



HAL
open science

Optimisation et développement d'un processus automatisé de dessin d'un plan d'intérieur, à partir d'un nuage de points 3D classifié

Rémy Sompayrac

► To cite this version:

Rémy Sompayrac. Optimisation et développement d'un processus automatisé de dessin d'un plan d'intérieur, à partir d'un nuage de points 3D classifié. Sciences de l'ingénieur [physics]. 2023. dumas-04412778

HAL Id: dumas-04412778

<https://dumas.ccsd.cnrs.fr/dumas-04412778>

Submitted on 23 Jan 2024

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
ECOLE SUPERIEURE DES GEOMETRES ET TOPOGRAPHES

MEMOIRE

présenté en vue d'obtenir

le DIPLOME D'INGENIEUR CNAM

SPECIALITE : Géomètre et Topographe

par

Rémy SOMPAYRAC

Optimisation et développement d'un processus automatisé de dessin d'un plan
d'intérieur, à partir d'un nuage de points 3D classifié.

Soutenu le 7 Septembre 2023

JURY

Madame Joëlle DUROY	Président du jury
Monsieur Nicolas MIDONNET	Maître de stage
Monsieur Ghyslain FERRE	Enseignant référent

Remerciements

Tout d'abord, je tiens à remercier mon maître de stage et associé de l'entreprise SIRAGUSA, Monsieur MIDONNET, qui m'a accompagné tout au long du projet avec une qualité d'écoute et une bienveillance à mon égard.

En effet, à travers cette période de 5 mois dans ce milieu professionnel qu'est le cabinet de Géomètre-Expert Siragusa à Montpellier, j'ai pu totalement m'investir dans la recherche et l'exploration des différentes solutions à ma problématique dans une atmosphère calme et propice au travail.

Je remercie également M. Ferré, pour son aide et ses remarques constructives concernant le plan de travail et son soutien pour me guider vers la fin de mes recherches.

J'exprime ma reconnaissance à toutes les personnes qui m'ont permis de vivre et de participer à ce stage très enrichissant, tant sur le plan humain que sur le plan professionnel.

Je remercie enfin, de tout mon cœur, ma mère, mon père, mes frères et mes sœurs, qui m'ont toujours soutenu et accompagné dans toutes mes échéances, toutes les étapes de ma vie et spécialement ma mère, qui continuera à veiller spécialement sur moi, de là-haut.

Glossaire

DXF (Drawing Exchange Format) : Le DXF est un format d'échange développé par Autodesk pour assurer l'interopérabilité entre AutoCAD et les autres logiciels de DAO.

PIC : Plan d'Intérieur Compensé

RANSAC (RANdomSAmple Consensus) : Méthode de reconnaissance de formes grâce à des échantillons aléatoires

Segmentation : Segmentation (d'un nuage de points) : subdivision de l'ensemble des points 3D en sous-ensembles (sous-nuages de points) homogènes, suivant des critères prédéfinis [19]

SLAM : Algorithmes intégrés à un robot pour lui permettre de créer une carte de son environnement et de s'y repérer au même instant.

[19] Tania LANDES et al, 2011.

