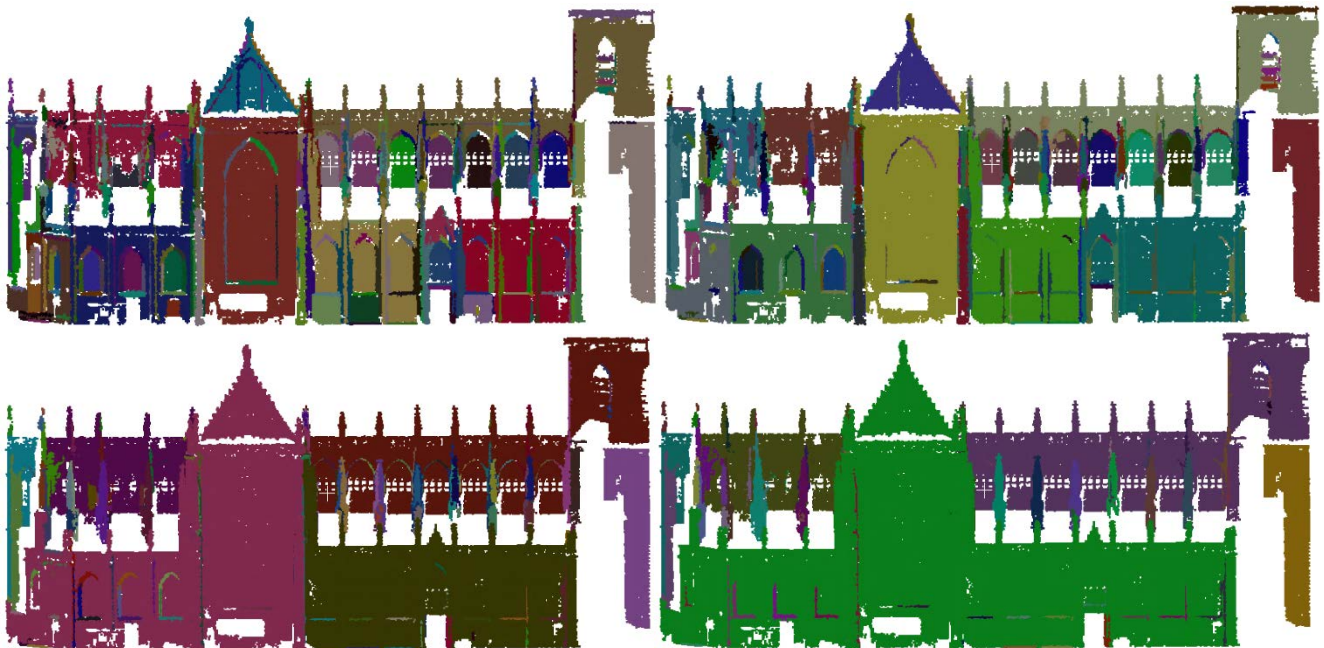
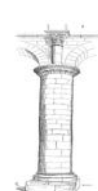


Figure 47 Segmentation façade Sud : a)  $C=1.5 \cdot 10^{-1}$  b)  $C=2 \cdot 10^{-1}$  c)  $C=3 \cdot 10^{-1}$  d)  $C=5 \cdot 10^{-1}$

Le dernier critère d'assimilation sera important dans le cadre de formes non planimétriques. Son influence est très importante mais les variations dans le paramètre d'entrée ne conditionnent pas la majorité du résultat. En effet, un choix adapté sera entre 0.05 et 0.15 cm pour des ouvrages archéologiques afin de grouper efficacement les morceaux aux connectivités évidentes.

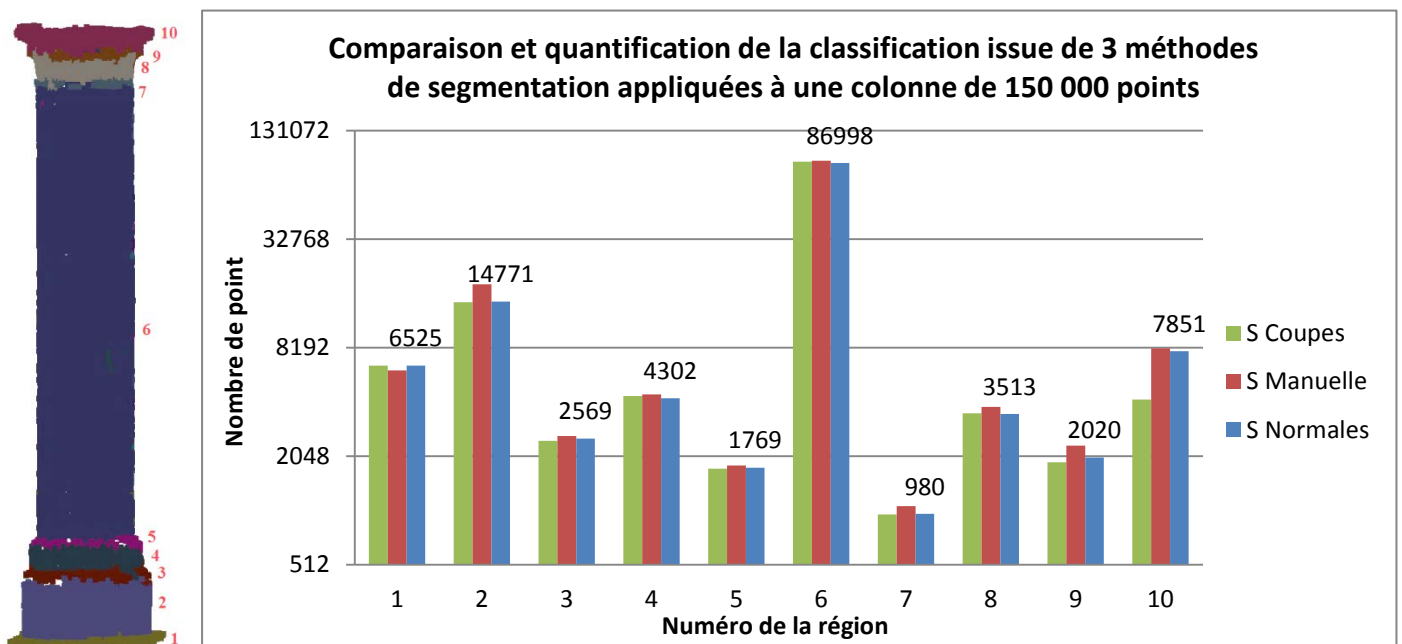


Il est également important de souligner qu'une optimisation de l'algorithme reste possible pour en étendre l'utilisation à des nuages de points très volumineux (complexité  $n^2$  apparente). En effet, au vu des analyses temporelles, il est encore possible de réduire considérablement le temps d'exécution du script.

Nb de points	Temps (s)
12552	15.26
14000	17.43
15000	19.96
16000	21.71
17000	24.66
18000	30.08
19000	29.8
20000	32.2
25000	45.82
30000	64.54
35000	91.34
40000	116.77
50000	178.34
80000	502.41
100000	854.75

Figure 48 Analyse temporelle sur la segmentation d'une colonne - pas = 0.05

Afin de quantifier les points bien classifiés, j'ai comparé les résultats obtenus concernant la colonne avec la segmentation issue de la méthode initialement définie de projection de cercles, et une segmentation manuelle.



Les écarts de classification entre les différentes méthodes sont inférieurs à 10% et sont dû à certains artéfacts :

- La présence de bruit : Sa détection et mise à l'écart est bien meilleure dans le cas d'une segmentation par normale, tandis qu'une segmentation manuelle ne permet que l'écart des zones flagrantes. Sur le segment 10, le fort écart avec la segmentation coupe provient de la mise à l'écart de points hors tolérance pour la reconstruction de cercles, et renvoie un résultat donc faux.
- Imprécision de pointé : La segmentation manuelle est imprécise par exemple pour détecter le sol
- Nombre de région : Le nombre de régions créées varient suivant la méthode.

Ces fortes similarités et les faibles écarts permettent de conclure et valider la segmentation par normales proposée.





## 5 MODÉLISATION, TEXTURISATION ET NIVEAUX DE DÉTAILS

### 5.1 Utilisation des différentes modélisations

Une fois l'ensemble hiérarchisé et structuré, nous pouvons nous affaier à l'étape 5 de la chaîne de production : la modélisation.

Grâce à la segmentation, nous allons traiter le nuage de points en différentes entités. Chaque segment sera alors modélisé indépendamment, limitant ainsi la base du maillage aux points pertinents.

Afin de répondre à une approche multi-perspective, il va être intéressant de modéliser de différentes manières avec différents niveaux de détails. Ainsi, nous aurons plusieurs modèles pour différentes applications (support d'une exploitation future par l'Unité de Géomatique).

Il est important de définir les finalités pour pouvoir adapter les modélisations et les contraintes de production des différents modèles. L'usage archéologique impose une précision centimétrique qui de fait rendrait la modélisation de l'ensemble de la cathédrale inutilisable dûes aux simplifications induisant une perte de détail trop importante pour une manipulation fluide. C'est pourquoi les différentes modélisations seront l'objet de commandes particulières, identifiées au moyen du nuage de point traité et transformé en visite virtuelle WebGL.

Néanmoins, dans une finalité de recherche de meilleures techniques de modélisation, les travaux effectués à cette étape à petite échelle seront extensibles et constitueront une base solide pour des modélisations futures à grande envergure. J'ai donc décidé de réaliser une comparaison entre modélisation globale et post-segmentation pour quantifier et qualifier les apports de la segmentation en termes de précision de modélisation.

### 5.2 Modélisation de l'objet dans sa totalité

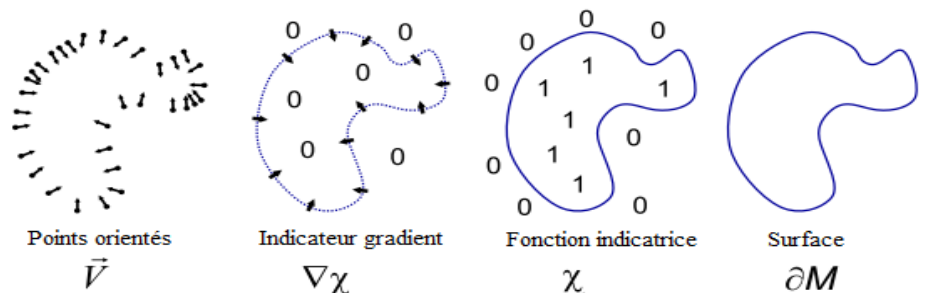
Pour obtenir un modèle maillé et texturé de bonne qualité, je me suis appuyé sur une méthodologie simple et structurée. La finalité n'est pas de mailler l'ensemble de la scène, mais de s'attacher à modéliser les parties qui intéressent les archéologues.

#### 5.2.1 Création du maillage brut : MeshLab

Dans un premier temps, un maillage a été réalisé pour décrire les différents objets tels que saisis en utilisant le logiciel open-source MeshLab. Étant donné le nombre important de points et les faiblesses du logiciel, j'ai dû jongler entre nuages échantillonnés et complet pour obtenir un modèle 3D complet. Cependant, ceci ne se fait pas au détriment de la précision, mais permet un gain de temps considérable au niveau algorithmique.

Avant de commencer le traitement, il a été nécessaire de faire un inventaire des différentes méthodes de maillage existantes, pour en dégager la méthode la plus adaptée (voir Annexe 7 pour résumé des méthodes). La reconstruction de surface comme solution d'une équation de Poisson ((Kazhdan *et al.*, 2006) sera ici la méthode la plus précise et la plus propre pour les objets considérés. L'algorithme utilisé résout un système linéaire discret bien conditionné dont la complexité temporelle et spatiale est fonction du nombre de points traités :

$$\Delta \tilde{\chi} = \nabla \cdot \vec{V}$$
<sup>31</sup>  
 où  $\nabla \vec{V} = \frac{\partial v}{\partial x} \vec{e}_x + \frac{\partial v}{\partial y} \vec{e}_y + \frac{\partial v}{\partial z} \vec{e}_z$  est le gradient de  $\vec{V}$ ,  
 $\tilde{\chi}$  est la fonction scalaire recherchée,  
 $\vec{V}$  est le vecteur champs issu de la fonction scalaire.



La

Figure 49 Illustration de la reconstruction de Poisson 2D. Inspiré de (Kazhdan *et al.*, 2006)

reconstruction de poisson est une solution globale qui considère l'ensemble des données sans segmenter en régions

<sup>31</sup>  $\Delta \tilde{\chi} = \nabla^2 \tilde{\chi}$  et  $\nabla \cdot \vec{V} = \text{div } \vec{V}$

pour une reconstruction locale. La méthode est très robuste dans l'approximation des données bruitées et crée une surface lissée. La détection des détails et la précision de reconstruction ainsi que le volume et le nombre de triangle créé dépendent du niveau d'octree choisi. Pour illustrer ceci, voici un exemple réalisé sur un modèle en libre-service sur le Web :

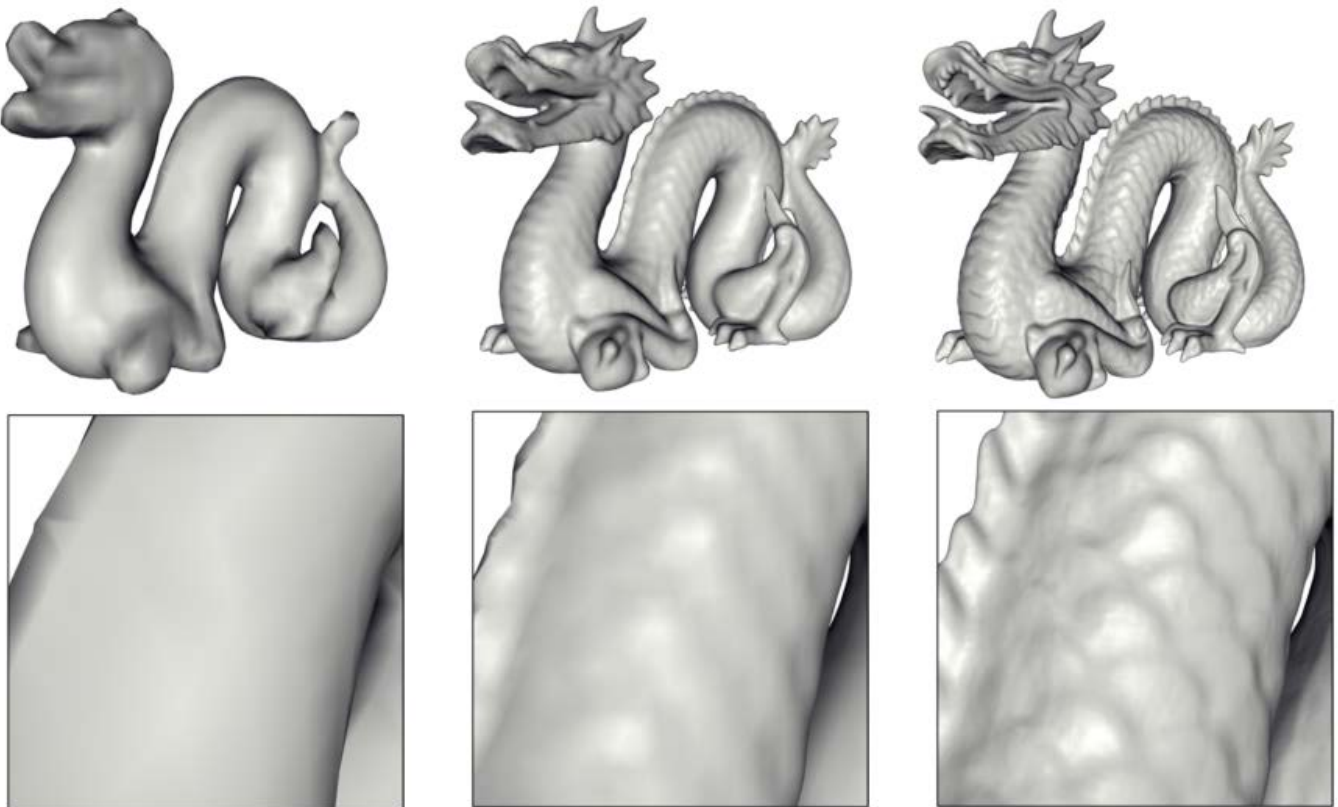


Figure 50 Reconstruction du modèle du dragon au niveau d'octree 6 (gauche) 8 (milieu) 10 (droite). Source (Kazhdan *et al.*, 2006)

La réalisation d'un maillage par la méthode poisson se fait en 6 étapes :

- Dans un premier temps, il faut importer le nuage de points à modéliser dans le logiciel MeshLab. Pour limiter les temps de chargement, de traitement et le volume de données, il est préférable d'importer le nuage au format .ply<sup>32</sup> (transformation possible via CloudCompare de ASCII → ply)
- Ensuite, nous calculons les normales des points par KNN (K-plus-proches-voisins). Cette étape est problématique pour des gros nuages de points (dans MeshLab) c'est pourquoi il est intéressant d'échantillonner le nuage de manière homogène (La méthode d'échantillonnage de Disk-Poisson est particulièrement efficace et permet d'obtenir un échantillon homogène qui conserve les propriétés essentielles du nuage) pour calculer les normales des points de l'échantillon. Nous pouvons alors transférer les propriétés des normales de l'échantillon vers le modèle. Le nuage de point initial se trouve avec des normales bien plus lissées, mais ceci n'est pas un problème pour la modélisation et le calcul de la surface Poisson (cela peut cependant poser un problème pour l'utilisation de shaders<sup>33</sup>)
- La reconstruction Poisson peut maintenant être réalisée. Dû à des bugs relativement importants, il est conseillé de prendre un niveau d'octree (lié au niveau de détail) de 9 pour une balance équilibrée entre détail et temps de traitement. Nous obtenons ainsi un modèle 3D watertight et topologiquement correct.
- Pour augmenter la précision et la fidélité de notre modèle vis-à-vis du nuage de point, il est intéressant de sous échantillonner notre modèle maillé. Ceci alourdira le modèle en créant davantage de triangles, mais permettra par la suite un meilleur rendu de la réalité. La méthode LS3 permet de subdiviser les surfaces suivant des boucles et l'orientation des normales.
- Puis, le transfert des propriétés via « vertex attribute transfer » permet de projeter le maillage sur le nuage de point pour épouser d'une meilleure façon la géométrie du nuage de point.
- Finalement, l'application d'une texture produit un modèle complet (détaillé dans la partie V.2.C)

<sup>32</sup> Ply : PoLYgon file format. Plus léger que le format ASCII et mieux géré dans MeshLab.

<sup>33</sup> Processus de rendu pour décrire l'absorption/diffusion de la lumière, les ombrages, la réflexion/réfraction,...

Pour résumer de manière illustrée la réalisation d'un maillage complet, voici la méthode appliquée à une colonne :

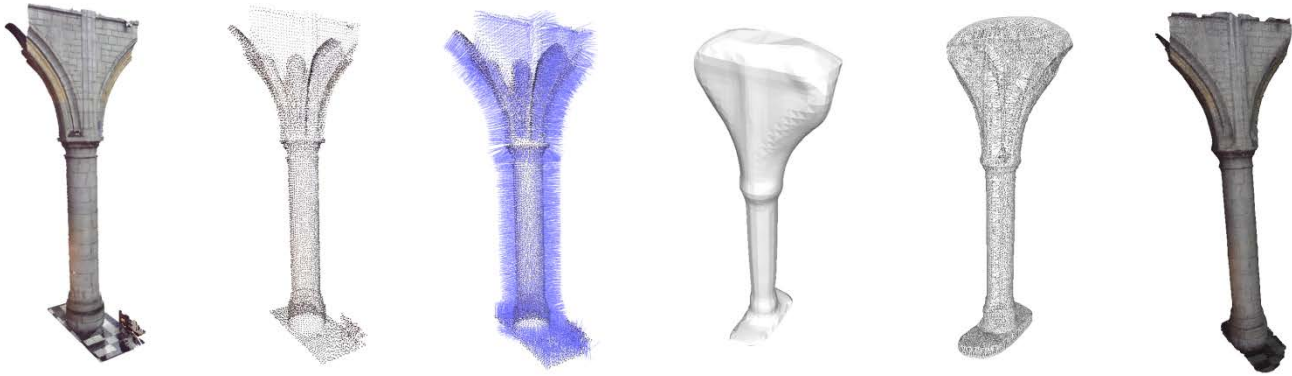


Figure 51 Les 6 étapes pour la réalisation d'un modèle 3D texturé de qualité par reconstruction Poisson

### 5.2.2 Simplifications et corrections

Il faut maintenant corriger le modèle et le simplifier pour qu'il soit optimisé et « propre ». Pour cela nous utilisons un outil extrêmement puissant, simple et reconnu : les surfaces de subdivision. Leur utilisation permet de gérer des topologies quelconques et un raffinement local et multi-résolution (très pratique pour la gestion de différents niveaux de détails) tout en étant stable pour les calculs. De plus, la compression et la transmission du maillage en font un outil de choix pour la simplification.

Cependant, il s'agit fondamentalement d'un maillage et l'on ne manipule jamais la surface, (on obtient une subdivision du maillage initial). Les problèmes de calculs de distance et l'impossibilité de la subdivision inverse demande une utilisation intelligente pour éviter des sauts trop importants entre deux subdivisions successives. L'utilisation parallèle de l'outil d'effondrement d'arêtes (Collapse edge), permet de grandement simplifier le maillage créé, mais rend souvent le maillage topologiquement incorrect (Hussain, 2009). Ces opérations améliorent le rapport taille/qualité du maillage créé. En général, il est possible de réduire de 30% la taille de fichier de sortie de manière à ce que les simplifications ne soient pas trop importantes réduisant la qualité du modèle.

