



**HAL**  
open science

# Segmentation, classification et modélisation à partir de données issues de capteurs dynamiques

Rémi Reboul

► **To cite this version:**

Rémi Reboul. Segmentation, classification et modélisation à partir de données issues de capteurs dynamiques. Sciences de l'ingénieur [physics]. 2014. dumas-01179612

**HAL Id: dumas-01179612**

**<https://dumas.ccsd.cnrs.fr/dumas-01179612>**

Submitted on 23 Jul 2015

**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**

**REBOUL Rémi**

---

Segmentation, classification et modélisation à partir de données issues de  
capteurs dynamiques

**Soutenu le 09 juillet 2014**

---

**JURY**

**PRESIDENT :** M. Rani EI MÉOUCHE

**MEMBRES :** M. Cyril MICHON, Maître de stage  
Mme Elisabeth SIMONETTO, Professeur référent  
M. Jean-Michel FOLLIN  
M. Eric LABERGERIE  
M. Laurent POLIDORI  
M. Yannick ARDOUIN  
M. Yvan LIGOUT  
M. Hugo MEUNIER

## **Remerciements**

Tout d'abord je souhaite remercier mon maître de stage, Monsieur Cyril MICHON, pour sa disponibilité et ses conseils durant toute la période de ce TFE.

Je souhaite exprimer ma gratitude envers l'entreprise Fit Conseil pour m'avoir fait confiance et m'avoir permis de réaliser ce TFE au sein de leur société.

Je tiens à remercier Monsieur Guillaume Richard, Monsieur Emmanuel Gouraud et Monsieur Allan Rossignol du service topographie d'avoir pris le temps de répondre à mes questions et pour avoir partagé leurs connaissances à propos des logiciels Terrasolid et FME.

J'adresse mes remerciements à Madame Elisabeth SIMONETTO, professeur référent de mon TFE, pour le suivi de ce projet et ses conseils ainsi qu'à l'ensemble du corps enseignant.

Je remercie également Monsieur Nicolas LANDRIVON de la société TECHNODIGIT pour m'avoir permis d'utiliser le logiciel 3DReshaper pendant toute la durée de mon TFE.

Enfin, je tiens à dire merci par-dessus tout à mes parents, mon frère et mes proches pour leur soutien permanent tout au long de mes études.

## **Glossaire**

Segmentation d'un nuage de points : Etape de traitement qui consiste à découper un nuage de points en régions homogènes appelées clusters. (Ensemble de points connexes).

Classification : Méthode qui permet de labéliser les différents objets (sous-ensembles), ici produits par la segmentation.

Lasergrammétrie : Détermination de la forme et des dimensions d'un objet à partir de données laser.

Maillage : Représentation d'une image ou d'une surface par des éléments polygonaux (triangles, rectangles...).

## Table des matières

Remerciements.....	2
Glossaire.....	2
Table des matières.....	3
Introduction.....	5
<b>I ÉTAT DE L'ART .....</b>	<b>9</b>
I.1 LES SYSTEMES DE CARTOGRAPHIE MOBILES ( <i>MOBILE MAPPING SYSTEMS</i> ) .....	9
I.1.1 Le système de position et d'orientation .....	9
I.1.2 Les capteurs de perception.....	9
I.1.2.1 Les scanners .....	9
I.1.2.2 Les caméras.....	10
I.1.3 Les systèmes de cartographie mobile existants.....	10
I.2 LE TRAITEMENT DES DONNEES .....	12
I.2.1 Segmentation et classification.....	13
I.2.2 La modélisation.....	16
I.2.3 Le texturage .....	17
I.2.4 Etude de quelques logiciels de modélisation et de texturation.....	18
I.3 LES SIG ET LA 3D .....	20
I.3.1 Les formats 3D.....	20
I.3.2 Les modélisations 3D urbaines .....	21
<b>II SEGMENTATION ET CLASSIFICATION.....</b>	<b>23</b>
II.1 LE SOL.....	23
II.1.1 Méthode .....	23
II.1.2 Evaluation de la classification .....	25
II.2 LES FAÇADES.....	27
II.2.1 Méthode .....	27
II.2.2 Évaluation de la segmentation .....	29
II.3 LES OBJETS URBAINS .....	30
II.3.1 Méthode .....	30
II.3.2 Evaluation de la segmentation .....	31
II.4 CONCLUSION .....	31
<b>III GENERATION D'UN MODELE 3D.....</b>	<b>32</b>
III.1 MODELISATION D'UNE FAÇADE.....	32
III.1.1 Modélisation par maillage.....	32
III.1.2 Modélisation à partir de primitives géométriques .....	34
III.1.2.1 Création d'un plan.....	34
III.1.2.2 Digitalisation.....	35
III.1.2.3 Modélisation 3D.....	36
III.1.3 Comparaison des deux méthodes .....	37
III.2 MODELISATION D'UN ENSEMBLE DE FAÇADES .....	38
III.2.1 Création de plans.....	38
III.2.2 Détection de contours.....	38
III.2.3 Modélisation de contours .....	40
III.2.4 Évaluation de la modélisation .....	42
III.2.4.1 Évaluation qualitative .....	42
III.2.4.2 Évaluation quantitative.....	43

III.2.5	Modification de contours .....	44
III.3	MODELISATION D'OBJETS URBAINS .....	45
III.3.1	Calcul des coordonnées du point d'insertion .....	45
III.3.2	Calcul de l'orientation.....	46
III.3.3	Insertion sous Rhinocéros 3D .....	47
III.4	CONCLUSION .....	47
	Conclusion et perspectives .....	48
	Bibliographie.....	50
	Poster scientifique .....	52
	Table des annexes .....	53
	Annexe 1 Macro sol .....	54
	Annexe 2 Script FME modélisation de contours.....	55
	Annexe 3 Modélisation et insertion automatique des balcons .....	57
	Annexe 4 Script FME modélisation d'objets ponctuels.....	61
	Annexe 5 Script FME modélisation d'objets linéaires.....	63
	Annexe 6 Script Python modélisation d'objets urbains .....	65
	Liste des figures .....	67
	Liste des tableaux.....	69

## **Introduction**

### **Présentation de l'entreprise d'accueil**

J'ai réalisé mon stage de fin d'études au sein de FIT Conseil, une société anonyme de géomètres experts dont le siège social est situé à Nantes (44) et qui exerce son activité professionnelle depuis 1968.

Fit Conseil est partenaire du groupe FIT qui comprend FIT ESIC et Memoris.

FIT ESIC (plus de 170 collaborateurs) est une société d'ingénierie spécialisée dans les études, la maîtrise d'œuvre, le suivi des travaux, l'assistance technique, l'expertise des mesures topographiques et la métrologie industrielle.

Memoris (plus de 40 collaborateurs) est une société spécialisée dans la constitution de systèmes d'information dans les domaines de l'aménagement et de la gestion du territoire qui compte plus de 40 collaborateurs.

FIT Conseil compte 240 collaborateurs répartis dans 7 agences en France et offre ses services aux particuliers, aux entreprises et aux collectivités en France et à l'étranger.

Il s'agit d'une société spécialisée dans la topographie, l'acquisition et le traitement des données aériennes, les activités foncières, les réseaux et la description des patrimoines techniques et immobiliers, ce qui représente un potentiel d'activité très étendu.

L'entreprise ne cesse d'investir dans de nouvelles technologies lui permettant de gagner des marchés dans de nouveaux domaines.

En juin 2007, FIT Conseil a formalisé sa démarche qualité par la certification ISO 9001-2000. En juin 2014, cette certification a été renouvelée selon la norme ISO 9001-2008 et depuis octobre 2011, FIT Conseil est certifié Sécurité et Environnement (14001 et 18001).

### **Contexte de l'étude**

#### **La cartographie 3D**

Depuis plusieurs années déjà la troisième dimension a fait son apparition dans les systèmes d'information géographique (SIG) et s'est ouvert au grand public rendant la demande de modélisation 3D de plus en plus forte.

Ces données ne cessent d'évoluer se retrouvant aujourd'hui dans de multiples applications :

- La communication et le tourisme à travers des visites virtuelles

La 3D va nous permettre de découvrir un site (ville, quartier...) de manière intuitive en nous plongeant dans une immersion virtuelle.



Figure 1: Photographie de la ville de Paris [GoogleEarth]

➤ La planification et la gestion urbaine

La 3D est un outil d'aide à la décision pour les collectivités publiques permettant de mettre en scène tout projet de construction, d'aménagement ou de renouvellement urbain.

La ville du Havre a par exemple exploité sa propre maquette 3D pour mener à bien un projet de renouvellement urbain.



Figure 2: Réaménagement de l'ilot "Jules Vallés" dans la ville du Havre [Banaszak et Koehl., 2008]

➤ La navigation

La principale utilisation de la cartographie pour la plupart des gens réside dans le fait de se déplacer et d'arriver à bon port.

Comme l'illustre la photo ci-dessous, un nombre croissant de GPS intègrent la donnée 3D dans leur système de navigation mobile, qui permet à l'utilisateur de s'orienter de façon très réaliste.



Figure 3: GPS utilisant de la donnée cartographique 3D [www.navigadget.com]

- La simulation des risques (inondations, tremblements de terre...)

À partir d'une maquette 3D, il est possible de simuler la hauteur du niveau d'eau partout dans la ville y compris dans les constructions souterraines de type parking ou tunnel.

Il est également possible de visualiser l'impact d'un tremblement de terre.



Figure 4: Simulation 3D d'une inondation à la confluence de la Seine et la Marne [Xavier Opigez IAU-îdF, 2013]

## Les différentes méthodes de relevés 3D

### **La photogrammétrie**

Historiquement cette méthode est basée sur la perception humaine du relief par observation stéréoscopique, principe expérimenté par Aimé Laussedat (officier topographe) dès le 19<sup>ème</sup> siècle.

Ce procédé permet d'obtenir un modèle tridimensionnel à partir d'un ensemble d'images couvrant la même zone et prises sous différents points de vue.

Aujourd'hui, à partir de l'association des points images du même point de la scène, il est possible de réaliser une reconstruction 3D et ainsi d'obtenir un nuage de points.

### **La radargrammétrie**

Il s'agit d'une technique équivalente à la photogrammétrie mais qui va utiliser des images radar pour déterminer les caractéristiques et les formes du terrain.

L'imagerie radar repose sur la mesure du trajet aller-retour de l'émission d'impulsions micro-onde (longueurs d'onde comprises entre 1 mm et 1 m).

### **La lasergrammétrie**

Cette technique repose sur l'acquisition de données laser donc sur l'émission et la réception de lumière (impulsions dans le visible et le proche infrarouge).

Afin d'obtenir des renseignements sur les dimensions des différents éléments d'aménagement urbain, le gabarit des rues et les espaces disponibles au sein du domaine public, de plus en plus de villes comme celle de Genève par exemple [Gaillet et al., 2013] demandent une représentation en trois dimensions laissant apparaître les éléments de façade, le mobilier urbain ainsi que les panneaux de signalisation.

Les moyens de relevé topographique ont donc dû s'adapter à cette nouvelle demande laissant apparaître les scanners laser 3D mobiles (*Mobile Mapping Systems*). Cette technique permet en effet d'effectuer des relevés laser en environnement urbain avec un rendement nettement plus



important que les systèmes fixes et avec précision du détail plus grande que les systèmes aéroportés.

## **Problématique**

Depuis l'acquisition d'un laser scanner mobile terrestre (Streetmapper de 3DLM), l'entreprise offre des prestations répondant à l'acquisition terrestre haute résolution, autrement dit, qui comprennent la livraison des nuages de points géoréférencés et classifiés ainsi que la livraison des images géoréférencées.

Aujourd'hui, le besoin grandissant en modélisation 3D nous amène à nous poser la question suivante : comment passer d'un semi de points 3D acquis par MMS à un modèle 3D exploitable?

Afin de répondre à ce besoin, des méthodes de segmentation et de classification automatiques permettant de différencier les éléments urbains (routes, trottoirs, façades, végétation, mobilier urbain) seront étudiées et proposées sur des échantillons de la ville de Rennes. Par la suite deux méthodes de modélisation seront comparées et il sera proposé une solution permettant d'automatiser une des étapes de la modélisation choisie.

Le but est de vérifier si, à partir de données extraites des capteurs dynamiques, il est possible de générer un modèle permettant de satisfaire à des critères de précision topographique, de réalisme et de facilité d'utilisation.

Pour des raisons d'identification et de réalisme, les objets représentant le mobilier urbain comme les lampadaires, les bornes ou les panneaux peuvent être modélisés en 3D. Il est donc nécessaire de réfléchir sur une méthode d'import automatique de cellules 3D à partir d'une bibliothèque existante.

# I État de l'art

Ce chapitre a pour objet de renseigner le lecteur sur les différentes phases de traitement permettant d'aboutir à un modèle 3D réaliste à partir de données acquises avec un système de cartographie mobile. Une étude sera également portée sur les différents logiciels de modélisation, les formats 3D et les représentations de villes numériques existantes.

Je commencerai par faire un point sur les systèmes de cartographie mobiles.

## I.1 Les systèmes de cartographie mobiles (*Mobile Mapping Systems*)

La plupart des systèmes de cartographie mobiles sont composés de deux éléments :

- Un système de position et d'orientation
- Des capteurs de perception

Certains systèmes utilisent des données images seules, c'est par exemple le cas du système mis en place par Google pour acquérir les données utilisées dans Street View (*Google Maps*).

### I.1.1 Le système de position et d'orientation

Pour connaître la position et l'orientation du véhicule en tout point de sa trajectoire, on utilise un système généralement composé d'un récepteur GNSS, d'une centrale inertielle et d'un odomètre. Ce système est appelé POS (*Position and orientation system*).

Pour déterminer le positionnement, la méthode utilisée par la cartographie mobile est le mode cinématique différentiel en temps différé [Morel, 2013]. Dans de bonnes conditions, la précision de géoréférencement est de l'ordre de quelques centimètres mais peut rapidement se détériorer en cas de non fixation des ambiguïtés entières (GDOP élevé).

Pour obtenir les informations concernant l'orientation, la centrale inertielle (*Inertial Measurement Unit, IMU*) est constituée de trois gyromètres et de trois accéléromètres. Associés à un calculateur réalisant l'intégration en temps réel des angles d'attitudes, de la vitesse et de la position, la centrale inertielle devient un système de navigation inertielle (*Inertial Navigation System, INS*).

Le POS est donc un élément important pour un système de cartographie mobile car la précision des points 3D acquis est grandement dépendante de la précision du géoréférencement.

### I.1.2 Les capteurs de perception

#### I.1.2.1 Les scanners

Le principe du scanner laser est de mesurer la distance qui le sépare d'une surface en utilisant la lumière comme source d'information. L'avantage de cette technologie est l'obtention rapide et automatique de points en 3D avec une densité très élevée contrairement à un tachéomètre.

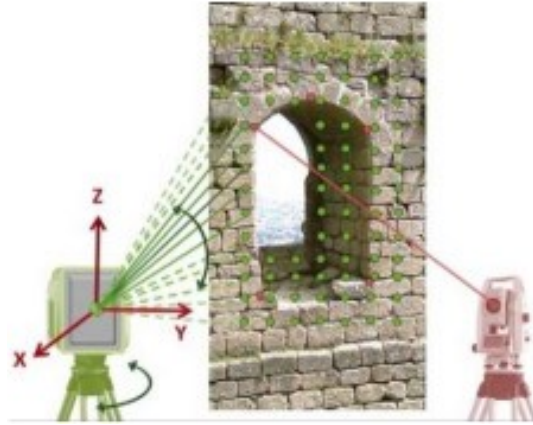


Figure 5: Relevé d'une ouverture à partir d'un scanner laser terrestre (à gauche) et au tachéomètre sans réflecteur (à droite) [Landes et Grussenmeyer., 2011]

On va pouvoir distinguer trois catégories de scanners laser :

- Les scanners laser à temps de vol  
« Le scanner laser émet un faisceau laser pulsé et la distance est déterminée par la mesure du temps que met le train d'impulsion pour faire l'aller-retour ». [Cali et al., 2013]
- Les scanners laser à différence de phase  
« Le scanner laser émet un faisceau laser continu modulé et la distance à l'objet est déterminée par la différence de phase entre la modulation de lumière reçue et envoyée » [Cali et al., 2013]. Cette technique est adaptée à la mesure d'objets peu éloignés.
- Les scanners laser à triangulation  
La distance entre le scanner laser et l'objet est calculée à partir d'une résolution de triangle, méthode adaptée pour numériser des objets à une courte distance.

### I.1.2.2 Les caméras

La plupart du temps, le scanner laser est associé à plusieurs caméras permettant la prise d'images continues dans toutes les directions et de façon panoramique.

Ces images permettront par la suite la colorisation ou le texturage des nuages de points pour les systèmes comprenant un scanner laser et d'effectuer des mesures de stéréovision pour les systèmes basés sur la photogrammétrie.

Les types de caméras utilisées sont des caméras numériques comprenant généralement de petites résolutions et un temps d'exposition faible : cela permet d'optimiser le nombre d'images.

### I.1.3 Les systèmes de cartographie mobile existants

Depuis le premier système de cartographie mobile opérationnel appelé GPSVan<sup>TM</sup> développé en 1991 par le centre de cartographie à l'université d'état de l'Ohio [Ellum et El-Sheimy., 2002], les systèmes se sont développés et ont évolué pour répondre à une demande de plus en plus grande vis-à-vis de la modélisation 3D urbaine.

Paramètres	StreetMapper 3DLM	Leica Pegasus	Optech Lynx SG1	Riegl VMX-450
<b>POS</b>	Terra control	Novatel SPAN	POS-LV 520	POS-LV 510
Fréquence	512Hz	200 Hz	200 Hz	...
Odomètre	Oui	Oui	Oui	Oui
<b>Laser</b>	2 scans 2D 360°	2 scans 2D 360°	2 scans 2D 360°	2 scans 2D 360°
Fréquence max	2x550 KHz	2x1000 KHz	2x600 KHz	2x550 KHz
Rotation max	200 Hz	100 Hz	250Hz	200Hz
<b>Caméras</b>	Ladybug 5	6 caméras	4 caméras	6 caméras
Résolution	6x5MP	6x2MP	4x5MP	6x5MP
Frames/sec	10 fps	8 fps	2 fps	8 fps
Champs de vision	360°	360°x70°	360°	6x(80°x65°)

Figure 6: Tableau des caractéristiques des différents systèmes de cartographie mobile existants. [3DLM, 2014], [Leica, 2014], [Optech, 2014], [Riegl, 2014]. Le système utilisé dans le cadre de ce TFE est le StreetMapper de 3DLM.

L'évolution rapide des capteurs rend déjà obsolète le tableau présenté ci-dessus.

Après avoir présenté les différents systèmes de cartographie mobile, nous allons maintenant nous intéresser à la qualité des données acquises.

Un projet nommé « EuroSDR Mobile Mapping-Road Environment Mapping using Mobile Laser scanning » développé par l'organisation européenne pour la recherche des données spatiales (*European Organization for Spatial Data Research (EuroSDR)*) a été mené sur le site d'Espoonlahti (quartier de la ville d'Espoo en Finlande), son objectif étant d'évaluer la qualité de mesure de différents systèmes de scanners laser mobiles terrestres sous de bonnes conditions GNSS. [Kaartinen et al., 2013]

Un réseau géodésique a été utilisé comme référence pour l'analyse de la qualité des données. Les différents modèles de scanners laser mobiles commerciaux ou de recherche qui ont été comparés dans ce projet sont les suivants : le RIEGL VMX-250, le Streetmapper 360, l'Optech Lynx, le FGI Roamer et le FGI Sensei.

Le premier test concerne la qualité des données altimétriques. Pour cela, un modèle digital de surface du sol a été produit en utilisant des stations totales et des scanners laser terrestres. Le modèle compte 3283 points de référence. L'outil « Output control report » du logiciel TerraScan (*Terrasolid, 2005*) a été utilisé pour la comparaison des points. Les résultats obtenus sont les suivants :

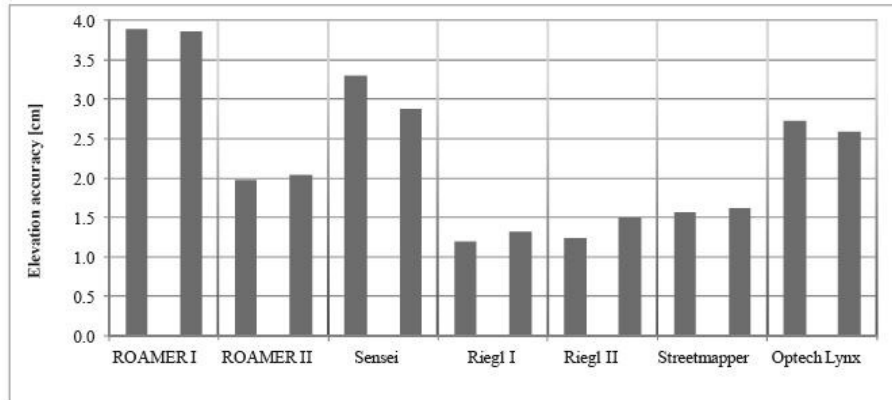


Figure 7: Valeurs de la précision d'élévation en cm pour les différents scanners laser mobiles utilisés. La colonne de gauche pour un sens de marche anti-horaire (CCW) et la colonne de droite pour un sens de marche horaire (CW). [Kaartinen et al., 2013]

Nous pouvons remarquer que les meilleurs systèmes peuvent atteindre des précisions comprises entre 1 et 2 centimètres, pour des distances de 35 mètres en moyenne.

Le second test concerne la précision planimétrique : 273 références ont été mesurées. La précision planimétrique a été évaluée en mesurant une cible de référence dans le nuage de points reçu et en calculant les différences en Est et en Nord.

	ROAMER I		ROAMER II		RIEGL I		RIEGL II		Streetmapper		Optech Lynx	
	CCW	CW	CCW	CW	CCW	CW	CCW	CW	CCW	CW	CCW	CW
<b>N</b>	124	120	136	120	169	178	193	192	190	177	160	160
<b>Mean</b>	3.4	3.5	2.2	1.8	2.3	1.5	1.6	2.0	1.7	1.8	3.2	3.7
<b>Min</b>	0.4	0.5	0.2	0.3	0.1	0.0	0.1	0.0	0.0	0.2	0.1	0.3
<b>Max</b>	8.7	11.8	6.7	5.1	11.7	11.7	11.5	10.6	9.4	9.7	12.3	13.2
<b>Std</b>	1.8	2.2	1.1	0.9	1.7	1.3	1.4	1.5	1.5	1.3	1.8	2.3

Figure 8: Valeurs de la précision planimétrique en cm pour les différents systèmes de scanners laser mobiles utilisés dans les deux sens de la marche. [Kaartinen et al., 2013]

N représente le nombre de points de références utilisés et Std (standard déviation) l'écart type. Nous pouvons constater que la précision planimétrique oscille autour des 2 centimètres. Il est même possible d'améliorer cette précision en utilisant des points de calage situés de manière régulière sur le projet.

Il est nécessaire de connaître l'ordre de précision des points 3D obtenus lors de l'acquisition car cette précision va se répercuter sur les étapes de traitement des données.

La différence entre les données acquises à partir de systèmes d'acquisition mobiles et de systèmes d'acquisition fixes réside dans le bruitage et la structuration de celles-ci. En effet un nuage de points acquis par un système de lasergrammétrie mobile présentera une densité variable et du bruit de mesure compte tenu du déplacement et du nombre de passages du véhicule lors de l'acquisition.

## 1.2 Le traitement des données

Bien que l'acquisition des données soit relativement rapide (préparation de la mission, acquisition dynamique), la phase de traitement s'avère longue et complexe. Elle comporte six étapes :

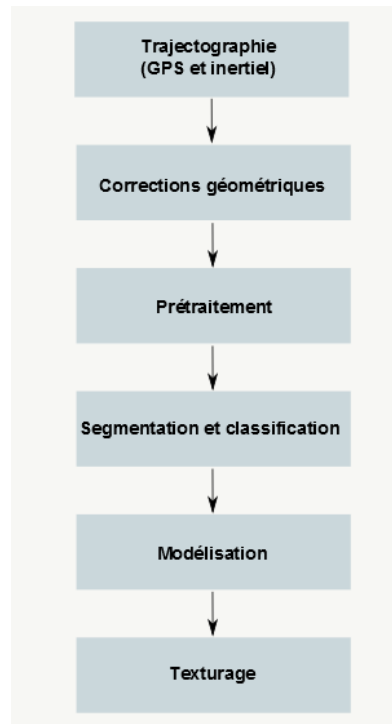


Figure 9: Différentes étapes réalisées lors du traitement des données

Seules les trois dernières étapes vont être abordées par la suite.