Table des matières

I	ETAT DE L'ART	7
I.1	ACQUISITION DE LA DONNEE.....	7
I.1.1	Levé traditionnel.....	7
I.1.2	Levé à la station totale	7
I.1.3	Levé par Scanner 3D.....	8
I.1.3.1	Scanner fixe.....	8
I.1.3.2	Scanner mobile.....	9
I.2	DONNEES ET INTERPRETATION, LE NUAGE DE POINTS 3D.....	10
I.3	APPROCHES EXISTANTES DE DESSIN AUTOMATISE OU SEMI-AUTOMATISE DE PLAN D'INTERIEUR .	11
I.3.1	Applications semi-automatisées engendrant le plan d'intérieur.....	11
I.3.1.1	Plan d'Intérieur Compensé (PIC)	12
I.3.1.2	Logiciel Build2map.....	12
I.3.1.3	TopoDot	12
I.3.2	Travaux universitaires et recherches sur la reconnaissance de formes	12
I.3.2.1	RANSAC (RANDOM SAmple Consensus).....	14
I.3.2.2	Transformée de Hough.....	16
II	DEMARCHE MISE EN ŒUVRE.....	18
II.1	CONTEXTE DE L'ETUDE	19
II.1.1	Choix de l'environnement de travail.....	19
II.1.2	Langage de programmation	19
II.1.2.1	Bibliothèques et modules utilisés	19
II.1.3	Mise en place d'une méthode de modélisation des éléments	20
II.1.4	Données fournies en amont.....	21
II.2	DEVELOPPEMENT DE LA SOLUTION.....	22
II.2.1	Réalisation de la coupe du nuage de points	23
II.2.2	Opérations de prétraitement de la coupe.....	25
II.2.3	Reconnaissance de la géométrie des murs	26
II.2.3.1	Test de l'algorithme de RANSAC.....	26
II.2.3.2	Mise en place de la Transformée de Hough	27
II.2.3.3	Reconnaissance des coins et des points de contour	30
II.2.3.4	Implémentation d'un algorithme de regroupement (Clustering)	31
II.2.3.5	Formation des lignes de murs à partir des segments alignés	33
III	ANALYSE DES RESULTATS.....	35
III.1	ANALYSE TEMPORELLE	35
III.2	ANALYSE DES PRECISIONS AU LONG DU PROCESSUS.....	35
III.2.1	Détection des droites à l'aide de la Transformée de Hough.....	36
III.2.2	Extraction des coins et points de contour.....	37
III.2.3	Segments issus des résultats précédents et du Clustering DBSCAN	38
III.2.4	Fusion des segments alignés	39
III.3	AMELIORATIONS FUTURES POSSIBLES	40
III.3.1	Intégration des contraintes associées au plan d'intérieur	40
III.3.1.1	Perpendicularité et parallélisme des murs	40
III.3.1.2	Contraintes de construction standard.....	41
	Conclusion.....	42
	Références bibliographiques.....	43
	Liste des figures.....	47
	Liste des tableaux	47

Introduction

Les subdivisions d'immeubles : mise en copropriété, division en volumes, les projets architecturaux, les certification Loi Carrez ou diagnostic immobilier, représentent une part importante des missions d'un cabinet de Géomètre-Expert. La réalisation de plans d'intérieur est indispensable afin de mener à bien ces missions.

Le plan d'intérieur permet de représenter en deux dimensions les éléments structurels du bien immobilier, à savoir, la géométrie des pièces, les murs et leurs épaisseurs, les ouvertures, éventuellement les hauteurs sous-plafond, et d'autres données numériques. Ils sont représentés soit à l'échelle du 1/100^e, soit à l'échelle plus grande du 1/50^e de manière à faire apparaître plus de détails. Il est composé de plusieurs éléments dont la désignation et la surface des pièces, avec des cotes périmétriques par pièce. Le plan réalisé est coté, à l'échelle et orienté.

De plus, depuis le 29 Juillet 2022, en accord avec un arrêt de la Cour de Cassation, les plans annexés aux actes de copropriété sont réaffirmés comme relevant du monopole des géomètres-experts. le géomètre-expert est ainsi le professionnel habilité à établir les documents technique et juridique relatifs à la création et la gestion de la copropriété dont les plans d'intérieur relatifs à la création de la copropriété[28] .

Cependant, le traitement des données collectées à la conception du plan peut représenter un temps important en raison du volume des données et du processus de traitement. Par conséquent, j'ai été missionné par le cabinet SIRAGUSA, situé à Montpellier dans le but de générer, à partir d'un nuage de points 3D classifié, un plan d'intérieur 2D de manière automatique. L'avantage étant de mettre fin aux tâches répétitives et au traitement trop long.