### 5.2.3 Texturisation

Dans un but d'automatisation et de gain de temps, une technique de texturation à partir des composantes RGB du nuage de point a été mise en place et testée. En effet, les photos qui serviront à la texturisation ont été obtenues à partir de l'appareil photo intégré au laser Scanner. Ainsi, pour les récupérer au format JPG, il est possible de les exporter via le logiciel SCENE. Cependant, ceci est problématique car impose de recalculer manuellement toutes les photos désirées sur les modèles maillés, ce qui en plus d'augmenter le volume horaire, fait apparaître une nouvelle source d'erreur possible (en cas de texturisation approximative). Ainsi, j'ai décidé de réaliser une texturation à partir du nuage de point colorisé (tous les points sont projetés sur les matrices des images photographiées correspondantes). Le problème ici n'est alors que celui de la résolution des images. En effet, selon la résolution, les informations colorimétriques seront justes ou approximatives. Néanmoins, en considérant le nombre de points, cela ne posera pas de problèmes pour la texturation.

Ainsi, il est possible de créer une texture à partir du nuage de point en utilisant la couleur de chaque point alors considéré comme un vertex. Il faut au préalable réaliser une paramétrisation du modèle, pour pouvoir attribuer la texture. Par un transfert d'attribut à texture entre le nuage de point initial très dense au modèle 3D simplifié, on crée alors une texture à la résolution de notre choix (2048x2048 donne une texture de haute résolution), pour obtenir un modèle texturé final.

L'inconvénient de cette méthode concerne les zones d'ombres qui ne possèdent pas d'information colorimétrique. L'algorithme réalise alors une interpolation pour la couleur attribuée à la zone en question.

La texture est ensuite enregistrée au format png (jpeg pour une plus grande compression), et sera appelé au moyen du fichier .tml créé pour lier les fichiers rasters au fichier obj. (Voir Annexe 8)

Le modèle final est donc un fichier .obj lisible et utilisé par la majorité des programmes d'infographie 3D, ce qui permet une utilisation et une diffusion majeure de notre objet modélisé.

### 5.3 Modélisation hybride post-segmentation

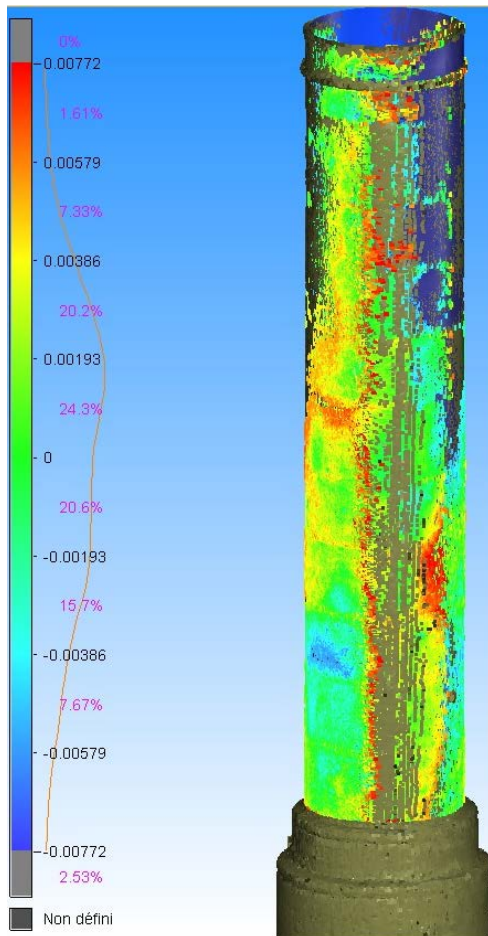


Figure 52 Ecart entre le segment principal du nuage de point et le cylindre associé

Dans le cadre d'une validation et d'une optimisation de la phase de traitement-modélisation, j'ai réalisé un modèle hybride maillage/géométries. La segmentation permet de considérer indépendamment les différents segments pour la modélisation et d'adapter la précision de modélisation suivant le segment considéré.

La création d'un modèle reconstruit par un mélange de formes géométriques et de maillages est un atout pour la simplicité et la diminution du volume des données, sans omettre la précision. Pour chaque entité segmentée, toutes les différentes formes sont générées (cylindre, cône, plan, ...) à partir de la région, et la forme retenue est celle qui minimise au mieux l'écart type. Si aucune forme ne convient réellement (c'est à dire si l'écart type le plus adapté est supérieur à 5 mm) alors le segment est maillé par la méthode Ball-Pivoting (Bernardini & Mittleman, 1999).

Son principe est très simple : A partir d'un triangle origine, l'algorithme crée une balle de rayon défini par l'utilisateur autour des 3 sommets jusqu'à toucher un autre point formant un nouveau triangle. Il fonctionne similairement à une croissance de région et s'arrêtera une fois toutes les arrêtes possibles testées. Ainsi, il se base sur les points existant et n'interpole pas une nouvelle surface, qui est essentielle pour conserver la précision liée au levé. De plus, appliquée aux différents segments, l'algorithme est très rapide et offre la plus grande fiabilité vis-à-vis du nuage de point.



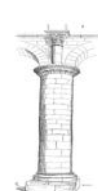
Figure 53  
Modèle hybride

Le résultat obtenu est un ensemble de segments que l'on regroupe par création de triangles liant les morceaux adjacents. La distance entre les sommets aux extrémités des régions est calculée et la liaison se fait lorsque le critère de distance minimale est satisfait. L'ensemble des géométries et maillages sont alors groupés dans un ensemble d'un seul tenant de haute-fidélité. Aucune simplifications/corrections n'ont besoin d'être appliquées, le maillage est propre et prêt à accueillir une texture.

### 5.4 Étude comparative

Dans un premier temps, il est nécessaire de noter la différence du volume pour un même fichier .obj modélisé suivant les deux méthodes précitées. Alors qu'un maillage par la méthode de poisson occupe 20 Mo (pour environ 1 million de points), un maillage hybride occupera 4 Mo (Soit une réduction de 80% du volume des données). Mais qu'en est-il des précisions ?

Dans un premier temps, j'ai réalisé un maillage par méthode Ball-Pivoting pour obtenir un modèle étalon, qui naturellement donne le maillage le plus proche du nuage de point. En effet, il n'y a pas de nouveaux sommets créés, les points du nuage devenant vertex du maillage.



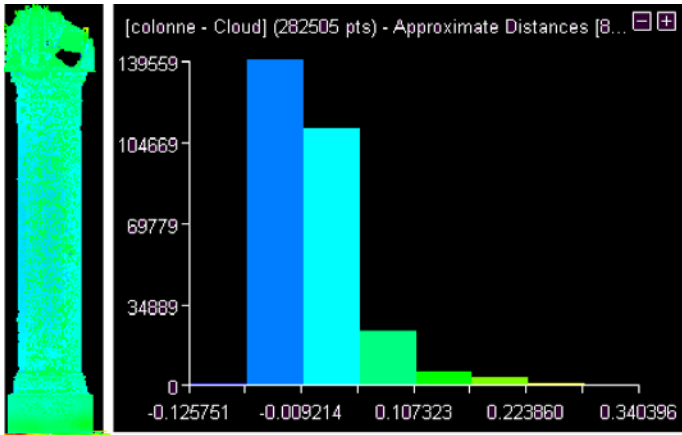


Figure 54 Ecart de distance au nuage de point de la modélisation Poisson

Ceci n'est pas le cas avec la modélisation hybride, où les écarts les plus marquants se font au niveau de la zone problématique dû à un recalage imparfait mettant en avant une double épaisseur (cf. Figure 55). Ainsi, la forme minimisant l'écart type a conservé le cylindre le mieux adapté ce qui montre des écarts visibles à ce niveau. Néanmoins, dans l'ensemble la modélisation montre des précisions très bonnes et homogènes : l'écart de distance moyen est de 4.8 mm avec plus de 90% des points compris entre 0 et 3 cm.

Voici en Figure 54 les résultats issus d'une comparaison avec la reconstruction poisson. On réalise que le modèle relativement fidèle au nuage dans les régions simples s'écarte de la réalité lorsque le nombre de détails augmente (chapiteaux, carreaux sol). Les précisions sont néanmoins bonnes, l'écart de distance moyen est de 1,73 cm, avec plus de 60% des points compris entre 0 et 3 cm. On remarque également le caractère robuste au bruit de la reconstruction poisson, et l'hétérogénéité des précisions suivant les détails.

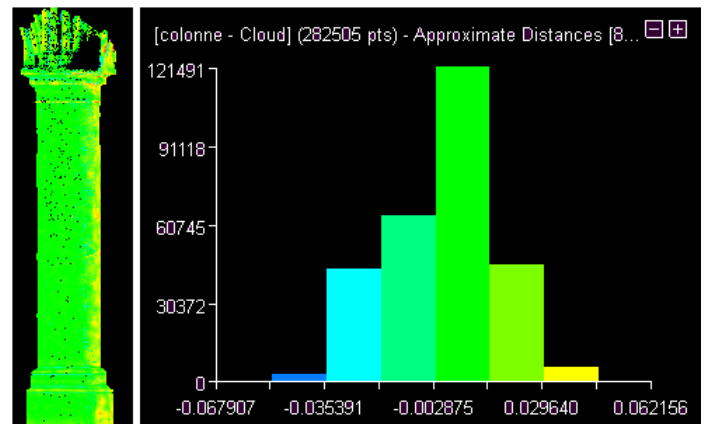
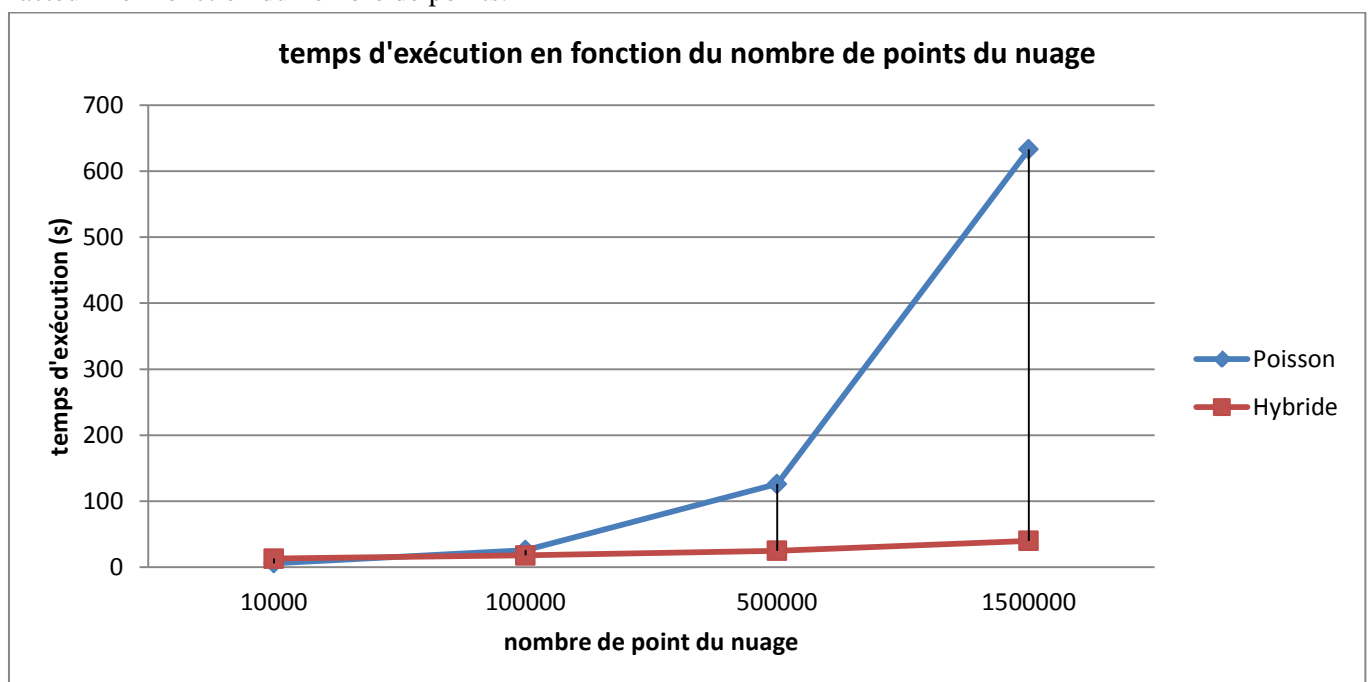


Figure 55 Ecart de distance au nuage de point de la modélisation hybride

Ainsi, en plus de permettre l'obtention d'un modèle plus léger, la modélisation hybride issue d'une segmentation préalable permet l'obtention d'un modèle pertinent et précis. Il permet également d'identifier clairement les zones de bruit ou sujettes à un recalage approximatif.

En amont, l'exécution et les traitements réalisés sont plus rapides dans le cas d'une modélisation hybride lorsque le nombre de points devient plus important. La reconstruction de poisson augmente grossièrement d'un facteur 4 en fonction du nombre de points.



## 6 VERS DIFFÉRENTS SUPPORTS POUR DIFFÉRENTES UTILISATIONS

### 6.1 Une application archéologique de précision

Pour développer une application simple, interactive, universelle, diffusable et qui ne dépende pas (ou peu) de la configuration, j'ai développé une chaîne de traitement qui aboutit à la création d'un fichier PDF 3D. Ce fut le système proposé aux archéologues pour réaliser les mesures nécessaires directement sur le modèle 3D maillé.

#### 6.1.1 Le format U3D et la création PDF 3D

Il est maintenant possible d'intégrer des éléments 3D à nos fichiers PDF au format U3D. Il s'agit d'un type de fichier qui conserve tous les paramètres 3D, et qui peut être intégré à l'environnement PDF. Très pratique, ceci permet d'effectuer tout type de mesures, de basculer entre plusieurs modes d'affichages et perspectives, de créer des annotations, des liens logiques à l'aide de boutons, de localiser des commentaires et bien d'autres possibilités.

Il est alors possible de manipuler des fichiers texturés (malheureusement les nuages de points ne sont pas pris en compte) très lourds, comprenant plusieurs milliers/millions de triangles permettant de conserver un détail minimal pour une application de précision. L'utilisation d'un nuage de point segmenté intelligemment est donc une prérogative pour obtenir un modèle d'une haute qualité.

En moyenne, un fichier PDF 3D (.pdf) est 80% plus léger que le même fichier au format obj (non texturé), présentant un atout considérable. Néanmoins, le passage vers un modèle .pdf demande la modification manuelle du fichier .tml associé au fichier .obj pour une gestion de la texture, car Meshlab utilise des paramètres d'entrées étalonnés différemment (voir Annexe 8)

Modèle maillé et texturé uniquement grâce à un nuage de points obtenu avec le laser scanner

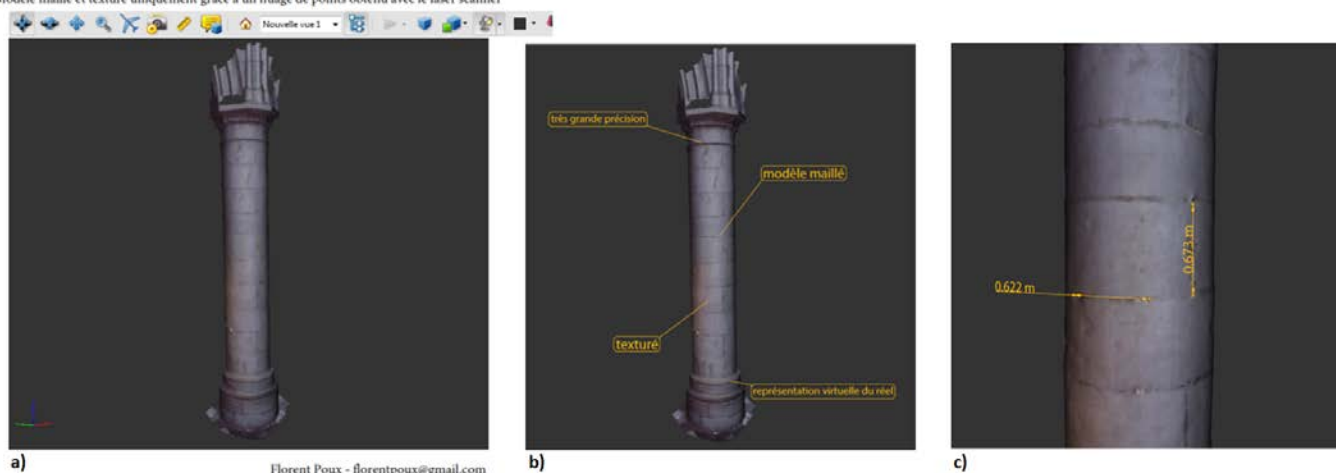


Figure 56 Exemple de PDF3D créé pour une colonne. a) Vue interface - b) Commentaires - c) détail et cotations

Il est difficile de trouver un logiciel de traitement 3D qui permette l'export au format U3D. Quelques solutions payantes sous forme de plugins existent pour ajouter ce format aux exports possibles de quelques logiciels (Autocad, Solidworks, ...), mais ne considèrent pas les logiciels de traitement utilisés.

La solution proposée demande l'utilisation du logiciel MeshLab qui propose l'export au format U3D.

En résumé, voici la chaîne de traitement idéale :

- **Consolidation : Logiciel constructeur (FARO SCENE, Trimble Realworks, Leica Cyclone)**  
*Export au format ASCII (.txt, .xyz, .asc, .ptc, ...)*
- **Échantillonnage + Segmentation : MATLAB + CloudCompare**  
*Export au même format*
- **Modélisation + texturation : MeshLab ou 3DReshaper (licence payante)**  
*Export au format obj*
- **Transmission : MeshLab + Adobe Acrobat Pro**  
*U3D + PDF*

### 6.1.2 Des possibilités d'exploitation étendues

Pour les archéologues ceci est particulièrement adapté. J'ai donc par l'intermédiaire de commandes ciblées pu fournir un document de travail en 3 dimensions. Les supports transmis furent :

- 1 – Un nuage de point échantillonné et le logiciel CloudCompare
- 2 – Les zones sélectionnées et traitées par la segmentation pour l'intégration dans un modèle sémantique de donnée, pour une reconnaissance adaptée
- 3 – Des fichiers PDF 3D de chaque chapiteau modélisé.

Afin de leur permettre une utilisation optimale des outils mis à disposition, nous leur avons donné une formation complète sur les différentes utilisations des fonctions désirées et des logiciels fournis. Ainsi, il leur était possible d'effectuer tout type de mesure 3D sur des objets d'intérêt inaccessibles (exemple Figure 57). Leur principale attache concernait les chapiteaux des colonnes, qu'ils ont ainsi pu étudier en profondeur et rentrer efficacement dans leur base de données.

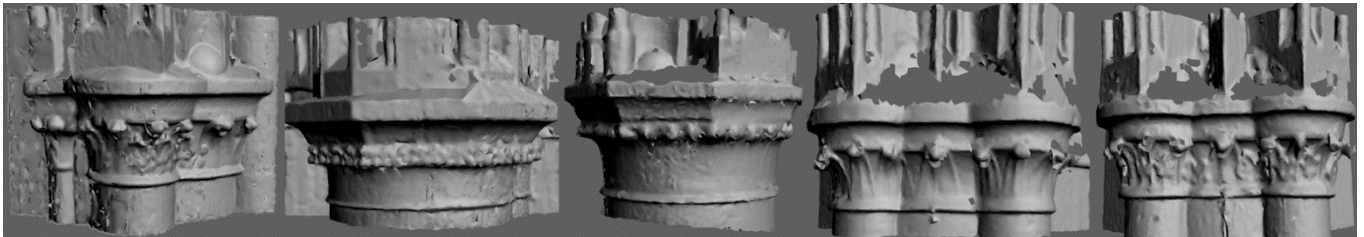


Figure 57 Exemples de feuillages modélisés et diffusés au moyen de PDF3D

## 6.2 Visite virtuelle WebGL

### 6.2.1 Le contexte

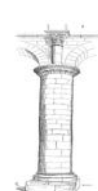
Au vu de l'envergure de la cathédrale, et les moyens mis à ma disposition l'application d'une visite virtuelle classique a dû être reconsidérée. En effet, ne disposant que du logiciel 3DReshaper en version démonstration pour 30 jours (ainsi qu'une version démonstration de VRmesh), la modélisation à grande échelle n'a pu être entreprise que durant cette période. C'est pourquoi j'ai décidé de changer l'optique de la visite virtuelle : développer une application qui permettrait la navigation instantanée dans le nuage de point colorisé.

Étant donné la grande densité de points (résolution spatiale inférieure à 1 mm) et le niveau de détail obtenu (Scans de distance adaptée d'une précision sur point de l'ordre de 5 mm en accordance avec (Boehler *et al.*, 2003)) le résultat est saisissant. Il est possible de contempler les coups de pinceaux sur les tableaux et les œuvres d'art, les vitraux de différentes couleurs sont illuminés, les voûtes et points inaccessibles (scannés à 20-25 mètres) sont retranscrits avec grande fidélité et la possibilité de se déplacer en 3 dimensions permet d'observer la richesse de la cathédrale Saint-Paul.

Néanmoins, la publication sur internet d'une visite dans un nuage de 500 millions de points impose des contraintes majeures pour permettre une expérience utilisateur fluide au temps de chargement instantané :

- Gestion du volume des données
- La visualisation d'un nuage de point dynamique
- L'évolution interactive et fluide de l'utilisateur
- L'intégration sur le Web de la visite virtuelle

A l'heure actuelle, aucun moteur, plugin ou application permettant la publication sur internet de modèles 3D ne permet l'affichage fluide de nuage de points volumineux. De plus, l'inconvénient majoritaire des solutions les plus courantes sur internet pour des visites interactives (Unity Web Player, Flash, ...) est un temps de chargement relatif au volume des données (un nuage de point de plus de 500 millions de points demanderait plusieurs heures de chargement).



Pour pallier ces différents problèmes j'ai ainsi décidé de développer une application WebGL à l'aide des travaux réalisés par (Preiner *et al.*, 2012) et des codes open-source fournis par le groupe de recherche autrichien TERAPOINTS Point Cloud Research de « Vienna University of Technology ».

## 6.2.2 La création de l'environnement WebGL

Les tests ont été réalisés sur serveur local, mais seront prochainement diffusés au public sur le site web de l'unité de Géomatique de l'Université de Liège. Dans un premier temps, la structure du site Web Html et l'organisation de la feuille de style CSS a été créé pour accueillir une interface efficace. J'ai organisé le site Web de façon à accéder aisément à la visite virtuelle WebGL. Un menu latéral composé de 4 images réagit interactivement sur simple clic pour afficher la visite sélectionnée. Ainsi, pas de changement de page, et la possibilité d'être immédiatement plongé dans la visite virtuelle (voir Figure 58).