### I.2.1 Segmentation et classification

La segmentation est une étape importante car la qualité du modèle final va dépendre étroitement de celle-ci.

Dans un premier temps, la segmentation a été utilisée dans le traitement d'images dont on peut différencier trois méthodes : la segmentation par seuillage, par frontière et par région. Ces méthodes ont ensuite été adaptées pour être utilisées dans la segmentation de points 3D.

Beaucoup de travaux portent sur l'automatisation de la segmentation car cette étape peut s'avérer longue, fastidieuse et imprécise si l'on dispose d'un nombre de points importants. Avant d'étudier différentes méthodes de segmentation, une approche permettant de classier automatiquement les différents éléments urbains est présentée.

#### ➤ Classification d'objets

Un lieu urbain présente des ensembles différents tels que les bâtiments, la chaussée, les trottoirs, les tunnels, les véhicules ou encore les arbres. L'université de Tokyo [Manandhar et Shibasaki, 2001] a proposé des solutions permettant de classier ces différents ensembles en se basant sur la densité des points.

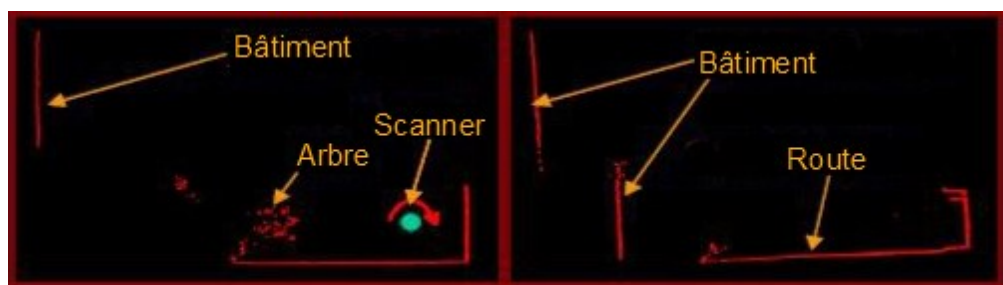


Figure 10: Points issus d'un balayage laser. [Manandhar et Shibasaki, 2001]

Nous pouvons constater sur la Figure 10 que les points issus d'objets naturels (arbres, buissons) sont écartés compte-tenu du fait qu'ils décrivent des formes géométriques irrégulières tandis que les points issus d'ensembles réalisés manuellement (routes, bâtiments) sont denses du fait qu'ils représentent une certaine forme géométrique.

En plus de la distance entre les points, [Manandhar et Shibasaki, 2001] s'appuient également sur l'orientation géométrique d'un ensemble de points, notamment pour l'extraction de la route et des faces de bâtiments. En effet, ces dernières seront représentées par des points alignés verticalement alors que la route sera visualisée par des points alignés sur une pente horizontale aux alentours de 1%. L'algorithme ne peut cependant pas délimiter la limite entre deux faces de bâtiments différents.

➤ Segmentation par objet

Une méthode de segmentation urbaine est fondée sur une approche à base de super-voxels qui sont liés entre eux pour former des objets par une méthode de chaînage. [Aijazi et al., 2012]

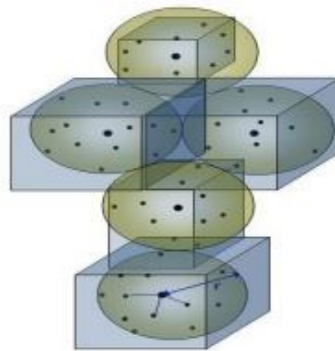


Figure 11: Regroupement de plusieurs points 3D en voxels parallélépipédiques de taille maximale  $2r$ . La taille des voxels est définie par les valeurs extrémales des coordonnées des points voisins rassemblés au sein du voxel. [Aijazi et al., 2012]

Après avoir converti les voxels en super voxels à partir de certaines propriétés (normales aux surfaces, couleur RVB, intensité, primitives géométriques...), ceux-ci sont segmentés en objets suivant un algorithme basé sur une méthode de chaînage. Cette méthode a l'avantage de réduire l'ensemble des données ainsi que le temps de post-traitement au détriment de la précision globale. Les différents objets créés pourront par la suite être classifiés en fonction de leurs caractéristiques géométriques et locales (la normale à la surface, le barycentre et le centre géométrique, la couleur et l'intensité, la forme).

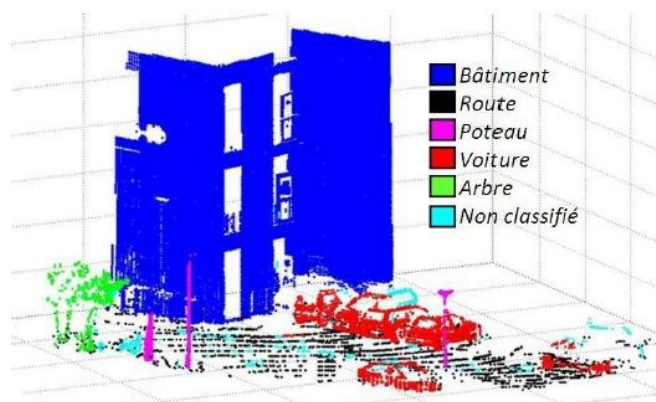


Figure 12: Résultat d'une classification obtenue après création de voxels et segmentation. [Aijazi et al., 2012]

➤ Segmentation par reconnaissance de formes

D'autres méthodes de segmentation sont basées sur la détection de formes géométriques (plan, cylindre...). Cette détection peut se faire de façon automatique en utilisant notamment les algorithmes RANSAC [Fischler et Bolles., 1981] et la transformée de Hough [Hough, 1962].

L'algorithme RANSAC a été développé au départ par [Fischler et Bolles., 1981] comme un outil d'analyse d'image et de lissage de données. Aujourd'hui, l'algorithme a évolué et permet de réaliser une segmentation automatique sur un nuage de points LiDAR. Cet algorithme va permettre, de façon itérative, d'extraire des primitives géométriques en sélectionnant aléatoirement des groupes de points. Son point fort réside dans sa robustesse qui permet de traiter des données bruitées et qui le rend intéressant dans le traitement des données LiDAR.

La transformée de Hough est une technique de reconnaissance de forme qui a été inventée en 1962 par Paul Hough pour le traitement d'images numériques mais qui peut également aujourd'hui s'appliquer aux données laser (transformée de Hough 3D). [Vosselman et al., 2004] a par exemple utilisé cette transformée pour segmenter des données laser acquises sur des équipements industriels. L'un des principaux défauts de cette méthode est le temps de calcul qui peut s'avérer long si l'on dispose de données importantes, souvent le cas pour un nuage de points.

[Boulaasal, 2010] a par exemple développé une segmentation basée sur la reconnaissance de la primitive plan à partir de données acquises avec un scanner laser terrestre en s'appuyant sur la combinaison de l'algorithme RANSAC et d'un algorithme de croissance de région.

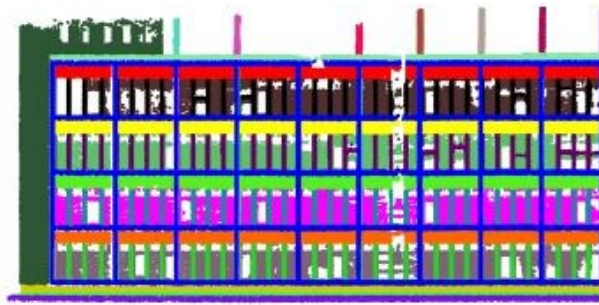


Figure 13: Segmentation automatique de la façade de l'INSA de Strasbourg. Chaque couleur se rapporte à un segment plan. [Boulaasal, 2010]

➤ Segmentation basée sur la croissance de surfaces

Une autre approche consiste à utiliser un algorithme de croissance de surface pour segmenter un nuage de point.

[Vosselman et al., 2004] a par exemple utilisé un algorithme de croissance de région pour segmenter les façades de bâtiments. Plusieurs variantes de cet algorithme se sont développées mais le principe reste le même et consiste à identifier des « surfaces graines » (surfaces de départ) pour ensuite les faire croître vers les points voisins.

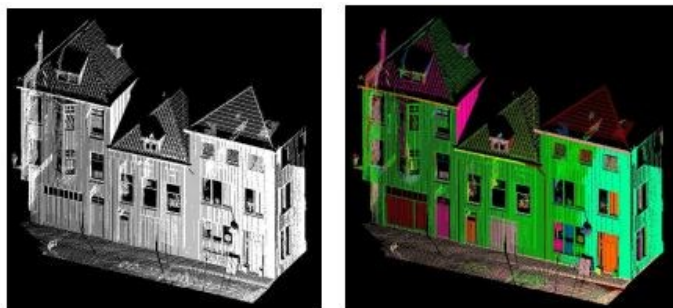


Figure 14: Façade segmentée avec l'algorithme de croissance de région. A gauche: Numérisation de la façade par scanner laser terrestre; A droite : résultat de la segmentation. [Pu et al., 2007]



Bien que cet algorithme présente des avantages en terme de rapidité d'implémentation, les données bruitées ne favorisent pas la sélection juste des « graines de départ ». Cela dit plusieurs solutions ont été proposées pour améliorer sa fiabilité en intégrant par exemple un critère de planéité globale [Yu et al., 2008].

### I.2.2 La modélisation

« Le but de la modélisation est de construire et de créer des liens entre des points caractéristiques du nuage, afin de réaliser un tout, qui peut être un bâtiment, une façade ou tout autre objet » [Boulaasal, 2010].

Nous allons distinguer deux types de modélisation utilisés dans les logiciels de traitement de nuage de points :

- La modélisation par maillage (ou polygonale)

Le principe de cette modélisation est de constituer des facettes entre les différents points du nuage en utilisant généralement des algorithmes de triangulation (de Delaunay).

L'avantage de cette modélisation réside dans sa fidélité à représenter la réalité d'un modèle. Nous allons pouvoir parler ainsi de modèle « *tel que saisi* ». Autrement dit, il s'agit de la modélisation du modèle en état tel qu'il a été levé lors de l'acquisition. L'inconvénient de ce type de modélisation réside dans la lourdeur des fichiers qui peut s'avérer être un réel handicap notamment quand il y a une multitude d'objets à représenter.

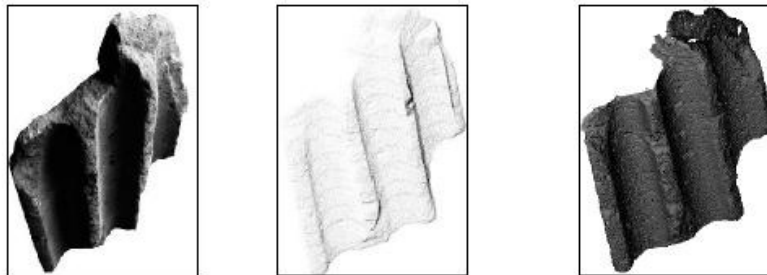


Figure 15: Exemple de modélisation par maillage-photographie (gauche)-nuage de points (centre)-maillage (droite). [Begrache, 2003]

À partir de la Figure 15, nous pouvons remarquer que ce type de modélisation s'adapte parfaitement aux objets de formes géométriques complexes. Ce type de modélisation est d'ailleurs utilisé pour la représentation de monuments, de statues ou de tout autre objet complexe avec de nombreux détails qu'il est nécessaire d'archiver ou de restaurer. Le rendu peut se faire sous la forme d'un modèle filaire ou d'un modèle maillé surfacique.

- La modélisation géométrique (à partir de modèles)

« La modélisation géométrique est basée sur la construction géométrique d'éléments identifiables et descriptibles d'un point de vue mathématique dans le nuage de points » [Boulaasal, 2010].

L'objet va donc être représenté à partir de primitives géométriques (plan, sphère, cône,...), ce qui va simplifier la modélisation et va la faire rentrer dans la catégorie des modèles « *tels que construits* ».



Figure 16: Exemple de modélisation géométrique-photographie (gauche)-nuage de points (centre)-modélisation géométrique (droite). [Begrliche, 2003]

À travers cet exemple, nous pouvons remarquer que la modélisation ne représente pas les dégradations que l'objet a subi au cours du temps mais va représenter celui-ci sous une forme géométrique parfaite.

Cette modélisation géométrique est souvent utilisée dans les scènes industrielles qui sont constituées en grande partie de lignes de tuyauterie. Le rendu peut se faire sous la forme d'un modèle surfacique ou volumique.

La modélisation géométrique peut être obtenue à partir de différentes méthodes :

- A partir d'une définition paramétrique des objets. Chaque objet est défini à partir de différents paramètres ou caractéristiques mathématiques.
- Par le biais de primitives géométriques qui peuvent être obtenues de façon manuelle ou automatique à partir d'un nuage de points.

Certains modèles cumulent ces deux types de modélisation leur permettant de disposer d'un maillage par triangulation pour les éléments complexes comme les statues par exemple et de primitives géométrique pour la représentation d'éléments plus simples. Le modèle de rue représente une combinaison de ces deux types de modélisation.

La reconstruction tridimensionnelle peut être réalisée à partir d'images, de nuages de points ou d'une utilisation conjointe de ces deux procédés. Je ne m'attarderai pas sur les caractéristiques et l'évaluation de ces différentes méthodes de modélisation qui sont présentées dans la thèse de Lionel PÉNARD [Pénard, 2009]. Dans le cadre de cette étude, seules les données LiDAR seront utilisées pour la modélisation.

### **I.2.3 Le texturage**

Une fois l'étape de modélisation terminée, il est nécessaire d'appliquer une texture sur le modèle généré afin de le rendre le plus réaliste possible. Il est possible d'effectuer un texturage simple basé sur des motifs répétitifs choisis dans une bibliothèque de textures ou un texturage réaliste obtenu à partir d'images numériques.

Le plaquage automatique des photos sur le modèle nécessite de connaître la position et l'orientation de chaque image, ce qui est le cas pour les photos acquises à partir d'un système d'acquisition mobile. Lorsque ce n'est pas le cas, une autre solution consiste à créer un panorama (assemblage de plusieurs photographies) et à le redresser de façon à venir le plaquer sur le modèle.

De nombreux articles traitent de la texturation automatique. C'est le cas de l'article [Benitez et Baillard, 2009] qui aborde trois méthodes basées sur le *Ray-tracing* et le *Z-buffering* permettant de sélectionner automatiquement un jeu d'images avant d'appliquer la texture. Nous retiendrons la méthode du *3D Ray-Tracing* car c'est celle qui donne les résultats les plus concluants.

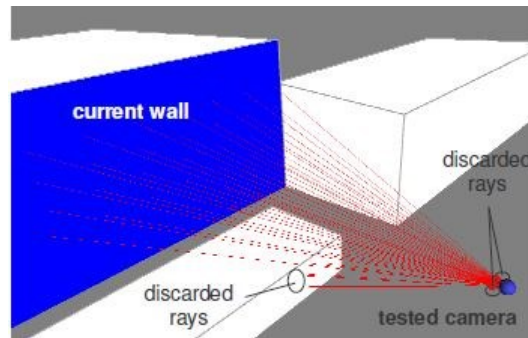


Figure 17: Principe du 3D ray tracing. [Benitez et Baillard., 2009]

Pour chaque mur une liste de caméra est présélectionnée à partir de quatre critères : la distance entre la façade et la position de la caméra, la position de la façade dans le demi-plan face à la caméra, la position du mur par rapport à la caméra et l'angle entre l'axe de prise de vue et la normale du bâtiment. L'analyse s'effectue mur par mur et non pas caméra par caméra. Une fois la caméra présélectionnée, les rayons qui intersectent une face plus proche que la face qui est en cours d'analyse sont rejetés.

[Bénitez et al., 2010] propose une méthode permettant de rectifier automatiquement les problèmes d'occlusion sur les textures de façades à partir de données issues d'un système de cartographie mobile. La méthode comprend trois étapes :

- la rectification géométrique de chaque image de façade.
- la détection des occlusions « prédites » et « non prédites » à partir des données laser et image.
- la fusion des images rectifiées des occlusions.

Les occlusions « prédites » sont causées par les immeubles ou les murs à l'intérieur du model 3D et les occlusions « non prédites » sont causées par les objets qui ne sont pas modélisés comme les piétons ou les véhicules.

#### **I.2.4 Etude de quelques logiciels de modélisation et de texturation**

##### ➤ 3DReshaper

3DReshaper est un logiciel de traitement de nuages de points, spécialisé dans le maillage 3D, la reconstruction de surfaces et la comparaison tridimensionnelle, il est produit par la compagnie Technodigit (France) faisant partie du groupe Hexagon Metrology depuis 2009 [Technodigit, 2014].

Concernant la modélisation 3D, ce logiciel permet de créer des surfaces à partir de points via la modélisation polygonale ou via la création de surfaces NURBS (Non Uniform Rational Basis Splines). Il sera également possible d'appliquer l'image ou la texture sur le modèle 3D.

Le logiciel n'a apparemment pas de limitation en ce qui concerne la taille des nuages de points ou des maillages qu'il peut prendre en charge. Il dispose de nombreuses fonctions intéressantes comme la segmentation de points automatique et la détection de lignes caractéristiques, ce qui le rend particulièrement adapté pour la réalisation de ce projet de fin d'étude. Fit Conseil travaille actuellement avec le logiciel de traitement de point Terrascan de Terrasolid [Terrasolid, 2014], il faut donc s'assurer de la compatibilité entre les formats d'exports de Terrascan et d'import de 3DReshaper. Nous pouvons remarquer ci-dessous que 3DReshaper permet d'importer des nuages de points au format \*.las, format d'export de Terrascan.

	EXPORT	IMPORT
Nuage de points	Fichiers ASCII (*.asc / *.csv ...) Fichiers binaires (*.nsd) PCDMIS format (*.dms) Leica Geosystems (*.pts) IGES (*.igs)	Fichiers ASCII (asc / csv / xyz / xyz- raw...) Leica Geosystems (*.pts - .pcs - .pcst - .pcg - .pcsx - .pcix - .pci - .pci - .ptx) Leica Nova MSD50 (*.sdb / *.xml) ShapeGrabber (*.3pi) STL file Binary (*.nsd) Tool path (*.iso) AutoDesk DXF (*.dxf) STL (*.stl) DMIS file (*.dms / *.rnr) GSCAN file (*.gsn) Confocal Scanning Laser Microscope (*.csv) NIKON Nexiv video measuring system (*.csv) Konica Minolta Vivid (*.cdm) Perceptron SWB/SWL (*.swb / *.swl) Konica Minolta CDK (*.cdk) Leica Tscan + Steinbichler (*.ac) Données de points LIDAR (*.las) Autres ASCII (*.*) Zoller and Fröhlich (*.zfs - *.zfc) PLY points sans facettes (*.ply) ESRI ASCII (format raster *.asc) FARO (*.fls - *.fws) POLYWORKS (*.psl) Coreview (*.atr / *.ftr)
Maillage	Format STL Ascii et binaire (*.stl) Binary PBI (*.pbi) DXF 3Dface (*.dxf) Ascii POLY (*.poly) Uniquement les sommets (Vertices) (*.asc) DXF polyligne (*.dxf) Universal IDEAS (*.unv) OBJ (*.obj) STEP (*.stp) Ascii Leica format (*.msh) VRML 2 (*.wrl / *.vml / *.iv) MDL (*.mdl) Object File Format (*.off)	STL (*.stl) Binary PBI (*.pbi) DXF 3Dface uniquement (*.dxf) Ascii POLY (*.poly) OBJ (*.obj) Ascii Leica (*.msh) Minolta Vivid (*.cdm) MDL (*.mdl) VRML (*.wrl / *.vml / *.iv) PLY (*.ply) OFF (*.off)
Contour / Section	IGES DXF polyligne ASCII	IGES DXF polyligne Binary MLI (*.mli) Coreview (*.edg / *.xml)
Modèle CAO	IGES STEP	IGES STEP

Figure 18: Formats d'import et d'export pris en charge par 3DReshaper. [ @Technodigit, 2014 ]

### ➤ Rhinocéros3D

Rhinocéros est un modèleur 3D s'adaptant aussi bien à des projets de construction qu'à des projets de design comme l'automobile par exemple.

Rhinocéros se base sur l'outil NURBS pour la modélisation de courbes 3D ce qui lui permet de représenter avec précision des objets géométriques simples (lignes, cercles, ellipses...) et des géométries complexes de forme libre comme les parties du corps humain.

Ce logiciel est capable de travailler avec des nuages de points conséquents [ @rhino3d, 2014 ]. (Fit Conseil utilise déjà Rhinocéros pour la texturation de données acquises par LiDAR aéroporté, il sera donc intéressant de l'utiliser pour le plaquage d'images terrestres.)

### ➤ Meshlab

Meshlab est un logiciel libre (opensource) de traitement de maillage 3D développé depuis 2005 par l'ISTI et le CNR dans le but de pouvoir manipuler et éditer des modèles 3D conséquents (provenant de scans 3D). Il permet également d'ajouter une texture sur les maillages.

### ➤ Sketchup

Sketchup est un logiciel de modélisation 3D, d'animation et de cartographie orienté vers l'architecture qui a été racheté en 2012 par la société Trimble. Ce logiciel se caractérise par des outils simples (rotation, extrusion, déplacement...) qui le rend différent des modèleurs 3D classiques.

Il existe deux versions de Sketchup, une version gratuite nommée « SketchUp Make » et la version professionnelle intitulée « SketchUpPro » disposant de modules d'animation et d'un plus large choix de formats d'export 3D.

L'un des points forts de Sketchup est sa facilité d'utilisation, notamment pour des travaux de création sur des volumes et la présence de nombreuses extensions (plugin) pouvant être téléchargées librement sur internet ou écrites en langage « ruby » apportant des fonctions supplémentaires au logiciel. De plus la bibliothèque 3D Warehouse (banque d'images 3D) permet d'importer de nombreux symboles 3D dans un projet. SketchUp est également compatible avec Google Earth permettant de visualiser le modèle créé dans son environnement et son voisinage. Jusqu'en 2013, Sketchup disposait d'un plugin (Pointools) lui permettant de traiter des nuages de points mais depuis sa disparition il n'est malheureusement plus possible de modéliser à partir d'un nuage de points.

➤ 3D Studio Max

3D Studio Max est un logiciel de modélisation et d'animation 3D qui est développé par la société Autodesk pour des applications professionnelles très perfectionnées. C'est un des logiciels de référence pour les infographistes et les concepteurs de jeux vidéo et de films.

Ce logiciel permet l'obtention d'un modèle surfacique et de formes maillées ainsi que l'attribution et l'édition des textures.

Il faut préciser que 3D Studio Max n'est pas compatible avec le format CityGML car LandExplorer, un autre logiciel d'autodesk le permet déjà.

➤ Blender

Blender est un des seuls logiciels de modélisation, d'animation et de rendu 3D en accès libre (open source). Il a été réalisé pour être intuitif et permet d'ajouter des effets et du réalisme à la scène.

L'un des avantages de Blender est de disposer d'une importante base de plugins lui permettant d'exporter dans de nombreux formats.

### **I.3 Les SIG et la 3D**

Les modèles urbains générés par l'acquisition de données cartographiques ont pour but d'être intégrés dans des SIG par la suite, il est donc normal de se pencher sur les différents formats 3D existants et les différentes réalisations de villes numériques.

#### **I.3.1 Les formats 3D**

➤ X3D

Extensible 3D (X3D) a été créé par le consortium Web3D pour succéder au format VRML97 en 2005. Il appuie son vocabulaire sur un langage de type XML lui permettant une meilleure interactivité avec les pages web.

Le X3D peut être exprimé en deux langages, le VRML et le XML. Il a été étudié pour la définition de contenus interactifs en 3D et leur communication en temps réel. Il peut donc par exemple être utilisé dans les SIG, la CAO ou encore les simulations visuelles.

➤ LandXML

LandXML est un format de fichier XML ouvert et public. C'est un format de fichier standard pour le partage de données de conception d'ingénierie civile et de mesure de relevés topographiques.