En 2021, l'entreprise a été missionnée par l'agglomération et la ville de Montpellier pour réaliser les plans d'intérieurs de leurs infrastructures publiques. La perspective d'un

[28] La copropriété est définie, selon la loi du 10 Juillet 1965, comme « tout immeuble bâti ou groupe d'immeubles bâtis dont la propriété est répartie, entre plusieurs personnes, par lots comprenant chacun une partie privative et une quote-part de parties communes, lesquelles sont indissociables »

processus d'automatisation de création de plans d'intérieur, à partir d'un nuage de points est ainsi née, face aux importants travaux à réaliser et dans un souci de productivité.

Des travaux universitaires et scientifiques ont été menés afin de pouvoir assister au mieux l'utilisateur dans le traitement du nuage de points brut, issu de l'acquisition et de créer un processus semi-automatique, voire automatique, dans l'idéal.

Ce TFE s'inscrit donc à la suite de travaux universitaires réalisés sur le sujet et se base notamment sur le TFE de Victorien MALLEBAY, qui a développé un outil de classification automatique de nuages de points, au sein du cabinet SIRAGUSA et qui représente la base du travail réalisé.

Son travail sera ainsi poursuivi en vue d'implémenter un processus automatique de création de plan d'intérieur à partir d'un nuage de points classifié ainsi que la détection de certains éléments non abordés par certains auteurs de travaux précédents et notamment les ouvertures et hauteur ainsi que d'autres éléments apparents tels que des poutres, charpentes ou placards cloisonnés. L'objectif est que tout cela soit réalisé dans le respect de contraintes liées aux standards de construction (parallélisme des 2 côtés d'un mur, standards de dimensions des différents éléments...)

Le travail précédent a abouti à la réalisation de différents algorithmes, notamment des programmes d'entraînement de modèles, un algorithme de classification, un code pour l'interface et un script produisant un DXF.

Il conviendra de créer un programme, qui, à partir de la classification du nuage de points, créera un plan d'intérieur 2D, de manière automatique.

Ce rapport fera l'objet de 3 parties. Nous ferons tout d'abord un Etat de l'art concernant les données nécessaires à la génération du plan d'intérieur, les solutions semi-automatisées concernant le dessin du plan d'intérieur et les recherches universitaires et scientifiques menées sur le sujet, à travers les méthodes de reconnaissances de forme.

Nous aborderons ensuite la démarche mise en œuvre afin de proposer une nouvelle solution innovante et efficace. Nous aborderons les choix de paramètres et résultats, en détaillant la partie de création du dessin du plan d'intérieur 2D sous Autocad,

Pour terminer, nous conclurons quant aux résultats obtenus en proposant les améliorations possibles et les perspectives d'évolution concernant la méthode produite.

I ETAT DE L'ART

I.1 Acquisition de la donnée

Diverses méthodes de relevés permettent de concevoir les plans d'intérieur. Selon le type de relevé, nous pouvons collecter des données variées. Un travail universitaire qui s'intitule « Etude comparative des différentes méthodes du levé d'intérieur d'un bâtiment en vue de réhabilitation et analyse des besoins et des moyens d'un architecte dans le but de produire une modélisation 3D optimale »[9] a été réalisé et répertorie les différentes techniques et procédés d'acquisition de mesures en vue de la création du plan d'intérieur. On y reconnaît ainsi la méthode traditionnelle, la méthode par tachéomètre et enfin, celle du scanner 3D. L'étude en question a été réalisée afin de comparer ces différentes méthodes et il est intéressant de la prendre en compte ici, même si nous traitons les différentes solutions pour la réalisation d'un plan d'intérieur de manière descriptive.

I.1.1 Levé traditionnel

La méthode traditionnelle de lever d'intérieur consiste en la prise de cotes périmétriques d'une pièce à l'aide d'un distancemètre ou éventuellement d'un mètre. On relève ainsi les différentes distances, les épaisseurs des murs, les positions et largeurs des ouvertures et les décrochements. Ce travail peut se révéler long et fastidieux. De plus, cela nécessite le dessin d'un croquis. Dans la plupart des cas, le levé d'intérieur mène à la création ou à la modification des documents constitutifs d'une copropriété et requiert, de ce fait, de nombreuses mesures. Un levé traditionnel peut être source de nombreuses fautes liées à la notation des différents mesurages, à la réalisation du croquis et également à l'oubli de certaines mesures de distances.