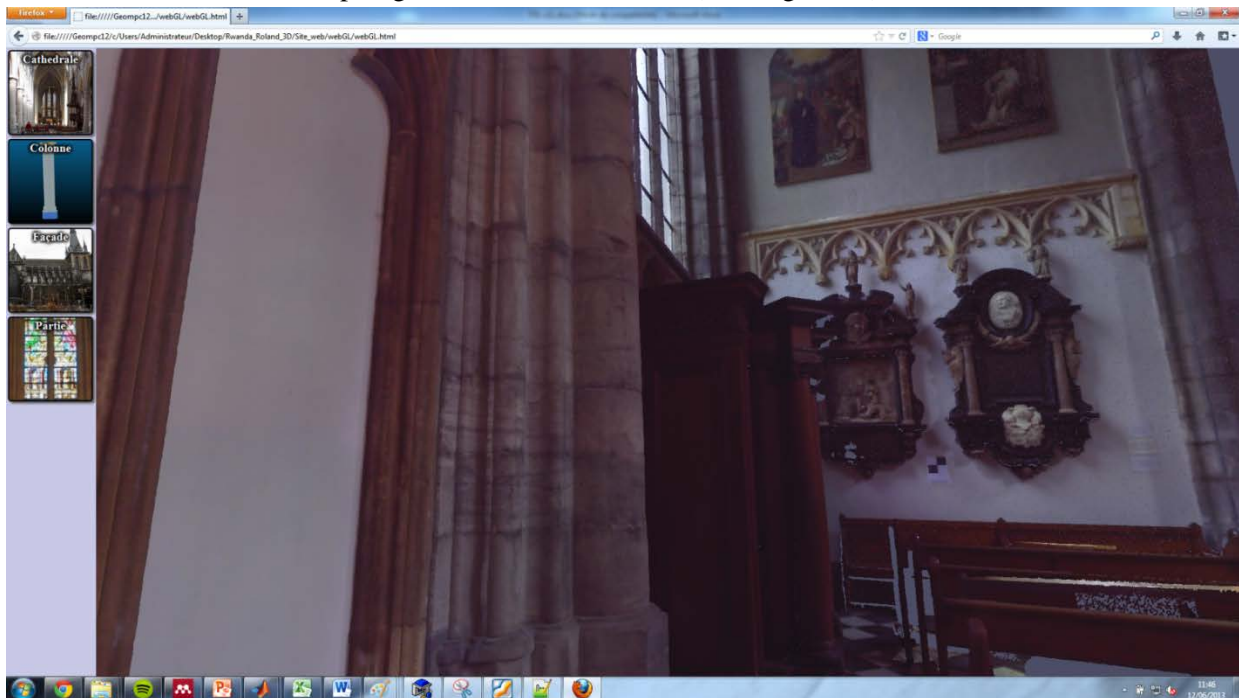


Figure 58 Capture d'écran réalisé de la visite virtuelle WebGL. Nuage de 500 millions de points en temps réel

Ensuite il a fallu organiser le nuage de point en octree, afin que l'affichage des points soit fonction de la position de la caméra. Ceci permet d'afficher et gérer seulement la partie visible par la caméra (repérée par son centre et son orientation), participant grandement à la fluidité de la visite. L'exécution d'un script Javascript<sup>34</sup> via la console windows permet de convertir rapidement le nuage de point total au format xyz ([X Y Z R G B] en partitions logiques d'octree. Il faut penser à changer de dossier (au moyen de la commande cd) et exécuter la ligne de commande suivante : `java -jar PotreeConverter.jar "chemin du fichier .xyz" "chemin du dossier de sortie"`.

Le temps d'exécution linéaire est fonction du nombre de point (environ 10 minutes pour 100 millions de points), au bout duquel un dossier est créé contenant tous les fichiers d'appel au nuage de point converti en format binaire pour une meilleure compression<sup>35</sup>. Ces fichiers d'appels sont la structure de l'octree, activés selon la position de la caméra. Pour l'intégration dans l'interface WebGL et Html, il faut modifier les fichiers sources suivant<sup>36</sup> :

- Pointcloud.js : Création des liens (Appel du fichier, référencement), paramétrage de la caméra (position initiale, paramètres de transformation) et définition de l'apparence des points (illumination, brillance, taille)
- Page.js : Gestion des shaders, des effets d'illumination, de l'affichage et du design des icônes de nuage de point (menu latéral), du nombre d'images par secondes

<sup>34</sup> Nécessite l'installation du JRE (Java Runtime Environment) pour pouvoir exécuter les programmes Java.

<sup>35</sup> en moyenne 75% de réduction de taille du fichier. Exemple du nuage de 500 millions : 20 Go en .xyz et 4 Go en .bin

<sup>36</sup> Utilisation de Notepad++ pour accéder au code source des fichiers javascripts (.js) html (.html) css (.css) et php (.php)



- Fichiers .js du dossier src (FrameBuffer.js, KeyListener.js, Viewport.js ...) : Gère toutes les interactions et l'affichage de la visite virtuelle.

### 6.2.3 La visite sur serveur

La paramétrisation des différents fichiers sur lesquels la visite WebGL est basée a été nécessaire pour arriver au résultat fluide en temps réel et sans temps d'attente. Les mouvements de caméra ont aussi dû être programmés pour une intuitivité et une bonne réactivité.

Le fichier potree.js (dans le dossier src) est le coordinateur de tous les développements Javascript, et est le fichier principalement édité. La ligne de code « Potree.Settings.LODMultiplier = 3.0 » a une grande influence sur la fluidité du nuage de point car elle gère les différents niveaux de détails : plus ce paramètre sera grand, plus le nombre de points affichés sera important pour une distance donnée à l'objet. A faible LOD il faudra se rapprocher de l'objet observé pour profiter de la même densité de point qu'à multiplicateur LOD élevé (voir Figure 59)



Figure 59 Visite de la cathédrale par différents niveaux de LOD : a) 5 ; b) 10 ; c) 25 ; d) 50

Ce multiplicateur sera donc d'une présence essentielle pour permettre l'utilisation de la visite virtuelle sur le plus grand nombre de supports (la carte graphique, la mémoire vive et le processeur conditionnant majoritairement la fluidité au vu du nombre de point affiché). La visite de la cathédrale Saint-Paul a donc été intégralement réalisée grâce à ces développements, et les images ci-dessous sont des captures d'écran du nuage de point en temps réel (et non des photos) lors de la visite virtuelle.

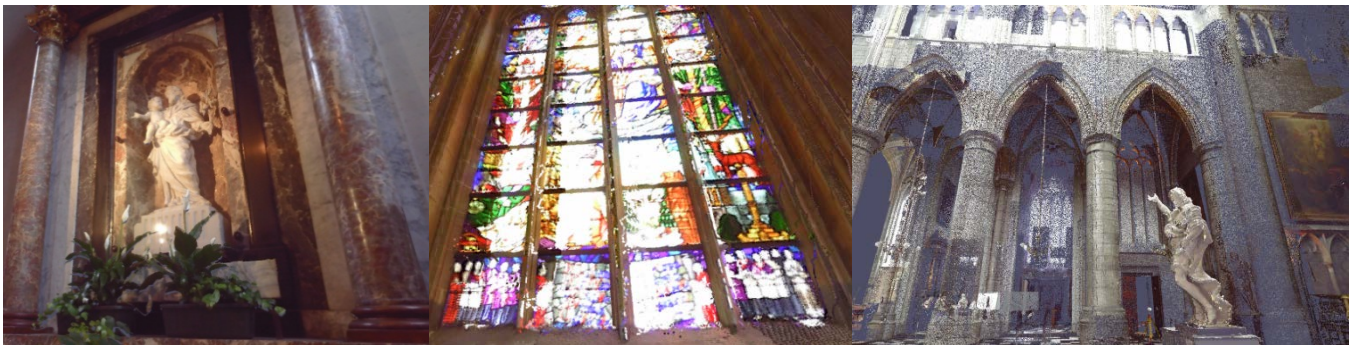


Figure 60 Capture d'écran de la visite virtuelle WebGL réalisée de la cathédrale Saint-Paul de Liège

## CONCLUSION

La lasergrammétrie est une science qui pour être maîtrisée et utilisée efficacement demande des connaissances et applications de domaines très variés. Ainsi, ce TFE est l'aboutissement et la mise en pratique de réflexions informatiques, topographiques, lasergrammétiques, infographiques, mathématiques et physiques.

L'acquisition de larges volumes de données au moyen d'un Scanner Laser n'est finalement qu'une mince partie -exigeant néanmoins précision et méthodologie- dans la reconstitution et la diffusion de modèles 3D. La chaîne de traitement définie et les méthodes développées jouent alors un rôle majeur dans la recherche d'une automatisation des processus de traitement de nuages de points. Nous avons ainsi proposé avec succès une méthode au coût minime et participant à une automatisation du processus de segmentation, essentielle pour une modélisation et analyse approfondie. Versatile car adaptée à des ouvrages d'arts aux détails fins autant qu'aux scènes simples composées de plans en tout genre, la méthode semi-automatique s'attache à une simplicité d'utilisation et de compréhension grâce à l'utilisation de paramètres intuitifs.

Différentes méthodes ont également été introduites pour la création de différents modèles pour différentes applications, diffusables aisément au moyen de PDF 3D. Les archéologues ont pu apporter une vision 3D et des mesures surfaciques à une science qui majoritairement s'appuie sur 2 dimensions. La modélisation hybride géométrie/maillage issue de l'analyse post-segmentation débouche sur un modèle simple léger mais précis, qui réduit le volume des données en conservant un critère de reconstitution du modèle réel. Lors de l'analyse avec le modèle maillé par la méthode Ball-Pivoting (la plus précise des méthodes comparées), nous avons vu la forte similarité et donc nous pouvons valider la modélisation.

Dans un souci applicatif complet, le développement d'une visite virtuelle WebGL innovante et saisissante a également été entrepris, ouvrant grandement les perspectives d'utilisation de nuages de points. L'affichage de millions/milliards de points RGB en temps réel sans temps de chargement via internet est réellement bénéfique dans la diffusion publique et l'utilisation de visites virtuelles.

Le développement de la méthode de segmentation proposée au sein du TFE, l'optimisation de l'algorithme et l'implémentation dans un logiciel (par exemple CloudCompare) en tant que plugin participerait grandement à la simplification de la chaîne de traitement, et à la recherche d'un gain de temps. Ceci nécessiterai alors un travail de développeur informatique, pour réduire le temps de traitement de l'algorithme proposé et « traduire » le script MATLAB optimisé dans un langage approprié à une intégration plugin.

En outre, il serait également intéressant de développer des algorithmes d'automatisation des processus de modélisation post-segmentation proposés. Ainsi, la chaîne de production de modèles 3D serait automatisée, complète et précise.

La réalisation de ce travail d'envergure sur la cathédrale Saint-Paul de Liège a réellement été une expérience passionnante de par son contexte mais également par la diversité des sujets approfondis. L'étude de la chaîne de production d'informations dans sa totalité m'a permis de mettre en parallèle les exigences des professionnels de la conservation du patrimoine avec l'analyse scientifique propre au métier d'ingénieur. Maîtriser différents logiciels complexes en quelques jours voir quelques heures fut une prérogative pour exploiter pleinement la science lasergrammétrique et ses applications.

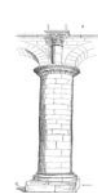
De plus, l'optique de recherche m'a particulièrement motivée et j'ai pu donner une conférence scientifique dans différentes langues dans le cadre de mes travaux de recherches (BGD 2013). Une publication prévue dans CIPA 2013 et une conférence en septembre sera l'aboutissement de ce TFE.

La polyvalence et la diversité bibliographique m'ont permis de compléter mes connaissances de différents domaines, en m'ouvrant des horizons variés grâce à une lecture en 4 langues (Français, Anglais, Espagnol, Allemand) d'ouvrages du monde entier. J'ai pu apprécier le potentiel de la discipline lasergrammétrique qui est devenue ma spécialité, et je peux aujourd'hui participer au rayonnement des technologies 3D.



## RÉFÉRENCES BIBLIOGRAPHIQUES

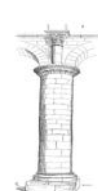
- Al Shawa M.**, 2006. Consolidation des nuages de points en lasergrammétrie terrestre. Strasbourg.
- Beder C., Förstner W.**, 2006. Direct solutions for computing cylinders from minimal sets of 3D points. *Computer Vision and Image Understanding*, pp. 1–12.
- Berkmann J., Caelli T.**, 1994. Computation of surface geometry and segmentation using covariance techniques. *IEEE Transactions on Visualization and Computer Graphics*, v. 35, pp. 848–853.
- Bernardini F., Mittleman J.**, 1999. The ball-pivoting algorithm for surface reconstruction. *Visualization and Computer Graphics, IEEE Transactions On*, v. 5, pp. 349–359.
- Bey A.**, 2011. Détection de cylindres dans les nuages de points de scènes industrielles basée sur l'utilisation de modèles 3D a priori. *Revue Electronique Francophone d'Informatique Graphique*, v. 5, pp. 1–10.
- Boehler W., Vicent M.B., Marbs A.**, 2003. Investigating laser scanner accuracy. *CIPA Symposium*, pp. 1–9.
- Borrmann D., Elseberg J., Lingemann K., Nüchter A.**, 2011. The 3D Hough Transform for plane detection in point clouds: A review and a new accumulator design. *3D Res*, v. 02, pp. 1–13.
- Boulaassal Hakim**, 2010. Segmentation et modélisation géométriques de façades de bâtiments à partir de relevés laser terrestres. École Doctorale Mathématiques, Sciences de l'Information et de l'Ingénieur. Strasbourg.
- Canciani M., Falcolini C., Saccone M., Spadafora G.**, 2013. From point clouds to architectural models: algorithms for shape reconstruction. *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, v. XL, pp. 25–26.
- Castillo E.**, 2013. Point Cloud Segmentation via Constrained Nonlinear Least Squares Surface Normal Estimates. *Recent UCLA Computational and Applied Mathematics Reports*, pp. 1–6.
- Chaperon T.**, 2002. Segmentation de nuage de points 3D pour la modélisation automatique d'environnements industriels numérisés. Ecole des Mines de Pairs.
- Chaperon T., Goulette F.**, 2001. Extracting cylinders in full 3D data using a random sampling method and the Gaussian image. *Computer Vision and Image Understanding*, pp. 35–42.
- Douadi L., Aldon M., Crosnier A.**, 2005. Variantes de l'Algorithme ICP pour le Recalage de Données 3D/Couleur. Montpellier.
- Douillard B., Underwood J.**, 2011. On the segmentation of 3D LIDAR point clouds. *Robotic and Automation (ICRA)*, pp. 2798 – 2805.
- Gerrard C.**, 2003. *Medieval archaeology: understanding traditions and contemporary approaches*. Routledge.
- Goldreich O.**, 2008. Computational complexity: a conceptual perspective. *ACM SIGACT News*,.
- Holz D., Behnke S.**, 2013. Fast range image segmentation and smoothing using approximate surface reconstruction and region growing. *12th International Conference on Intelligent Autonomous Systems*, Jeju Island.
- Hussain M.**, 2009. Efficient simplification methods for generating high quality LODs of 3D meshes. *Journal of Computer Science and Technology*, v. 24, pp. 604–608.
- Kakde H.**, 2005. *Range Searching using Kd Tree*.
- Kazhdan M., Bolitho M., Hoppe H.**, 2006. Poisson surface reconstruction. *Eurographics Symposium on Geometry Processing*, pp. 1–10.
- Klasing K., Althoff D.**, 2009. Comparison of surface normal estimation methods for range sensing applications. *Robotics and Automation*, pp. 3206–3211.
- Landes T., Grussenmeyer P.**, 2011. Les principes fondamentaux de la lasergrammétrie terrestre: systèmes et caractéristiques (partie 1/2). *XYZ*, v. 128, pp. 37–49.
- Landes T., Grussenmeyer P., Boulaassal H.**, 2011. Les principes fondamentaux de la lasergrammétrie terrestre: acquisition, traitement des données et applications (partie 2/2). *XYZ*, v. 129, pp. 25–38.
- Lerma J., Biosca J.**, 2005. Segmentation and filtering of laser scanner data for cultural heritage. *CIPA International Symposium*, pp. 1–6.
- Lewiner T, Mello V., Peixoto A., Pesco S.**, 2010. Fast generation of pointerless octree duals. *Computer Graphics Forum*, v. 29, pp. 1661–1669.
- Lewiner Thomas**, 2005. Mesh Compression from Geometry. Université Pierre et Marie Curie.



- Liu Y., Pottmann H., Wang W.**, 2006. Constrained 3D shape reconstruction using a combination of surface fitting and registration. *Computer-Aided Design*, v. 38, pp. 572–583.
- Luo D., Wang Y.**, 2008. Rapid extracting pillars by slicing point clouds. *Proc. XXI ISPRS Congress, IAPRS*, v. 37, pp. 215–218.
- Preiner R., Stefan J., Michael W.**, 2012. Auto Splats : Dynamic Point Cloud Visualization on the GPU. *Eurographics Symposium on Parallel Graphics and Visualization*, pp. 1–10.
- Rabbani T.**, 2006. Segmentation of point clouds using smoothness constraint. *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, v. 36, pp. 248–253.
- Rabbani T., Heuvel F. Van Den**, 2005. Efficient hough transform for automatic detection of cylinders in point clouds. *ISPRS WG III/3, III/4*, v. 3, pp. 60–65.
- Reshetyuk Y.**, 2009. Self-calibration and direct georeferencing in terrestrial laser scanning. Umea.
- Rusu R.**, 2010. Semantic 3D Object Maps for Everyday Manipulation in Human Living Environments. *KI-Künstliche Intelligenz*,.
- Rusu R., Blodow N.**, 2009. Close-range scene segmentation and reconstruction of 3D point cloud maps for mobile manipulation in domestic environments. *Intelligent Robots and Systems IROS IEEE/RSJ International Conference On*, pp. 1–6.
- Schnabel R., Wahl R., Klein R.**, 2006. Shape detection in point clouds. *Universität Bonn*,.
- Schnabel R., Wahl R., Klein R.**, 2007. Efficient RANSAC for Point Clouds. *Computer Graphics Forum*, v. 26, pp. 214–226. -Cloud
- Schnabel R., Wessel R., Wahl R., Klein R.**, 2008. Shape recognition in 3d point-clouds. *Conf. in Central Europe on Computer Graphics, Visualization and Computer Vision*,. p. 2.
- Tishchenko I.**, 2010. Surface Reconstruction from Point Clouds. Swiss Federal Institute of Technology Zurich.
- Wehr A., Lohr U.**, 1999. Airborne laser scanning—an introduction and overview. *ISPRS Journal of Photogrammetry and Remote Sensing*, v. 54, pp. 68–82.

## SITES INTERNET

Faro : [www.faro.com](http://www.faro.com)  
 Imag'ing : [www.imag-ing.com](http://www.imag-ing.com)  
 Leica : [www.leica-geosystems.com](http://www.leica-geosystems.com)  
 Tplm3D : [www.tplm-3d.fr](http://www.tplm-3d.fr)  
 Trimble : [www.trimble.com](http://www.trimble.com)  
 CloudCompare : [www.danielgm.net/cc](http://www.danielgm.net/cc)  
 Optech : [www.optech.ca](http://www.optech.ca)  
 Simple3D : [www.simple3D.com](http://www.simple3D.com)  
 Topcon : [www.topconpositioning.com](http://www.topconpositioning.com)  
 Autodesk : <http://www.autodesk.com>  
 TERAPOINTS Research : [www.cg.tuwien.ac.at](http://www.cg.tuwien.ac.at)



## *ANNEXES*



## Annexe 1 Caractéristiques du matériel FARO

Caractéristiques de performances Focus<sup>3D</sup>S

## Unité de mesure de distance

Intervalle d'ambiguïté : 153,49 m

Portée Focus<sup>3D</sup>S 120<sup>1</sup> : 0,60 m - 120 m en intérieur ou extérieur par éclairage faible et incidence normale sur une surface réfléchissante à 90 %Portée Focus<sup>3D</sup>S 20 : 0,60 m - 20 m à une incidence normale sur des surfaces mates réfléchissantes >10 %

Taux de scan (points /s) : 122 000 / 244 000 / 488 000 / 976 000

Incertitude de mesure<sup>2</sup> : ±2 mm à 10 m et 25 m, chacune avec une réflectivité de 90 % et 10 %

Bruit <sup>3</sup>	@10 m	@10 m - compression du bruit <sup>4</sup>	@25 m	@25 m - compression du bruit <sup>4</sup>
@ 90 % réfl.	0,6 mm	0,3 mm	0,95 mm	0,5 mm
@ 10 % réfl.	1,2 mm	0,6 mm	2,20 mm	1,1 mm

## Unité couleur

Résolution : couleur jusqu'à 70 mégapixels

Couleur dynamique : adaptation automatique de la luminosité

## Déflecteur

Champ de vision (vertical/horizontal) : 300° / 360°

Résolution (verticale/horizontale) : 0,009° (40 960 points 3D sur 360°) / 0,009° (40 960 points 3D sur 360°)

Vitesse max. de rotation du miroir : 5 820 rpm ou 97 Hz

## Laser (émetteur optique)

Performance du laser (cw Ø) : 20 mW (classe de laser 3R)

Longueur d'onde : 905 nm

Divergence du rayon : 0,19 mrad (0,011°)

Diamètre du rayon (à la sortie) : 3,0 mm, circulaire

## Gestion des données et commande

Mémoire des données : SD, SDHC™, SDXC™ ; carte de 32 GB fournie avec l'appareil

Commande du scanner : par écran tactile et WLAN

Nouvel accès WLAN : la commande du scanner à distance, la visualisation et le téléchargement des numérisations sont possibles sur des terminaux mobiles équipés de Flash®.