LandXML est pris en charge par un grand nombre d'applications des principaux éditeurs de logiciels de conception comme par exemple Autodesk avec Autocad civil 3D.

LandXML supporte les coordonnées en trois dimensions mais ne comprend pas les géométries volumétriques.

➤ **Objet3D**

OBJ est un format de fichier contenant la description d'une géométrie 3D qui a été définie par la société Wavefront Technologie. Ce format de fichier est ouvert et a été adopté par d'autres logiciels 3D (Maya, Blender, Meshlab, 3d Studio Max...).

➤ **PDF3D**

PDF3D est un format qui a été créé par la société Adobe dans le but de convertir des données 3D en documents interactifs PDF3D permettant une navigation libre 3D temps réel [PDF3D, 2014]. L'avantage de ce type de format réside dans la légèreté du poids des fichiers et de sa compatibilité avec Acrobat Reader (gratuit), ce qui peut facilement être lu par les utilisateurs. De plus les fichiers « .pdf » disposent d'une certaine protection d'échange car ils ne peuvent pas être modifiés par l'utilisateur.

➤ **COLLADA**

COLLADA définit un standard de schéma XML ouvert qui a été développé par Sony en 2004. Il s'agit d'un format d'échange pour les applications 3D interactives [COLLADA, 2014]. C'est le format qui a été choisi par Google pour l'importation de modèles 3D dans Google Earth.

➤ **CityGML**

Le format CityGML développé depuis 2002 par les membres du groupe « SIG 3D » dans le cadre de l'initiative allemande « Geodata Infrastructure North-Rhine Westphalia » est le standard officiel de l'OGC depuis 2008. Il s'agit d'un modèle de données ouvert reprenant la base du langage GML3 et permettant un échange de modèles 3D de villes.

CityGML ne représente pas seulement l'aspect graphique du modèle urbain, il prend également en compte la sémantique ce qui lui permet d'avoir de multiples informations sur les différents éléments du modèle (MNT, bâtiments, rivières, chemins de fer, mobilier urbain, réseaux de transport, végétation...).

Afin de permettre différents niveaux de modélisation, la visualisation du modèle différencie cinq niveaux consécutifs de détails (LOD), où les objets deviennent plus détaillés avec l'augmentation de LOD concernant la géométrie et la différenciation thématique.

Les différents niveaux de détail sont présentés dans le tableau suivant :

	LOD 0	LOD 1	LOD 2	LOD 3	LOD 4
Échelle du Modèle	Région	La ville	Quartier - Projets	Immeuble extérieur	Immeuble intérieur
Classes de précision	La plus faible	Faible	Moyenne	Élevée	Très Élevée
Précision absolue des point 3D (Position/Élévation)	Plus faible que LOD1	5/5m	2/2m	0.5/0.5m	0.2/0.2m
Structure des toits	Aucun	Plat	Type de toit	Représentation réelle	Représentation réelle
Mobilier urbain	Aucun	Objets importants	Prototypes	Représentation réelle	Représentation réelle
Végétation seule	Aucun	Objets importants	Prototypes, Élévation >6m	Prototypes, Élévation >2m	Prototypes, Représentation réelle
Couverture végétale	Aucun	50m x 50m	5m x 5m	<LOD2	<LOD2

Figure 19:CityGML: Tableau comparatif des cinq niveaux de détail. [Gröger et al., 2006]

### I.3.2 Les modélisations 3D urbaines

Comme nous avons pu le voir dans la norme CityGML, les modélisations 3D urbaines peuvent présenter différents niveaux de détails.

➤ Modèles virtuels 3D proposés par Ubick et Google Earth

Les modèles sont constitués à partir de prises de vues aériennes réalisées par cinq caméras dont quatre obliques. Les traitements s'appuient sur le savoir-faire d'InterAtlas et d'Acute3D (reconstitution 3D automatique avec Smart3Dcapture).



Figure 20: Visualisation de la ville de Paris à partir d'Ubick. [[@ubick, 2014](#)]

Comme nous pouvons le remarquer sur la Figure 20, cette modélisation a des limites. En effet, les arbres ne sont pas représentés correctement et les traits des façades, bien qu'ils laissent apparaître la troisième dimension, ne sont pas très nets. Nous pouvons parler alors de maquette 3D « classique » car la navigation s'effectue en survol à une altitude d'environ 400 mètres.

➤ Les maquettes 3D de type piéton

La maquette 3D de type piéton va permettre à l'utilisateur de s'immerger entièrement dans le modèle et donc d'en percevoir plus de détails. Ainsi, dans la maquette 3D de la principauté de Monaco présenté par le stand IGO lors du salon Imagina en février 2012, la modélisation laisse apparaître les balcons et les avancés des façades. Le mobilier urbain est aussi représenté de façon très réaliste (bornes, lampadaires, plots). Cette maquette a été réalisée à partir de photographies terrestres et aériennes de grande précision (la résolution spatiale des orthophotographies vraies est de 4.5 centimètres).



Figure 21:Maquette 3D de la principauté de Monaco. [[@gouv.mc](#)]

Ce type de modélisation est représentatif du niveau LOD3 présenté dans le format CityGML et le résultat de mon travail de fin d'étude tentera de s'en approcher.

## II Segmentation et classification

*Afin d'automatiser au maximum la modélisation des différents objets présents dans le nuage de points, il est nécessaire d'optimiser le processus de segmentation et de classification.*

*Dans cette partie, nous allons proposer différentes méthodes permettant d'extraire automatiquement les caractéristiques urbaines présentes dans un nuage de point acquis à partir du Streepmapper de 3DLM.*

### II.1 Le sol

#### II.1.1 Méthode

Pour le traitement du nuage de points, Fit Conseil utilise le logiciel Terrascan.

Terrascan est une application de la famille des logiciels Terrasolid permettant la gestion et le traitement de nuages de points LiDAR. Les formats d'import et d'export pris en charge par Terrascan sont les suivants : LAS, TerraScan binaire et les formats ASCII.

Ce logiciel est muni de différentes routines de classification permettant le filtrage semi-automatique de nuages de points à partir de certaines caractéristiques comme la couleur, l'intensité, la réflectance, l'élévation, l'angularité ou encore la planéité par exemple.

Ces différentes routines peuvent être combinées dans des macros permettant d'automatiser au maximum le processus de classification. Chaque ligne présente dans la macro correspond à une étape de traitement effectuée sur les points laser.

L'entreprise utilise actuellement ces différentes étapes de traitements pour classifier des données acquises par lasergrammétrie aérienne. Il est évident que les données laser acquises à partir du MMS ne présentent pas le même niveau de détail et de densité que celles-ci ce qui n'engendrera pas le même nombre de classe souhaité et les mêmes spécifications dans les critères de classification.

Pour la classification du sol, nous allons utiliser la routine « ground » présente dans le logiciel Terrascan.

Cette routine classe les points au sol à partir de la construction itérative d'un modèle de surface triangulée. La routine commence par construire un modèle initial en sélectionnant les points bas. Les triangles de ce modèle initial sont la plupart du temps en dessous du sol avec seulement les sommets qui touchent le sol. La routine va ensuite mouler le modèle vers le haut en ajoutant itérativement de nouveaux points qui vont faire suivre le modèle au plus près de la surface du sol.

Le paramètre « Max building size » permet de contrôler la sélection de points de départ en renseignant le paramètre maximum de la taille du bâtiment pour la recherche d'au moins un point sol.

Les paramètres d'itération vont permettre d'accepter ou non un point sur le modèle en fonction de sa proximité par rapport au plan du triangle.

Le paramètre « Terrain angle » renseigne la plus grande pente permise sur le terrain.

Le paramètre « Iteration angle » représente l'angle maximum entre un point, sa projection sur le plan du triangle et le sommet du triangle le plus proche.

Le paramètre « Iteration distance » permet de s'assurer que l'itération ne fait pas de grands sauts vers le haut lorsque la taille des triangles devient conséquente.



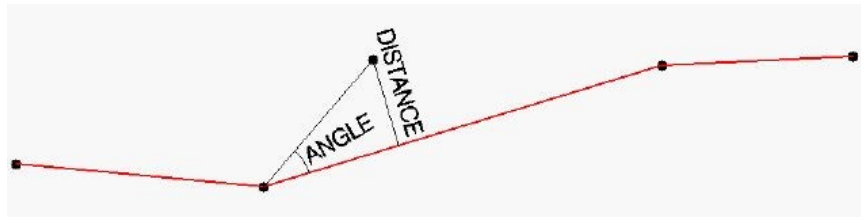


Figure 22: Schéma représentant les paramètres angle et distance de la routine « ground ». [Terrasolid, 2014]

Avant de lancer la routine « ground », il est indispensable d'identifier et de classer les points bas qui représentent des points faux situés au-dessous du sol et qui sont la plupart du temps dus à des réflexions spéculaires au niveau des vitres ou du sol mouillé. Si quelques points bas ne sont pas détectés ils seront alors pris en compte dans le modèle de départ du processus d'itération et la classification du sol s'en trouvera altérée. Pour ce faire, il est nécessaire de lancer l'étape de traitement « low point » plusieurs fois d'affilée dans la macro pour un ensemble de points et pour des points individuels. Cette étape va permettre d'identifier et de classer un point ou un ensemble de points isolé à partir d'un critère de distance et d'un voisinage que l'on doit renseigner.

Une fois les points bas identifiés, la routine « ground » peut être définie. Après plusieurs tests, il s'avère que les différents paramètres de cette routine ne nous permettent pas d'isoler les points appartenant à la route des points appartenant au trottoir. En effet, les différences de hauteur, largeur ainsi que les bateaux au niveau des trottoirs rendent complexe la création d'une classification correcte généralisée. Les trottoirs seront donc pris en compte dans la classe « sol » et il sera expliqué à la fin de ce chapitre « méthode » une façon simple et efficace de les classer manuellement.

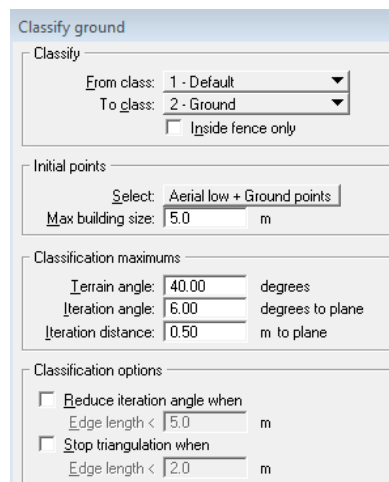


Figure 23: Interface de la routine "ground". [Terrascan]

Le paramètre « Max building size » est fixé à 5 mètres afin de limiter le rayon de recherche. Afin de prendre en compte les trottoirs dans la classe « sol » l'angle du terrain est renseigné avec une valeur de 40 degrés, le paramètre « Iteration angle » par une valeur de 6 degrés et le paramètre « Itération distance » par une valeur de 50 centimètres. Ces paramètres vont permettre de prendre en compte dans la classe « sol » tous les trottoirs quel que soit leur hauteur et leur largeur tout en préservant la jonction entre le sol et les façades comme l'indique la Figure 24.

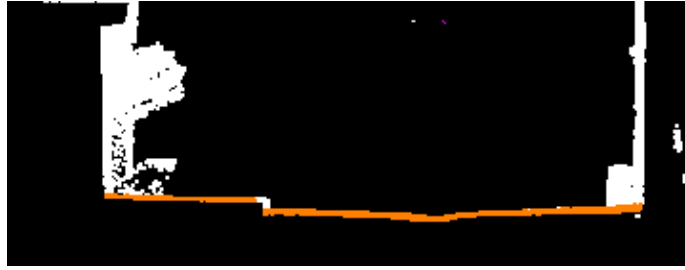


Figure 24: Vue en coupe du nuage de point. Les points de couleur marron représentent les points appartenant à la classe "sol" et les blancs décrivent les points non classés. [Terrascan]

Les points appartenant au sol sont affectés d'un certain bruit de mesure au moment de l'acquisition, ce qui fait que la couche sol est constituée d'une certaine épaisseur (confère figure 24). Cette épaisseur peut osciller entre 1 et 4 centimètres selon que le revêtement soit sec ou humide. Si le sol est humide, un phénomène de réflexion va augmenter le bruit de la mesure. Dans un cas général nous avons choisi de joindre à la classe « sol » tous les points de la classe « autre » situés 1 centimètre en dessous et 2 centimètres en dessus de celle-ci par le biais de la routine « by height for ground ».

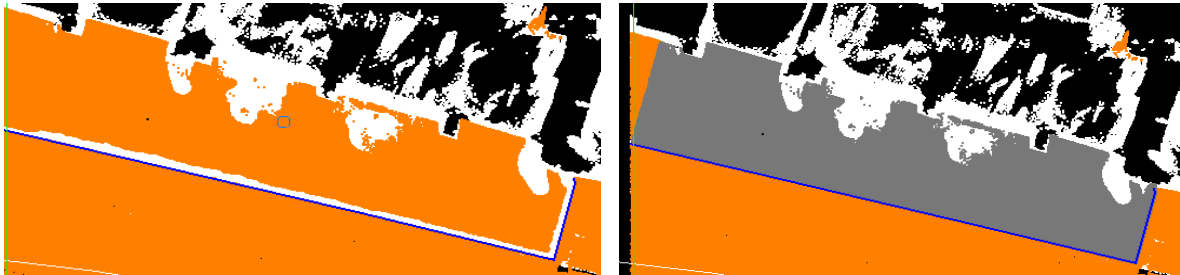


Figure 25: Classification d'un trottoir. Digitalisation du contour extérieur du trottoir à partir d'une polyligne (gauche) et résultat obtenu après l'étape de classification automatique (droite). [Terrascan]

Si l'on se positionne en vue de dessus la bordure de trottoir est facilement repérable car elle représente une ligne blanche au milieu des points appartenant au sol. Il est donc possible de tracer une polyligne sur le contour extérieur de la bordure puis de lancer la routine « by centerline » permettant de classer tous les points de la classe « sol » ainsi que les points de la classe « autre » situés à une certaine distance et à un certain côté de la polyligne dans la classe « trottoir ». Afin de pouvoir appliquer cette fonction de manière générale il faut séparer les polygones créés en fonction de leur appartenance à un côté de la rue (côté gauche ou côté droit) en les plaçant dans deux niveaux (calques) différents. La classification pourra par la suite être réalisée de façon automatique en intégrant cette étape de traitement dans la macro présentée en [Annexe 1](#).

Même si cette classification est réalisée de façon semi-automatique, elle se révèle rapide et efficace.

## II.1.2 Evaluation de la classification

Afin d'évaluer quantitativement la classification, il est nécessaire de comparer la classification semi-automatique avec une classification manuelle. Une classification manuelle permettant de distinguer les points appartenant à la classe « sol » des points appartenant à la classe « autre » est donc réalisée sur plusieurs échantillons de la ville de Rennes. La comparaison va s'effectuer sur environ 10 millions de points au total.

Une matrice de confusion calculée en pourcentage est alors construite :

Echantillon test	Vérité terrain (%)	
	autre	sol
autre	97,68	0,23
sol	2,32	99,77
total	100	100

Tableau 1: Matrice de confusion

Nous pouvons également extraire l'indice de précision globale de cette matrice :

$$\text{Précision globale} = \frac{T_{BC}}{T_C} * 100 = 98\%$$

Avec  $T_{BC}$  = nombre de points bien classés et  $T_C$  = nombre de points classés

L'analyse des valeurs de la matrice de confusion et de la précision globale nous permet d'affirmer que la classification est correcte même si un certain nombre de points de l'échantillon test « sol » (2,32%) sont mal classés.

En évaluant visuellement le résultat (évaluation qualitative), nous pouvons voir que certains points représentant l'intérieur des bâtiments sont classés dans la classe « sol », cependant l'ensemble de la classification est satisfaisante, une très grande partie des points bas a donc bien été identifiée.



Figure 26: Nuage de points non classifié en vue de dessus (gauche). Nuage de points classifié en vue de dessus (milieu) et vue de profil (droite) avec les points appartenant à la classe sol représentés par la couleur marron. [Terrascan]

Après avoir réalisé la classification semi-automatique il est nécessaire d'effectuer un contrôle visuel et de corriger les erreurs de classification éventuelles. Pour cela nous réalisons des coupes transversales ou nous dessinons aussi un polygone à partir de la vue de dessus et utilisons la routine « Inside shape » permettant de classifier tous les points à l'intérieur (ou à l'extérieur) de ce polygone ce qui peut être extrêmement rapide.

Une fois tous les points du sol bien classifiés, une autre classification peut être réalisée sur la classe « autre ».

## II.2 Les façades

### II.2.1 Méthode

Afin de classer au mieux l'ensemble des points appartenant aux façades il va falloir arriver à différencier les points d'intérêts des points indésirables. Les points indésirables vont représenter tous les points qui peuvent appartenir à des objets urbains (arbres, lampadaires), des éléments mobiles (voitures, piétons...) ou qui vont décrire l'intérieur des bâtiments en pénétrant à partir d'ouvertures transparentes tel que les baies vitrées et les fenêtres. Les points d'intérêts vont représenter tous les points appartenant à la façade y compris les balcons.

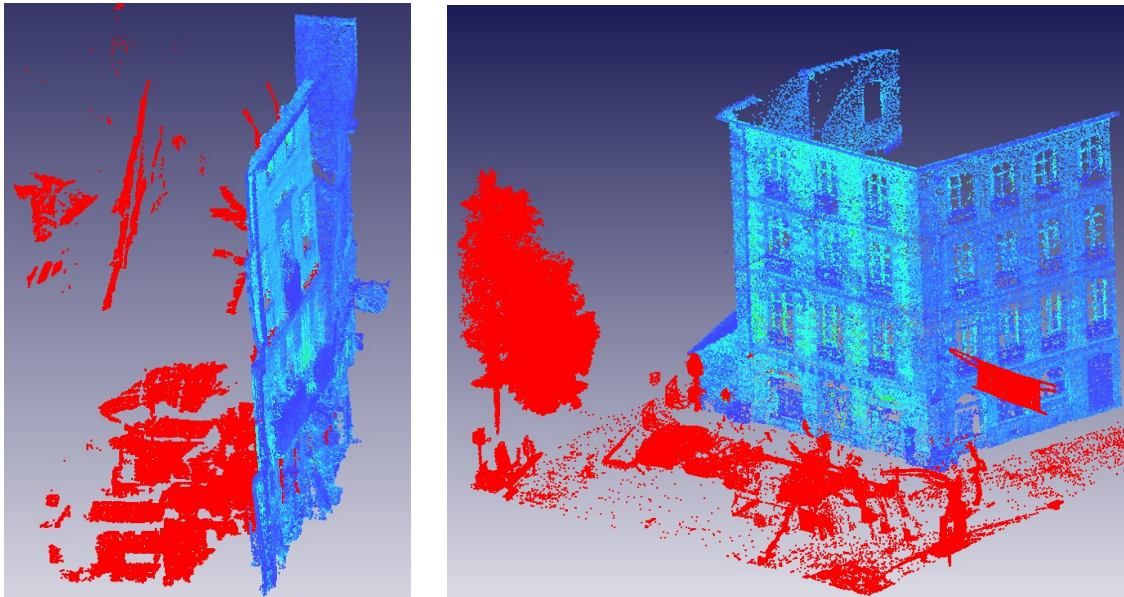


Figure 27: Visualisation des points indésirables (rouge) à partir de deux vues en perspective. [3DReshaper]

Pour ce type de classification [Manandhar et Shibasaki., 2001] se base sur le principe suivant : les points appartenant aux façades des bâtiments sont alignés verticalement et présentent une certaine densité du fait de leur construction par l'homme.

Une première solution consiste à calculer le vecteur normal sur l'ensemble du nuage de points à l'aide de l'action « compute normal vectors » dans le logiciel Terrascan puis à lancer une classification « by normal vector » en ne retenant dans la classe « façade » que les points ayant une dimension plane, mais des façades pas forcément planes et la planéité de la plupart des points indésirables ne rendent cette méthode que peu efficace.

Une deuxième solution basée sur la densité des points est donc étudiée. Il est possible sur Terrascan d'exporter un fichier grille avec des distances uniformes entre les points et une valeur de densité associée à chaque cellule de la grille via la commande « Export lattice model ». Après avoir déterminé un espacement grille de 5 centimètres et la valeur de densité moyenne permettant de distinguer au mieux points d'intérêts et points indésirables nous pouvons nous rendre compte que le résultat est correct lorsqu'il s'agit de façades simples sans balcon mais s'avère beaucoup moins bon lorsque l'on a affaire à des façades constituées de différents plans et balcons. Le temps passé à corriger la classification de ces dernières est conséquent, il est donc nécessaire de proposer une autre méthode basée sur des critères différents.

Après réflexion, nous constatons qu'une grande partie des points indésirables présente une discontinuité de quelques centimètres ou de quelques mètres vis-à-vis de la façade. Nous allons

donc utiliser le logiciel 3DReshaper et sa fonction de segmentation automatique à partir d'un critère de distance (présente dans l'outil « Nuages Filtrer/Exploser Nuage(s)»)

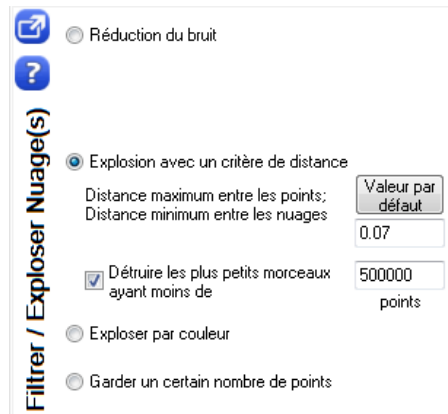


Figure 28: Fenêtre de l'outil "Filtrer/Exploser Nuage(s)" dans 3DReshaper

Cette fonction, avec l'option « Explosion avec un critère de distance » permet de détruire les morceaux ayant moins d'un certain nombre de points. Monter cette valeur à 500000 points permet de supprimer au maximum les éléments mobiles (piétons, voitures), le mobilier urbain et la végétation présente dans le nuage de point initial.

Pour déterminer au mieux le critère de distance nous allons commencer par calculer la densité des points situés sur le haut des façades de la ville de Rennes de façon à ce que les parties les moins denses des façades ne soient pas supprimées. Pour ce faire, le calcul est réalisé à partir de valeurs extrêmes en prenant une hauteur de façade de 15 mètres et une distance de 5 mètres entre les scanners laser et la façade. Il faut également prendre en compte le fait que les scanners laser se situent à une hauteur de 3 mètres au-dessus du sol et que leur fréquence d'acquisition est de 380000 points par seconde représentés par 200 profils. Un calcul est réalisé à partir de ces paramètres et pour des vitesses comprises entre 10 et 70 Km/h.

Vitesse (km/h)	Distance laser-façade (m)	Hauteur façade (m)	Distance latérale entre 2 points (cm)	Distance longitudinale entre 2 points (cm)
10	5	15	1.4	11.1
30	5	15	4.2	11.1
50	5	15	6.9	11.1
70	5	15	9.7	11.1

Tableau 2: Variation de la distance latérale et longitudinale entre deux points en fonction de la vitesse pour un scanner laser

À partir du Tableau 2 nous pouvons constater que la distance latérale augmente proportionnellement par rapport à la vitesse (coefficient multiplicatif égal à 0.14) mais que celle-ci ne dépasse pas la distance longitudinale entre 10 et 70 km/h. C'est donc la distance longitudinale que nous choisissons.

Le deuxième scanner situé de l'autre côté du véhicule ne permettra pas de diviser la distance prise en compte par deux mais permettra d'augmenter la densité d'environ un tiers.

La prise en compte des deux scanners laser permettra d'obtenir une distance longitudinale minimum entre deux points d'environ 7,4 centimètres.

Après plusieurs tests effectués sur différentes zones de la ville de Rennes, nous pouvons constater que la segmentation du nuage de points avec un critère de distance de 7 centimètres donne un résultat convaincant.

## II.2.2 Évaluation de la segmentation

Visuellement, nous vérifions que la segmentation permet d'éliminer un maximum d'éléments indésirables de part et d'autre de la façade même si un certain nombre de points subsistent principalement à l'arrière de celle-ci.