I.1.2 Levé à la station totale

Les levés d'intérieur peuvent également faire l'objet d'un levé tachéométrique en effectuant des stations dans chacune des pièces. Ainsi, de nombreuses stations sont nécessaires ; ce qui implique une perte de temps considérable par rapport à un levé d'intérieur traditionnel, en considérant que la mise en station de l'appareil peut prendre

[9]Stéphane FERNANDEZ-CUIREZ, 2013

dans la pratique moins de 5 minutes. Avec l'utilisation, d'une araignée et d'un trépied spécial, ce temps peut être réduit. Cette méthode n'est pas souvent utilisée, surtout quand il s'agit d'un levé de petits espaces intérieurs, tels que des appartements par exemple.

I.1.3 Levé par Scanner 3D

Au cours de ces dernières années, les technologies ont beaucoup évolué avec l'essor de l'utilisation des scanners 3D, l'acquisition des données a été simplifiée tout en permettant de passer d'un levé traditionnel, potentiellement source d'erreurs de retranscription à la capture d'un très grand nombre de points dans un espace en 3 dimensions, un nuage de points 3D.



Figure 1 Scanner fixe RTC360 [31]

I.1.3.1 Scanner fixe

Le scanner 3D est un appareil de mesure qui détermine la position d'une très grande quantité de points en 3D, dans un référentiel spécifique, celui de l'instrument de mesure. Concrètement, est émis un rayon laser qui, au contact de l'objet, est réfléchi vers le scanner. La méthode utilisée par ce scanner pour mesurer une distance est le décalage de phase entre le rayon émis et le rayon reçu [6]. Il permet la création d'un plan « tel que construit » à une précision directement dépendante de paramètres d'entrée comme le pas ou la résolution spatiale par exemple.

[31]Leica-geosystems, Leica RTC360 Solution de capture de la réalité 3D

[6]Gabriel BERGE, 2020

Le scanner 3d est apparu chez les géomètres au début des années 2000, permettant une très grande acquisition de données en un temps très restreint. Il n'a cessé de se développer ces dernières années et aujourd'hui, il fait partie intégrante du métier de géomètre. Le Scanner statique, à la manière du tachéomètre, nécessite une mise en station, en rendant vertical l'axe principal [29]. Cela permet de limiter les traitements sur le nuage de points a posteriori.

I.1.3.2 Scanner mobile

Plus récemment, de nouveaux appareils scanners laser mobiles ont été commercialisés. Ces instruments de mesure fonctionnent notamment à l'aide de la technologie SLAM : « Simultaneous Localisation And Mapping » (localisation et cartographie simultanées). Par leur taille réduite et leur légèreté, les scanners 3d mobiles sont utilisables dans tous les environnements, particulièrement les endroits complexes et fermés, et facilement transportables [8].

Cette technologie est composée d'une centrale inertielle ainsi que d'un scanner laser. La fusion de ces deux capteurs permet d'optimiser son fonctionnement. Le ZEB-REVO est un exemple de scanner mobile à main commercialisé par la société Geoslam [6].



Figure 2 Geoslam Zeb-REVO RT [6]

[29]aftopo, Lexique : 3.2 Calage, mise en station

[8]Nicolas DESCHATRETTE, 2017

[6]Gabriel BERGE, 2020

Il est constitué d'une part d'un scanner laser, dont le fonctionnement est semblable au scanner fixe et d'une centrale inertielle, couplée à un ordinateur qui permettent, grâce à la technologie SLAM de se repérer dans le nuage de points généré en temps réel. Un tel instrument de mesure enregistre environ 43 200 points/s ; ce qui est considérablement inférieur au nombre de points mesurés avec un scanner fixe. Sa portée peut aller jusqu'à 30 m en intérieur et 15 m en extérieur, même si en pratique, on passe près des murs.

I.2 Données et interprétation, le nuage de points 3D

Le scanner 3D est composé d'un télémètre laser pour transmettre et acquérir un faisceau, d'un ou plusieurs composants de déviation du faisceau laser, de systèmes de mesure des informations de distance et de direction et d'une partie pour traiter et enregistrer les informations transmises.