## Multi-Capteurs

Compensateur bi-axial : Nivelle chaque numérisation ; précision 0,015° (plage de mesure ±5°)

Capteur de hauteur : Un baromètre électronique permet de calculer la hauteur relative par rapport à une valeur de référence et de l'attribuer aux numérisations.

Boussole : La boussole électronique fournit aux numérisations des données d'orientation par rapport aux points cardinaux. Une fonction de calibration est disponible.



<sup>1</sup> Dépend de la lumière ambiante qui peut être une source de bruit. Une lumière vive comme celle du soleil peut réduire la portée effective du scanner. Par éclairage faible, la portée peut dépasser 120 m par incidence normale sur des surfaces hautement réfléchissantes. <sup>2</sup> L'incertitude de mesure est définie comme l'erreur maximale mesurée par le scanner sur cette distance depuis son point d'origine jusqu'au point sur la cible planaire. <sup>3</sup> Le bruit est défini comme l'écart standard des valeurs sur le meilleur plan d'ajustement. <sup>4</sup> Un algorithme de compression du bruit peut être activé pour calculer la moyenne des points moyens dans des sets de 4 ou 16, comprimant ainsi le bruit des données brutes par un facteur de 2 à 4. Informations susceptibles d'être modifiées sans indication préalable.

## Général

Alimentation électrique : 19 V (alimentation externe),  
14,4 V (batterie interne)

Consommation électrique : respectivement 40 W et 80 W  
(pendant la charge de la batterie)

Autonomie de la batterie : jusqu'à 5 heures

Température : 5° - 40°C

Humidité : sans condensation

Connecteur du câble : situé dans le support du scanner

Poids : 5,0 kg

Dimensions : 240 x 200 x 100 mm

Maintenance / Calibration : une fois par an

Sans erreur de parallaxe : Oui



Contract Holder

Figure 61 Documentation technique du scanner FARO FOCUS<sup>3D</sup>. Source : www.faro.com

## Annexe 2 Comparaison des suites logicielles de traitement de nuages de points

### SCENE

SCENE est un logiciel constructeur de la société FARO, et permet le traitement de nuages de points en étant spécialisé dans la consolidation des données. A l'issue des différents levés réalisés avec le scanner FARO FOCUS<sup>3D</sup>, nous obtenons différents fichiers .fls associés aux différents scans réalisés. L'avantage de ce type de fichier est la compression optimale des données, qui donne des fichiers relativement légers selon le nombre de points qu'ils contiennent. En contrepartie ils ne peuvent être traités que par le logiciel constructeur (et aujourd'hui TRIMBLE suite au contrat OEM et l'accord de distribution des scanners 3D signé avec FARO<sup>37</sup>).

Le logiciel permet notamment le recalage des scans entre eux, par une méthode automatique de compensation des déviations par moindres carrés, en s'appuyant sur des sphères homologues. Bien que pratique car automatique, cela impose de faire « confiance » à l'algorithme du logiciel qui ne fournit pas beaucoup d'information sur la méthode de recalage, et apparie très souvent des points qui ne devraient pas être homologues, ni même détectés (exemple : détection de 75 sphères dans une scène ou n'étaient présentes que 7 sphères). Dans notre cas, nous n'avons pas pu utiliser ceci car certaines sphères avaient des hauteurs différentes entre 2 scans, et le renommage automatisé donnait des résultats aberrants. Manuellement, il est possible de rentrer les 7 paramètres d'une transformation pour imposer un recalage particulier.

SCENE est très performant pour afficher des milliards de points mais ne permet pas la mesure directe sur la vue 3D, ce qui en limite grandement l'usage après recalage. De plus, il ne couvre que la partie consolidation et importation des données, imposant une exportation en .asc, .xyz, .bin, .pts, ... pour la gestion par un logiciel tiers.

+	-
Traite de gros nuages de points (>1 Milliard) (64bits)	Ne traite que la partie consolidation
Gère les .fls	Payant
Permet le recalage des scans	Interface pas très intuitive
Large gamme d'exports possibles	Mode vue 3D limité
Très bonne compression des données	Un rapport d'état du recalage peu parlant
Possibilité de transformation 4-6 paramètres	Incohérence de certains algorithmes

### REALWORKS

Le logiciel TRIMBLE REALWORKS (constructeur) s'affaire à l'ensemble de la chaîne de traitement de nuage de points. Cependant, contrairement à SCENE, REALWORKS ne gère pas des ensembles de point trop importants (car version 32bits, donc ne prenant en compte que 3Go de mémoire RAM) et réalisera un décimage au-delà de 10 000 000 de points. Son interface claire et épurée découpe la chaîne de traitement en 3 étapes via un menu déroulant accessibles à tout moment :

#### Recalage

Tout comme le logiciel FARO SCENE, il permet un recalage à base de cibles ou de nuages, et comprend des outils de géoréférencement. Il fournit également un rapport clair et détaillé concernant le recalage effectué, et des méthodes de recalage automatique à base de sphères, plans, cibles, ...

#### OfficeSurvey

Cet onglet permet d'effectuer presque toutes les opérations de manipulation sur un nuage de point dont les essentielles : Segmentation (manuelle par sélection rectangulaire ou polygonale) ; Échantillonnage (aléatoire, par scans, par intensité, par discontinuités en offrant un outil de détection d'arrêtes performant, ou par intensité) ; Création de plans de coupes (multiples, suivant un plan, un axe, et permettant la création automatique de polygones 2D); Dessin de courbes de niveaux; Calculs de profils en long, en travers; Dessin de polygones; Calculs de volume; ...

<sup>37</sup> Pour plus d'informations, voir <http://investor.trimble.com/releasedetail.cfm?ReleaseID=703766>



De plus, des outils de mesures et d'analyse (surface, écarts, ...) adaptés, de création d'ortho projection et d'inspection 3D permettent une analyse des nuages de points. Des outils de modélisations par reconnaissance de forme (pointage manuel et utilisation très basique encore émergente) ou pour par une méthode de triangulation 2D qui agit sur la projection d'un nuage de points 3D sur une surface 2D (Modélisation très approximative qui n'est pas encore au point) permettent de dégager une géométrie de ces nuages de point.

### Modélisation

Ce dernier onglet permet la modification et la gestion des géométries, à l'aide de fonction comme la duplication ou l'intersection.

C'est un logiciel très performant et intuitif, qui demanderait une plus grande précision des algorithmes utilisés ainsi qu'une méthode de modélisation par triangulation plus aboutie pour pouvoir réaliser le traitement de nuages de points seulement dans un environnement unique.

+	-
Logiciel relativement complet	Ne gère pas de gros nuages de points (32bits)
Interface claire et complète	Pas de segmentation automatique
Prise en charge des .fls	Peu d'automatisation (reconnaissance de forme,...)
Modélisation géométrique avec rapport de précision	Outil de maillage aux résultats non convaincant et gestion des géométries basique
Outils de coupes très performant et utile	Pas de texturation possible
Orthophoto et orthoprojection	Payant

### 3DReshaper

Ce logiciel de la société française Technodigit est très performant pour créer un modèle maillé et texturé à partir d'un nuage de point. Il propose également des options de recalage (Best fit, 6 paramètres, suivant des axes) et de nettoyage poussé.

L'intérêt principal de ce logiciel réside dans la partie avancée de modélisation qu'il propose. Les outils de création de géométrie et de modèles maillés sont aboutis, et donnent des résultats vraiment convaincants. De plus, il offre des possibilités d'édition de ces géométries au moyen d'outils simples à prendre en main, présentés dans une interface intuitive.

Il s'agit donc d'un logiciel vraiment complet, qui souffre principalement de ne pas être un logiciel constructeur, et donc pas le mieux adapté pour la consolidation liée aux différents scanners 3D.

+	-
Interface épurée	Pas de reconnaissance de sphères
Nettoyage, Réduction du bruit, Échantillonnage Recalage, Maillage	Pas de recalage lié aux cibles utilisées lors de levés 3D
Outil relativement complet	Payant
Méthode de maillage très efficace et modification très facile et pertinente	Pas d'outils de segmentation automatique
Manipulation des points et calcul des distance aisé	Outil d'extraction de formes ne permet pas l'extraction sur des formes non segmentées
Texturation possible et précise	Payant
Extraction de lignes caractéristiques	Payant
Outil de coupe performant	

### CloudCompare

Ce logiciel open-source a le gros avantage sur ses concurrents d'être gratuit (contrairement à tous les logiciels cités ci-dessus dont une simple licence coûte entre 6000€ et 15000€). Il est très efficace pour le traitement de gros nuages de points, et pour les manipulations, modifications et l'analyse de nuages de points. Ce qui est agréable dans cet environnement, c'est que toutes les fonctions utilisées, renseignent l'utilisateur sur quel algorithme est utilisé. Actuellement, il est possible d'utiliser la librairie PCL, pour par exemple déterminer les





normales et courbes, ou encore d'échantillonner suivant la distance, ou d'affiner le modèle par MLS<sup>38</sup> ... Tous les algorithmes utilisés sont clairs et précis, et indiquent le nom de la méthode utilisée. Il existe également un algorithme de segmentation par reconnaissance de forme (RANSAC) en version plugin, ou de modélisation par reconstruction de poisson (Très exigeant).

Il existe beaucoup d'outils statistiques pour étudier le nuage de point, et intégrer de nouvelles données aux données existantes.

Ce logiciel n'a pas prétention et n'a pas pour but de traiter l'ensemble de la chaîne de traitement, mais il devrait être la référence pour l'analyse et le traitement de nuage de point après consolidation. Étant un logiciel open source, il est possible d'intégrer des développements propres et plugins fonctionnant dans le logiciel, ce qui étend considérablement son utilisation avec les outils appropriés.

+	-
Gratuit	Demande de bonnes connaissances
Très performant	Ne permet pas l'importation de fichiers propriétaires
Technique et efficace	Pas d'outils de création de coupes
Export et import de plusieurs formats	Pas d'outils de consolidation/recalage
Gestions très fluide de gros nuages de points	
Intégration de la librairie PCL <sup>39</sup>	
Information avancées sur les fonctions utilisées	
Outils statistiques et d'analyse intéressants	

Étant donné le manque de formation sur ce logiciel, j'ai créé un tutoriel de prise en main (voir annexe 2)

### MeshLab

MeshLab est un logiciel libre avancé de traitement de maillages. Il permet ainsi à partir d'un nuage de point (format .ply, .xyz, .asc ...) de créer un modèle surfacique, qui pourra par la suite être texturé et utilisé pour différentes applications virtuelles. Il est totalement gratuit, et propose des outils extrêmement puissants pour la création de maillages et géométries à partir des nuages de points.

Très technique et avancé, il est difficile d'accès pour les non-initiés : l'interface n'est pas très claire, la manipulation non intuitive et l'utilisation des différents algorithmes demandent de solides connaissances pour rentrer des paramètres adaptés. Cependant, lorsque maîtrisé, les possibilités de nettoyage, échantillonnage, maillage, texturation et de mapping en font un outil de choix (là encore, les algorithmes utilisés sont renseignés et brièvement décrits).

Seulement, bien que le potentiel de ce logiciel soit élevé, la présence de bugs majeurs entraîne une expérience utilisateur désagréable et frustrante, qui empêche l'utilisation de ce logiciel pour un certain nombre d'applications.

Un tutoriel pour la création d'un modèle maillé et texturé à partir d'un nuage de point coloré a été créé (voir annexe 3)

+	-
Gratuit	Demande de bonnes connaissances
Un potentiel conséquent	Des bugs majeurs aléatoires
Technique et efficace	Interface et interaction manuelle limitée
Des algorithmes testés et approuvés	Agit en amont de la chaîne de traitement
Une large gamme d'outils pour un usage adapté à toute situation	Au-delà de quelques millions de points, la gestion et le traitement bug souvent
Information sur les fonctions utilisées	Adapté pour des

### Suite Autodesk

La suite Autodesk est une solution très complète dans les domaines de CAO/DAO. Cependant, ce qui nous intéresse ici est un module spécifique, Kubit PointCloud Pro, pour la gestion des nuages de points. Ce plugin s'insère dans l'environnement Autocad et permet la gestion simple de nuages de plusieurs millions de points.

<sup>38</sup> Séquence de longueur maximale (Maximum Length Sequence), voir (Herman, 2006)

<sup>39</sup> Point Cloud Library : projet open source pour le traitement de nuage de points. Pour plus d'informations, voir <http://pointclouds.org/>

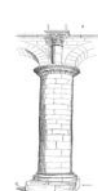


Il permet la manipulation des nuages issus de tous types de scanners depuis différents formats (hors formats constructeur). Plusieurs nuages de points peuvent être édités, et les coupes peuvent être facilement réalisées suivant des courbes AutoCAD ou plans définis par l'utilisateur. La version Pro permet de modéliser suivant différentes méthodes, et inclus une détection des collisions, ainsi que des outils de modélisation géométrique relativement complets.

Il s'agit d'un logiciel intéressant pour l'approche AutoCAD des nuages de points, qui permet d'utiliser des outils AutoCAD et d'effectuer simplement des mesures sur les différentes vues disponibles.

Les fonctionnalités proposées sont pour le moment assez basiques mais un développement futur permettrait aux utilisateurs d'AutoCAD de garder leurs repères sans se perdre dans la jungle des logiciels de traitement 3D.

+	-
Prise en main logicielle facile	Échantillonnage balbutiant
Opère dans l'environnement AutoCAD	Pas d'outils de segmentation
Plusieurs outils de modélisation	Peu d'automatisation (reconnaissance de forme,...)
Outils de coupes très performant et utile	Payant pour avoir accès aux fonctionnalités intéressantes
Création d'orthophoto aisée	



## Annexe 3 Tutoriel CloudCompare


CloudCompare est un logiciel très facile à prendre en main, et est doté d'une interface intuitive.

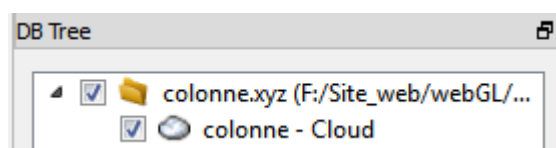
Dans ce tutoriel, je vais développer les fonctions utiles pour l'édition de nuages de points obtenus par scanner Laser.

CloudCompare permet d'importer des nuages de points et des maillages de différents formats :

Extension	Type	P	M	RGB	NG	N	S	Autre	Description
asc, txt, neu, xyz, etc.	ascii	✓		✓	✓	✓	✓		nuage de points ASCII
bin	binaire	✓		✓		✓	✓		nuage(s) de points, format binaire spécifique à CloudCompare
ply	ascii / binaire	✓	✓	✓	✓	✓	✓	✓	maillage (Stanford)
obj	ascii		✓			✓		✓	maillage (Wavefront)
soi	ascii	✓			✓				nuage(s) (Soisic, Mensi)
(c)bin	binaire	✓		✓					nuage (C. Hernandez)
pn	binaire	✓				✓			nuage (point + normale)
pvc, pcv	binaire	✓					✓		nuage (point + valeur)
icm	ascii							✓	association nuages/images

P : Points  
M : Maillages  
RGB : Couleurs (Red, Green, Blue)  
NG : Niveaux de Gris  
N : Normales  
S : Scalaires

Une fois le nuage de point importé (via le raccourci ) et chargé, il est immédiatement visible dans la fenêtre de visualisation, et apparaît dans l'arbre de données DB Tree. On note la création d'un dossier spécifique pour chaque nuage de point importé. De plus, si le nuage de point fait plus de 128 millions de points, CloudCompare divise le fichier initial en plusieurs nuages de 128 millions de points maximums.




Property	State/Value
<b>CC Object</b>	
Name	colonne - Cloud
Unique ID	4
Children	0
Visible	<input checked="" type="checkbox"/>
Colors	<input checked="" type="checkbox"/>
Show name (in 3D)	<input type="checkbox"/>
Current Display	3D View1
Box dimensions	X: 1.70135 Y: 1.73611 Z: 7.55991
Box center	X: -0.00792602 Y: 0.00275549 Z: 0.0075531
<b>Cloud</b>	
Points	2,000,000
Global shift	(0.00;0.00;0.00)
Point size	Default

Lorsque le nuage ou un objet créé est sélectionné, une bounding box filaire jaune apparaît autour, et l'on peut voir les propriétés de l'objet dans la fenêtre propriété. Y sont détaillés le nom du nuage, le nombre d'enfants, l'affichage courant, les dimensions de la bounding box, le nombre de points, ...

Ce n'est qu'alors qu'il est possible d'éditer le nuage de point sélectionné. Bien qu'il soit possible d'effectuer des transformations (translations et rotations) pour aligner et recalibrer des nuages de points, le but de ce tutoriel est de donner les outils nécessaires à une édition postérieure à la consolidation. Ainsi, je vais détailler plusieurs fonctions utiles à ce processus, et largement utilisée pour obtenir les résultats

de ce travail de fin d'études.



**Point Picking** : Cette fonctionnalité permet d'effectuer plusieurs types de mesures directes sur le nuage de point ou maillage. Il fait apparaître le bandeau , qui offre la possibilité (de gauche à droite), d'obtenir les informations d'un point (ID, Coordonnées, Couleur, ...), la distance entre deux points, les informations d'un triangle constitué par trois points (angles, aire, segments, ...), définir un label rectangulaire, sauvegarder les mesures effectuées dans un fichier texte.



**Point List Picking** : Cette fonctionnalité permet de sélectionner plusieurs points qui seront ajoutés à une liste qu'il est possible d'exporter au format texte.




**Subsample (Échantillonnage)** : Sans doute la fonction la plus utilisée, elle permet d'accéder à plusieurs méthodes d'échantillonnages :

- Aléatoire : méthode la plus rapide qui permet un premier échantillonnage de manière aléatoire lorsque la densité de point est bien trop élevée et que l'on veut réaliser un support aux futurs échantillonnages. Pour des nuages particulièrement volumineux, il s'agit d'une méthode efficace en temps de traitement.
- Espace : Méthode qui consiste à échantillonner selon une distance minimale entre deux points. Cette méthode est chronophage pour de larges nuages de points ou lorsque la distance voulue n'est pas adaptée, puisque dans un premier temps l'algorithme utilisé impose la construction d'un Octree pour ensuite travailler dessus. Cette méthode permet un contrôle absolu de la densité désirée en fonction de la précision voulue. Néanmoins, pour des nuages de point trop volumineux, l'algorithme peut demander plusieurs minutes voir quelques heures avant de renvoyer un résultat.
- Octree : Cette méthode permet un échantillonnage régulier suivant un le schéma d'un octree (suivant un axe z). Il y a 10 niveaux possibles d'échantillonnages, qui fournissent un échantillonnage similaire à des coupes transversales, plus ou moins nombreuses suivant le niveau de subdivision déterminé.



**Composants connectés** : Cet outil permet de segmenter le nuage de point en plusieurs morceaux qui comportent des points connectés les uns aux autres, c'est-à-dire ayant une distance inférieure au seuil déterminé suivant le niveau de l'octree choisi (de 1 à 10). Il est possible de déterminer le nombre de point minimum qu'un segment peut comporter pour être créée. Cet outil est particulièrement utile pour éliminer le bruit ou des points aberrants.



**Segmentation** : Cet outil permet d'effectuer une segmentation manuelle par sélection rectangulaire ou polygonale. Il fait apparaître un bandeau  qui une fois la sélection effectuée (clique gauche pour ajouter un point au polygone de sélection; clique droit pour fermer le polygone), permet de conserver activement la partie à l'intérieur ou à l'extérieur de la sélection. Les différents segments sont créés, et ne suppriment pas l'original (qui est conservé).



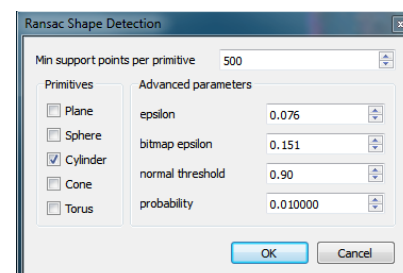
**Estimation de normales et courbures** : Cet algorithme issu du plugin PCL (Point Cloud Library) permet de calculer rapidement et de manière robuste les normales et courbures de chaque points suivant deux approches :

- Rayon de recherche : cherche les voisins pour l'estimation de la normale et de la courbure de chaque point dans un rayon défini par l'utilisateur
- KNN : Utilise une recherche par plus proches voisins pour l'estimation de la normale et de la courbure.

La deuxième méthode avec l'utilisation des 30 voisins les plus proches est la plus robuste pour une estimation optimale.



**Détection et segmentation RANSAC** : Utilise l'algorithme défini dans (Schnabel *et al.*, 2007) pour détecter les formes géométriques d'un nuage de point. Bien que les différents paramètres et les différentes primitives reconnaissables permettent théoriquement une grande flexibilité, la détection de forme ne permet pas dans notre cas de détecter les différents cylindres d'une colonne.