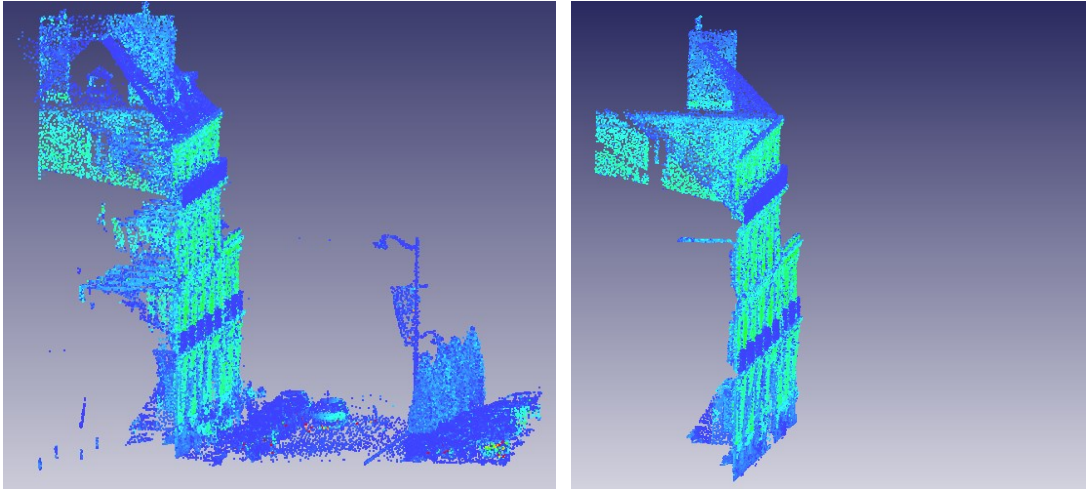


Figure 29: Nuage de points initial (gauche) et nuage de points segmenté (droite). [3DReshaper]

Afin d'évaluer quantitativement la segmentation, celle-ci est comparée avec une segmentation manuelle réalisée sur plusieurs endroits de la ville de Rennes.

Une matrice de confusion calculée en pourcentage est alors construite :

Échantillon test	Vérité terrain (%)	
	autre	façade
autre	65	0
façade	35	100
total	100	100

Tableau 3: Matrice de confusion

Nous pouvons également extraire l'indice de précision globale de cette matrice :

$$OA = \frac{T_{BC}}{T_c} * 100 = 89\%$$

Au vue de ces résultats, nous pouvons constater que cette méthode ne permet pas d'éliminer tous les points « indésirables » car un nombre important de points de la façade se retrouvent finalement dans la classe « autre ». Cette valeur pourrait être réduite en diminuant le critère de distance choisi mais l'on perdrait à ce moment-là des points de façade, ce n'est donc pas judicieux car il est plus facile de supprimer des points que d'en ajouter. La suppression des points restant devra donc être réalisée manuellement.

## II.3 Les objets urbains

### II.3.1 Méthode

Après avoir classifié le sol et les façades des bâtiments, nous devons maintenant nous pencher sur la classification des différents objets urbains tels que les arbres et le mobilier urbain (panneaux de signalisation, lampadaires...). Après plusieurs tests, nous nous rendons compte que le logiciel Terrascan n'est pas adapté à ce type de classification plutôt complexe en termes de forme, couleur et densité.

Comme pour la segmentation des façades, nous allons utiliser l'outil « Nuages Filtrer/Exploser Nuage(s) » du logiciel 3DReshaper. Les morceaux comportant moins de 2000 points seront détruits de façon à nettoyer le nuage. La distance à prendre en compte pour avoir une segmentation réussie est de 15 centimètres. En dessous de cette distance, on remarque que certains objets vont être segmentés en différentes parties ce qui n'est pas le but escompté.

Afin de segmenter au mieux les différents objets urbains, il est nécessaire d'importer uniquement dans 3DReshaper tous les points situés à 20 centimètres au-dessus du sol (route + trottoir) de façon à couper la liaison sol-objet comme nous pouvons le voir sur la Figure 30. Cette discontinuité de 20 centimètres n'aura pas de conséquence sur la modélisation qui sera effectuée par la suite.

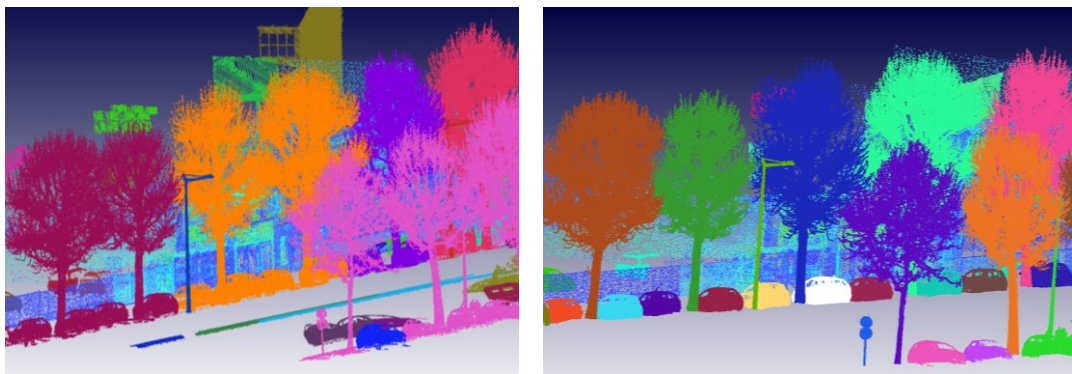


Figure 30: Aperçu de la segmentation réalisée sur un ensemble de points excepté la classe « sol » (à gauche) et à 20 centimètres au-dessus de la classe « sol » (à droite). [3DReshaper]

Après avoir réalisé la segmentation automatique, chaque objet est associé individuellement à un nuage de points. Les différents nuages appartenant à la même classe peuvent ensuite être fusionnés manuellement à partir de l'outil « Nuages → Fusionner Nuage(s) ». Il est également possible de nettoyer le nuage de points en supprimant tous les éléments mobiles présents au moment de l'acquisition comme les voitures ou les piétons.

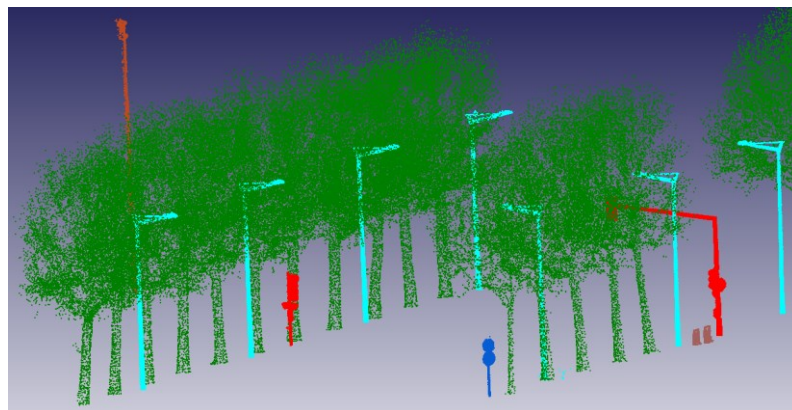


Figure 31: Nuage de point nettoyé et classifié. [3DReshaper]

### **II.3.2 Evaluation de la segmentation**

Dans la majorité des cas, la segmentation automatique par critère de distance permet de distinguer chaque objet les uns des autres. Il peut arriver que deux objets soient collés et qu'il ne soit pas possible de les séparer automatiquement comme nous pouvons l'observer sur la Figure 32.



Figure 32: Limite de la segmentation automatique. [3DReshaper]

Sur cette capture, un piéton s'est intercalé entre un lampadaire et des voitures au moment de l'acquisition ce qui n'assure pas une discontinuité de 15 centimètres entre les objets et entraîne par conséquent une erreur de segmentation. Il faudra donc dans ce type de cas dissocier les différents objets manuellement.

## **II.4 Conclusion**

Dans cette première partie, plusieurs méthodes de segmentation et de classification semi-automatiques ont été proposées.

Nous avons pu constater que le logiciel Terrascan permet une classification efficace lorsqu'il s'agit d'objets relativement plans (sol, façades simples) mais montre rapidement ses limites lorsque l'on cherche à classifier des objets complexes (mobiliers urbains, arbres, façades non planes) là où la segmentation automatique par critère de distance proposée par le logiciel 3DReshaper s'avère être la meilleure solution. Une fois cette étape de traitement réalisée, nous pouvons travailler sur la modélisation.



## III Génération d'un modèle 3D

*Dans une première partie deux types de modélisation seront mises en œuvre et comparées sur une façade de la ville de Rennes. Dans une deuxième partie le type de modélisation choisie sera appliqué sur un échantillon test (ensemble de façades) et il sera proposé ensuite une méthode d'automatisation concernant la détection et la modélisation de contours. Enfin une démarche permettant de modéliser les objets urbains sera présentée.*

### III.1 Modélisation d'une façade

#### III.1.1 Modélisation par maillage

Après avoir segmenté la façade en question, il est nécessaire d'effectuer une réduction du bruit de façon à éliminer un maximum de points isolés et de points faux qui pourraient venir perturber le maillage. Pour ce faire nous utilisons le filtre « Réduction du bruit » présent dans l'outil « Nuages/Filtrer/Exploser Nuage(s) » de 3DReshaper qui permet de supprimer les points épars dans le nuage en ajustant une certaine valeur d'intensité.

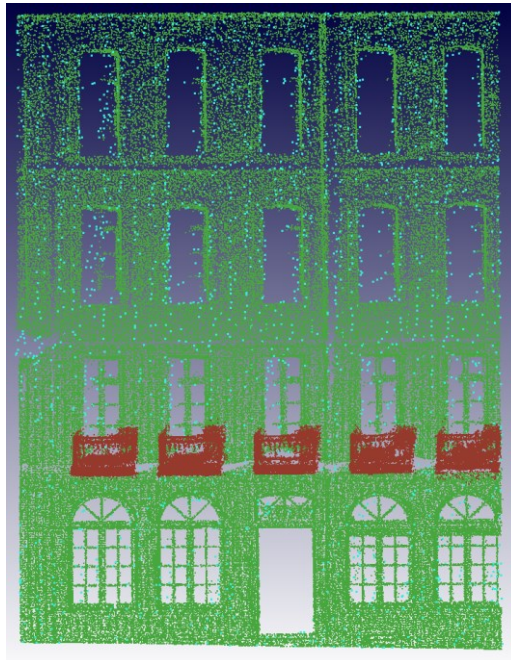


Figure 33: Réduction du bruit (points de couleur bleu turquoise) pour une valeur d'intensité égale à 20. [3DReshaper]

Il est utile de commencer à réaliser un maillage avec les paramètres par défaut de façon à pouvoir observer ce qui se passe. Nous pouvons remarquer qu'un pas d'échantillonnage trop petit va entraîner un nombre de trous important et ne permettra pas de représenter entièrement la façade. Pour créer une surface sans trous, nous allons choisir de créer un maillage avec un échantillonnage régulier de 20 centimètres.

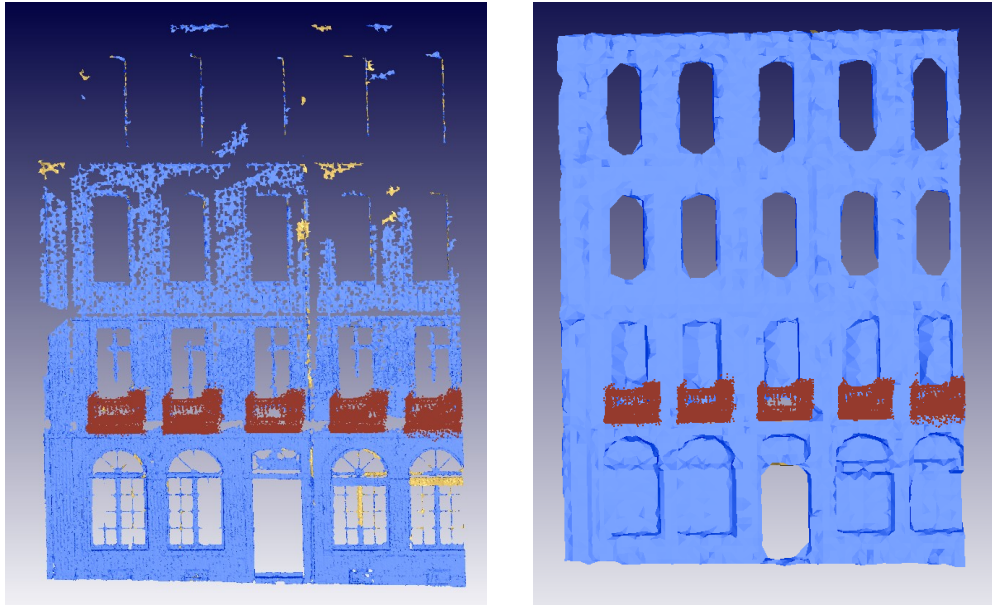


Figure 34: Maillage réalisé avec un échantillonnage régulier de 3 centimètres (gauche) et de 20 centimètres (droite). [3DReshaper]

Le maillage peut maintenant être affiné par le biais de l'outil « Affinage à erreur de corde » et en choisissant la méthode de maillage adaptée. Les deux méthodes de maillage proposées sont les suivantes : « prendre les points du nuage » et « interpoler de nouveaux points ». La première méthode propose de créer un maillage à partir des points du nuage et d'une erreur cordale (erreur de déviation) qu'il faut renseigner tandis que la seconde propose d'exercer un raffinement à erreur de corde en interpolant de nouveaux points qui vont se baser sur la meilleure forme du maillage à créer.

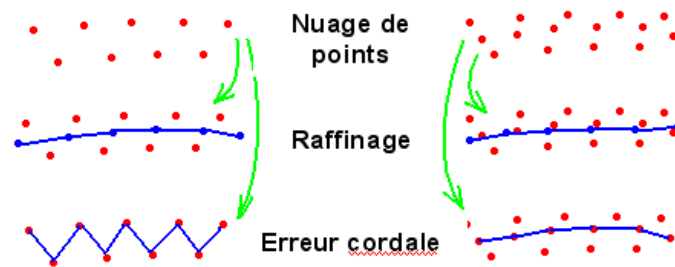


Figure 35: Différence entre "raffinage" et "erreur cordale". [@Technodigit, 2014]

Après différents tests, nous constatons qu'il est difficile d'obtenir un résultat correct à partir des points du nuage car le nuage est bruité (deux passages ont été réalisés au moment de l'acquisition) et dense.

C'est donc la méthode de raffinement qui est retenue et qui donne le meilleur résultat pour une valeur choisie de 3 millimètres. Une fois le maillage affiné, il est nécessaire de le parfaire en éliminant les pics, en bouchant les trous et en effectuant un lissage à certains endroits. Certains contours mal représentés, notamment sur la partie haute de la façade doivent également être retravaillés et redessinés. Un maillage sous contrainte permettra par la suite d'affiner le modèle au niveau de ces contours.

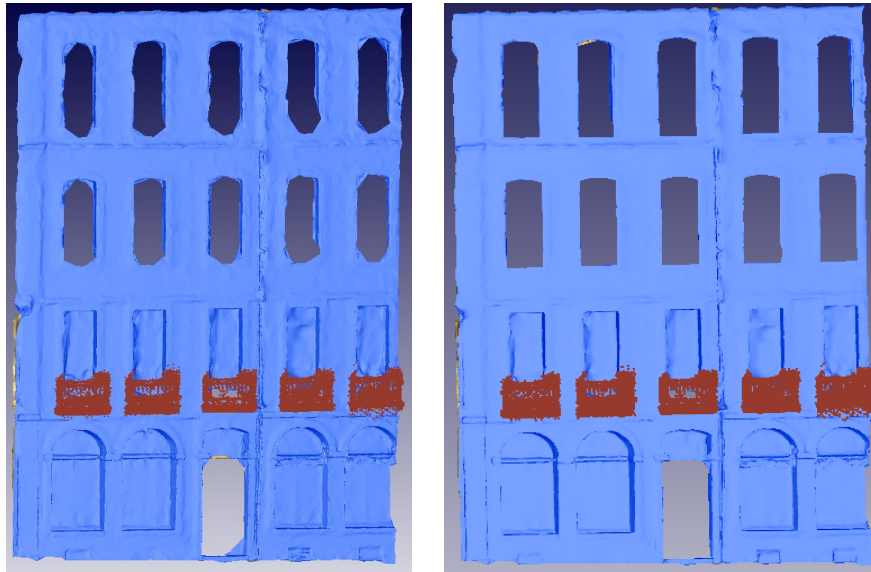


Figure 36: Maillage obtenu après interpolation (gauche) et maillage retravaillé (droite). [3DReshaper]

Compte tenu de leur complexité, le maillage des balcons est réalisé à part. Après plusieurs tests, nous nous apercevons qu'il n'est pas possible de réaliser un maillage précis permettant de distinguer le détail des balcons (barreaux en fer forgé) à cause du bruit présent dans le nuage de points et du manque d'information (trop peu de points au niveau des détails). Nous allons donc suivre la même méthode que précédemment nous permettant d'obtenir au final une modélisation simplifiée des balcons. C'est donc la texture qui devra amener le détail.

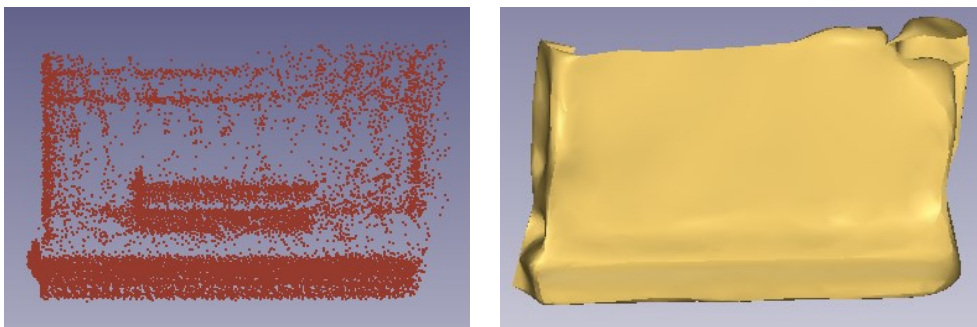


Figure 37: Nuage de points représentant un balcon (gauche) et maillage obtenu (droite). [3DReshaper]

Une fois le maillage de la façade terminé, nous obtenons un fichier dont la taille est 26766 KO, ce qui est conséquent pour la représentation d'une unique façade.

### III.1.2 Modélisation à partir de primitives géométriques

#### III.1.2.1 Création d'un plan

Comme précédemment, il est intéressant d'effectuer une réduction du bruit de manière à supprimer les points isolés et de nettoyer le nuage de points. Celui-ci peut également être nettoyé manuellement à partir de l'outil «Nuages → Nettoyer/Séparer Nuage(s)» présent dans 3DReshaper.

L'outil « Mesure → Meilleure Forme → Plan » va permettre de créer un plan moyen de la façade à partir du nuage de points. Il est possible de lancer un calcul automatique de la normale à ce plan ou de forcer la normale suivant les axes X, Y et Z. Partant du principe qu'un plan de façade n'est pas

forcément vertical, nous allons utiliser la fonction « calculer la normale ». Avant le lancement du calcul, il est possible d'écarter les points les moins bons (les plus éloignés) de sorte à ce qu'ils ne viennent pas fausser le calcul. Il n'est donc pas utile de dissocier les balcons du plan de façade avant le lancement du traitement car ceux-ci ne seront pas pris en compte dans le calcul du plan moyen.

Pour le projet de la ville de Rennes, deux passages ont été réalisés avec le MMS au niveau de chaque rue. La précision voulue par le cahier des clauses techniques particulières est de 5 à 10 centimètres en position XYZ et de 1 centimètre en position relative. Pour faire correspondre au mieux les deux passages effectués, un ajustement relatif a été réalisé montrant une cohérence entre eux à 3 centimètres près pour 95% du projet. Nous devons donc tenir compte au mieux de cette précision lors du calcul de la normale et ajuster le nombre de mauvais points à éliminer pendant le calcul pour obtenir un écart type aux alentours de 3 centimètres.

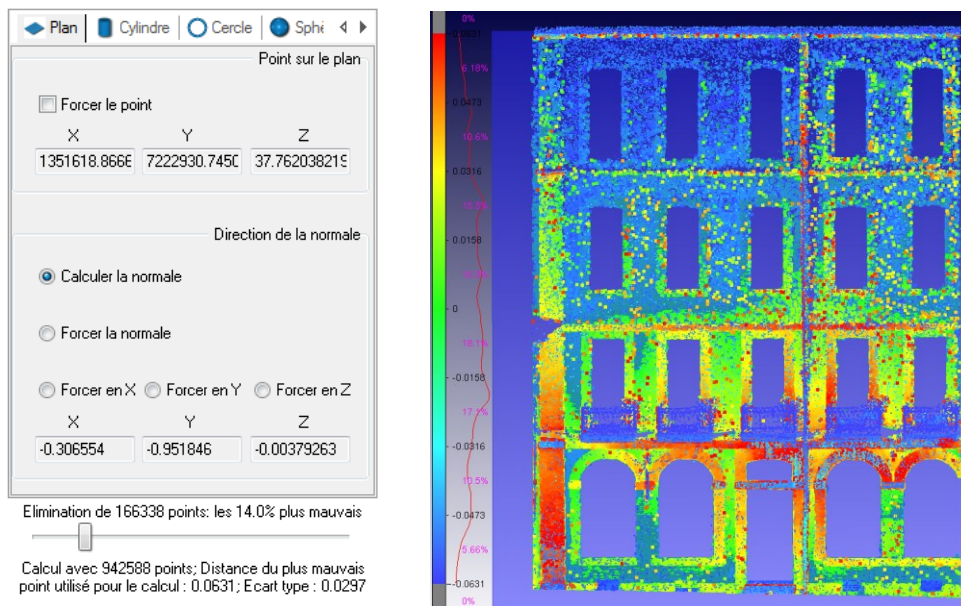


Figure 38: Calcul de la normale permettant d'extraire le meilleur plan à partir du nuage de points. [3DReshaper]

Après avoir créé le modèle plan, nous allons projeter le nuage de points sur celui-ci de façon à pouvoir par la suite digitaliser les différents contours des objets présents sur la façade. Pour ce faire nous utilisons l'outil « Création → Projection » permettant de projeter un nuage, un point, une ligne ou une polyligne sur une entité (plan, maillage ou surface).

### III.1.2.2 Digitalisation

Il est possible de modéliser manuellement par l'intermédiaire de polygones ou de formes (lignes, rectangles, cercles) le contour extérieur de la façade ainsi que les contours des portes et des fenêtres mais ce processus va s'avérer extrêmement long si l'on doit modéliser toutes les façades des bâtiments à l'échelle d'une ville.

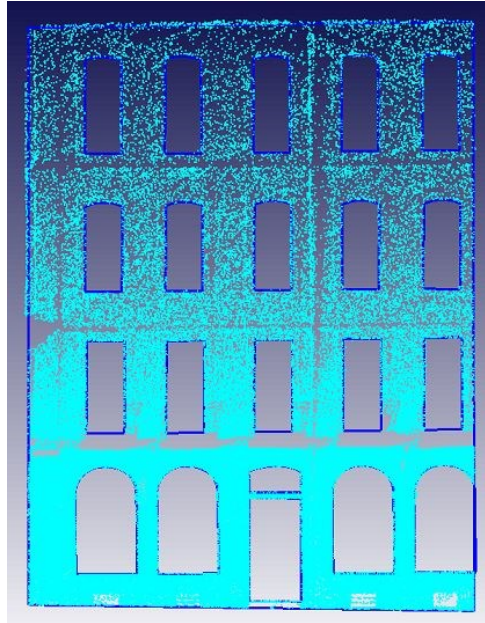


Figure 39: Digitalisation des contours sur le nuage de points projeté sur le plan créé. [3DReshaper]

Une fois les différents contours digitalisés, il est possible d'effectuer un maillage à partir d'un groupe de contours à l'aide de l'outil « Maillage sous contraintes », qui permet d'aboutir à un résultat net et précis.

### III.1.2.3 Modélisation 3D

Pour donner l'effet de profondeur au niveau des portes et des fenêtres, des extrusions de 20 centimètres sur les différents contours sont réalisées. Il s'agit là en effet d'obtenir un modèle grossier et simplifié de la façade, 20 centimètres étant en général la profondeur moyenne constatée au niveau des portes et des fenêtres.