Les éléments de déviation assurent respectivement les mouvements horizontaux et verticaux des miroirs du scanner. Avant de passer à la phase de traitement, un premier produit est formé sous forme de matrice dont chaque point est positionné selon ses angles de vue (φ , Θ) et sa distance par rapport au scanner. La partie traitement consiste à extraire les coordonnées cartésiennes 3D (X , Y , Z) de chaque point en fonction de ses coordonnées sphériques mesurées (r , φ , Θ). Cela est visible sur la figure 1.

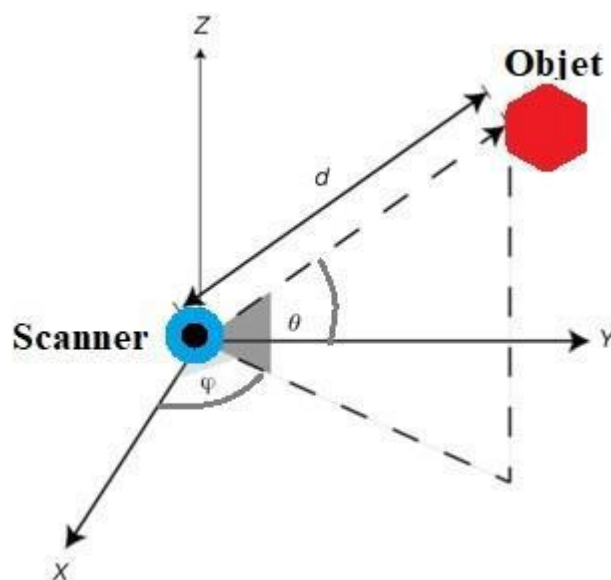


Figure 3 Coordonnées sphériques d'un scanner laser terrestre

Un nuage de points, même brut, permet déjà une multitude d'usages. En effet, par l'intermédiaire de logiciels spécialisés (comme Cyclone, 3DR, Scene ou Realworks), un nuage de points est une représentation complète et précise du bâti dans laquelle rechercher

des informations [30]. Par informations, nous faisons ici référence aux inspections (recherche de détails, analyses de surfaces, etc.) pouvant être menées dans une scène 3D, ou à la possibilité de prendre des côtes, surfaces et dimensions dans celle-ci. Un exemple ci-dessous montre ce à quoi ressemble un nuage de points 3D brut (figure 2).



Figure 4 Nuage de points 3d brut [22]

I.3 Approches existantes de dessin automatisé ou semi-automatisé de plan d'intérieur

Dans la littérature, il existe de nombreuses approches de dessin automatisé ou semi-automatisé de dessin d'intérieur. En effet, de nombreuses solutions commerciales sont présentes sur le marché. Et ces dernières se basent sur des outils de reconnaissance de formes pour la plupart. Nous étudierons donc le principe de fonctionnement de ces dernières.

I.3.1 Applications semi-automatisées engendrant le plan d'intérieur

Nous avons à notre disposition des moyens semi-automatiques de réaliser un plan d'intérieur. Ce sont des solutions optimisées pour le métier qui permettent un gain de temps car le plan est réalisé directement sur le terrain.

[30]My digital building, Générer et exploiter le nuage de points 3D d'un bâtiment

[22] QINGZHAO et al ,2023

I.3.1.1 Plan d'Intérieur Compensé (PIC)

PIC (Plan d'Intérieur Compensé) est un logiciel qui a été lancé en 2007 et développé par la société TopStation. Ce module est implémenté à une tablette sur laquelle on rentre les différentes mesures, avec une compensation des mesures par la méthode des moindres carrés en cours de levé. La limite de cette méthode est la superficie qui ne peut excéder 20 000m² (V. MALLEBAY), ce qui est adapté à la plupart des cas de levé d'intérieur. Les différentes contraintes que le logiciel traite sont les contraintes de distance, de hauteur, d'alignement, d'angle droit, de parallélisme et de même axe.[33]

I.3.1.2 Logiciel Build2map

Une autre solution est le logiciel Build2Map, développé par Sogelink, qui permet un levé rapide, avec une interface qui présente des pictogrammes. Il s'agit d'un programme assez intuitif. Concernant les mesures, il n'est pas nécessaire de prendre les points dans les angles, seuls 2 points dans chaque pan de mur sont requis. Le logiciel effectue ensuite lui-même les intersections de segment. De même, avec les mesures au sol et au plafond, la hauteur sous plafond est déterminée automatiquement.