## Annexe 4 Rapport Technique de Levé et recalage de la Cathédrale Saint-Paul de Liège

Instrument : Scanner Laser FARO FOCUS<sup>3D</sup>

Journées terrains : 5 jours

Nombre de scans total : 66 scans

### 12-13-14 Juin 2012 : Levé intérieur de la cathédrale

Nombre de scans : 61 scans

Références : S13 – S14 – S20 – S1 – S19 – S15

3 jours de levés : 12 Juin/13 Juin/14 Juin 2012

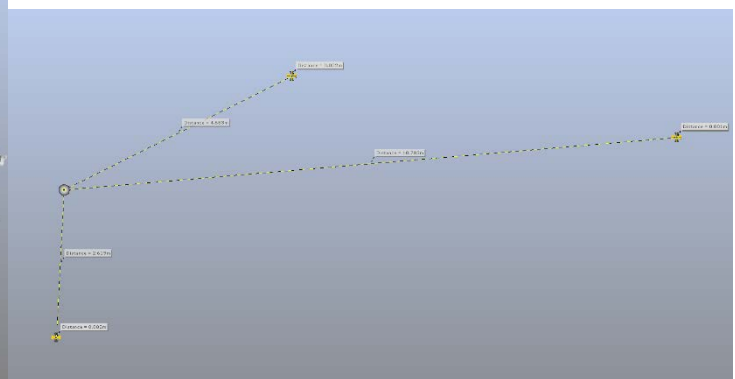
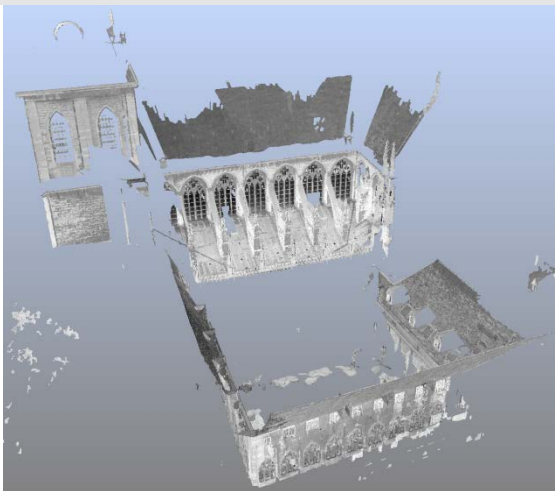
Coordonnées globales : <input checked="" type="checkbox"/>	
Nom de sphère :	S8
Nom :	SphereFit
Rayon :	0.0725 [m]
Position :	1019.875972 987.342655 126 [m]
Nombre de points de scan:	392
Déviaton de rayon :	0 [mm]
Déviaton de position :	1.104055 [mm]
Distance de point de scan:	0.848149 [mm]
Dérive de point de scan:	-0.041677 [mm]

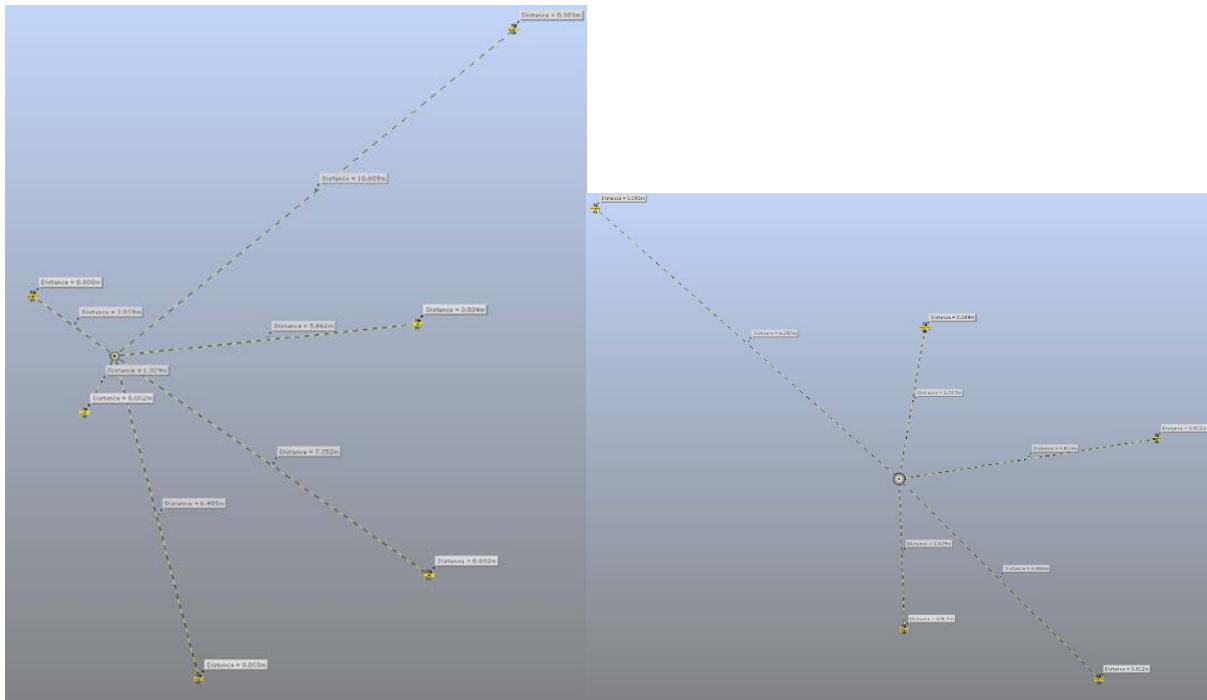
References

- S1
- S8
- S9
- TRN\_S1
- TRN\_S15
- TRN\_S7

Rapport de recalages :

### Toit





/Scans/CALAGE\_OK/CLOITRE\_HAUT/ScanManager

GestScanning Résultats de scanning Tensions de réf.

Tensions pesées Hiérarchie totale

Référence	Tens...	Scanning 1	Scanning 2
CH_S4	0.0057	Cathédrale_Liège.3_Scan_005	Cathédrale_Liège.3_Scan_002
CH_S1	0.0053	Cathédrale_Liège.3_Scan_005	Cathédrale_Liège.3_Scan_006
CH_S13	0.0050	Cathédrale_Liège.3_Scan_001	Cathédrale_Liège.3_Scan_004
CH_S8	0.0049	Cathédrale_Liège.3_Scan_006	Cathédrale_Liège.3_Scan_005
CH_S14	0.0047	Cathédrale_Liège.3_Scan_006	Cathédrale_Liège.3_Scan_005
CH_S1	0.0043	Cathédrale_Liège.3_Scan_005	Cathédrale_Liège.3_Scan_002
CH_S1	0.0041	Cathédrale_Liège.3_Scan_006	Cathédrale_Liège.3_Scan_002
CH_S4	0.0037	Cathédrale_Liège.3_Scan_003	Cathédrale_Liège.3_Scan_005
CH_S9	0.0033	Cathédrale_Liège.3_Scan_004	Cathédrale_Liège.3_Scan_005
CH_S1	0.0032	Cathédrale_Liège.3_Scan_001	Cathédrale_Liège.3_Scan_006
CH_S1	0.0032	Cathédrale_Liège.3_Scan_001	Cathédrale_Liège.3_Scan_005
CH_S8	0.0031	Cathédrale_Liège.3_Scan_005	Cathédrale_Liège.3_Scan_007
CH_S5	0.0026	Cathédrale_Liège.3_Scan_001	Cathédrale_Liège.3_Scan_004

Statistiques coefficientées

Principal :  Écart :

Min :  Max :

Enregistrement fin




/Scans/CALAGE\_OK/CLOITRE\_HAUT/ScanManager

GestScanning Résultats de scanning Tensions de réf.

Tensions pesées Hiérarchie totale

Référence	Tens...	Scanning 1	Scanning 2
◊ CH_S8	0.0031	Cathédrale_Liège.3_Scan_005	Cathédrale_Liège.3_Scan_007
◊ CH_S5	0.0026	Cathédrale_Liège.3_Scan_001	Cathédrale_Liège.3_Scan_004
◊ CH_S3	0.0024	Cathédrale_Liège.3_Scan_001	Cathédrale_Liège.3_Scan_002
◊ CH_S9	0.0023	Cathédrale_Liège.3_Scan_001	Cathédrale_Liège.3_Scan_004
◊ CH_S4	0.0021	Cathédrale_Liège.3_Scan_003	Cathédrale_Liège.3_Scan_002
◊ CH_S8	0.0021	Cathédrale_Liège.3_Scan_006	Cathédrale_Liège.3_Scan_007
◊ CH_S7	0.0018	Cathédrale_Liège.3_Scan_001	Cathédrale_Liège.3_Scan_002
◊ CH_S2	0.0017	Cathédrale_Liège.3_Scan_003	Cathédrale_Liège.3_Scan_002
◊ CH_S9	0.0016	Cathédrale_Liège.3_Scan_001	Cathédrale_Liège.3_Scan_005
◊ CH_S1	0.0014	Cathédrale_Liège.3_Scan_001	Cathédrale_Liège.3_Scan_002
◊ CH_S12	0.0011	Cathédrale_Liège.3_Scan_006	Cathédrale_Liège.3_Scan_007
◊ CH_S15	0.0010	Cathédrale_Liège.3_Scan_006	Cathédrale_Liège.3_Scan_007
◊ CH_S16	0.0004	Cathédrale_Liège.3_Scan_006	Cathédrale_Liège.3_Scan_007

Statistiques coefficientées

Principal :   Écart :

Min :  Max :

Enregistrement fin


/Scans/CALAGE\_OK/CLOITRE\_HAUT/ScanManager

GestScanning Résultats de scanning Tensions de réf.

Tensions pesées Hiérarchie totale

Référence	Tens...	Scanning 1	Scanning 2
◊ CH_S9	0.0016	Cathédrale_Liège.3_Scan_001	Cathédrale_Liège.3_Scan_005
◊ CH_S1	0.0014	Cathédrale_Liège.3_Scan_001	Cathédrale_Liège.3_Scan_002
◊ CH_S12	0.0011	Cathédrale_Liège.3_Scan_006	Cathédrale_Liège.3_Scan_007
◊ CH_S15	0.0010	Cathédrale_Liège.3_Scan_006	Cathédrale_Liège.3_Scan_007
◊ CH_S16	0.0004	Cathédrale_Liège.3_Scan_006	Cathédrale_Liège.3_Scan_007
⊗ Inclinomètre	0.0000	Cathédrale_Liège.3_Scan_005	Cathédrale_Liège.3_Scan_006
⊗ Inclinomètre	0.0000	Cathédrale_Liège.3_Scan_001	Cathédrale_Liège.3_Scan_006
⊗ Inclinomètre	0.0000	Cathédrale_Liège.3_Scan_002	Cathédrale_Liège.3_Scan_006
⊗ Inclinomètre	0.0000	Cathédrale_Liège.3_Scan_003	Cathédrale_Liège.3_Scan_006
⊗ Inclinomètre	0.0000	Cathédrale_Liège.3_Scan_006	Cathédrale_Liège.3_Scan_007
⊗ Inclinomètre	0.0000	Cathédrale_Liège.3_Scan_004	Cathédrale_Liège.3_Scan_006
⊗ Inclinomètre	0.0000	Cathédrale_Liège.3_Scan_004	Cathédrale_Liège.3_Scan_005
⊗ Inclinomètre	0.0000	Cathédrale_Liège.3_Scan_005	Cathédrale_Liège.3_Scan_007

Statistiques coefficientées

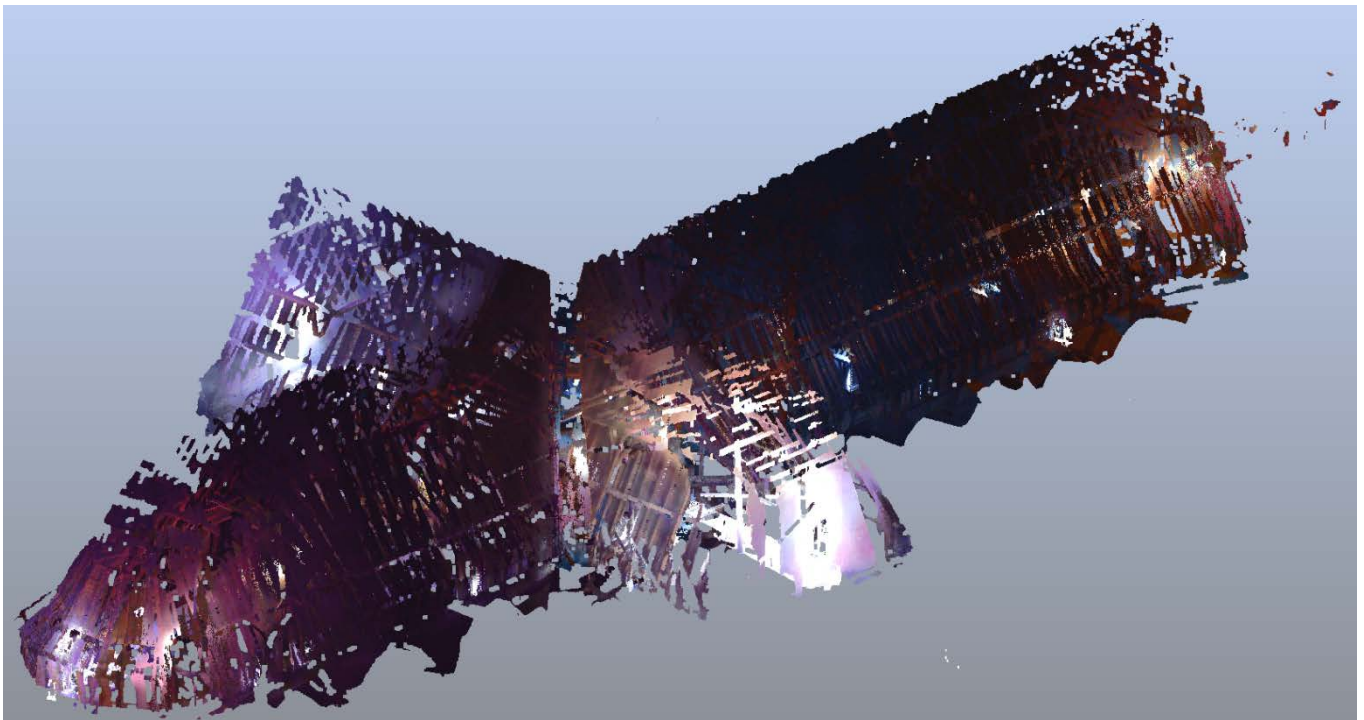
Principal :   Écart :

Min :  Max :

Enregistrement fin



### Grenier



Coordonnées globales:

Nom de scan: Cathédrale\_Liège\_3\_Scan\_008

Nom: ScanFit

Inclinomètre:  Nivelier scan selon de l'inclinomètre

Position: 1009.602574 970.066512 126.01321 [m]

Axe d'orientation: 0.003747268 -0.012290779 0.99991

Angle: 31.818754 [°]

Tension normalisée: 3.859208

Distance de point: 3.859208 [mm]

Dérive de point: 0 [mm]

Décalage longueur: 0.013209 [mm]

Décalage angulaire: 0.009186 [°]

Décalage orthogonal: 2.212087 [mm]

Décalage selon de l'inclinomètre: 0 [°]

Objet	Dist. [mm]	Long. [mm]	Angulaire
/Scans/CALAGE_O...	3.7964	-1.9183	0.0115
/Scans/CALAGE_O...	2.6098	0.9705	0.0073
/Scans/CALAGE_O...	8.6607	-7.9829	0.013
/Scans/CALAGE_O...	9.7381	9.4884	0.0134

Affichage

Coordonnées globales:

Nom de scan: Cathédrale\_Liège\_3\_Scan\_009

Nom: ScanFit

Inclinomètre:  Nivelier scan selon de l'inclinomètre

Position: 1027.513458 963.911487 126.15504 [m]

Axe d'orientation: -0.0016273981 -0.0042201985 0.999

Angle: 132.670939 [°]

Tension normalisée: 6.201247

Distance de point: 6.201247 [mm]

Dérive de point: 0 [mm]

Décalage longueur: -0.578803 [mm]

Décalage angulaire: 0.026798 [°]

Décalage orthogonal: 2.293646 [mm]

Décalage selon de l'inclinomètre: 0 [°]

Objet	Dist. [mm]	Long. [mm]	Angulaire
/Scans/CALAGE_O...	3.7964	-1.1608	0.0342
/Scans/CALAGE_O...	2.6098	-2.0105	0.0396
/Scans/CALAGE_O...	8.6607	-8.5509	0.0184
/Scans/CALAGE_O...	9.7381	9.407	0.015

Affichage

Coordonnées globales:

Nom de scan: Cathédrale\_Liège\_3\_Scan\_010

Nom: ScanFit

Inclinomètre:  Nivelier scan selon de l'inclinomètre

Position: 995.587776 977.13453 126.37984 [m]

Axe d'orientation: -0.0020876507 -0.00057867689 0.999

Angle: 102.973556 [°]

Tension normalisée: 1.635744

Distance de point: 1.635744 [mm]

Dérive de point: 0 [mm]

Décalage longueur: 0.346431 [mm]

Décalage angulaire: 0.018425 [°]

Décalage orthogonal: 1.346407 [mm]

Décalage selon de l'inclinomètre: 0 [°]

Objet	Dist. [mm]	Long. [mm]	Angulaire
/Scans/CALAGE_O...	1.5731	0.1531	0.0239
/Scans/CALAGE_O...	2.1645	1.5685	0.0065
S1	1.542	-0.9228	0.0286
S2	1.4952	1.0724	0.0157

Affichage

Coordonnées globales:

Nom de scan: Cathédrale\_Liège\_3\_Scan\_012

Nom: ScanFit

Inclinomètre:  Nivelier scan selon de l'inclinomètre

Position: 986.916446 981.157179 126.340633 [m]

Axe d'orientation: 0.0088468253 0.0019615439 -0.999

Angle: 78.932139 [°]

Tension normalisée: 2.191157

Distance de point: 2.191157 [mm]

Dérive de point: 0 [mm]

Décalage longueur: 0.15963 [mm]

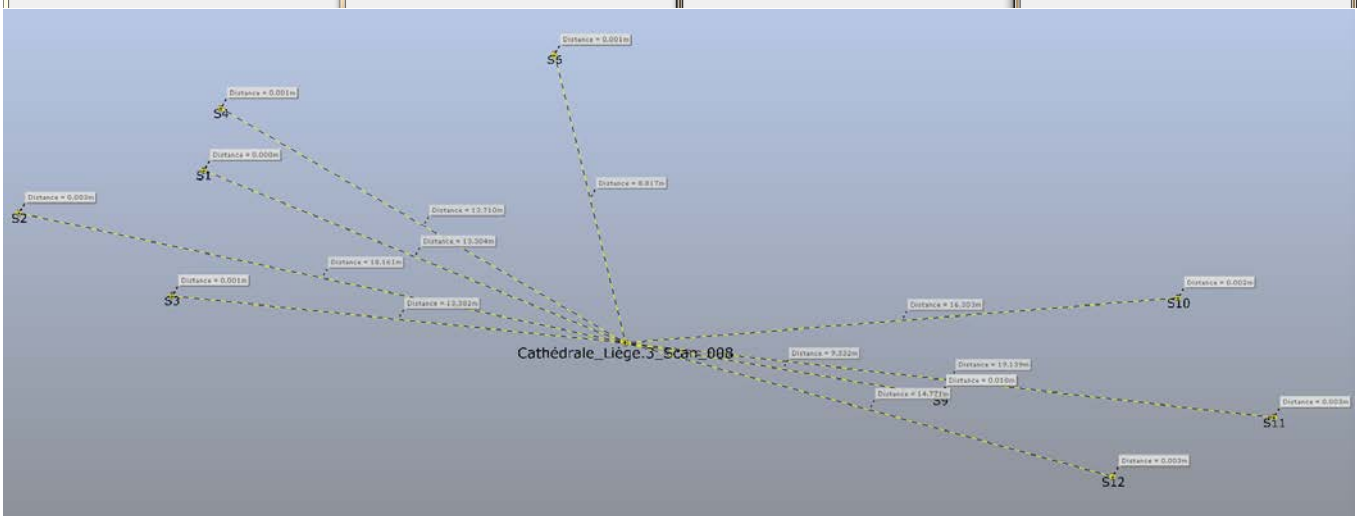
Décalage angulaire: 0.010024 [°]

Décalage orthogonal: 1.826835 [mm]

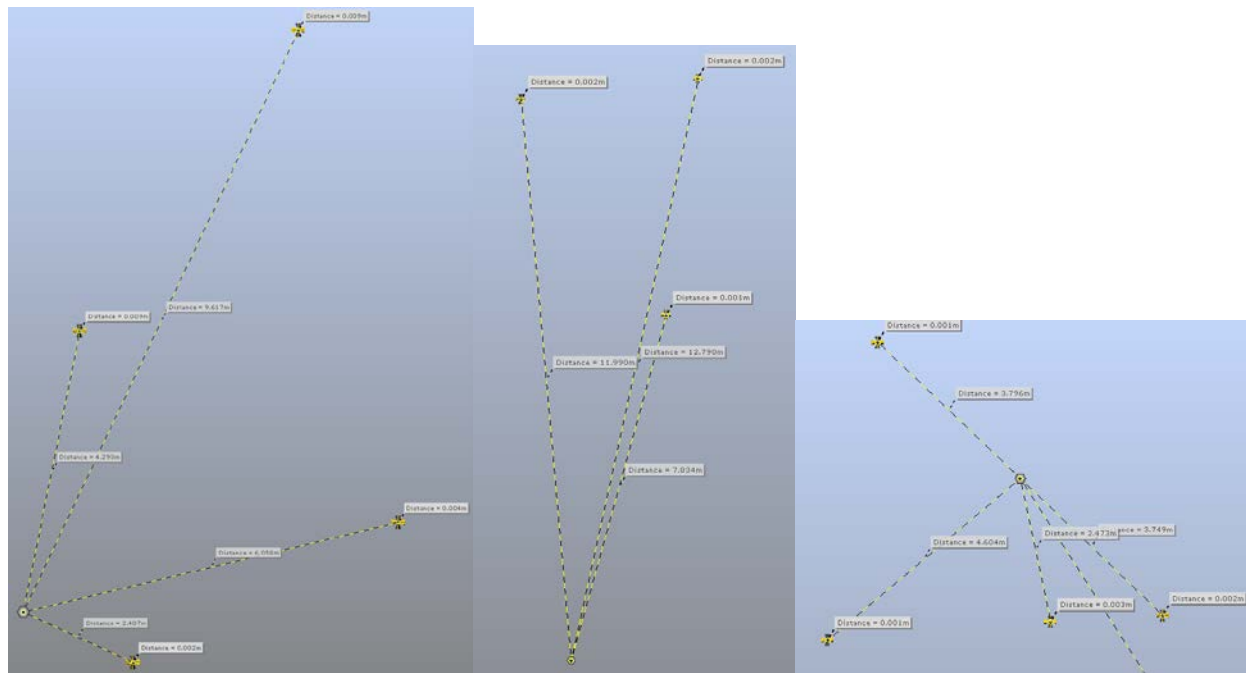
Décalage selon de l'inclinomètre: 0 [°]

Objet	Dist. [mm]	Long. [mm]	Angulaire [°]
S1	2.4489	1.7949	0.008
S2	1.7804	-0.1631	0.013
S3	2.3431	-1.1529	0.0091

Affichage







/Scans/CALAGE\_OK/Toiture/ScanManager

GestScanning Résultats de scanning Tensions de réf.