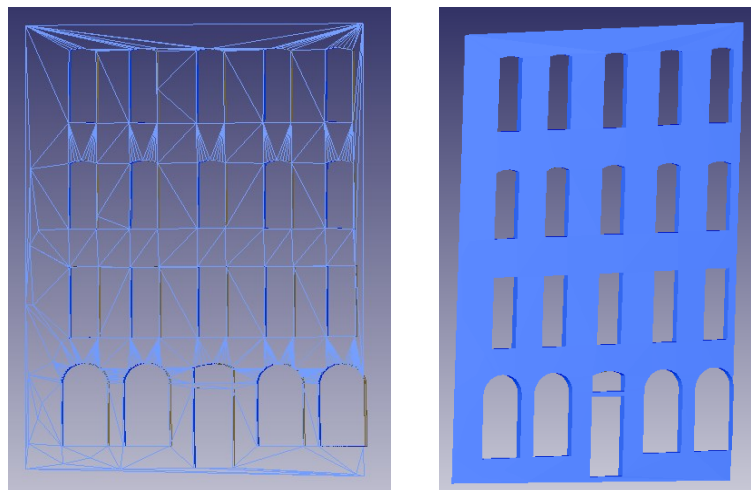


Figure 40: Modélisation de façade obtenue après avoir réalisé le maillage sous contrainte et extrudé les contours. Représentation filaire (gauche) et représentation flat (droite). [3DReshaper]

Par la suite, le maillage réalisé sera importé dans le logiciel SketchUpMake (open source) au format « dxf ». Il est important au moment de l'importation du fichier de choisir l'unité « mètre » dans les options et de cocher la case « fusionner les faces coplanaires ».

L'un des intérêts majeur du logiciel SketchUpMake réside dans le fait de pouvoir importer une multitude d'éléments 3D via la librairie 3D Warehouse. Nous allons donc importer un modèle de

balcon sous la forme d'une cellule 3D et le mettre à l'échelle avant de l'intégrer sur notre modèle de façade.

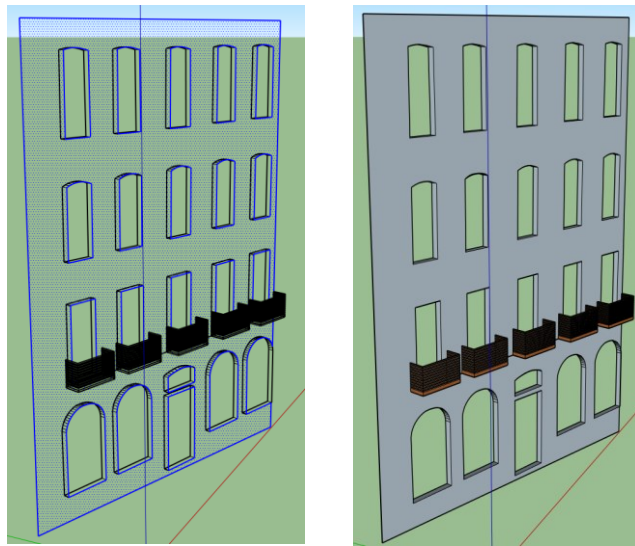


Figure 41: Modélisation d'une façade sous SketchUpMake. Représentation filaire (gauche) et représentation ombragée (droite)

Nous pouvons remarquer sur la représentation filaire ci-dessus que toutes les faces coplanaires ont bien été fusionnées et que la façade est désormais représentée par une seule face. Le poids du fichier Sketchup est de 353 KO.

### III.1.3 Comparaison des deux méthodes

Pour résumer au mieux les deux types de modélisation un tableau présentant les caractéristiques de chacune d'elle est réalisé.

Modélisation par maillage	Modélisation à partir de primitives géométriques
Représentation détaillée de l'architecture même si ce n'est pas le cas pour les balcons	Représentation simplifiée
Poids du fichier élevé	Poids du fichier relativement léger
Temps important passé pour trouver les bons paramètres et retravailler les imperfections du maillage	Peu de temps passé pour générer le modèle 3D
Bonne précision entre le maillage et le nuage de points quel que soit l'endroit sur le maillage	Précision entre le nuage de points et le plan qui peut varier d'un endroit à l'autre

À partir de ce tableau nous pouvons constater que chaque méthode présente des avantages et des inconvénients. Pour modéliser toutes les façades à l'échelle d'une ville, nous avons besoin d'un modèle 3D léger et facile d'utilisation donc nous allons opter pour la modélisation à partir de primitives géométriques.

## III.2 Modélisation d'un ensemble de façades

Après avoir réalisé la modélisation sur une façade, nous devons maintenant trouver une méthode nous permettant de modéliser un ensemble de façades de manière précise et efficace. Un ensemble de façades de la ville de Rennes d'environ 150 mètres est donc choisi comme échantillon test. Après réflexion, nous souhaitons nous pencher vers l'automatisation de la modélisation de contours de portes et de fenêtres car c'est une étape qui, réalisée manuellement peut s'avérer très longue.

### III.2.1 Création de plans

Après avoir supprimé les points indésirables, il est nécessaire de segmenter manuellement l'ensemble des façades à chaque fois qu'il y a un changement de plan. Chaque nuage de points peut par la suite être projeté sur le plan correspondant comme cela a été expliqué dans la partie III.1.2.1.

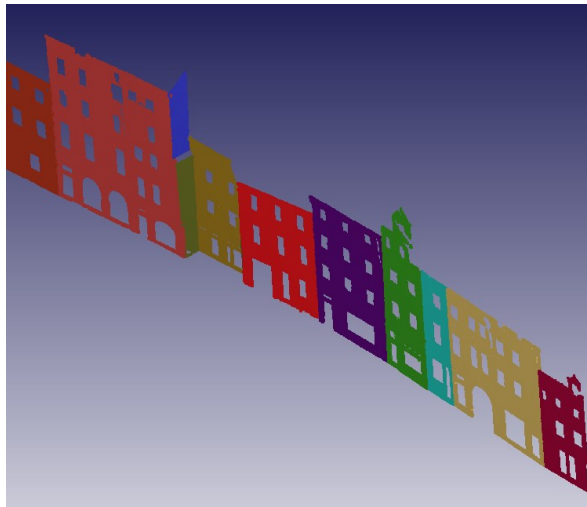


Figure 42: visualisation du nuage de points segmenté et projeté sur différents plans. [3DReshaper]

Les différents nuages peuvent ensuite être fusionnés par le biais de l'outil « Nuages → Fusionner Nuage(s). »

### III.2.2 Détection de contours

La détection de contours commence par la création d'un maillage. Il faut donc réaliser un maillage permettant de combler un maximum de trous tout en préservant une certaine précision au niveau des contours. Un maillage en deux étapes doit par conséquent être réalisé.

Pour la première étape et après plusieurs tests, nous optons pour un échantillonnage régulier de 15 centimètres.

Pour la deuxième étape, le maillage peut être affiné à partir des points du nuage car la présence de bruit est moins élevée que dans la partie III.1.1 car le nuage de points a été plaqué sur un plan. Cependant des points isolés persistent au niveau des contours. Pour éviter de les prendre en compte, nous avons choisi de calculer « l'erreur cordale avec les meilleurs points uniquement » ce qui a pour conséquence de sélectionner uniquement les points les moins bruités lors de la création du maillage.

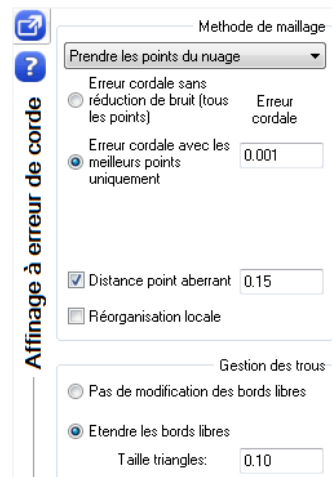


Figure 43: Méthode de maillage utilisé. [3DReshaper]

Après avoir affiné le maillage, nous nous apercevons que les angles des contours sont coupés et que les côtés ne sont pas représentés de façon rectiligne mais oscillent de manière plus ou moins importante suivant la qualité des données.

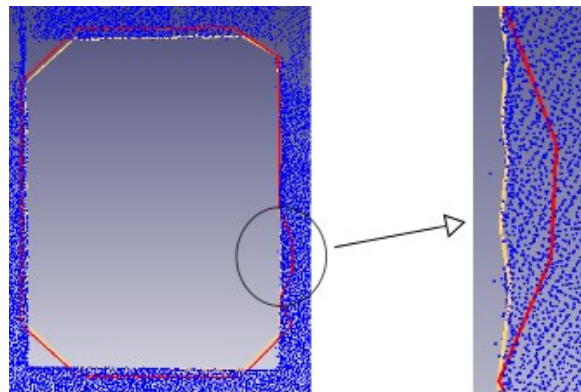


Figure 44: Contour extrait à partir du maillage initial de 15 centimètres (rouge) et après affinage du maillage (jaune). [3DReshaper]

Après avoir créé le maillage, il est nécessaire de boucher les trous apparents ne représentant pas des portes ou des fenêtres de façon à ce qu'il n'y ait pas de données fausses lors de l'extraction automatique des contours. Cette dernière action peut être réalisée à partir de la fonction « Tous contours/trous de l'entité cliqué » présente dans l'outil « Polyligne → Contour/Trou » de 3DReshaper.

Nous pouvons constater après visualisation que certains contours du fait du manque de données ne peuvent pas être extraits automatiquement, il faudra donc les digitaliser manuellement. Ces contours sont au nombre de 15%.

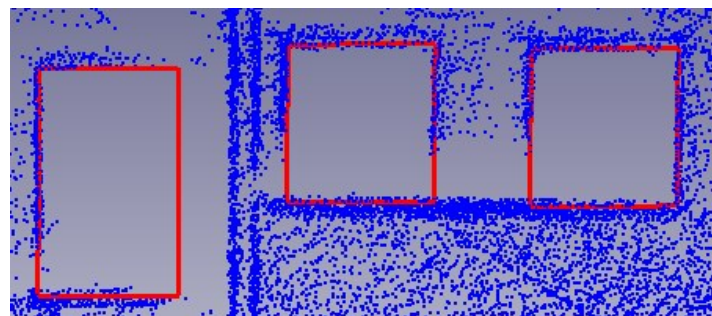


Figure 45: Visualisation des contours digitalisés manuellement sur le nuage de points. [3DReshaper]



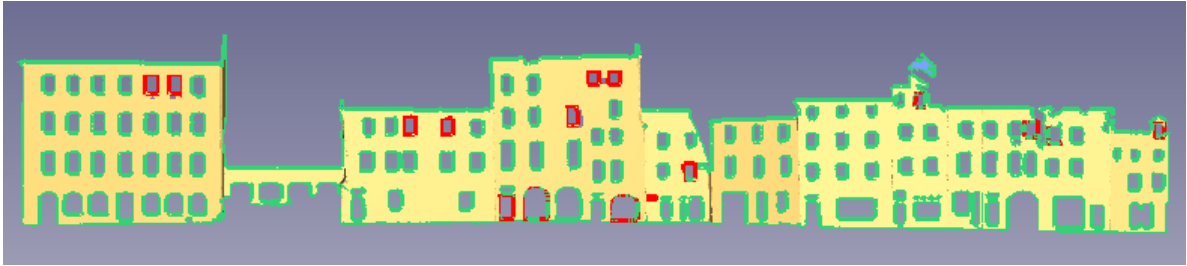


Figure 46: Visualisation des contours extraits automatiquement (vert) et digitalisés manuellement (rouge). [3DReshaper]

### III.2.3 Modélisation de contours

Après réflexion, nous avons décidé de mettre en place une méthode permettant de représenter chaque contour de porte et de fenêtre par un rectangle (voir [annexe 2](#)). Le rectangle est en effet une forme géométrique que l'on retrouve fréquemment au niveau des ouvertures de façades et qui peut être facilement modifiable si l'un de ses côtés doit être transformé en arc de cercle. Le fait d'utiliser une forme géométrique simple permet également d'obtenir un modèle léger.

Compte tenu de la précision des données (4-5 centimètres) et de la précision recherchée (10 centimètres), nous avons convenu que représenter un contour par son rectangle englobant pourrait s'avérer être une solution efficace.

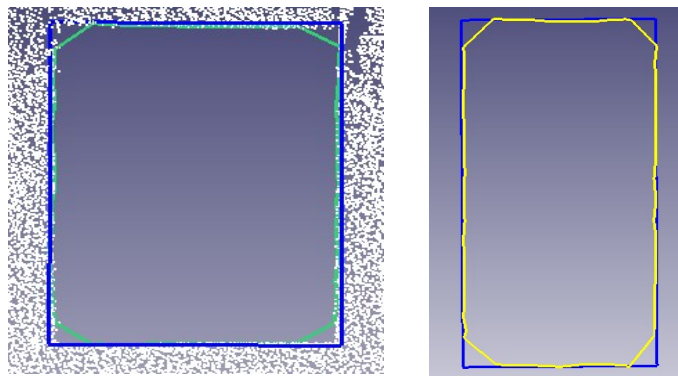


Figure 47: Représentation de contours à partir de rectangles englobants digitalisés manuellement. [3DReshaper]

Pour mettre en place cette modélisation, nous allons utiliser l'outil FME Workbench du logiciel FME Desktop qui permet de concevoir graphiquement des traitements que l'on souhaite effectuer sur des données de différentes sortes.

Cet outil comprend de nombreux « transformers » pouvant être définis comme des objets qui permettent de restructurer des entités. Parmi eux on trouve le transformer « BoundingBoxAccumulator » qui va créer le rectangle minimal englobant 2D d'un ensemble d'entités (points, lignes, polygones) dont les côtés sont parallèles aux axes X et Y.

En visualisant les données à partir de l'outil FME Universal Viewer, nous pouvons nous rendre compte qu'il faut commencer par appliquer une rotation sur chaque entité autour de l'axe Z pour obtenir une orientation parallèle à l'axe des X.

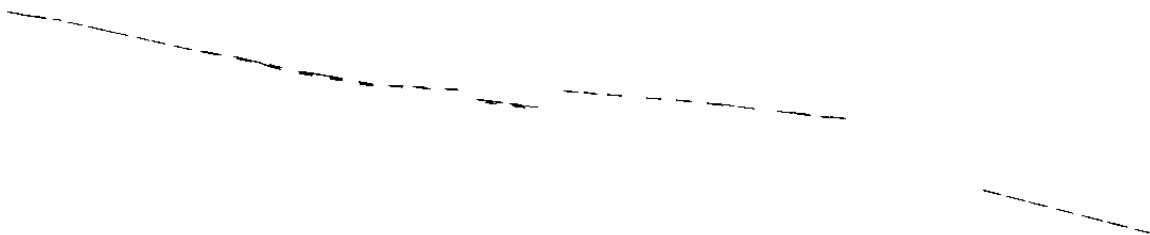


Figure 48: Visualisation des contours extraits automatiquement sous FME Universal Viewer

Chaque contour étant représenté par un polygone, il est possible pour chacun d'eux d'extraire les coordonnées  $X_{min}$ ,  $X_{max}$ ,  $Y_{min}$  et  $Y_{max}$  en utilisant le transformeur « BoundsExtractor ». À partir de ces coordonnées nous pouvons calculer l'angle d'inclinaison de chaque entité par rapport à l'horizontal en appliquant la formule suivante:

$$angle = \text{atan} \frac{(Y_{max} - Y_{min})}{(X_{max} - X_{min})} = \text{atan} \frac{\Delta Y}{\Delta X}$$

Suivant l'inclinaison de chaque polygone, l'angle calculé peut changer de signe au moment de la rotation, les deux cas sont présentés dans la Figure 49.

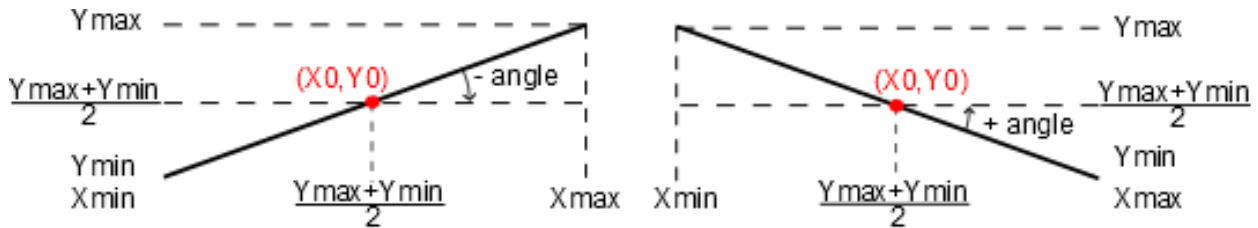


Figure 49: Détermination du signe de l'angle à utiliser dans la rotation (vue de dessus)

Pour chaque polygone, une rotation est effectuée dans les deux sens à partir du transformeur « Rotator » qui permet de faire pivoter chaque entité autour de l'axe des Z sans affecter l'information d'élévation et à partir d'un point d'origine défini par les coordonnées  $(X_0, Y_0)$ .

En analysant l'angle de verticalité d'un certain nombre de façades nous constatons que celui-ci oscille entre  $0^\circ$  et  $1,5^\circ$  sauf exception, ce qui représente pour un contour de 2m x 2m (dimensions maximales d'une porte ou fenêtre), un décalage de 5,3 cm entre le pied et le haut de la fenêtre pour un défaut de verticalité de  $1,5^\circ$ .

Pour garder la valeur de l'angle de rotation qui nous intéresse nous commençons par effectuer un premier test où l'on applique une rotation de  $90^\circ$  pour les contours où  $\Delta X < 5,5$  cm pour éviter de se retrouver dans un cas particulier où les deux solutions pourraient être acceptées. Le fait de créer un rectangle englobant à partir d'un contour pas complètement vertical n'altérera quasiment pas la qualité de la modélisation. Un second test consiste à conserver l'angle de rotation utile pour les autres contours où  $\Delta X > 5,5$  cm et  $\Delta Y < 5,5$  cm.

Une fois cette étape exécutée, une rotation de 90 degrés suivant l'axe des X peut être réalisée par le biais du transformeur « 3DAffiner » qui permet de réaliser des transformations affines 3D sur les coordonnées des entités.

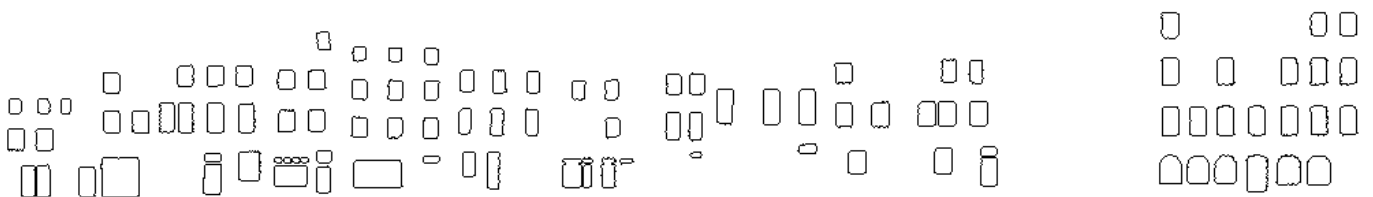


Figure 50: Visualisation des entités après avoir effectué les deux transformations. [FME Universal Viewer]

Nous pouvons dès à présent lancer le transformeur « BoundingBoxAccumulator ».

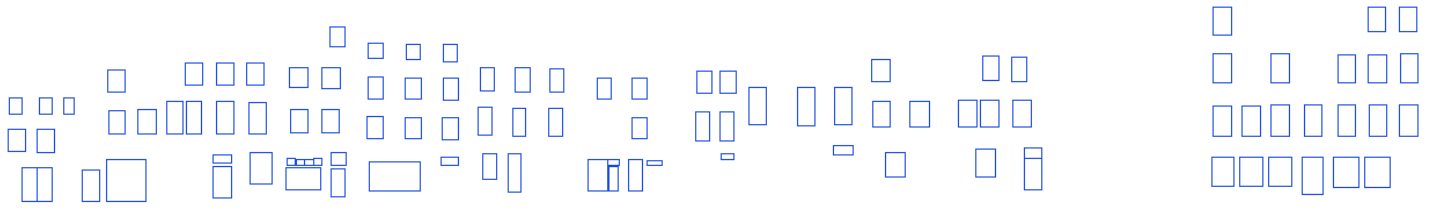


Figure 51: Visualisation des rectangles englobants. [FME Universal Viewer]

Pour rattacher la troisième coordonnée aux entités, nous employons le transformeur « 3DForcer » qui va attribuer aux coordonnées Z de chaque entité la valeur de l'attribut spécifié. Une valeur de z est donc affectée à chaque rectangle englobant.

Afin de retrouver les coordonnées X et Y de départ, il est nécessaire d'exécuter les transformations inverses des deux rotations effectuées en commençant par la rotation sur l'axe des X.

Les nouvelles entités créées peuvent par la suite être exportées au format « dxf » avant d'être importées dans le logiciel 3DReshaper. Pour que chaque entité retrouve toutes ses coordonnées Z de départ nous utilisons l'outil « création → Projection » qui va réaliser une projection normale de toutes les entités sur le maillage créé précédemment.



Figure 52: Visualisation de entités créées après projection sur le maillage. [3DReshaper]

Il est également possible de modéliser automatiquement le contour extérieur des façades mais, après plusieurs essais, nous constatons qu'il est impossible d'obtenir un résultat net et précis car les données sont très bruitées au niveau des hauts et des bas de façades. Il est donc plus judicieux de modéliser manuellement le contour extérieur des bâtiments.

### III.2.4 Évaluation de la modélisation

Des évaluations qualitatives et quantitatives vont être réalisées sur les contours modélisés automatiquement.

#### III.2.4.1 Évaluation qualitative

L'évaluation qualitative des différents modèles vectoriels obtenus consiste à comparer visuellement ces modèles avec le nuage de point.

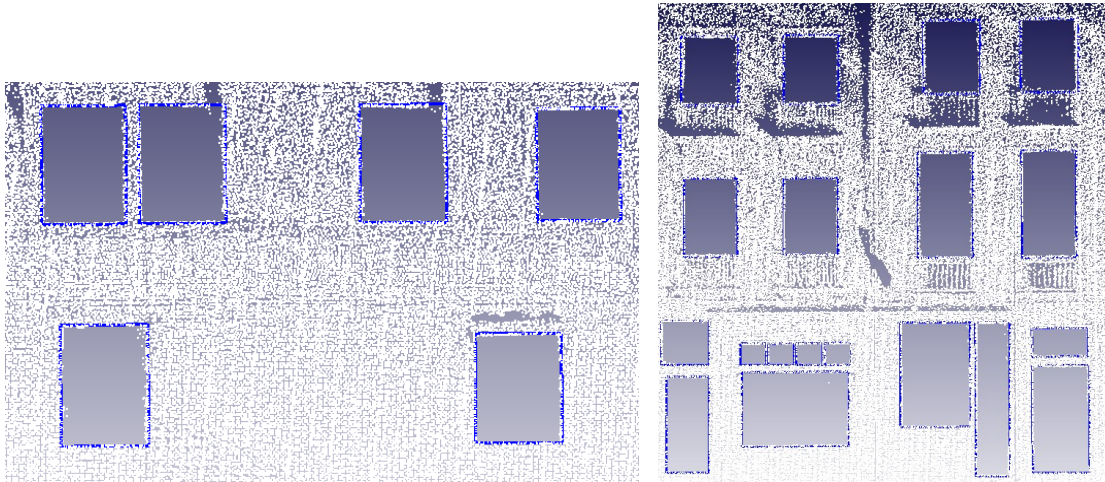


Figure 53: Superposition du nuage de point et des contours modélisés automatiquement. [3DReshaper]

Nous pouvons constater (Figure 53) que la superposition du nuage de point avec les modèles vectoriels 3D obtenus est satisfaisante.

En visualisant les fenêtres munies de balcons, nous constatons que les points bas des contours modélisés ne correspondent pas à la limite de la fenêtre qui ne peut d'ailleurs pas être réellement identifiable à partir des données issues de l'acquisition. Le balcon est un masque d'acquisition par rapport au bas de la fenêtre. Cela représente une des limites de la méthode adoptée.

### III.2.4.2 Évaluation quantitative

Pour quantifier la qualité de la modélisation automatique, nous allons commencer par évaluer la probabilité de détection de contours (portes et fenêtres) puis nous estimerons la précision obtenue à partir d'un modèle de référence digitalisé manuellement.

$$\text{Probabilité de détection} = \frac{\text{nombre contours détectés}}{\text{nombre contours réels}} = 98\%$$

Nous pouvons en déduire que la probabilité de détection est satisfaisante, cependant lorsque deux fenêtres sont très proches il peut arriver qu'elles soient toutes les deux représentées par un unique contour.