Ce logiciel semble assez similaire à PIC, avec une compensation multi pièces et paraît être une solution semi-automatisée commerciale satisfaisante.

I.3.1.3 TopoDot

TopoDOT est un logiciel conçu et développé pour convertir des nuages de points 3D en dessins (type DAO) pouvant être directement livrés aux clients. Son objectif principal est d'extraire des données vectorielles à partir de ces nuages. Le logiciel permet d'utiliser des photographies pour faciliter la visualisation des zones à vectoriser [10].

I.3.2 Travaux universitaires et recherches sur la reconnaissance de formes

Bien qu'un processus d'automatisation de dessin de plan d'intérieur ne soit pas encore totalement exploré dans les différents outils commerciaux, il s'agit d'un sujet qui a été développé dans divers mémoires et travaux universitaires. Bien souvent, ces travaux

[33]Jsinfo, Plan d'Intérieur Compensé (PIC)

[10]Thibault NOELLAT, 2019

sont réalisés en vue de produire une maquette BIM et non un plan d'intérieur mais les méthodes en présence sont également les mêmes, notamment concernant la détection d'éléments tels que les murs en particulier.

Néanmoins, étant donné le fort intérêt des cabinets de Géomètres Experts, certains mémoires relatifs au sujet sont confidentiels pendant une période variant entre 2 et 10 ans. Il est important de noter la curiosité et l'attention légitimes suscitées par ces recherches.

Victorien Mallebay a réalisé son projet de Fin d'Etudes dans le cabinet Siragusa en créant un processus de classification automatique d'un nuage de points. Il a notamment obtenu de très bons résultats de classification avec son programme d'Intelligence Artificielle. Le premier travail effectué par V. MALLEBAY a été celui de la segmentation sémantique des nuages de points de manière manuelle en associant une classe à chaque région de points segmentée. Ces nuages de points classifiés ont constitué un jeu de données d'entraînement pour le programme d'Intelligence Artificielle utilisé, à savoir l'algorithme Random Forest. Il s'agit d'une architecture de Machine Learning, une branche de l'IA. L'intelligence artificielle est décrite par Chollet [15] comme « l'effort d'automatisation des tâches intellectuelles normalement effectuées par des humains ». Afin d'automatiser la détection et la classification des différents éléments du nuage de points, en commençant par les murs, l'IA est une solution qui s'est imposée. Une grande partie du travail a donc été de créer une classification automatique du nuage de points avec Random Forest.

Par la suite, un programme primaire a été développé concernant le dessin des murs à partir du nuage de points classifié. Cette première ébauche a permis d'identifier des points clés du dessin de plan 2D à partir d'un nuage de points, tels que les bibliothèques nécessaires ou encore le besoin d'un algorithme tel que RANSAC afin de déterminer les plans de murs par exemple. Cependant, le programme en tant que tel est très loin d'être fonctionnel.

Les murs dessinés sont ajustés avec la méthode des moindres carrés non linéaires avec une régression linéaire pour le 2D et une régression plane pour le 3D.

[15]François CHOLLET, 2017

Dans ses travaux de recherche, Hélène Macher [5] distingue trois méthodes de segmentation, régulièrement utilisées pour déterminer les modèles mathématiques de formes spécifiques (comme par exemple des plans, des lignes, des cercles...). Nous retrouvons ainsi RANSAC, la Transformée de Hough et enfin la Croissance de régions.

Ces deux premières sont régulièrement implémentées dans les travaux de recherche pour la reconnaissance de formes géométriques. Nous allons donc les présenter ci-dessous.