Tensions pesées  Hiérarchie totale

Référence	Tens...	Scanning 1	Scanning 2
◊ G_S9	0.0097	Cathédrale_Liège.3_Scan_008	Cathédrale_Liège.3_Scan_009
◊ G_S12	0.0087	Cathédrale_Liège.3_Scan_008	Cathédrale_Liège.3_Scan_009
◊ G_S10	0.0038	Cathédrale_Liège.3_Scan_008	Cathédrale_Liège.3_Scan_009
◊ G_S11	0.0026	Cathédrale_Liège.3_Scan_008	Cathédrale_Liège.3_Scan_009
◊ G_S4	0.0011	Cathédrale_Liège.3_Scan_008	Cathédrale_Liège.3_Scan_010
◊ G_S1	0.0011	Cathédrale_Liège.3_Scan_010	Cathédrale_Liège.3_Scan_012
◊ G_S6	0.0010	Cathédrale_Liège.3_Scan_008	Cathédrale_Liège.3_Scan_010
◊ G_S3	0.0009	Cathédrale_Liège.3_Scan_010	Cathédrale_Liège.3_Scan_012
◊ G_S3	0.0008	Cathédrale_Liège.3_Scan_008	Cathédrale_Liège.3_Scan_012
◊ G_S2	0.0008	Cathédrale_Liège.3_Scan_008	Cathédrale_Liège.3_Scan_012
◊ G_S1	0.0007	Cathédrale_Liège.3_Scan_008	Cathédrale_Liège.3_Scan_012
◊ G_S3	0.0006	Cathédrale_Liège.3_Scan_008	Cathédrale_Liège.3_Scan_010
◊ G_S2	0.0005	Cathédrale_Liège.3_Scan_010	Cathédrale_Liège.3_Scan_012

Statistiques coefficientées

Principal :  Écart :

Min :  Max :

Enregistrement fin



/Scans/CALAGE\_OK/Toiture/ScanManager

GestScanning Résultats de scanning Tensions de réf.

Tensions pesées Hiérarchie totale

Référence	Tens...	Scanning 1	Scanning 2
◊ G_S2	0.0008	Cathédrale_Liège.3_Scan_008	Cathédrale_Liège.3_Scan_012
◊ G_S1	0.0007	Cathédrale_Liège.3_Scan_008	Cathédrale_Liège.3_Scan_012
◊ G_S3	0.0006	Cathédrale_Liège.3_Scan_008	Cathédrale_Liège.3_Scan_010
◊ G_S2	0.0005	Cathédrale_Liège.3_Scan_010	Cathédrale_Liège.3_Scan_012
◊ G_S1	0.0005	Cathédrale_Liège.3_Scan_008	Cathédrale_Liège.3_Scan_010
◊ G_S2	0.0004	Cathédrale_Liège.3_Scan_008	Cathédrale_Liège.3_Scan_010
⊗ Inclinomètre	0.0000	Cathédrale_Liège.3_Scan_008	Cathédrale_Liège.3_Scan_009
⊗ Inclinomètre	0.0000	Cathédrale_Liège.3_Scan_009	Cathédrale_Liège.3_Scan_012
⊗ Inclinomètre	0.0000	Cathédrale_Liège.3_Scan_008	Cathédrale_Liège.3_Scan_010
⊗ Inclinomètre	0.0000	Cathédrale_Liège.3_Scan_009	Cathédrale_Liège.3_Scan_010
⊗ Inclinomètre	0.0000	Cathédrale_Liège.3_Scan_010	Cathédrale_Liège.3_Scan_012
⊗ Inclinomètre	0.0000	Cathédrale_Liège.3_Scan_008	Cathédrale_Liège.3_Scan_012

Statistiques coefficientées

Principal :  Écart :

Min :  Max :

Enregistrement fin

### Triforium : partie Nord

/Scans/CALAGE\_OK/Trifo/TRIFO\_NORD/ScanManager

GestScanning Résultats de scanning Tensions de réf.

Tensions pesées Hiérarchie totale

Référence	Tens...	Scanning 1	Scanning 2
◊ TRN_S10	0.0070	Cathédrale_Liège.2_Scan_027	Cathédrale_Liège.2_Scan_028
◊ TRN_S5	0.0063	Cathédrale_Liège.2_Scan_023	Cathédrale_Liège.2_Scan_024
◊ TRN_S9	0.0063	Cathédrale_Liège.2_Scan_023	Cathédrale_Liège.2_Scan_028
◊ TRN_S7	0.0060	Cathédrale_Liège.2_Scan_023	Cathédrale_Liège.2_Scan_027
◊ TRN_S9	0.0060	Cathédrale_Liège.2_Scan_023	Cathédrale_Liège.2_Scan_027
◊ TRN_S7	0.0050	Cathédrale_Liège.2_Scan_023	Cathédrale_Liège.2_Scan_028
◊ TRN_S13	0.0049	Cathédrale_Liège.2_Scan_024	Cathédrale_Liège.2_Scan_028
◊ TRN_S3	0.0048	Cathédrale_Liège.2_Scan_023	Cathédrale_Liège.2_Scan_024
◊ TRN_S12	0.0047	Cathédrale_Liège.2_Scan_023	Cathédrale_Liège.2_Scan_024
◊ TRN_S3	0.0043	Cathédrale_Liège.2_Scan_023	Cathédrale_Liège.2_Scan_028
◊ TRN_S5	0.0043	Cathédrale_Liège.2_Scan_023	Cathédrale_Liège.2_Scan_025
◊ TRN_S5	0.0038	Cathédrale_Liège.2_Scan_024	Cathédrale_Liège.2_Scan_025
◊ TRN_S1	0.0024	Cathédrale_Liège.2_Scan_023	Cathédrale_Liège.2_Scan_027

Statistiques coefficientées

Principal :  Écart :

Min :  Max :

Enregistrement fin

Création de  
Création de


/Scans/CALAGE\_OK/Trifo/TRIFO\_NORD/ScanManager

GestScanning Résultats de scanning Tensions de réf.

Tensions pesées Hiérarchie totale

Référence	Tens...	Scanning 1	Scanning 2
TRN_S5	0.0038	Cathédrale_Liège.2_Scan_024	Cathédrale_Liège.2_Scan_025
TRN_S1	0.0034	Cathédrale_Liège.2_Scan_023	Cathédrale_Liège.2_Scan_027
TRN_S3	0.0032	Cathédrale_Liège.2_Scan_023	Cathédrale_Liège.2_Scan_025
TRN_S3	0.0030	Cathédrale_Liège.2_Scan_023	Cathédrale_Liège.2_Scan_027
TRN_S7	0.0027	Cathédrale_Liège.2_Scan_027	Cathédrale_Liège.2_Scan_028
TRN_S1	0.0025	Cathédrale_Liège.2_Scan_023	Cathédrale_Liège.2_Scan_025
TRN_S1	0.0020	Cathédrale_Liège.2_Scan_023	Cathédrale_Liège.2_Scan_026
TRN_S3	0.0019	Cathédrale_Liège.2_Scan_024	Cathédrale_Liège.2_Scan_027
TRN_S3	0.0019	Cathédrale_Liège.2_Scan_024	Cathédrale_Liège.2_Scan_025
TRN_S14	0.0018	Cathédrale_Liège.2_Scan_029	Cathédrale_Liège.2_Scan_030
TRN_S9	0.0018	Cathédrale_Liège.2_Scan_027	Cathédrale_Liège.2_Scan_028
TRN_S3	0.0017	Cathédrale_Liège.2_Scan_027	Cathédrale_Liège.2_Scan_028
TRN_S3	0.0016	Cathédrale_Liège.2_Scan_025	Cathédrale_Liège.2_Scan_028

Statistiques coefficientées

Principal :   Écart :

Min :  Max :

Enregistrement fin


/Scans/CALAGE\_OK/Trifo/TRIFO\_NORD/ScanManager

GestScanning Résultats de scanning Tensions de réf.

Tensions pesées Hiérarchie totale

Référence	Tens...	Scanning 1	Scanning 2
TRN_S3	0.0016	Cathédrale_Liège.2_Scan_025	Cathédrale_Liège.2_Scan_028
TRN_S6	0.0015	Cathédrale_Liège.2_Scan_027	Cathédrale_Liège.2_Scan_028
TRN_S1	0.0014	Cathédrale_Liège.2_Scan_026	Cathédrale_Liège.2_Scan_027
TRN_S7	0.0014	Cathédrale_Liège.2_Scan_029	Cathédrale_Liège.2_Scan_030
TRN_S1	0.0010	Cathédrale_Liège.2_Scan_025	Cathédrale_Liège.2_Scan_027
TRN_S3	0.0007	Cathédrale_Liège.2_Scan_024	Cathédrale_Liège.2_Scan_028
TRN_S15	0.0007	Cathédrale_Liège.2_Scan_029	Cathédrale_Liège.2_Scan_030
TRN_S1	0.0006	Cathédrale_Liège.2_Scan_025	Cathédrale_Liège.2_Scan_026
TRN_S3	0.0005	Cathédrale_Liège.2_Scan_025	Cathédrale_Liège.2_Scan_027
TRN_S8	0.0004	Cathédrale_Liège.2_Scan_027	Cathédrale_Liège.2_Scan_028
TRN_S4	0.0001	Cathédrale_Liège.2_Scan_025	Cathédrale_Liège.2_Scan_026
Inclinomètre	0.0000	Cathédrale_Liège.2_Scan_023	Cathédrale_Liège.2_Scan_027
Inclinomètre	0.0000	Cathédrale_Liège.2_Scan_027	Cathédrale_Liège.2_Scan_028

Statistiques coefficientées

Principal :   Écart :

Min :  Max :

Enregistrement fin

### Triforium : Partie Sud



/Scans/CALAGE\_OK/Trifo/TRIFO\_SUD/ScanManager

GestScanning Résultats de scanning Tensions de réf.

Tensions pesées Hiérarchie totale

Référence	Tens...	Scanning 1	Scanning 2
◊ TRS_4	0.0134	Cathédrale_Liège.2_Scan_016	Cathédrale_Liège.2_Scan_018
◊ TRS_12	0.0133	Cathédrale_Liège.2_Scan_016	Cathédrale_Liège.2_Scan_020
◊ TRS_5	0.0109	Cathédrale_Liège.2_Scan_018	Cathédrale_Liège.2_Scan_019
◊ TRS_11	0.0108	Cathédrale_Liège.2_Scan_017	Cathédrale_Liège.2_Scan_018
◊ TRS_4	0.0098	Cathédrale_Liège.2_Scan_018	Cathédrale_Liège.2_Scan_020
◊ TRS_2	0.0069	Cathédrale_Liège.2_Scan_019	Cathédrale_Liège.2_Scan_020
◊ TRS_2	0.0069	Cathédrale_Liège.2_Scan_017	Cathédrale_Liège.2_Scan_020
◊ TRS_13	0.0062	Cathédrale_Liège.2_Scan_016	Cathédrale_Liège.2_Scan_020
◊ TRS_2	0.0061	Cathédrale_Liège.2_Scan_016	Cathédrale_Liège.2_Scan_017
◊ TRS_4	0.0056	Cathédrale_Liège.2_Scan_016	Cathédrale_Liège.2_Scan_020
◊ TRS_6	0.0054	Cathédrale_Liège.2_Scan_016	Cathédrale_Liège.2_Scan_018
◊ TRS_1	0.0053	Cathédrale_Liège.2_Scan_018	Cathédrale_Liège.2_Scan_019
◊ TRS_1	0.0050	Cathédrale_Liège.2_Scan_016	Cathédrale_Liège.2_Scan_019

Statistiques coefficientées

Principal :  Écart :

Min :  Max :

Enregistrement fin

/Scans/CALAGE\_OK/Trifo/TRIFO\_SUD/ScanManager

GestScanning Résultats de scanning Tensions de réf.

Tensions pesées Hiérarchie totale

Référence	Tens...	Scanning 1	Scanning 2
◊ TRS_1	0.0050	Cathédrale_Liège.2_Scan_016	Cathédrale_Liège.2_Scan_019
◊ TRS_3	0.0048	Cathédrale_Liège.2_Scan_016	Cathédrale_Liège.2_Scan_017
◊ TRS_2	0.0048	Cathédrale_Liège.2_Scan_017	Cathédrale_Liège.2_Scan_019
◊ TRS_2	0.0045	Cathédrale_Liège.2_Scan_016	Cathédrale_Liège.2_Scan_019
◊ TRS_15	0.0040	Cathédrale_Liège.2_Scan_020	Cathédrale_Liège.2_Scan_021
◊ TRS_15	0.0038	Cathédrale_Liège.2_Scan_020	Cathédrale_Liège.2_Scan_022
◊ TRS_2	0.0031	Cathédrale_Liège.2_Scan_016	Cathédrale_Liège.2_Scan_020
◊ TRS_7	0.0028	Cathédrale_Liège.2_Scan_018	Cathédrale_Liège.2_Scan_022
◊ TRS_7	0.0026	Cathédrale_Liège.2_Scan_018	Cathédrale_Liège.2_Scan_021
◊ TRS_7	0.0025	Cathédrale_Liège.2_Scan_018	Cathédrale_Liège.2_Scan_020
◊ TRS_7	0.0023	Cathédrale_Liège.2_Scan_016	Cathédrale_Liège.2_Scan_018
◊ TRS_7	0.0022	Cathédrale_Liège.2_Scan_016	Cathédrale_Liège.2_Scan_020
◊ TRS_17	0.0021	Cathédrale_Liège.2_Scan_016	Cathédrale_Liège.2_Scan_017

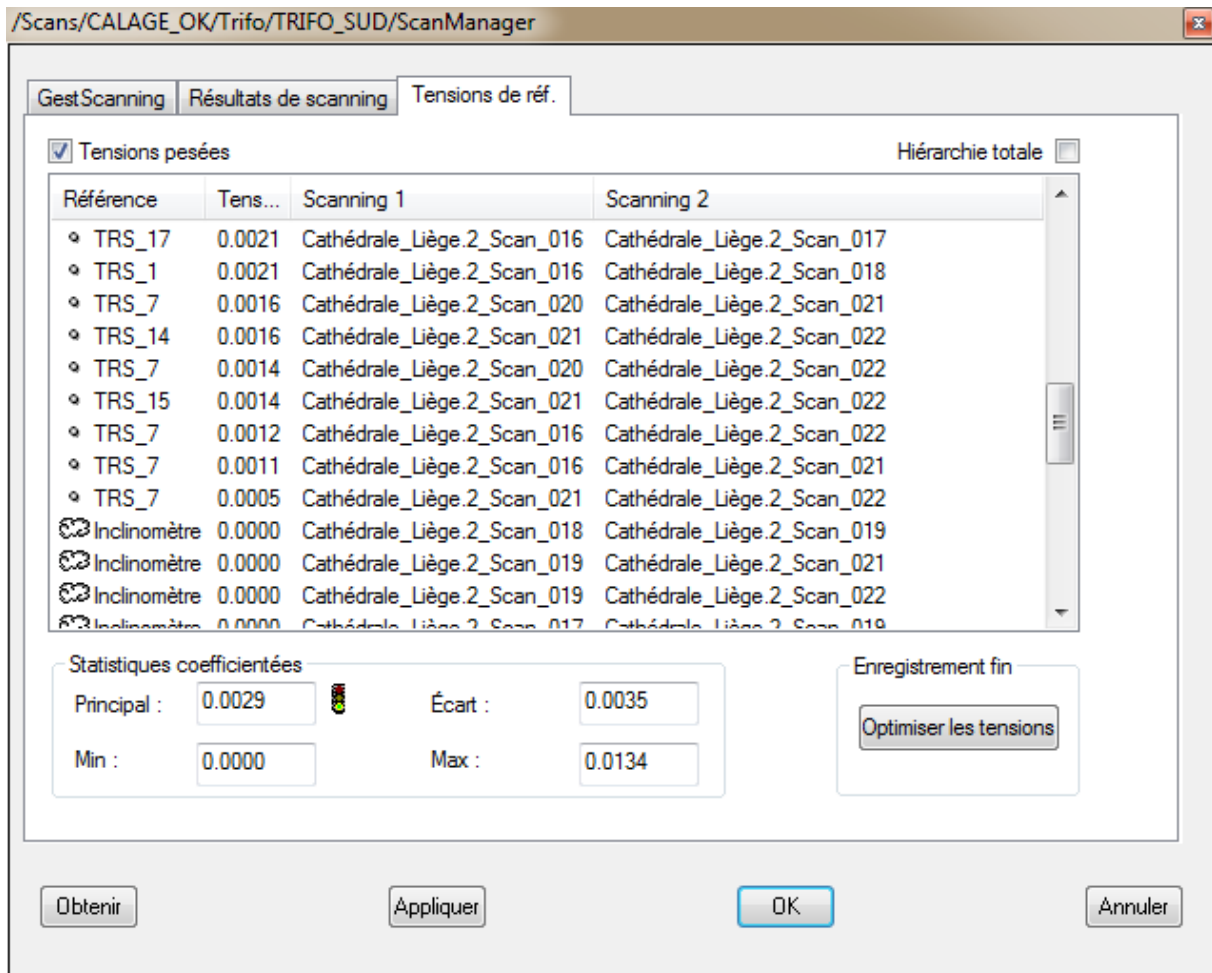
Statistiques coefficientées

Principal :  Écart :

Min :  Max :

Enregistrement fin





### Résumé :

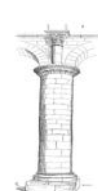
Ces tensions ne peuvent pas être interprétées comme des distances dans des unités standard de mesure. Plus la tension est proche de zéro plus le recalage est précis. Au moyen de l'outil ScanFit, il nous est possible de contrôler les distances (en mm) entre les positions des références dans le système général et le système scanning. Ainsi, nous voudrions avoir une différence inférieure à 10 mm pour obtenir un recalage relatif dans les tolérances, et conserver les résultats obtenus.

En pratique, lorsque les sphères et les cibles sont bien distribuées nous obtenons entre 1 et 5 mm de différence. Les distances maximales observées sont de 9 mm. Le feu tricolore (vert) indique ceci.

vert : < 10mm

orange:  $\geq 10\text{mm}$  et  $\leq 20$  mm

rouge : >20mm



The image displays 12 windows of the ScanFit software interface, arranged in a 3x4 grid. Each window shows the configuration and data for a specific scan (Scan\_001 to Scan\_012). The windows are titled 'ScanFit' and contain the following information:

- Coordonnées globales:** Global coordinates and orientation.
- Position:** X, Y, Z coordinates in meters.
- Angle:** Scan angle in degrees.
- Tension normalisée:** Normalized tension in mm.
- Distance de point:** Point distance in mm.
- Dérive de point:** Point drift in mm.
- Décalage longueur:** Length offset in mm.
- Décalage angulaire:** Angular offset in degrees.
- Décalage orthogonal:** Orthogonal offset in mm.
- Décalage selon de l'inclinomètre:** Offset according to the inclinometer in degrees.
- Objet:** A table listing object coordinates (Dist., Long., Angulaire) for various points.

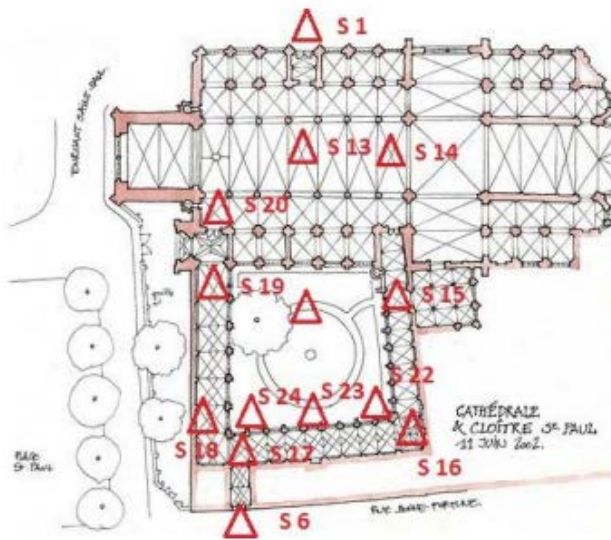
In the center of the grid, there is a 3D visualization of a scan of a stone wall. Yellow markers are visible on the wall surface, indicating specific points of interest or measurement locations.