Pour évaluer la précision du modèle obtenu par rapport au nuage de points, nous allons calculer pour chaque rectangle modélisé, la distance entre chaque coin détecté et coin vrai et en déduire la distance moyenne et l'écart type.

Pour n=100 mesures :

$$\text{Distance moyenne} = \bar{x} = \frac{1}{n} \sum_{i=1}^n x_i = 1,9 \text{ cm}$$

$$\text{Ecart type} = \sigma = \sqrt{\frac{1}{n-1} \sum_{i=1}^n (x_i - \bar{x})^2} = 1,3 \text{ cm}$$

A partir de ces résultats, nous pouvons en conclure que la précision des contours est de l'ordre de 3 centimètres, ce qui s'avère être satisfaisant car la précision de pointé d'une digitalisation manuelle est de l'ordre de 1 à 2 centimètres.

### III.2.5 Modification de contours

Après avoir réalisé la modélisation automatique, il est nécessaire de modifier les ouvertures qui ne présentent pas une forme rectangulaire.

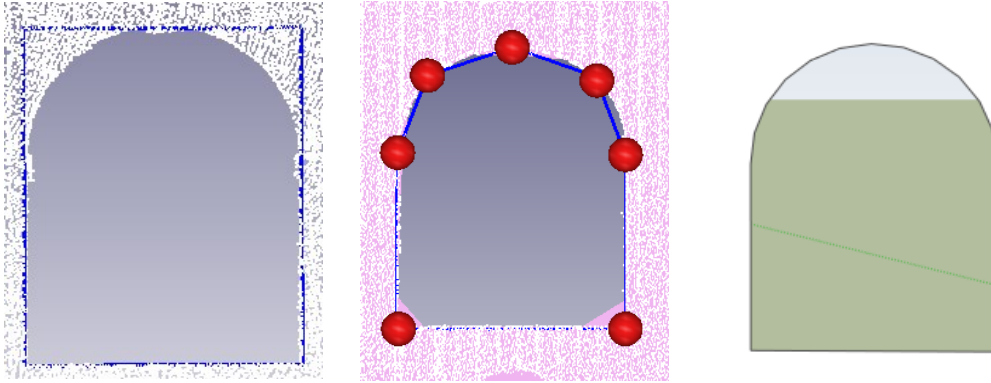


Figure 54: Représentation d'un contour automatique (gauche). Modification du contour sous 3DReshaper (milieu) et sous SketchUpMake (droite).

Le logiciel 3DReshaper ne dispose pas d'outils permettant de dessiner des arcs de cercles mais d'un outil permettant de déformer un contour à partir de plusieurs points. Après avoir sélectionné le maillage et la ou les polygones, il est possible de projeter la partie que l'on souhaite modifier sur le maillage en activant l'option « Projection sur le polyèdre sélectionné ». Cette partie courbe pourra ensuite être transformée facilement en arc de cercle sous le logiciel SketchUpMake, ce qui permettra d'obtenir une forme géométrique simple.

Les endroits présentant un manque de données peuvent également être représentés par des faces. Les balcons sont digitalisés sur 3DReshaper et seront insérés de façon automatique sous forme de cellules 3D dans Rhinocéros 3D (méthode présentée dans l'annexe 3). Le temps passé pour modéliser cet ensemble de façades à partir du nuage de points brut est d'environ 4h. Le modèle 3D peut être par la suite inséré dans Rhinocéros 3D pour être enregistré au format CityGML.



Figure 55: Modélisation obtenue sous 3DReshaper après maillage sous contrainte (haut), modélisation obtenue sous SketchUpMake (milieu) et modélisation finale obtenue sous Rhinocéros 3D (bas).

### III.3 Modélisation d'objets urbains

#### III.3.1 Calcul des coordonnées du point d'insertion

➤ Les objets ponctuels

Pour commencer, nous allons récupérer les différents objets 3D dans la librairie 3D Warehouse accessible depuis le logiciel SketchUpMake.

Nous devons maintenant mettre en place une méthode permettant d'obtenir les coordonnées (X, Y et Z) du point d'insertion de ces différents objets à partir des nuages de points obtenus après classification (voir partie II.3.1). Nous allons modéliser les arbres et les candélabres car ce sont des objets que l'on rencontre fréquemment en bordure de route.

Avec 3DReshaper, nous pouvons changer la couleur d'un nuage de points selon l'axe Z en définissant le nombre de paliers souhaité puis, par la suite, exploser le nuage de points par couleurs. Cela nous permet d'obtenir les parties basses de chaque objet. Nous pouvons dès à présent effectuer un maillage sur chaque entité et extraire le meilleur cylindre de chaque maillage.

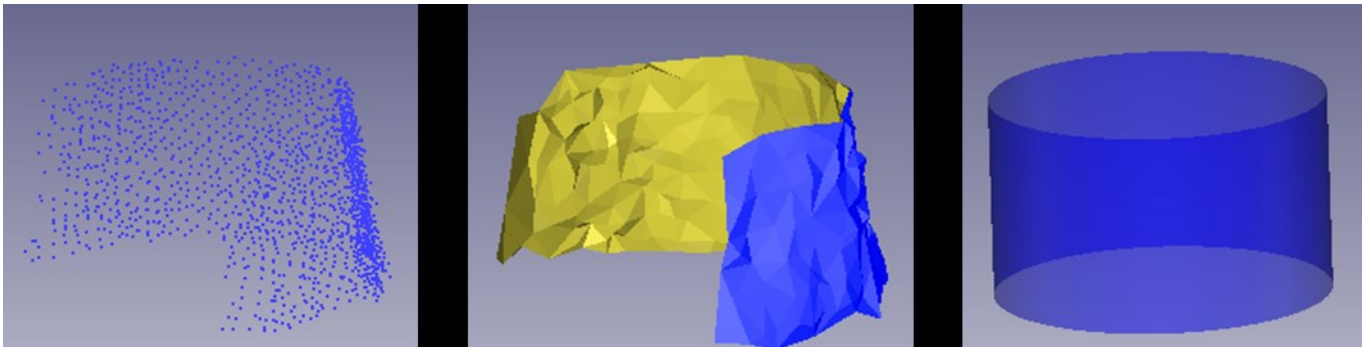


Figure 56: Etapes présentant le passage d'un nuage de points à un cylindre (de gauche à droite). [3DReshaper]

Ces différents cylindres peuvent par la suite être intégrés dans le logiciel FME Workbench au format « dxf ». En utilisant le transformer « CenterofGravityReplacer », le centre de gravité de chaque objet peut être calculé, nous permettant d'obtenir les coordonnées planimétriques du point d'insertion.

Sur Terrascan, après avoir allégé le nuage de points correspondant aux classes « sol » et « trottoir », nous exportons celui-ci sous la forme d'un raster par le biais de la fonction « export lattice model », qui permet d'obtenir les valeurs d'élévation de chaque cellule de la grille.

Sur FME, les différents points créés pour chaque entité pourront être projetés verticalement sur le MNT puis interpolés par l'intermédiaire du transformer « PointOnRasterValueExtractor », nous obtenons ainsi pour chaque objet la valeur altimétrique du point d'insertion.

➤ Les objets linéaires

Comme objets linéaires nous allons modéliser les abris de bus.

Les abris de bus vont être digitalisés manuellement à partir du nuage de points sur le logiciel Microstation selon un ordre de saisie.

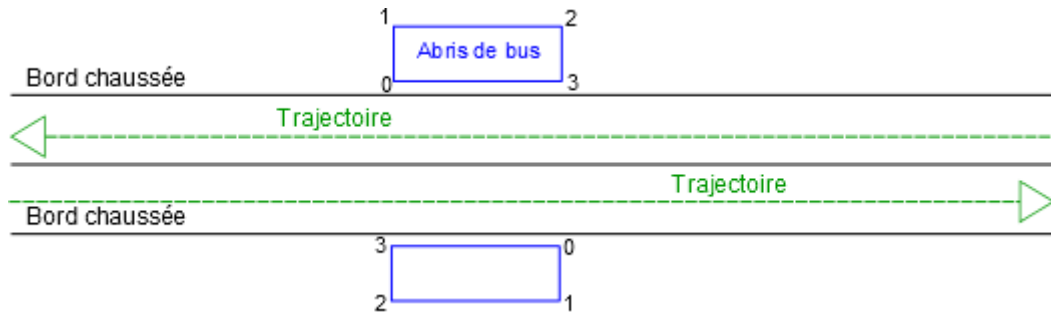


Figure 57: Ordre de saisie choisie pour la digitalisation

Il est impossible, sous Rhinocéros 3D, d'insérer un bloc par trois points. Le point d'insertion de chaque entité sera donc comme pour les objets ponctuels le centre de gravité. Pour que la cellule 3D vienne plaquer le MNT sur les quatre coins du bloc nous calculons le vecteur normal au plan défini par les points 0, 1 et 3. Ces valeurs seront renseignées lors de l'insertion du bloc.

### III.3.2 Calcul de l'orientation

➤ Les candélabres

Pour que l'orientation de chaque candélabre soit perpendiculaire à la route, nous allons utiliser les trajectoires du véhicule lors de l'acquisition (voir Figure 58). En effet, après visualisation de plusieurs portions de routes, nous nous rendons compte que les candélabres sont situés sur le côté droit de la trajectoire.

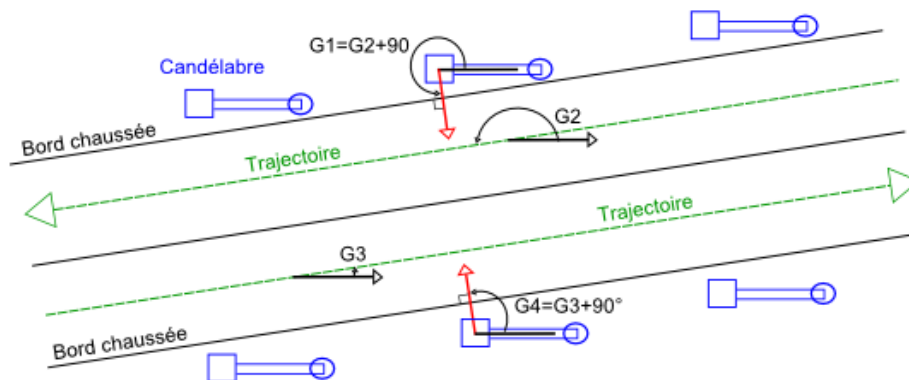


Figure 58: Orientation des candélabres par rapport aux trajectoires

Le transformateur « NeighborFinder » permet de trouver la trajectoire (ligne) la plus proche de chaque entité (candélabre) et calcule l'angle entre le point interpolé de la ligne la plus proche de chaque candélabre et le vertex suivant sur la ligne. Cet angle correspondant à l'orientation de la trajectoire auquel nous allons ajouter une constante de 90 degrés pour avoir l'orientation des candélabres (voir Figure 58).

Pour les autoroutes comportant des candélabres double branches situés sur la partie centrale, le calcul d'orientation reste le même, nous allons changer uniquement le symbole 3D à importer.

➤ Les abris de bus

L'orientation de chaque cellule est calculée en fonction de la trajectoire de la même manière que pour les objets ponctuels mais cette fois ci avec l'ajout d'une constante de 180 degrés.

### III.3.3 Insertion sous Rhinocéros 3D

Il est possible sous Rhinocéros5 via la commande EditPythonScript, d'éditer un script en langage python en ayant accès aux bibliothèques de fonctions python et Rhino (rhinoscriptsyntax).

➤ Les objets ponctuels

Nous obtenons en sortie du script FME un fichier texte comprenant les coordonnées X, Y et Z du point d'insertion de l'objet ainsi que son type (arbre, candélabre) et son orientation. (Voir script FME présent en [annexe 4](#)).

Les différents symboles sont insérés dans un calque qui leur est propre à l'échelle (1,1,1).



Figure 59: Aperçu des arbres et des candélabres insérés sous Rhinocéros3D

➤ Les objets linéaires

Nous obtenons en sortie du script FME un fichier texte comportant les coordonnées X, Y et Z du point d'insertion de l'objet, le type (abris de bus), l'orientation, les trois coordonnées du vecteur normal ainsi que les coefficients d'échelles en X et en Y. (Voir script FME présent en [annexe 5](#)). Chaque bloc est donc inséré suivant ses dimensions réelles (longueur et largeur) dans le dessin.

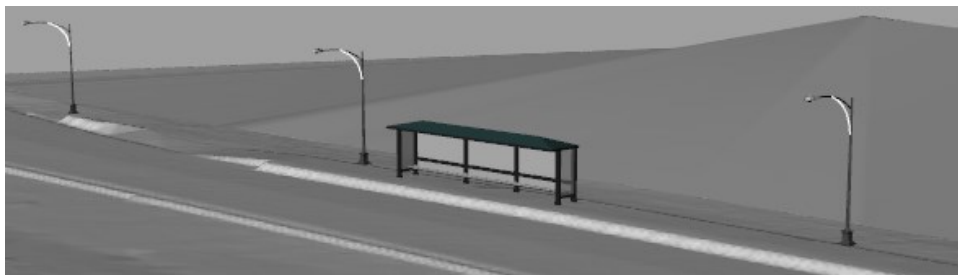


Figure 60: Aperçu d'un abri de bus inséré sous Rhinocéros 3D

Tous les symboles (ponctuels et linéaires) sont insérés automatiquement sur Rhinocéros 3D à partir du script python présenté en [annexe 6](#).

### III.4 Conclusion

Dans cette troisième partie, nous avons pu comparer deux méthodes de modélisation différentes en terme d'utilisation, de précision, de temps passé et de rendu final. La modélisation simplifiée à



partir de primitives géométriques s'avère être la solution adéquate pour mener un projet à l'échelle d'une ville.

Une méthode de modélisation automatique de contour de portes et de fenêtres a été mise en place permettant un gain de temps, un allègement du modèle final tout en restant dans des critères de précision acceptables.

Enfin, des méthodes ont été proposées pour insérer de façon semi-automatique des symboles 3D à partir de nuages de points permettant ainsi une représentation réaliste que l'on peut retrouver dans une maquette 3D de type piéton.

## **Conclusion et perspectives**

L'objectif principal de ce Travail de Fin d'Études était de savoir si, à partir d'un nuage de points 3D issu d'une acquisition MMS, il était possible d'obtenir un modèle 3D exploitable répondant à des critères de précision topographique, de réalisme et de facilité d'utilisation.

Une méthode de classification semi-automatique précise et efficace permettant d'identifier les points du nuage appartenant à la route et aux trottoirs a été mise en place à partir du logiciel Terrascan.

Pour distinguer les façades et les objets urbains (arbres, candélabres, panneaux), des méthodes de segmentation par critères de distances donnant des résultats plutôt convaincants ont été réalisées sur 3DReshaper. Concernant les façades, cette méthode ne permet pas d'éliminer la totalité des points indésirables et de segmenter chaque façade individuellement.

Une modélisation de façade sous la forme d'un maillage simplifié et léger a été réalisée sous 3DReshaper.

Une modélisation automatique permettant d'assimiler chaque ouverture de façade à un rectangle a été créée à partir des logiciels FME et 3DReshaper. Cette méthode a pour avantage de générer un modèle léger et de pouvoir traiter un ensemble de façades en même temps avec une précision de l'ordre de 3 centimètres sur les contours rectangulaires, ce qui est acceptable pour une modélisation 3D urbaine de type piéton. Une méthode permettant de modéliser les bacons de manière semi-automatique a également été proposée, ce qui accentue la rapidité et le réalisme de la modélisation de façades

Il n'est pas possible de modéliser les toits des bâtiments à partir des données acquises par MMS. Les toits devront donc être modélisés par photogrammétrie ou lasergrammétrie aérienne. Cependant la précision des données n'étant pas la même il faudra effectuer un calage de celles-ci par rapport aux façades mais cela n'affectera pas la maquette 3D finale.

Pour modéliser de manière automatique et réaliste les objets urbains, un premier « programme » FME a permis de renseigner les coordonnées du point d'insertion, le type et l'orientation de l'élément urbain pour les arbres et les candélabres.

Un second « programme » FME renseignant le point d'insertion, le type, le vecteur normal au plan, la longueur et la largeur a été créé pour les abris de bus.

Un script Python a ensuite permis d'insérer automatiquement ces objets sous la forme de cellules 3D dans Rhinocéros3D et ainsi d'obtenir une représentation réaliste du mobilier urbain. De plus les informations obtenues pour chaque objet (coordonnées, type, longueur, largeur...) pourront être intégrés par la suite au format CityGML qui prend en compte la sémantique.

Avec la standardisation et l'interopérabilité du format CityGML, les différentes sources de données pourront être répertoriées dans un même fichier dans tout logiciel de visualisation 3D.

Afin de pouvoir segmenter chaque façade les unes des autres de manière automatique, il faudrait utiliser non pas le nuage de points mais les images géoréférencées et calibrées obtenues par les

caméras car un changement de façade provoque la plupart du temps un changement d'intensité et une discontinuité radiométrique dans l'image.

Pour faciliter les extrusions au niveau des ouvertures de façades, il serait possible à partir de 3 points d'assimiler chaque façade à un rectangle par optimisation stochastique en imposant des contraintes au niveau des jonctions entre chaque façade. Cela permettrait de réaliser des extrusions façade par façade et non plus contour par contour ce qui engendrerait un gain de temps considérable lors de la modélisation et allégerait un peu plus le modèle final.

Les scripts FME produits peuvent également s'appliquer pour la représentation d'autres objets ponctuels (poubelles et les panneaux) et linéaires (les bancs).

Afin d'avoir une représentation lisse et simplifiée du trottoir et de la chaussée, il est possible comme nous pouvons le voir dans [Terrasse, 2013] de remplacer le MNT par un modèle numérique de route qui rendra la maquette piétonne plus esthétique et plus légère.

D'un point de vue personnel, ce travail de Fin d' Études m'a apporté beaucoup. Au terme de ces 20 semaines au sein de la société Fit Conseil, j'ai acquis de l'expérience dans les domaines de la lasergrammétrie mobile et de la modélisation 3D. Ce travail m'a permis de découvrir de nouveaux logiciels comme 3DReshaper, Rhinocéros 3D et Terrascan permettant de traiter et de modéliser un nuage de points.

Cette expérience m'a également appris à travailler de manière autonome car j'ai dû organiser les étapes de mon projet de manière cohérente dans les phases de recherche et de production.

Me prédestinant au métier de géomètre expert où le foncier occupe une place prépondérante, je me suis rendu compte que la programmation est une compétence qui peut s'avérer utile au vue des nombreuses possibilités qui en résultent et qu'il ne faut donc pas la négliger.

## **Bibliographie**

### *Ouvrages, articles et thèses*

- [Aijazi et al., 2012] Aijazi, A.K., Checchin, P., Trassoudaine, L. (2012). Segmentation et classification de points 3D obtenus à partir de relevés laser terrestres : une approche par super-voxels. Acte du dix-huitième congrès francophone AFRIF-RFIA, Lyon, France, 1.
- [Banaszak et Koehl., 2008] Banaszak, O., Koehl, M. (2008). SIG et 3D au service des collectivités territoriales : l'expérience de la ville du Havre. Revue XYZ n°114, p.50-58
- [Begrache, 2003] Begrache, R. (2003). Profil - du nuage de points au modèle 3D, contribution de l'élaboration d'un outil d'aide à la modélisation, cas d'un entablement composite. DEA, Modélisation et Simulation des espaces Bâti. MAP-CRAI (NANCY).
- [Benitez et Baillard., 2009] Benitez, B., Baillard, C. (2009). Automated selection of terrestrial images from sequences for the texture mapping of 3D city models. *IAPRS & SIS, XXXVIII (Part3/W4)* :97-102, 2009.
- [Bénitez et al., 2010] Benitez, B., Denis, E., Baillard, C. (2010). Automatic production of occlusion-free rectified façade texture using vehicle-based imagery. In *Photogrammetric Computer Vision and Image Analysis*, volume A, Paris, France.
- [Boulaassal, 2010] Boulaassal, H. (2010). Segmentation et modélisation géométrique de façades de bâtiments à partir de relevés laser terrestres. Thèse de doctorat, Université de Strasbourg, Strasbourg.
- [Ellum et El-Sheimy., 2002] Ellum, C., El-Sheimy, N., (2002). Land-based mobile mapping systems. *Photogrammetric Engineering & Remote Sensing (PE&RS)*.
- [Fischler et Bolles., 1981] Fischler, M. et Bolles, R. (1981). Random sample consensus : a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381\_395.
- [Gaillet et al., 2013] Gaillet, M., Henry, J., Niggeler, L. (2013). Relevé de corps de rue à Genève. *Géomatique Expert*, n°91, pages 50-55.
- [Gröger et al., 2006] Gröger, G., Kolbe, T., Czerwinski, A. (2006) Candidate opengis implementation specification. Technical Report OGC 07-062, Open Geospatial Consortium Inc.
- [Hough, 1962] Hough. (1962). Method and Means for Recognizing Complex Patters. U.S. Patent 3.069.654.
- [Kartinen et al., 2013] Kartinen, H., Hyypä, J., Kukko, A., Lehtomäki, M., Jaakkola, A., Vosselman, G., Oude Elberink, S., Rutzinger, M., Pu, S., Vaaja, M. (2013). Mobile Mapping-Road Environment Mapping using Mobile Laser Scanning. Official Publication N°62.
- [Landes et Grussenmeyer., 2011] Landes, T., Grussenmeyer, P. (2011). Les principes fondamentaux de la lasergrammétrie terrestre : systèmes et caractéristiques (partie 1/2). Revue XYZ n°119, p. 37-49.
- [Manandhar et Shibasaki, 2001] Manandhar, D., Shibasaki, R. (2001). Auto-extraction of urban features from vehicle-borne laser data, *IAPRS*, Vol. 34, Part 4, « geospatial Theory, Processing and Applications ».
- [Pénard, 2009] Pénard, L. (2009). Reconstruction de façades de bâtiments par appariement dense d'images terrestres géoréférencées haute résolution. Thèse de doctorat, Université de Paris Descartes, Paris.
- [Pu et al., 2007] Pu, Shi., Vosselman, George. (2007). Extracting windows from terrestrial laser scanning. *ISPRS Workshop on Laser Scanning 2007 and SilviLaser 2007*, Espoo, September 12-14, 2007, Finland.
- [Terrasse, 2013] Terrasse, G. (2013). Production automatisée de maquette 3D de type piéton. Mémoire, INSA, Strasbourg.
- [Vosselman et al., 2004] Vosselman, G., Gorte, B.G.H., Sithole, G., Rabbani, T. (2004). Recognising structure in laser scanner point clouds. *IAPRS*, vol.46, part 8/W2, Freiburg, Germany, October 4-6, pages 33-38.
- [Yu et al., 2008] Yu, G., Grossberg, M., Wolberg, G., Stamos, I. (2008). Think globally, cluster locally : a unified framework for range segmentation. *Proceedings of 3DPVT'08-the Fourth Symposium on 3D Data Processing, Visualization and transmission*. 8 pages.

### *Documents de formation*

[Morel, 2013] Morel, L. (2013). GNSS et Géoréférencement. ESGT.

[Cali et al., 2013] sous la coordination de Simonetto, E. (2013). Cali, J., Chandelier, L., Héno, R., Polidori, L. Photogrammétrie. ESGT

### *Sites Internet*

[@3DLM, 2014] (2014). The Streetmapper. URL: <http://www.3dlasermapping.com/>.