I.3.2.1 RANSAC (RANdom SAmple Consensus)

RANSAC [18] est une méthode robuste itérative qui est utilisée pour estimer les paramètres d'un modèle mathématique que ce soit une droite, un plan un cercle par exemple, à partir d'un ensemble de points considérés, qui contient des outliers et du bruit. Les points pertinents par rapport au modèle sont détectés s'ils respectent un seuil de distance minimale au modèle. Ces derniers sont appelés « inliers » et les points aberrants sont nommés « outliers ». Par itérations successives, le meilleur modèle est estimé lorsque le nombre minimal d'inliers est atteint.

Les paramètres de RANSAC sont : un seuil t définissant la distance maximale entre les points et le modèle, un nombre d'itérations maximum N et un seuil S donnant le nombre de points minimal devant être associé au résultat.

L'algorithme de base de RANSAC peut être résumé comme suit , selon Hélène Macher[5] .

1. Sélectionner aléatoirement le minimum de points requis pour déterminer les paramètres du modèle.
2. Calculer les paramètres du modèle.
3. Déterminer l'ensemble consensus associé au modèle, qui est l'ensemble des données vérifiant le modèle en considérant le seuil t .

[5]Hélène MACHER, 2017

[18]FISCHER et BOLLES, 1981

Si le nombre de données associées au modèle de l'ensemble consensus, également appelé *score*, dépasse un seuil S , réestimer les paramètres du modèle¹⁶ en utilisant tous les *inliers* et terminer.

4. Sinon répéter les étapes 1 à 4 (maximum de N itérations). Si les itérations arrivent à terme sans avoir atteint le score minimal, le résultat sera le modèle qui aura le meilleur score.

La figure 5 aide également à mieux comprendre RANSAC afin de trouver la meilleure droite dans le nuage de points (rouge), de manière itérative, à partir de deux points aléatoires (en vert). Nous voyons les points inliers (bleu) situés dans la distance seuil par rapport à la droite.

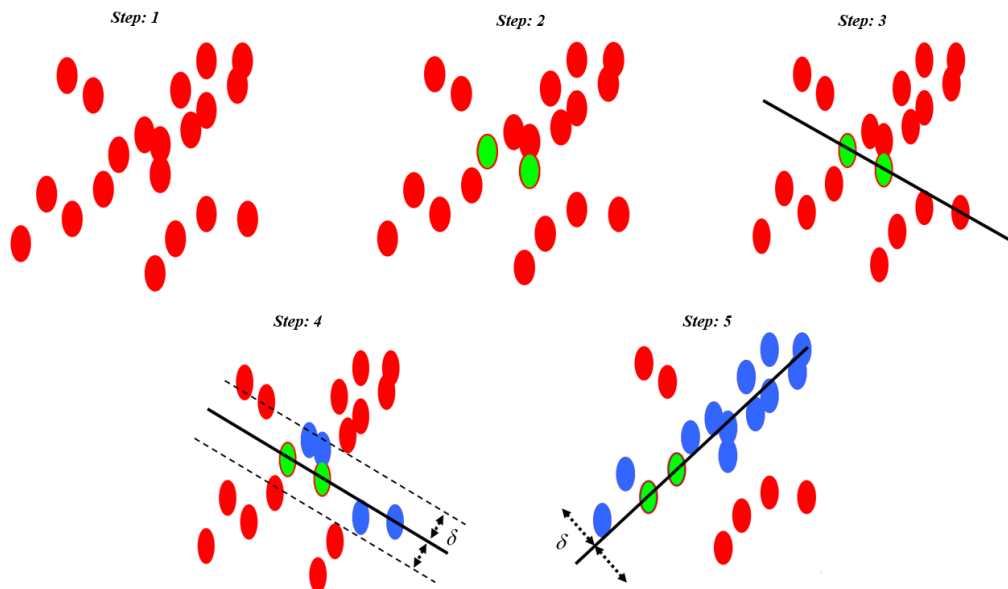


Figure 5 Algorithme de RANSAC pour trouver la meilleure droite dans un nuage de points bruité [27]

Par exemple, Hakim Boulaassal [2] a réalisé sa thèse sur la vectorisation automatique d'un plan de façade et il effectue une recherche des paramètres des droites sur la façade à l'aide de RANSAC. Ses résultats sont concluants.