### Problèmes rencontrés :

- Différence de hauteur sphères entre les différents jours
- Disparition de sphères entre plusieurs scans
- Placement mal distribué concernant certains scanings (emploi de cibles naturelles moins précises)
- Incohérence de la méthodologie de levé pour le rez-de-chaussée de la cathédrale
- 




**28 Mars 2013 : Cloître extérieur**





Nombre de scans : 2 scans

Références : S21 – S22 – S23

	Référence	Coordonnées	Local (m)	Lambert 72 (m)	H Sphère (m)
	S21	X	980.685	235082.820	1.550
Y		956.893	148401.715		
Z (DNG – H)		100.352	63.190		
	S22	X	988.123	235100.669	1.290
Y		937.971	148391.988		
Z (DNG – H)		100.326	63.164		
	S23	X	977.206	1.360	
Y		940.958			
Z (DNG – H)		99.961			

### Rapport de recalage & validation :

Erreur résiduelle	Delta X	Delta Y	Delta Z	Erreur d'ajustement
<b>1.05 mm</b>				
--	--	--	--	0.94 mm
1.03 mm	-0.09 mm	-0.51 mm	-0.89 mm	0.72 mm
0.76 mm	-0.55 mm	0.42 mm	-0.31 mm	2.39 mm
1.26 mm	0.19 mm	-0.92 mm	0.84 mm	0.64 mm
1.15 mm	0.44 mm	1.00 mm	0.36 mm	0.61 mm
--	--	--	--	0.63 mm
--	--	--	--	0.75 mm
--	--	--	--	0.70 mm
--	--	--	--	0.86 mm
--	--	--	--	1.36 mm
<b>1.05 mm</b>				
1.15 mm	-0.44 mm	-1.00 mm	-0.36 mm	0.82 mm
1.03 mm	0.09 mm	0.51 mm	0.89 mm	0.86 mm
--	--	--	--	0.72 mm
0.76 mm	0.55 mm	-0.42 mm	0.31 mm	0.70 mm
1.26 mm	-0.19 mm	0.92 mm	-0.84 mm	0.51 mm
--	--	--	--	0.68 mm
--	--	--	--	0.61 mm

Problèmes rencontrés :



- Échafaudage
- Coordonnées Lambert 72 pour station 23 non connues

### 29 Mars 2013 : Façade Sud

Nombre de scans : 3 Scans



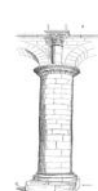
Références : S2 – S3 – Slibre

	Référence	Coordonnées	Local (m)	Lambert 72 (m)	H Sphère (m)
	S2	X	1027.349	235090.130	
		Y	1001.209	148465.637	
		Z - (DNG - H)	99.963	62.801	1.730
	S3	X	1049.243	235119.383	
		Y	981.783	148464.806	
		Z - (DNG - H)	100.044	62.882	1.760
	Slibre	X		235069.5215	
		Y		148465.12	
		Z - (DNG - H)		63.219	1.522*

\*Différence Sphère/Scan=22mm, Hauteur instrument rentrée=1.5m

Problèmes rencontrés :

- Les stations S1 et S11 ont sauté
- La station S12 est trop loin de la zone de levé
- La station S10 est cachée par angle bâtiment
- Bruit inhérent au passage du public/voitures





## Annexe 5 Algorithme MATLAB de recalage

Voici l'algorithme de consolidation à partir de sphères créé dans Matlab. Il permet de calculer les paramètres de transformation entre deux systèmes à partir de points homologues.

```

format long;
%Recalage et consolidation de nuages de points dans des systèmes
%différents( sans facteur d'échelle) : calcul des 6 paramètres de
%transformation

% I - Données

%Chargement des coordonnées des points homologues (P=repère 1= jour2, Q=repère2 =
jour3)

P=load('img2_cloi.txt');
Qb=load('img1_cloi.txt');

%Calcul des gisements

%Repère 1 & 2

taille=size(P);

ang=zeros(taille(1)-1,1); %Allocation pour augmenter vitesse de traitement
dx1=zeros(taille(1)-1,1);
dy1=zeros(taille(1)-1,1);
dx2=zeros(taille(1)-1,1);
dy2=zeros(taille(1)-1,1);
G1t=zeros(taille(1)-1,1);
G2t=zeros(taille(1)-1,1);
for i=1:(taille(1)-1); %R1
    dx1(i)=P(i+1,1)-P(i,1);
    dy1(i)=P(i+1,2)-P(i,2);
    dx2(i)=Qb(i+1,1)-Qb(i,1); %R2
    dy2(i)=Qb(i+1,2)-Qb(i,2);

    G1(i)=2*atan(dx1(i)/(sqrt(dx1(i)^2+dy1(i)^2)+dy1(i)));
    dx2(i)=Qb(i+1,1)-Qb(i,1); %R2
    dy2(i)=Qb(i+1,2)-Qb(i,2);
    G2(i)=2*atan(dx2(i)/(sqrt(dx2(i)^2+dy2(i)^2)+dy2(i)));
    ang(i)=G2(i)-G1(i);
end

%a=mean(ang);
a=mean(ang);

%Recalage approximatif du repère 2 sur le repère 1 :

Rz=[cos(a) -sin(a) 0; sin(a) cos(a) 0; 0 0 1];
Qt=Rz*Qb';
Q=Qt';

```



```

%Traitement par moindres carrés pour résoudre le système Ax=B

A=zeros(taille(1),6);
for i=1:taille(1)
    A(i*3-2,1)=1;
    A(i*3-1,2)=1;
    A(i*3,3)=1;
    A(i*3-2,5)=-Q(i,3);
    A(i*3-2,6)=Q(i,2);
    A(i*3-1,4)=Q(i,3);
    A(i*3-1,6)=-Q(i,1);
    A(i*3,4)=-Q(i,2);
    A(i*3,5)=Q(i,1);
end

B=zeros(taille(1),1);
for i=1:taille(1)
    B(i*3-2,1)=P(i,1)-Q(i,1);
    B(i*3-1,1)=P(i,2)-Q(i,2);
    B(i*3,1)=P(i,3)-Q(i,3);
end

X=zeros(6,1);

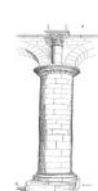
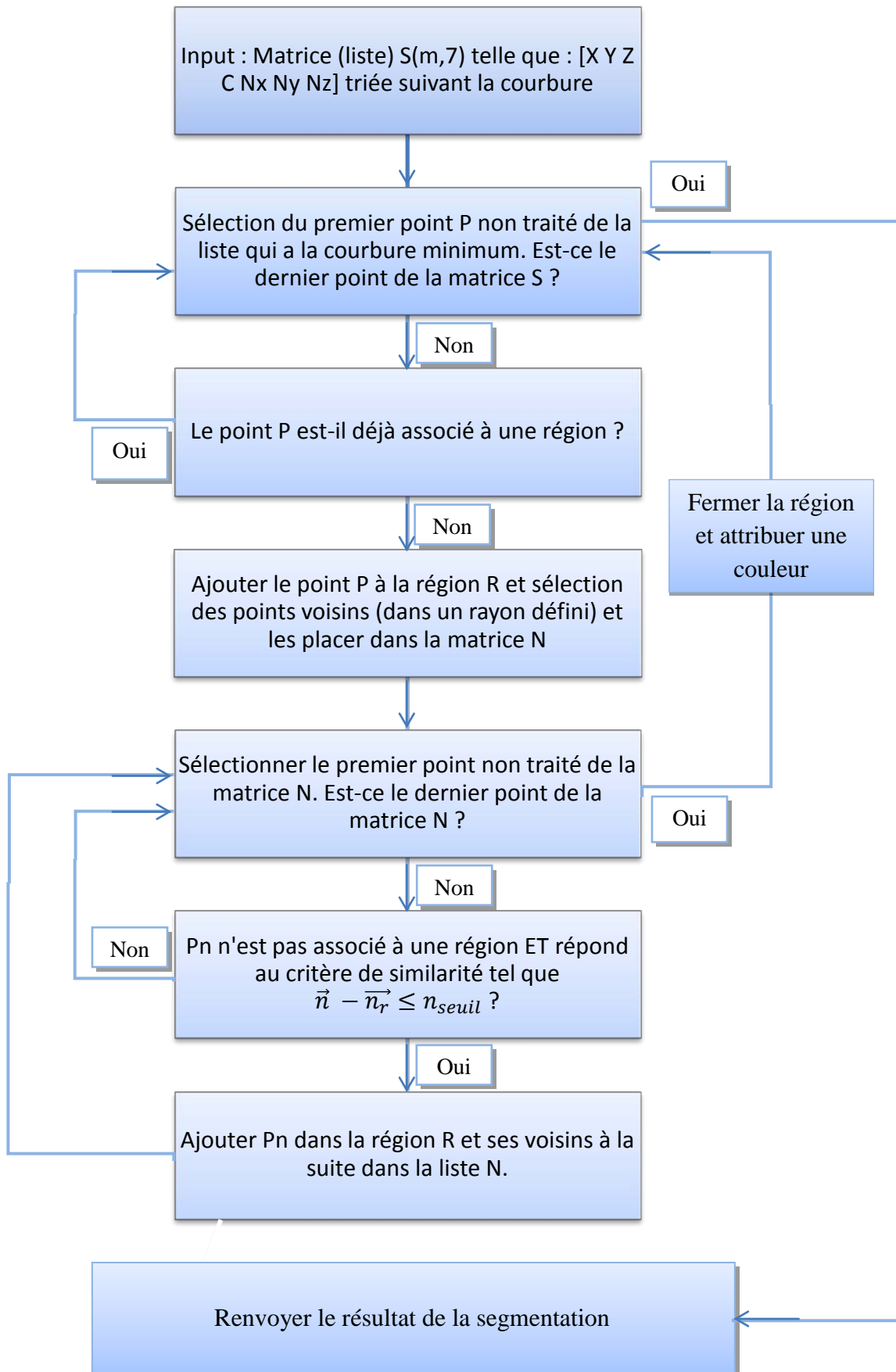
N=A'*A;
X=inv(N)*(A'*B);           %Solution
V=A*X-B                   %Résidus
S02=V'*V/(taille(1)-6)    %Variance
S0=sqrt(S02);
QCX=inv(N);               %matrice des cofacteurs des paramètres ajustés Q^x
COV=QCX*S02;             %Variances et covariances COV

%Solutions, angles en degrés décimaux :
Initialisation=a*180/pi
Tx=X(1,1)
Ty=X(2,1)
Tz=X(3,1)
ax=X(4,1)*180/pi
ay=X(5,1)*180/pi
az=X(6,1)*180/pi
G2-G1

```



## Annexe 6 Algorithme de segmentation créé



```

disp('-----Segmentation 2-----');

S=flipdim(sortrows(R2,8),1); %Tri Décroissant en fonction de la taille des régions
(nb de points)
%nb_knn=20;
ang=0.4;
tranche=0.10;
pas=1.5;
Rf2=[];
Rtemp=[];
for i=1:size(S,1);
    if (mean(ismember(S(i,1),Rtemp))~=1);
        R=[];
        N=[];
        R=[R;S(i,:)]; %Incrémentation
        %idx=knnsearch(S(:,2:4),S(i,2:4),'k',nb_knn);
        idx=rangesearch(S(i,2:4),pas,S(:,2:4));
        %B=abs(S(idx(m,1),4)-S(i,4));
        for m=1:size(idx,1);
            if abs(S(idx(m,1),4)-S(i,4))<=tranche
                N=[N;S(idx(m,1),:) S(i,1)];
                %N=[N;S(idx(m,1),:) repmat(S(i,1),size(idx,1),1)];
            end
        end
        idx=[];
        disp(i);
        j=1;
        while j<=size(N,1);
            %for j=1:10000 %size(S,1);
            if (mean(ismember(N(j,1),Rtemp))~=1 &&
mean(ismember(N(j,1),R(:,1))~=1); % && abs(N(j,7)-S(N(j,9),7))<=8*10^(-2));
                %{
                norm(N(j,5:7)-S(N(j,9),5:7))<=norm(10*[1 1 1]));
                %abs(N(j,8)-S(N(j,9),8))<=0.4*(N(j,8)+S(N(j,9),8))/2);
                %abs(N(j,7)-S(N(j,9),7))<=20*10^(-1));
                %abs(dot(N(j,5:7),S(N(j,9),5:7)))>=cos(ang));
                %}
                R=[R;N(j,1:8)];
                %idx=knnsearch(S(:,2:4),N(j,2:4),'k',nb_knn);
                idx=rangesearch(N(j,2:4),pas,S(:,2:4));
                A=[];
                A=S(idx,1); %Sélection des indices de Rf correspondant aux voisins
de N(j)
                for k=1:size(1,1);
                    if (mean(ismember(A(k,1),N(:,1)))==0 && abs(S(idx(k,1),4)-
N(j,4))<=tranche);
                        N=[N;S(idx(k,1),:) N(j,1)];
                    end
                end
            end
            j=j+1;
        end
        Rf2=[Rf2;{R}];
        Rtemp=[Rtemp;R(:,1)];
        N=[];
    end
end

```

Le regroupement final et l'écriture peuvent alors permettre une exploitation des résultats obtenus.

```

disp('-----Regroupement-----');
A=[];
idx=[];
Segm=[];
Rtemp=[];

```



```

for i=1:size(Rf2,1);
    A=Rf2{i};
    idx=A(:,1);
    R=cell2mat(Rf(idx,1));
    Segm=[Segm;{R}];
    Rtemp=[Rtemp;R];
end
disp(' Done');

disp('-----Ecriture-----');
E=zeros(size(Rtemp,1),11);
%E=[X Y Z C Nx Ny Nz n°ligne_S R G B]
cmp=rand(size(Segm,1),3);
for i=1;
    E(1:size(Segm{i},1),1:8)=Segm{i};
    E(1:size(Segm{i},1),9:11)=repmat(cmp(i,:),size(Segm{i},1),1);
    k=size(Segm{i},1);
end
for i=2:size(Segm,1);
    f=k+size(Segm{i},1);
    E(k+1:f,1:8)=Segm{i};
    E(k+1:f,9:11)=repmat(cmp(i,:),size(Segm{i},1),1);
    k=f;
end
H=zeros(size(Rtemp,1),6);
H(:,1:3)=E(:,1:3);
H(:,4:6)=round(E(:,9:11)*100);
dlmwrite('a_1475000_colonne_parfaite_segmentation2_test_final.asc', H, '\t')
disp(' Done');
disp('-----> Succès de la segmentation');

disp('-----Une region un fichier-----');
H=zeros(size(Rtemp,1),6);
i=1;
cmp=round(rand(size(Segm,1),3)*100);
for i=1:size(Segm,1);
    A=[];
    B=[];
    A=[cell2mat(Segm(i,1)) repmat(cmp(i,:),size(Segm{i},1),1)];
    B(:,1:3)=A(:,1:3);
    B(:,4:6)=A(:,9:11);
    fileName = sprintf('2_segmentation_%d.asc',i);
    dlmwrite(fileName,B,'\t');
end

```



## Annexe 7 Méthodes de maillage et paramétrisation

Voici les différents algorithmes utilisés et testés pour la modélisation :

### Modélisation « Marching cubes » (Lorenson & Cline, 1987)

Cet algorithme parcourt le champ scalaire, prenant huit points à la fois (définissant ainsi un cube imaginaire), et détermine les polygones à créer pour représenter une partie de l'isosurface (analogue en 3D d'une courbe de niveau) contenue dans ce cube.

Ceci fonctionne en créant un index dans un tableau précalculé des 256 configurations de polygones possibles () dans un cube, en traitant chacune des 8 valeurs scalaires comme un bit dans un nombre entier de 8 bits. Si la valeur scalaire est supérieure à la valeur de l'isosurface (i.e., est à l'intérieur de la surface), alors le bit correspondant est mis à 1, sinon il est mis à 0. La valeur finale après le test des 8 points est l'index de la bonne configuration polygonale dans le tableau précalculé.

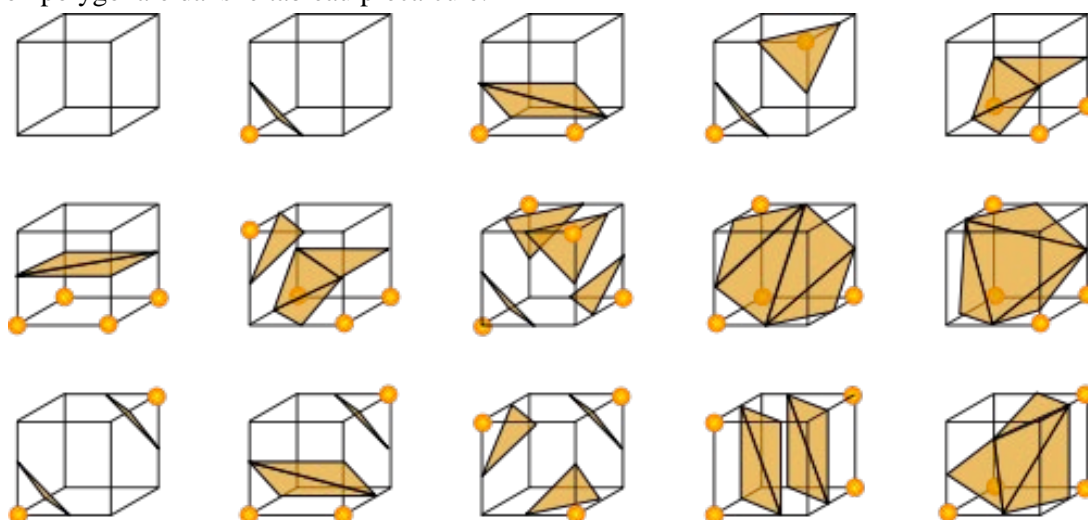


Figure 62 Marching Cubes. Source : (Lorenson & Cline, 1987)

Finalement, chaque sommet des polygones créés est placé à sa position finale le long de l'arête du cube, en interpolant linéairement les deux valeurs scalaires connectées par cette arête.

La valeur à chaque point du champ scalaire est aussi utilisée pour calculer le vecteur normal de l'isosurface passant en ce point. Ce calcul est basé sur le gradient du champ. Il est donc possible d'interpoler ces valeurs le long de chaque arête de chaque cube de façon à obtenir la normale des points sur la surface. L'interpolation permet d'éviter un calcul analytique du gradient pour une position quelconque. Le calcul des normales permet l'ombrage de l'objet par la suite.

### Triangulation de Delaunay

A partir d'un semis de point, le principe est de réaliser une triangulation avec des faces triangulaires les plus équilatérales et petites possibles. L'objectif est de maximiser le plus petit angle de tous les triangles du maillage. Raisononnons en 2 dimensions : Partant d'un ensemble  $E$  de points du plan, la triangulation de DELAUNAY est telle qu'aucun de ces points n'est à l'intérieur du cercle circonscrit d'un des triangles. Un cercle circonscrit est considéré comme vide s'il ne contient aucun point (à part sur son périmètre). Enfin, la condition de DELAUNAY est remplie si tous les cercles circonscrits des triangles du réseau sont vides. En trois dimensions le principe est le même sauf qu'on utilise des sphères circonscrites.



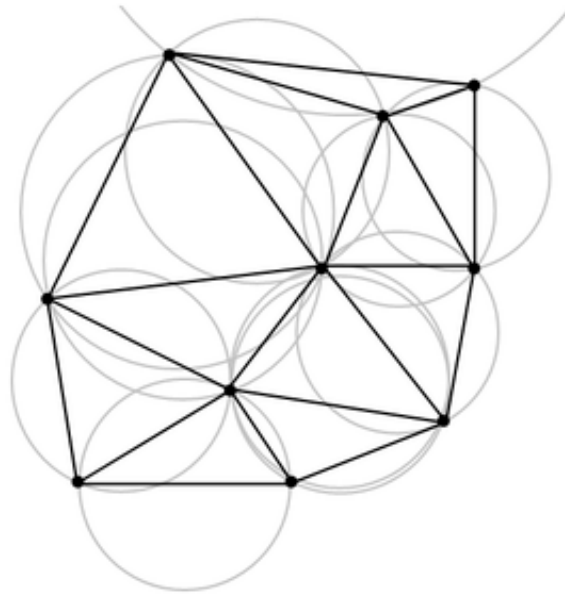


Figure 63 Une triangulation de Delaunay avec les cercles circonscrits en gris. Source : Wikipedia

Cependant, les techniques plus récentes telles que « Marching Triangle » ou « Ball-Pivoting » utilisant une structure plus flexible pour reconstruire les surfaces de nuage de point sont en général plus efficaces que la Triangulation de Delaunay (Pour une précision similaire)

#### « Ball-pivoting Algorithm » (BPA)

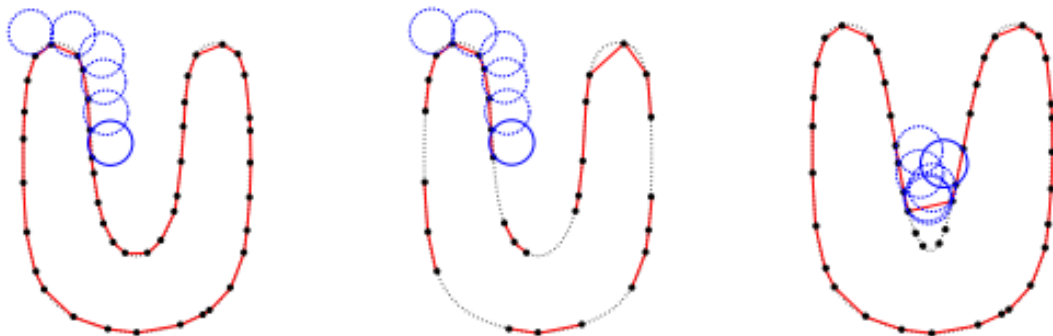


Figure 64 L'algorithme Ball-Pivoting en 2D. Un cercle de rayon  $r$  pivote (gauche, quand la densité est trop faible, création de trous (milieu), quand la courbure est trop importante, certains points ne seront pas atteints (droite). Source : (F. Berdanini & al)

L'algorithme « Ball-Pivoting » comme défini partie V calcule un maillage par triangulation à partir d'un nuage de point. La faible quantité de mémoire requise par le BPA, son efficacité et la qualité des résultats obtenus en font un outil de choix pour un maillage précis. Robuste et flexible, les différents tests réalisés sur des échantillons de plusieurs millions de points ont dévoilé la création de maillages rapides (environ 500 000 triangles par minute).

#### Reconstruction de Poisson (Kazhdan et al, 2006)

Comme nous l'avons vu partie V, la reconstruction de surface depuis un nuage de point orienté peut être résolu comme un problème de Poisson. Cette formulation de Poisson considère tous les points comme un ensemble unique, mettant en avant ses propriétés de robustesse face au bruit. D'un point de vue pratique, l'algorithme implémenté dans MeshLab est réellement pratique (quoique sujet aux bugs récurrents d'où l'obligation de réaliser des reconstructions par niveau d'octree amoindri), mais il m'est arrivé d'avoir des problèmes de non préservation du volume original (création à l'extérieur).



Figure 65 Reconstruction de la tête du Michelangelo de David, obtenu avec un niveau d'octree de 11. Source : Kazhdan et al, 2006)

### « Moving least-squares » (MLS)

Il s'agit d'une méthode de reconstruction de fonctions continues extraites d'un échantillon discontinu de points grâce au calcul de moindres carrés. L'implémentation dans MeshLab est basée sur la méthode définie par (Gianmauro et al, 2009), simple et efficace pour reconstruire des surfaces triangulées à partir de grands volumes de données orientées. Elle combine une subdivision de l'espace avec une mise en parallèle pour extraire des isosurfaces à intervalles réguliers. Néanmoins, bien qu'efficace pour une reconstruction géométrie fidèle à la réalité, les formes complexes reconstruites par cette méthode manque souvent de précision.