[@Leica, 2014] (2014). Leica Pegasus:One. URL: [http://www.leica-geosystems.com/en/Leica-PegasusOne\\_103551.htm](http://www.leica-geosystems.com/en/Leica-PegasusOne_103551.htm)

[@Optech, 2014] (2014). Optech Lynx SG1 Mobile Mapper. URL: <http://www.optech.com/index.php/product/lynx-sg1/>

[@Riegl, 2014] (2014). Riegl vmx-450 URL: <http://www.riegl.com/nc/products/mobile-scanning/produktdetail/product/scannersystem/10/>

[@Technodigit, 2014] (2014). URL: <http://www.technodigit.com/>

[@Terrasolid, 2014] (2014). Software for Processing LiDAR Point Clouds and Images. URL: <http://www.terrasolid.com/home.php>

[@rhino3d, 2014] (2014). URL: <http://www.rhino3d.com/fr/>

[@PDF3D, 2014] (2014). URL: [http://www.pdf3d.fr/index\\_fr.php](http://www.pdf3d.fr/index_fr.php)

[@COLLADA, 2014] (2014). URL: <https://collada.org/>

[@ubick, 2014] (2014). URL: <http://www.ubick.net/>

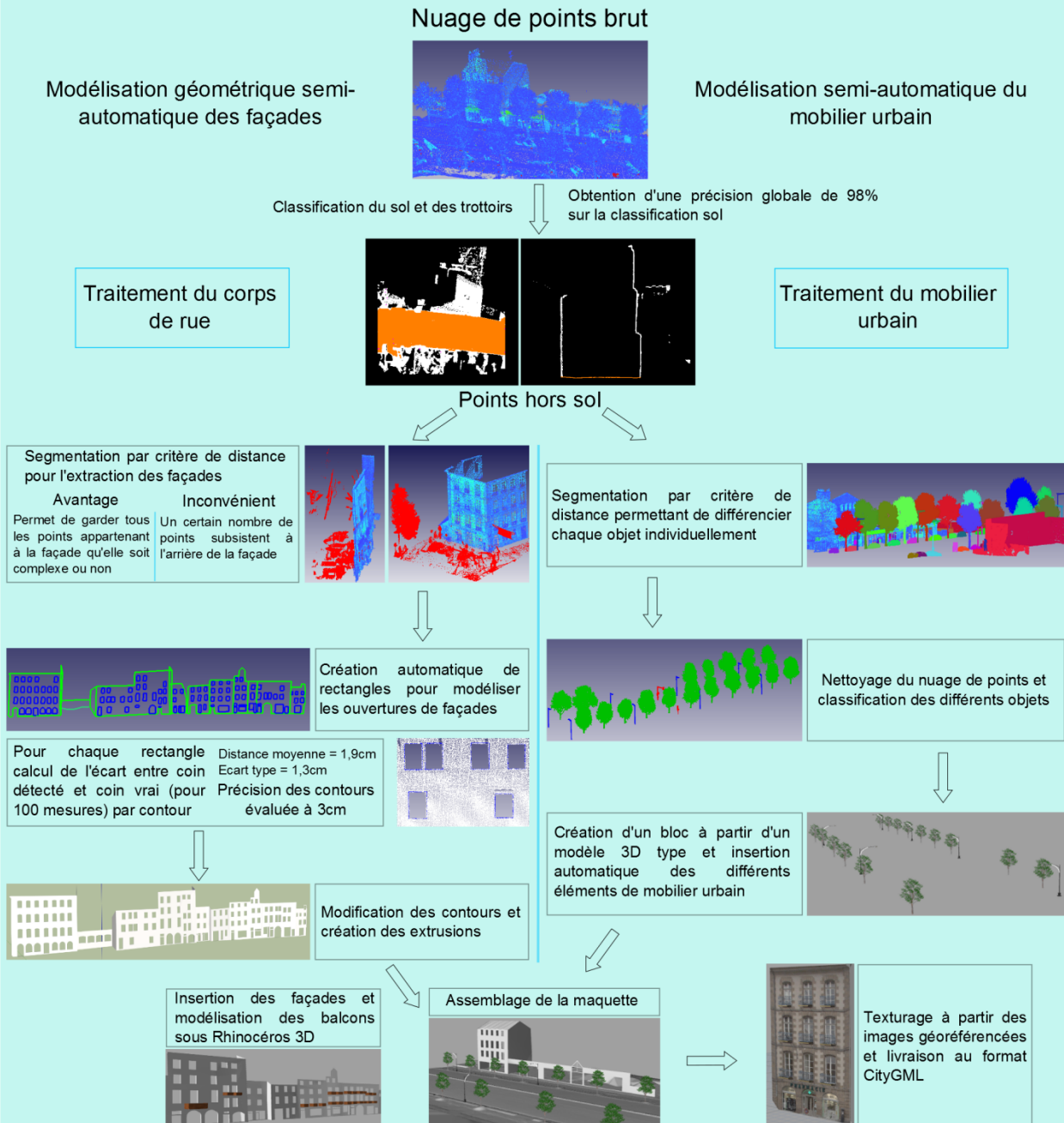
[@gouv.mc] (2012) URL: <https://www.gouv.mc/Action-Gouvernementale/La-Qualite-de-Vie/La-Prospective/Actualites/La-maquette-3D-de-Monaco-presentee-par-la-DPUM-a-Imagina>

## Poster scientifique

#### Objectifs de l'étude

- Produire et évaluer des méthodes de segmentation et de classification automatiques permettant de différencier les éléments urbains
- Comparaison de différentes méthodes de modélisation et automatisation de la modélisation des contours de portes et de fenêtres pour la méthode retenue
- Modéliser le mobilier urbain

Comment passer d'un semis de points 3D acquis par Mobile Mapping System à un modèle 3D exploitable?



#### Conclusion

- Processus semi automatique de classification (du sol et des trottoirs) et de segmentation (des façades et des objets urbains)
- Modèle 3D léger répondant à des critères de précision et de facilité d'utilisation
- Processus automatique de modélisation d'éléments de mobilier urbain avec l'insertion de cellules 3D sous Rhinocéros 3D

## **Table des annexes<sup>1</sup>**

Annexe 1 Macro sol .....	54
Annexe 2 Script FME modélisation de contours.....	55
Annexe 3 Modélisation et insertion automatique des balcons .....	57
Annexe 4 Script FME modélisation d'objets ponctuels .....	61
Annexe 5 Script FME modélisation d'objets linéaires.....	63
Annexe 6 Script Python modélisation d'objets urbains .....	65

## Annexe 1 Macro sol

```
1 { FnScanClassifyLow(1,7,99,0.03,0.30,0)
   FnScanClassifyLow(1,7,1,0.03,0.30,0)
   FnScanClassifyLow(1,7,99,0.03,0.10,0)
   FnScanClassifyLow(1,7,1,0.03,0.10,0)
   FnScanClassifyLow(1,7,99,0.03,0.30,0)
2 ← FnScanClassifyGround(1,2,1,5.0,40.00,6.00,0.50,0,5.0,0,2.0,0)
3 ← FnScanClassifyHgtGrd(2,100.0,1,2,-0.010,0.020,0)
4 ← FnScanClassifyIsolated("2",1,100,"2",0.50,0)
5 { FnScanClassifyCtrline("2",13,0,-1,0.00,10.00,0,-1.000,1.000,0,0.00,2.00,0,10,0,0,0,0,0,0)
   FnScanClassifyCtrline("2",13,0,1,0.00,10.00,0,-1.000,1.000,0,0.00,2.00,0,10,0,0,0,0,0,0)
   FnScanClassifyCtrline("1",13,0,-1,0.00,0.20,0,-1.000,1.000,0,0.00,2.00,0,10,0,0,0,0,0,0)
   FnScanClassifyCtrline("1",13,0,1,0.00,0.20,0,-1.000,1.000,0,0.00,2.00,0,10,0,0,0,0,0,0)
```

Etape 1 : Identification et classification des « points bas » dans la classe « low points »

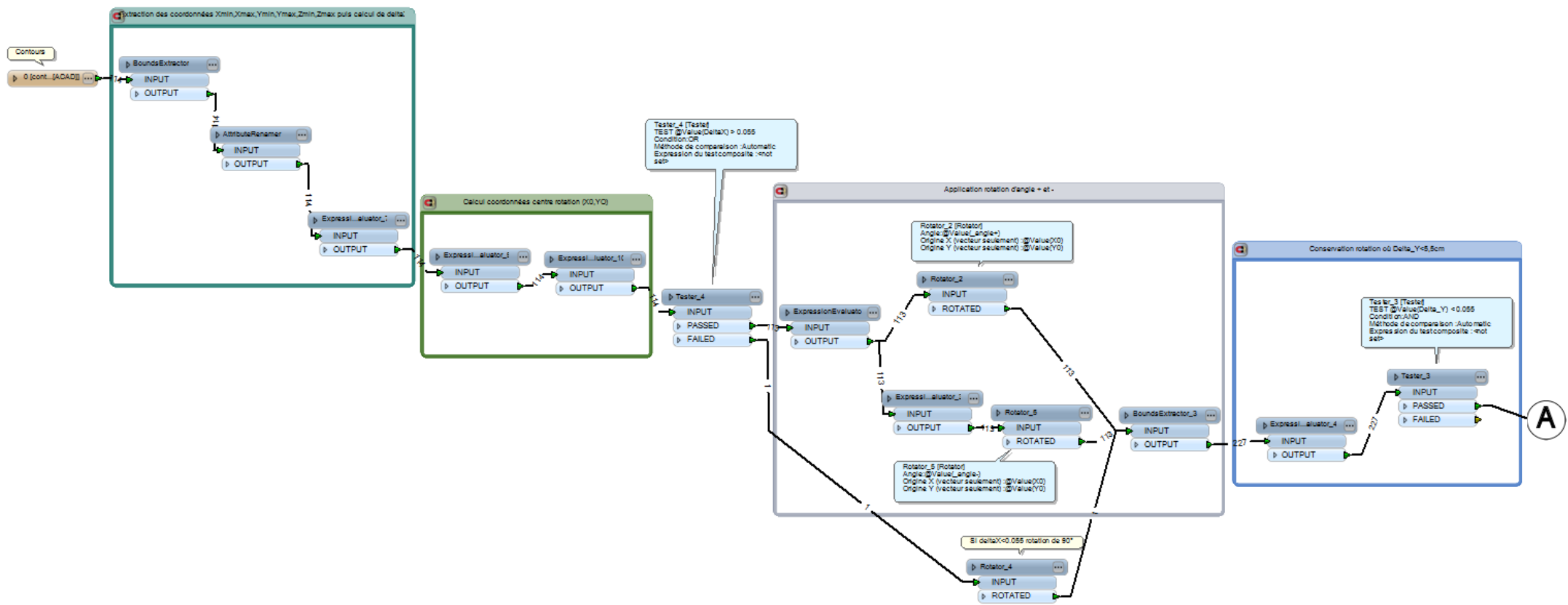
Etape 2 : Application de la routine « ground »

Etape 3 : Classification des points situés à une certaine hauteur du modèle « sol »

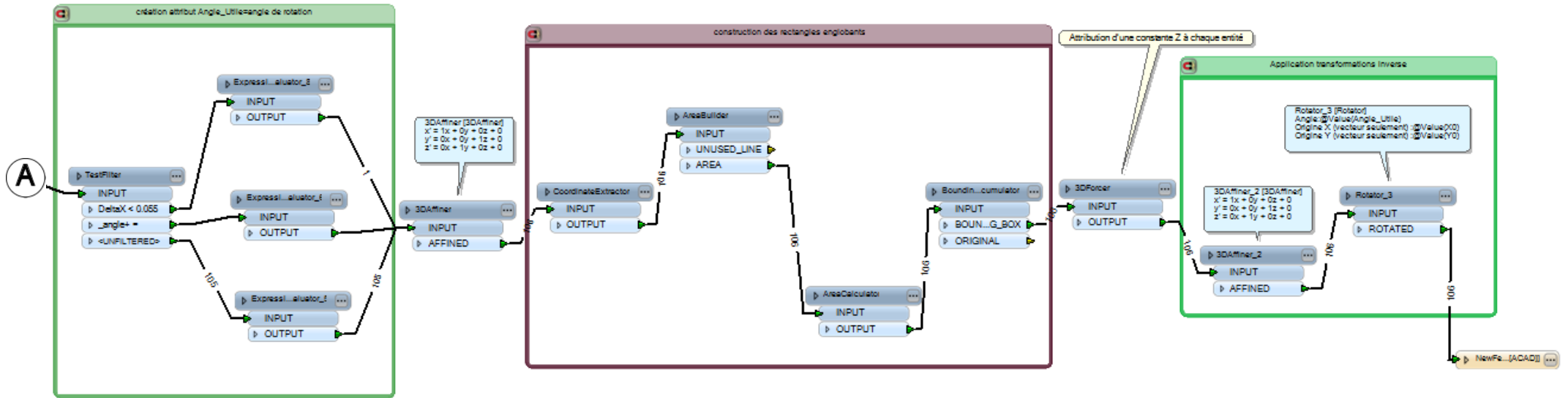
Etape 4 : Identification des points isolés appartenant à la classe sol et classification dans la classe « default »

Etape 5 : Classification des points appartenant à la classe « trottoir »

## Annexe 2 Script FME modélisation de contours

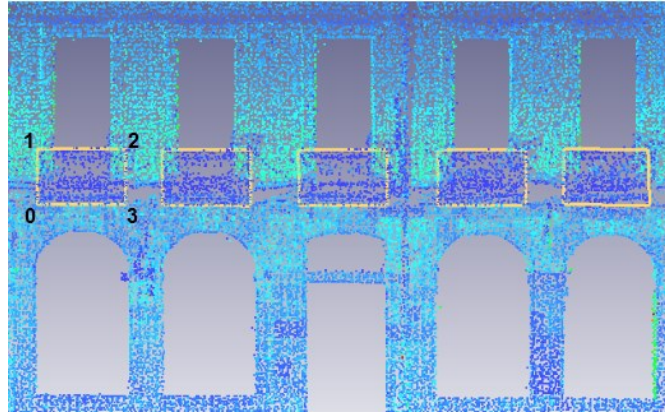




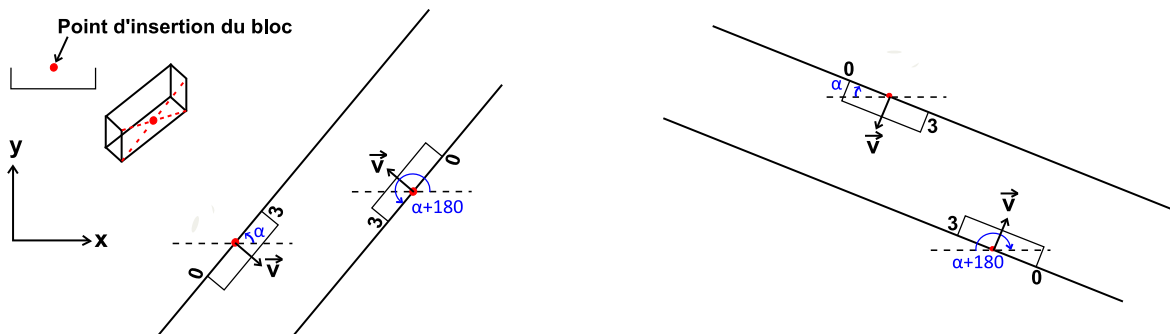


## Annexe 3 Modélisation et insertion automatique des balcons

Digitalisation des balcons sur 3DReshaper à partir du nuage de points plaqué sur le plan moyen de la façade selon un ordre de saisie :



On calcule sous FME le centre (point d'insertion du bloc), la longueur et la largeur de chaque rectangle ainsi que l'angle d'inclinaison par rapport à l'horizontale ( $\alpha = \text{atan} \frac{\Delta Y_{03}}{\Delta X_{03}}$ ) et le vecteur normal au plan de façade ( $\vec{01} \wedge \vec{03}$ ).



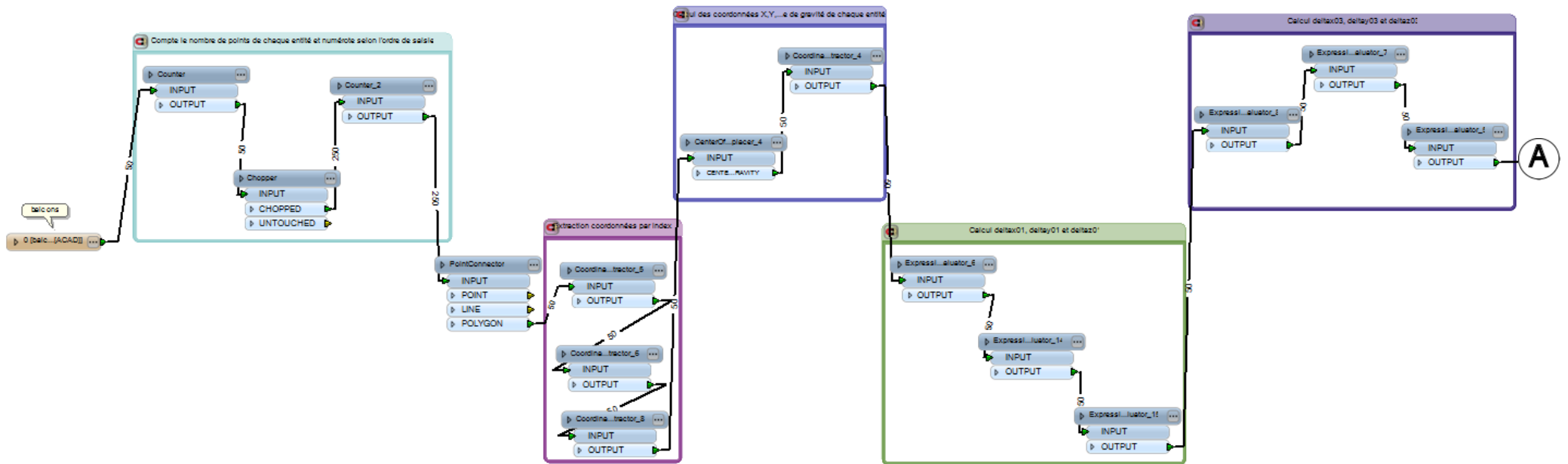
Pour que les balcons s'insèrent du bon côté de la façade nous ajoutons dans certain cas 180 degrés à l'angle  $\alpha$  calculé à partir des points 0 et 3. Nous pouvons remarquer qu'une constante de 180 degrés doit être prise en compte dans la rotation lorsque le signe du vecteur normal au plan de façade est positif en Y. Les coefficients d'échelles en X et en Z sont calculés par rapport aux dimensions du symbole 3D d'origine (bloc). Cependant la profondeur sera la même pour chaque balcon (valeur générale=0.50m).

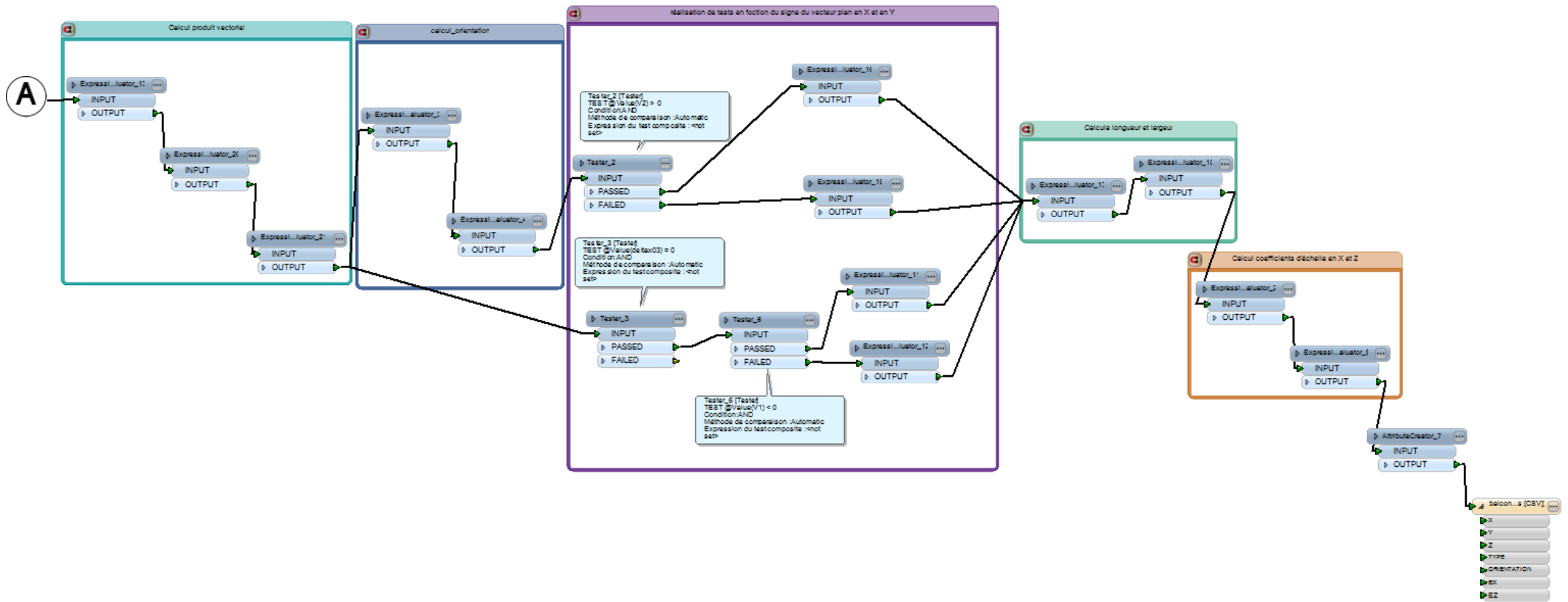
Il faudra cependant traiter le cas particuliers suivants :

- Si  $\Delta X=0$  : Alors si  $V_x < 0$  on ajoute une constante de 180°

Les symboles sont par la suite insérés automatiquement sous Rhinocéros3D et sous forme de cellules 3D à partir d'un script Python.

## Script FME





## Script Python

```
#!/usr/bin/python
# -*- coding:Utf-8 -*-

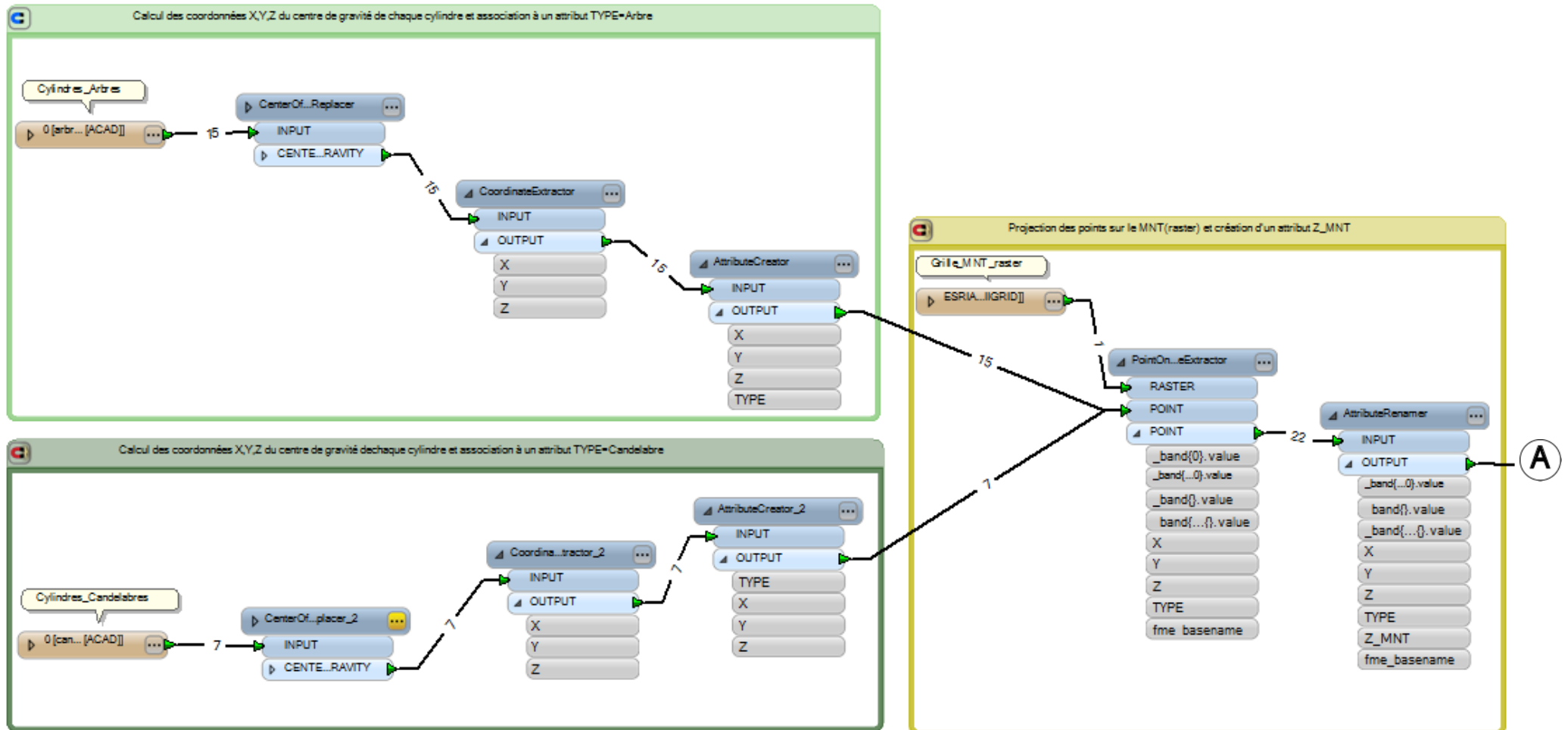
import rhinoscriptsyntax as rs
import math

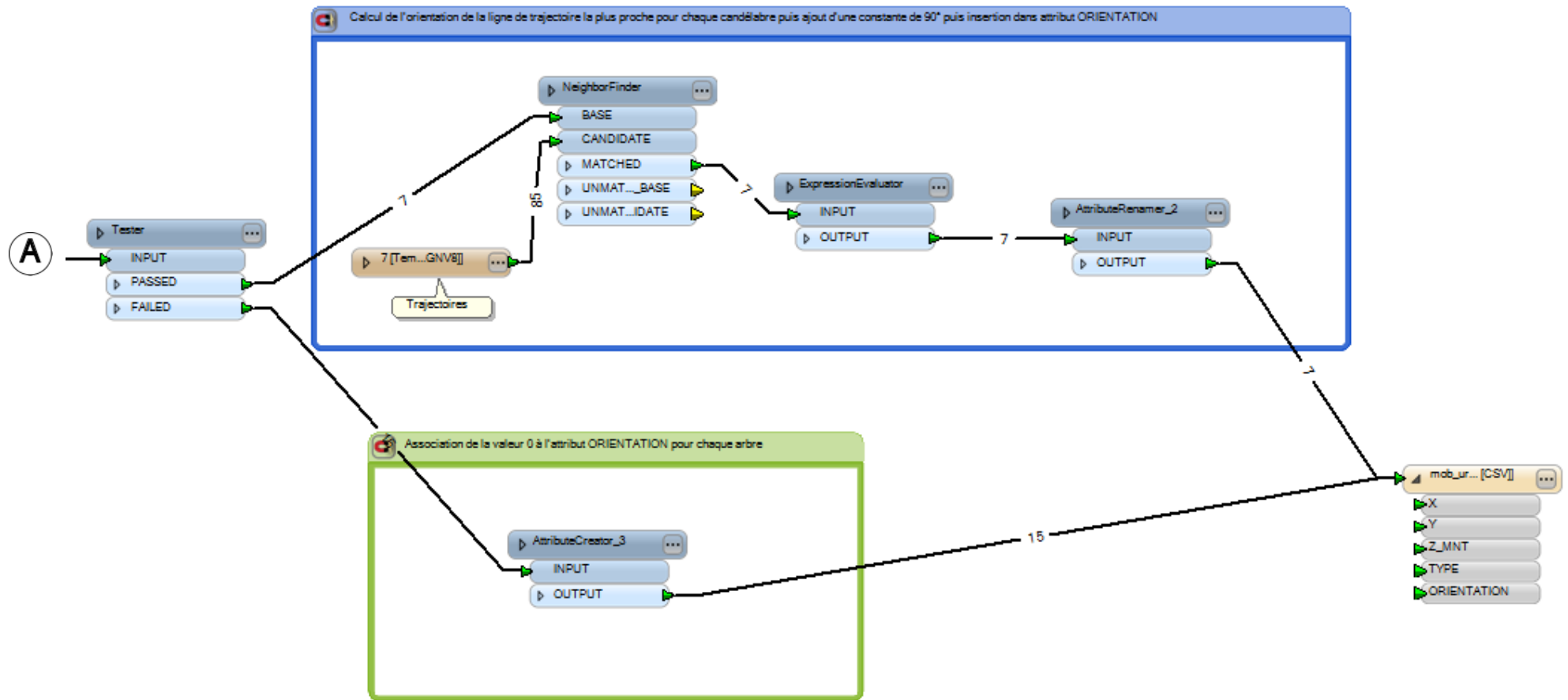
def ImportPoints():
    #demande à l'utilisateur le fichier à importer puis insert les
    différents objets linéaires dans les calques correspondants à partir de
    leurs attributs (X,Y,Z,TYPE,ORIENTATION, ex, ez).
    filter = "Text file (*.csv)|*.csv|All Files (*.*)|*.*||"
    filename = rs.OpenFileName("", filter)
    if not filename: return

    file = open(filename, "r")
    contents = file.readlines()
    N_lines=len(contents)
    for i in range(0,N_lines):
        items = contents[i].split(",")
        if items[3] == "balcon":
            x = float(items[0])
            y = float(items[1])
            z = float(items[2])
            ORIENTATION = float(items[4])
            ex = float(items[5])
            ez = float(items[6])
            rs.AddLayer ("Balcons", color=0, visible=True, locked=False,
parent=None) #création du calque "Balcon"
            rs.CurrentLayer ("Balcons")
            rs.RotateObject (rs.InsertBlock ("balcon", [x,y,z],
[ex,ez,1], 0, [0,0,1]), [x,y,z], ORIENTATION)
            file.close()
    ImportPoints()

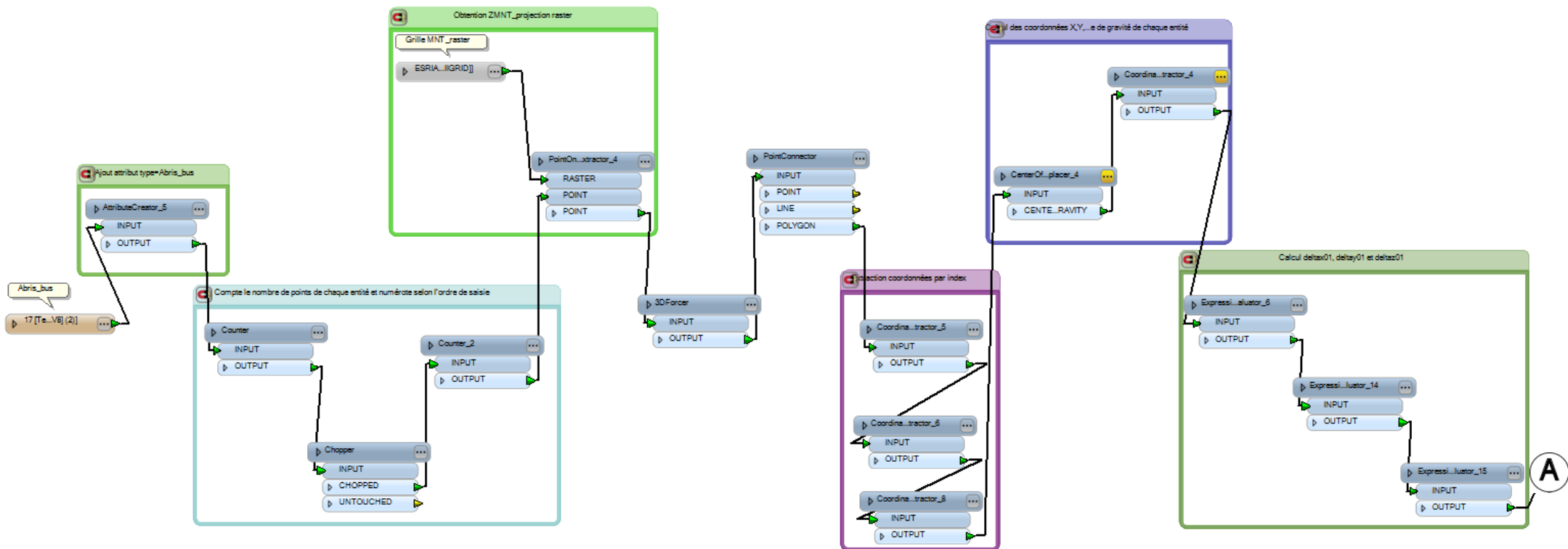
def Point_from_string(line): #définition des caractères présents dans une
ligne du fichier texte
    items = line.split(",")
    return items
Point_from_string("1,2,3,tt,5,6")
```

## Annexe 4 Script FME modélisation d'objets ponctuels

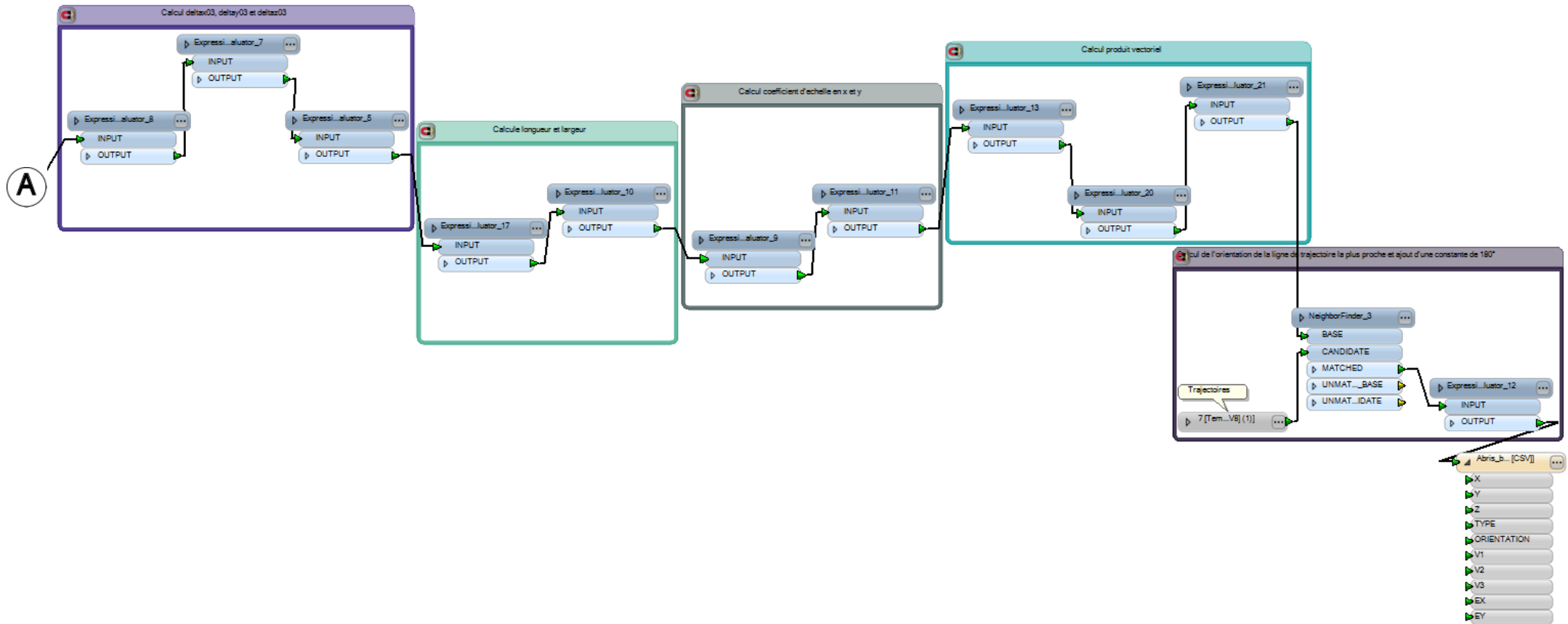




## Annexe 5 Script FME modélisation d'objets linéaires







## Annexe 6

### Script Python modélisation d'objets urbains

```
#!/usr/bin/python
# -*- coding:Utf-8 -*-

import rhinoscriptsyntax as rs #import de la bibliothèque
rhinoscriptsyntax

def ImportPoints():
    #demande à l'utilisateur le fichier à importer puis insert les
    différents objets ponctuels dans les calques correspondants à partir de
    leurs attributs (X,Y,Z,TYPE et ORIENTATION)
    filter = "Text file (*.csv)|*.csv|All Files (*.*)|*.*||"
    filename = rs.OpenFileName("", filter)
    if not filename: return

    file = open(filename, "r")
    contents = file.readlines()
    N_lines=len(contents)
    for i in range(0,N_lines):
        items = contents[i].split(",")
        if items[3] == "Arbre":
            x = float(items[0])
            y = float(items[1])
            z = float(items[2])
            orientation = float(items[4])
            rs.AddLayer ("Arbre", color=0, visible=True, locked=False,
parent=None) #création du calque "Arbre"
            rs.CurrentLayer ("Arbre")
            rs.InsertBlock (#Nom du bloc "Arbre", #point d'insertion
[x,y,z], #échelle [1,1,1], orientation, #vecteur normal [0,0,1])

            elif items[3] == "Candelabre":
                x = float(items[0])
                y = float(items[1])
                z = float(items[2])
                orientation = float(items[4])
                rs.AddLayer ("Candelabre", color=0, visible=True,
locked=False, parent=None)
                rs.CurrentLayer ("Candelabre")
                rs.InsertBlock ("Candelabre", [x,y,z], [1,1,1], orientation,
[0,0,1])
            file.close()
    ImportPoints()

def Point_from_string(line):
    #définition des caractères présents dans une ligne du fichier txt
    items = line.split(",")
    return items
Point_from_string("1,2,3,tt,5")

def ImportPoints():
    #demande à l'utilisateur le fichier à importer puis insert les
    différents objets linéaires dans les calques correspondants à partir de
    leurs attributs (X,Y,Z,TYPE,ORIENTATION,vecteur normal au plan
    (V1,V2,V3)et coefficients d'échelle (ex,ey))
```

```
filter = "Text file (*.csv)|*.csv|All Files (*.*)|*.*||"
filename = rs.OpenFileName("", filter)
if not filename: return

file = open(filename, "r")
contents = file.readlines()
N_lines=len(contents)
for i in range(0,N_lines):
    items = contents[i].split(",")
    if items[3] == "Abris_bus":
        x = float(items[0])
        y = float(items[1])
        z = float(items[2])
        orientation = float(items[4])
        v1 = float(items[5])
        v2 = float(items[6])
        v3 = float(items[7])
        ex = float(items[8])
        ey = float(items[9])
        rs.AddLayer ("Abris_bus", color=0, visible=True,
locked=False, parent=None)
        rs.CurrentLayer ("Abris_bus")
        rs.RotateObject (rs.InsertBlock ("Abris_bus", [x,y,z],
[ex,ey,1], 0, [v1,v2,v3]), [x,y,z], orientation)
        file.close()
ImportPoints ()

def Point_from_string(line):
    items = line.split(",")
    return items
Point_from_string("1,2,3,tt,5,6,7,8,9")
```

## Liste des figures

Figure 1: Photographie de la ville de Paris [GoogleEarth].....	6
Figure 2: Réaménagement de l'îlot "Jules Vallés" dans la ville du Havre [Banaszak et Koehl., 2008] .....	6
Figure 3: GPS utilisant de la donnée cartographique 3D [www.navigadget.com] .....	6
Figure 4: Simulation 3D d'une inondation à la confluence de la Seine et la Marne [Xavier Opigez IAU-îdF, 2013].....	7
Figure 5: Relevé d'une ouverture à partir d'un scanner laser terrestre (à gauche) et au tachéomètre sans réflecteur (à droite) [Landes et Grussenmeyer., 2011] .....	10
Figure 6: Tableau des caractéristiques des différents systèmes de cartographie mobile existants. [@3DLM, 2014], [@Leica, 2014], [@Optech, 2014], [@Riegl, 2014]. Le système utilisé dans le cadre de ce TFE est le StreetMapper de 3DLM.....	11
Figure 7: Valeurs de la précision d'élévation en cm pour les différents scanners laser mobiles utilisés. La colonne de gauche pour un sens de marche anti-horaire (CCW) et la colonne de droite pour un sens de marche horaire (CW). [Kartinen et al., 2013].....	12
Figure 8: Valeurs de la précision planimétrique en cm pour les différents systèmes de scanners laser mobiles utilisés dans les deux sens de la marche. [Kartinen et al., 2013].....	12
Figure 9: Différentes étapes réalisées lors du traitement des données .....	13
Figure 10: Points issus d'un balayage laser. [Manandhar et Shibasaki, 2001].....	13
Figure 11: Regroupement de plusieurs points 3D en voxels parallélépipédiques de taille maximale 2r. La taille des voxels est définie par les valeurs extrémales des coordonnées des points voisins rassemblés au sein du voxel. [Aijazi et al., 2012] .....	14
Figure 12: Résultat d'une classification obtenue après création de voxels et segmentation. [Aijazi et al., 2012].....	14
Figure 13: Segmentation automatique de la façade de l'INSA de Strasbourg. Chaque couleur se rapporte à un segment plan. [Boulaasal, 2010] .....	15
Figure 14: Façade segmentée avec l'algorithme de croissance de région. A gauche: Numérisation de la façade par scanner laser terrestre; A droite : résultat de la segmentation. [Pu et al., 2007] .....	15
Figure 15: Exemple de modélisation par maillage-photographie (gauche)-nuage de points (centre)-maillage (droite). [Begriche, 2003] .....	16
Figure 16: Exemple de modélisation géométrique-photographie (gauche)-nuage de points (centre)-modélisation géométrique (droite). [Begriche, 2003].....	17
Figure 17: Principe du 3D ray tracing. [Benitez et Baillard., 2009] .....	18
Figure 18: Formats d'import et d'export pris en charge par 3DReshaper. [@Technodigit, 2014]... ..	19
Figure 19: CityGML: Tableau comparatif des cinq niveaux de détail. [Gröger et al., 2006] .....	21
Figure 20: Visualisation de la ville de Paris à partir de Ubick. [@ubick, 2014].....	22
Figure 21: Maquette 3D de la principauté de Monaco. [@gouv.mc].....	22
Figure 22: Schéma représentant les paramètres angle et distance de la routine « ground ». [@Terrasolid, 2014] .....	24
Figure 23: Interface de la routine "ground". [Terrascan] .....	24
Figure 24: Vue en coupe du nuage de point. Les points de couleur marron représentent les points appartenant à la classe "sol" et les blancs décrivent les points non classés. [Terrascan].....	25
Figure 25: Classification d'un trottoir. Digitalisation du contour extérieur du trottoir à partir d'une polyligne (gauche) et résultat obtenu après l'étape de classification automatique (droite). [Terrascan].....	25
Figure 26: Nuage de points non classifié en vue de dessus (gauche). Nuage de points classifié en vue de dessus (milieu) et vue de profil (droite) avec les points appartenant à la classe sol représentés par la couleur marron. [Terrascan] .....	26
Figure 27: Visualisation des points indésirables (rouge) à partir de deux vues en perspective. [3DReshaper].....	27
Figure 28: Fenêtre de l'outil "Filtrer/Exploser Nuage(s)" dans 3DReshaper .....	28
Figure 29: Nuage de point initiale (gauche) et nuage de point segmenté (droite). [3DReshaper] ...	29

Figure 30: Aperçu de la segmentation réalisée sur un ensemble de points excepté la classe « sol » (à gauche) et à 20 centimètres au-dessus de la classe « sol » (à droite). [3DReshaper]	30
Figure 31: Nuage de point nettoyé et classifié. [3DReshaper]	30
Figure 32: Limite de la segmentation automatique. [3DReshaper]	31
Figure 33: Réduction du bruit (points de couleur bleu turquoise) pour une valeur d'intensité égale à 20. [3DReshaper]	32
Figure 34: Maillage réalisé avec un échantillonnage régulier de 3 centimètres (gauche) et de 20 centimètres (droite). [3DReshaper]	33
Figure 35: Différence entre "raffinage" et "erreur cordale". [@Technodigit, 2014]	33
Figure 36: Maillage obtenu après interpolation (gauche) et maillage retravaillé (droite). [3DReshaper]	34
Figure 37: Nuage de points représentant un balcon (gauche) et maillage obtenu (droite). [3DReshaper]	34
Figure 38: Calcul de la normale permettant d'extraire le meilleur plan à partir du nuage de points. [3DReshaper]	35
Figure 39: Digitalisation des contours sur le nuage de points projeté sur le plan créé. [3DReshaper]	36
Figure 40: Modélisation de façade obtenue après avoir réalisé le maillage sous contrainte et extrudé les contours. Représentation filaire (gauche) et représentation flat (droite). [3DReshaper]	36
Figure 41: Modélisation d'une façade sous SketchUpMake. Représentation filaire (gauche) et représentation ombragée (droite)	37
Figure 42: visualisation du nuage de points segmenté et projeté sur différents plans. [3DReshaper]	38
Figure 43: Méthode de maillage utilisé. [3DReshaper]	39
Figure 44: Contour extrait à partir du maillage initial de 15 centimètres (rouge) et après affinage du maillage (jaune). [3DReshaper]	39
Figure 45: Visualisation des contours digitalisés manuellement sur le nuage de points. [3DReshaper]	39
Figure 46: Visualisation des contours extrait automatiquement (vert) et digitalisés manuellement (rouge). [3DReshaper]	40
Figure 47: Représentation de contours à partir de rectangles englobants digitalisés manuellement. [3DReshaper]	40
Figure 48: Visualisation des contours extrait automatiquement sous FME Universal Viewer	40
Figure 49: Détermination du signe de l'angle à utiliser dans la rotation (vue de dessus)	41
Figure 50: Visualisation des entités après avoir effectué les deux transformations. [FME Universal Viewer]	41
Figure 51: Visualisation des rectangles englobants. [FME Universal Viewer]	42
Figure 52: Visualisation de entités créés après projection sur le maillage. [3DReshaper]	42
Figure 53: Superposition du nuage de point et des contours modélisés automatiquement. [3DReshaper]	43
Figure 54: Représentation d'un contour automatique (gauche). Modification du contour sous 3DReshaper (milieu) et sous SketchUpMake (droite).	44
Figure 55: Modélisation obtenu sous 3DReshaper après maillage sous contrainte (haut), modélisation obtenue sous SketchUpMake (milieu) et modélisation finale obtenue sous Rhinocéros 3D (bas).	44
Figure 56: Etapes présentant le passage d'un nuage de points à un cylindre (de gauche à droite). [3DReshaper]	45
Figure 57: Ordre de saisie choisie pour la digitalisation	46
Figure 58: Orientation des candélabres par rapport aux trajectoires	46
Figure 59: Aperçu des arbres et des candélabres insérés sous Rhinocéros3D	47
Figure 60: Aperçu d'un abri de bus inséré sous Rhinocéros 3D	47

## **Liste des tableaux**

Tableau 1: Matrice de confusion.....	26
Tableau 2: Variation de la distance latérale et longitudinale entre deux points en fonction de la vitesse pour un scanner laser .....	28
Tableau 3: Matrice de confusion.....	29

---

—

## RESUME

**Mots clés:** Lasergrammétrie, systèmes de cartographie mobile, segmentation, classification, modélisation, maillage

Aujourd'hui le besoin en modélisation 3D est de plus en plus important pour répondre à de multiples usages comme la communication ou l'urbanisme. Les systèmes de cartographie mobile se sont développés et permettent d'obtenir de vastes nuages de points. Ces données vont permettre de réaliser la plupart du temps une maquette 3D de type piéton laissant apparaître les éléments de façades (fenêtres, balcons...) et le mobilier urbain (candélabres, arbres...). Dans un premier temps les méthodes de classification et de segmentation seront étudiées pour distinguer le sol (routes et trottoirs), les façades et les éléments du mobilier urbain. Une démarche permettant de modéliser un ensemble de façades de façon précise et légère sera proposée ainsi qu'une méthode d'insertion automatique de cellules 3D sous le logiciel Rhinocéros 3D.

Le but de cette étude étant de vérifier la possibilité d'obtenir un modèle 3D exploitable à partir de données issues de capteurs dynamiques.

---

—

## SUMMARY

**Key words:** Lasergrammetry, mobile mapping systems, segmentation, classification, modelling, mesh

Today the need for 3D modeling is more and more important to meet multiple uses such as communication or urban planning. Mobile mapping systems have been developed and used to get wide point cloud to cover a city. These data will allow us to realize most of the time a 3D model of pedestrian type letting appear facade elements (windows, balconies...) and street furniture (candelabra, trees...). At first, classification and segmentation methods will be studied to separate the ground (roads and sidewalks) from facades and elements of street furniture. A process for modeling accurately and lightweightly a set of facades and a method of automatic insertion of 3D cells in the software Rhinocéros 3D will be implemented.

The purpose of this study was to verify the possibility of obtaining a usable 3D model from data from sensors dynamic.