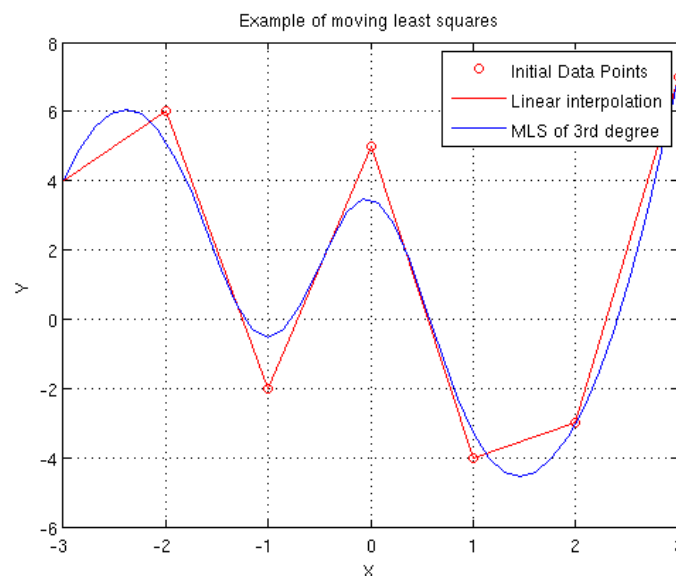


Figure 66 Exemple en 2D. Les cercles sont les échantillons et le polygone est une interpolation linéaire. Source : (Levin, 1998)



## Annexe 8 Principe du format .obj texturé

Pour obtenir un modèle 3D texture au format .obj, il est nécessaire de comprendre l'organisation et les liens logiques des différents éléments attachés.

Pour décrire un maillage texturé en utilisant le format de fichier .obj, il est nécessaire de respecter certaines exigences que je vais souligner et expliquer.

Dans un premier temps, il faut noter que la texture n'est pas incluse dans le .obj, mais est une image au format classique qui est appelée et associée au .obj au moyen d'un fichier matériaux :

- Le fichier .obj : colonne.obj
- Le mtl (material file) : colonne.mtl
- La texture (format jpg, png, ...) : colonne\_texture.jpg

Je vais maintenant décrire chaque fichier qu'il est possible d'éditer au moyen d'un éditeur de texte.

### Le fichier .obj

La première ligne indique le nom du fichier matériel, dans notre cas « colonne.mtl »

```
mtllib colonne.mtl
```

Les lignes qui commencent par un « v » constituent les coordonnées X,Y,Z des vertices.

```
v 0.000000 0.000000 -1.500000
v 0.000000 0.000000 -1.500000
v 0.000000 0.000000 -1.500000
v 0.000000 0.000000 -1.500000
:
:
:
```

Les lignes qui commencent par « vn » indiquent les normales des vertices. Dans notre exemple, chaque vertice est associée à une normale :

```
vn 0.000000 0.000000 -1.000000
vn 0.000000 0.000000 -1.000000
vn 0.000000 0.000000 -1.000000
vn 0.000000 0.000000 -1.000000
:
:
:
```

Les lignes ci-dessous commencent par « vt » et donnent les coordonnées de texture (l'UV mapping est à l'origine de ces coordonnées). Encore une fois, nous avons ici un nombre de coordonnées textures égal au nombre de points :

```
vt 0.000000 0.000000
vt 0.010000 0.000000
vt 0.020000 0.000000
vt 0.030000 0.000000
:
:
:
```



Notez que bien qu'ayant listé les vertices, normales et coordonnées de textures dans des blocs séparés dans cet exemple, ils peuvent être « mélangés » dans n'importe quel ordre. Leur index respectif qui est utilisé dans la prochaine section pour décrire les faces est simplement incrémenté selon leur position.

Ainsi, chaque ligne de cette section commence par un « f » donnant les 3 vertices créant chaque face. Le nombre commence à 1 (et non 0). Avant les informations de faces on retrouve le matériel utilisé, qui dans notre cas définit l'image utilisée pour la texture.

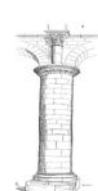
```
usemtl material0
f 1/1/1 2/2/2 103/103/103
f 1/1/1 103/103/103 102/102/102
f 2/2/2 3/3/3 104/104/104
:
:
:
```

Il est également possible d'ajouter des vertices, normales et coordonnées de texture mélangées avec la section descriptive des faces.

### Le fichier matériel .mtl

Le fichier matériel « colonne.mtl » définit les propriétés visuelles d'un maillage, les coefficients de réflexion (ambient, diffuse, specular), et notre intérêt ici concerne le nom de la texture. Le fichier peut être défini par un nombre de matériaux types, les faces du maillage définies plus haut héritent de ces valeurs après chaque usage de « usemtl ».

```
newmtl material0
Ka 1.000000 1.000000 1.000000
Kd 1.000000 1.000000 1.000000
Ks 0.000000 0.000000 0.000000
Tr 1.000000
illum 1
Ns 0.000000
map_Kd capsule0.jpg
```



## LISTE DES FIGURES

Figure 1 Chaîne de traitement d'un nuage de point obtenu par Laser Scanner	6
Figure 2 Matériel mis à disposition et utilisé	7
Figure 3 Cathédrale Saint-Paul de Liège : Façade Sud et Grande Nef	8
Figure 4 Plan de la cathédrale Saint-Paul	9
Figure 5 Principe triangulation	10
Figure 6 Méthode temps de vol inspiré de (Wehr & Lohr, 1999)	10
Figure 7 Principe du décalage de phase	11
Figure 8 Précision des méthodes de géoréférencement inspiré de (Reshetyuk, 2009)	12
Figure 9 Exemple de différents échantillons de différentes densités appliquées à un vitrail de la cathédrale	13
Figure 10 Estimation des normales d'un cube – source (Castillo, 2013)	14
Figure 11 Algorithme RANSAC appliqué à plusieurs colonnes via CloudComapre	14
Figure 12 Segmentation basée sur les surfaces. Source (Holz, 2013)	15
Figure 13 Approche de segmentation hybride. Source (Rusu & Blodow, 2009)	15
Figure 14 Modélisation d'une arcade	16
Figure 15 Modélisation de la façade Sud réalisée avec 3DReshaper	16
Figure 16 Modélisation géométrique d'une colonne de la cathédrale	17
Figure 17 Eléments d'une modélisation polygonale. Source (Pauly, 2003)	17
Figure 18 Représentation de l'UV Mapping par un exemple appliqué au globe terrestre. Source Tschmits	18
Figure 19 Anaglyphe	19
Figure 20 Réalité augmentée	19
Figure 21 Imprimante 3D Markerbot	20
Figure 22 Niveau 0	22
Figure 23 Triforium	22
Figure 24 Niveau 2	22
Figure 25 Le toit	22
Figure 26 Le cloître	22
Figure 27 Façade Sud	22
Figure 28 FARO FOCUS <sup>3D</sup>	23
Figure 29 Localisation du réseau de points d'appuis et stations de scans réalisées à l'intérieur/extérieur de la cathédrale	24
Figure 30 Nuage de point partiel de la cathédrale traité dans SCENE en a) Vue du dessus et b) Vue isométrique	25
Figure 31 Cible Damier	26
Figure 32 Cible Sphère	26
Figure 33 Erreur de géoréférencement sur 3 points d'appuis	27
Figure 34 Extrait du rapport d'erreur issu de Realworks	27
Figure 35 Réalisation des sections transversales (gauche) et modélisation paramétrique (droite) appliquée à une colonne	28
Figure 36 Principe théorique de l'approche par enveloppes convexes	29
Figure 37 Influence d'un problème de recalage sur la segmentation	30
Figure 38 Classification selon les normales	30
Figure 39 Classification selon la courbure	30
Figure 40 Nuage de point brut (gauche), segmentation 1 (milieu), segmentation 2 (droite)	31
Figure 41 Arbre Octree. Source (Peixoto, 2009)	32
Figure 42 Résultats et différents tests visualisation Matlab	35
Figure 43 Représentation graphique des différentes parties de l'algorithme et influence des paramètres d'entrée	36
Figure 44 Segmentation appliquée au château de Colonster levé par Scanner GX	37
Figure 45 Segmentation d'un extrait de 3 colonnes de la cathédrale Saint-Paul	37
Figure 46 Influence du pas sur une colonne. a) pas= 0.07 cm b) pas=0.02 cm	38
Figure 47 Segmentation façade Sud : a) C=1.5*10 <sup>-1</sup> b) C=2*10 <sup>-1</sup> c) C=3*10 <sup>-1</sup> d) C=5*10 <sup>-1</sup>	38

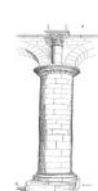
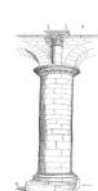


Figure 48 Analyse temporelle sur la segmentation d'une colonne - pas = 0.05	39
Figure 49 Illustration de la reconstruction de Poisson 2D. Inspiré de (Kazhdan <i>et al.</i> , 2006)	40
Figure 50 Reconstruction du modèle du dragon au niveau d'octree 6 (gauche) 8 (milieu) 10 (droite). Source (Kazhdan <i>et al.</i> , 2006)	41
Figure 51 Les 6 étapes pour la réalisation d'un modèle 3D texturé de qualité par reconstruction Poisson	42
Figure 52 Ecart entre le segment principal du nuage de point et le cylindre associé	43
Figure 53 Modèle hybride	43
Figure 54 Ecarts de distance au nuage de point de la modélisation Poisson	44
Figure 55 Ecarts de distance au nuage de point de la modélisation hybride	44
Figure 56 Exemple de PDF3D créé pour une colonne. a) Vue interface - b) Commentaires - c) détail et cotations	45
Figure 57 Exemples de feuillages modélisés et diffusés au moyen de PDF3D	46
Figure 58 Capture d'écran réalisé de la visite virtuelle WebGL. Nuage de 500 millions de points en temps réel	47
Figure 59 Visite de la cathédrale par différents niveaux de LOD : a) 5 ; b) 10 ; c) 25 ; d) 50	48
Figure 60 Capture d'écran de la visite virtuelle WebGL réalisée de la cathédrale Saint-Paul de Liège	48
Figure 61 Documentation technique du scanner FARO FOCUS <sup>3D</sup> . Source : <a href="http://www.faro.com">www.faro.com</a>	53
Figure 62 Marching Cubes. Source : (Lorenson & Cline, 1987)	77
Figure 63 Une triangulation de Delaunay avec les cercles circonscrits en gris. Source : Wikipedia	78
Figure 64 L'algorithme Ball-Pivoting en 2D. Un cercle de rayon r pivote (gauche, quand la densité est trop faible, création de trous (milieu), quand la courbure est trop importante, certains points ne seront pas atteints (droite). Source : (F. Berdanini & al)	78
Figure 65 Reconstruction de la tête du Michelangelo de David, obtenu avec un niveau d'octree de 11. Source : Kazhdan <i>et al.</i> , 2006)	79
Figure 66 Exemple en 2D. Les cercles sont les échantillons et le polygone est une interpolation linéaire. Source : (Levin, 1998)	79



# VERS DE NOUVELLES PERSPECTIVES LASERGRAMMÉTRIQUES : OPTIMISATION ET AUTOMATISATION DE LA CHAÎNE DE PRODUCTION DE MODELES 3D

Florent POUX

Ecole Supérieure des Géomètres et Topographes (ESGT), France – florentpoux@gmail.com

**MOTS CLÉS:** Nuage de points, segmentation, Scanner Laser, Reconstruction, Modélisation 3D, Conservation du patrimoine

## RÉSUMÉ

Le Scanner Laser, appareil de numérisation et d'acquisition 3D, possède des atouts indiscutables pour le relevé d'éléments architecturaux et de larges scènes. En quelques minutes, il est possible d'obtenir des millions de points 3D avec une densité élevée et une précision de l'ordre de quelques millimètres, quand auparavant cela demandait des semaines par tachéométrie. Cependant, ces avantages certains peuvent être ternis par une phase de post-traitement exigeante, due au traitement d'énormes quantités de données brutes non structurées.

Il apparaît clairement la nécessité de trouver des solutions pour optimiser cette phase afin de la rendre plus simple, plus rapide et automatique, pour arriver un jour à une interaction utilisateur minime.

C'est au sein de l'Unité de Géomatique de l'Université de Liège, en Belgique, que ce travail d'optimisation et d'automatisation de la chaîne de traitement des données 3D a été mené. Il s'inscrit dans un projet attaché à la modélisation de la Cathédrale Saint-Paul de Liège pour un usage d'archéologie du bâti médiéval. Ce projet vise à retranscrire - grâce à un nuage de plus de 2 milliards de points décrivant intégralement l'intérieur et partiellement l'extérieur de cette œuvre religieuse - la réalité archéologique d'un monument aux détails très fins : ornements, sculptures, vitraux, gabarit des pierres, peintures murales, mortiers, ...

Cependant, la transmission et l'intégration de ces données brutes dans des bases de données ou leur diffusion sont très lourdes compte tenu de la grande quantité de points. L'exploitation future de ce nuage de points exige alors un traitement précis et pertinent pour créer, structurer, simplifier, analyser et modéliser les données acquises.

La segmentation est une étape essentielle dans la chaîne de traitement de nuage de points. Elle définit la pertinence et la précision de la reconstruction d'un modèle 3D, tout en offrant un haut niveau d'analyse de la scène. Son but est d'extraire d'un large volume de données 3D différents groupes qui partagent un lien logique pour les considérer comme une même entité (des techniques existantes sont reportées dans (Schnabel *et al.*, 2008), (Rabbani & Heuvel, 2005), (Castillo, 2013)).

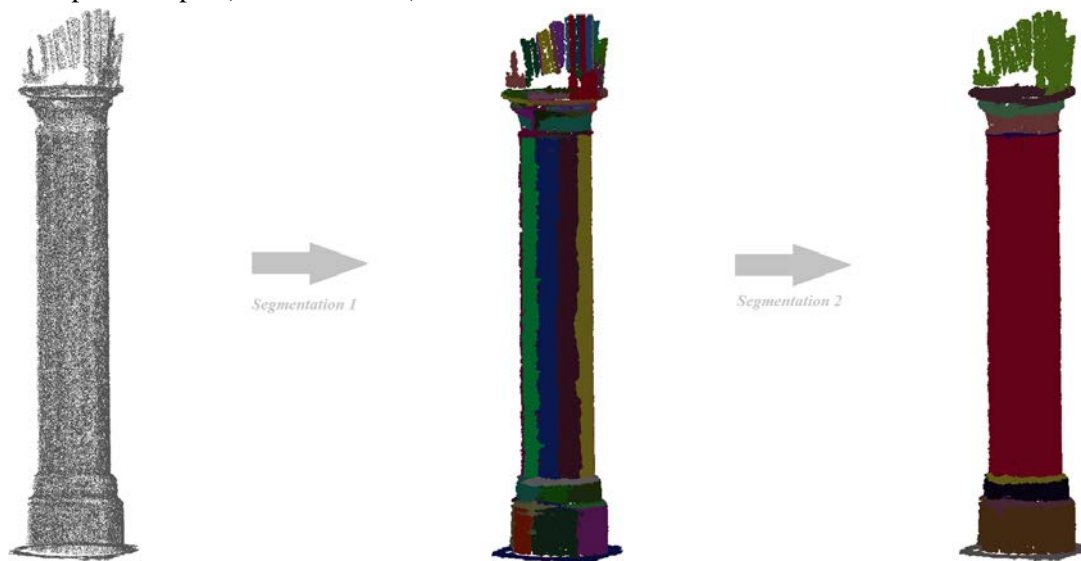
La méthode de segmentation mise en place s'appuie sur l'utilisation des normales. Elle s'articule sur une organisation en deux mouvements distincts. Dans un premier temps, la méthode regroupe par un processus de croissance de région tous les points qui ont la même normale. Puis, on étudie les relations (taille, connectivité, angle, symétrie, géométrie) entre chaque région et leurs voisins pour les regrouper dans un segment.

Nous obtenons finalement un nuage de point segmenté qui représente élégamment la scène et permet d'être la base d'une analyse future et d'une reconstruction.

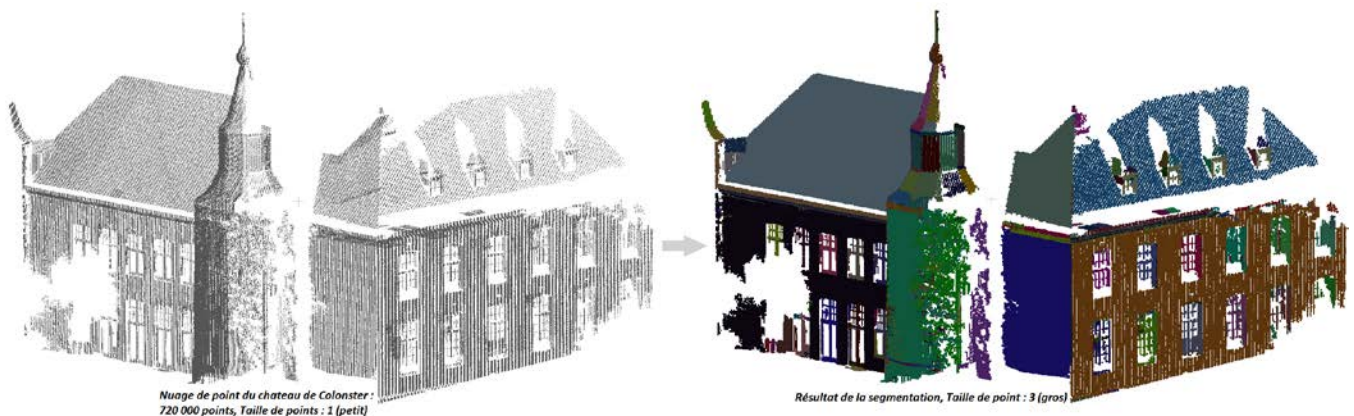
En plus d'être une méthode versatile, nous voulions un algorithme intuitif aux paramètres d'entrées qui ont du sens. Dans cette optique, nous avons décidé de créer une méthode de segmentation semi-automatique qui demande à l'utilisateur d'entrer 3 paramètres :

- Le critère de similarité : différence entre deux normales. Plus la valeur tend vers 0, plus le nuage de point sera segmenté (selon la qualité du nuage de point, choisir une valeur entre 0 et 1)
- Le rayon de recherche : Parce que la segmentation est basée sur la recherche de voisins selon une distance euclidienne, l'utilisateur est sollicité pour entrer une valeur qui constituera le rayon de recherche.
- Le détail minimum extrait : L'utilisateur rentre la valeur du détail minimum souhaité. Ce paramètre influe la seconde segmentation en adaptant une coupe pour limiter la recherche de voisins selon la symétrie, et ne dépassera pas la valeur entrée.

Nous avons testé le script développé sur différents nuages de points, et les résultats ont confirmé l'emploi à des applications très diverses. Sur des scènes très détaillées comme le nuage de 2 milliards de points de la cathédrale Saint-Paul, nous voulons un haut niveau de détail et une segmentation qui discerne les petites variations du rayon d'une colonne par exemple (voir ci-dessous)



Sur des scènes industrielles ou planaires, nous avons une reconnaissance rapide de ce qui est attendu, car la première partie de la segmentation regroupe les points qui appartiennent au même plan, et la deuxième partie réalisera un affinage (voir ci-dessous)



Différentes méthodes de modélisation post-segmentation sont introduites pour la création de différents modèles pour différentes applications, diffusables aisément au moyen de PDF 3D. Un modèle hybride simple, léger mais précis, permet la réduction du modèle tout en produisant un résultat précis.

Dans un souci applicatif complet, le développement d'une visite virtuelle WebGL innovante et saisissante a également été entrepris, ouvrant grandement les perspectives d'utilisation de nuages de points. L'affichage de millions/milliards de points RGB en temps réel sans temps de chargement via internet est réellement bénéfique dans la diffusion publique et l'utilisation de visites virtuelles, pour devenir une nouvelle façon de transmettre notre histoire aux futures générations.



# Vers de nouvelles perspectives lasergrammétriques : Optimisation et automatisation de la chaîne de production de modèles 3D

Mémoire d'Ingénieur ESGT, Le Mans 2013

---

## RÉSUMÉ

Les scanners lasers terrestres, présentant une rapidité de mise en place terrain, nécessitent des solutions pour optimiser et automatiser la phase de post-traitement chronophage.

La segmentation est une étape essentielle dans la chaîne de traitement de nuage de points. Elle définit la pertinence et la précision de la reconstruction d'un modèle 3D, tout en offrant un haut niveau d'analyse de la scène. Son but est d'extraire d'un large volume de données 3D différents groupes qui partagent un lien logique pour les considérer comme une même entité. La méthode semi-automatique développée extrait des formes d'un nuage de point représentant la cathédrale Saint Paul de Liège (Belgique), pour une étude en archéologie du bâti médiéval.

Nous obtenons un nuage de point segmenté qui représente élégamment la scène et en fait une base pour une analyse et reconstruction de haute qualité. Le développement d'une application de visite virtuelle au travers d'un développement WebGL devient une nouvelle façon de transmettre notre histoire aux futures générations.

**Mots clés : Nuage de points, Segmentation, Scanner Laser, Reconstruction, Extraction de formes, Conservation du patrimoine, Lasergrammétrie**

---

## SUMMARY

3D Laser scanner are very fast acquisition systems, but the huge amount of data collected demands processing optimization and automation.

Segmentation is an essential step in the production chain of point clouds processing. It defines the pertinence and accuracy of a future model reconstruction, as well as a high level analysis of any scene. Its goal is to extract from a large 3D dataset different groups of points that share a logical link to wisely consider them as one entity. The semi-automatic method developed extract shapes for an archeological study on medieval construction out of a point cloud acquired with a 3D Scanner laser, representing the cathedral Saint-Paul of Liège (Belgium).

We obtain a segmented point cloud that represents elegantly the scene and make it a base for a worthy analysis and reconstruction. The development of a virtual reality application through a WebGL development to promote cultural heritage transcend what exist nowadays, and become a new way of transmitting history to future generations.

**Key Words : Point cloud segmentation, Laser scanning, Monument Reconstruction, Feature extraction, Cultural Heritage, lasergrammetry**