Les paramètres jouent un rôle primordial dans le paradigme de RANSAC. Selon Hélène Macher[5] l'algorithme de RANSAC a démontré à quel point il est robuste et efficace

¹⁶Si le modèle à estimer est une droite, on peut réajuster les paramètres de la droite obtenue avec une régression linéaire, en utilisant tous les points inliers. Si le modèle est un plan, une régression plane est un moyen de réestimer le plan obtenu avec les inliers.

[27] Ransac algorithm,2020

concernant le temps de traitement et la gestion du bruit[24] mais il peut cependant être amélioré. L'un des problèmes avec RANSAC est qu'un mauvais choix des paramètres a un impact important sur le résultat. Par exemple si le seuil t est trop élevé, plusieurs solutions auront la même valeur pour C ; ce qui entraînera des estimations déplorables. D'un autre côté, si le seuil t choisi est trop bas, cela entraînera des ajustements instables. Dans le but de créer un processus de dessin automatique, RANSAC a été utilisé dans les travaux universitaires de Léa Talec [12] et V. Mallebay, respectivement pour estimer les paramètres des droites et pour évaluer les équations des plans de mur.

I.3.2.2 Transformée de Hough

La transformée de Hough est une technique utilisée pour détecter des formes géométriques dans une image, principalement des lignes, des cercles ou des ellipses. Elle a été développée par Paul Hough en 1962 [16] et est devenue une méthode populaire dans le domaine du traitement d'images et de la vision par ordinateur. Le principe de base de la transformée de Hough repose sur la représentation des formes géométriques dans un espace paramétrique plutôt que dans un espace cartésien traditionnel. Cela permet de simplifier la détection des formes, même en présence de bruit ou de segments de lignes partiellement visibles.

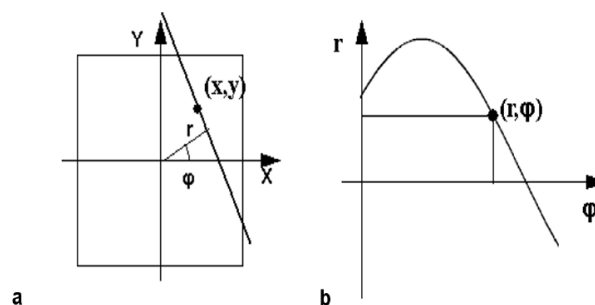


Figure 6 Transformation de Hough et changement de l'espace cartésien à l'espace paramétrique[21]

[2]Hakim BOULAASAL, 2010

[5] Hélène MACHER, 2017

[24]Tarsha_Kurdi et al.,2007

[12]Léa TALEC, 2017

[16]Paul HOUGH, 1962

[21]Jakub Stefan MARKIEWICZ et al, 2015

La procédure de la Transformée de Hough est effectuée sur des espaces de paramètres qui décrivent les formes que l'on veut déterminer ; chaque point augmente la valeur de l'accumulateur (compteur) dans la position correspondant aux paramètres auxquels il pourrait être lié.

Après traitement de tous les points de l'image, les valeurs maximales locales dans l'espace paramétrique décrivent les paramètres probables détectés des formes. Dans le cas le plus simple de la détection de lignes droites, les paramètres sont l'angle d'inclinaison de la ligne et ϕ et la distance par rapport à l'origine du système de coordonnées r , comme le montre la figure 6. Cette dernière nous montre le passage de l'espace cartésien à l'espace paramétrique effectué par la Transformée de Hough.

La Transformée de Hough a été implémentée dans de nombreuses études. LU et al. [20] ont par exemple développé une méthode de détection de segments à partir d'un nuage de points, de manière rapide, à partir d'images 2D. C'est une méthode relativement robuste au bruit et qui permet la détection de formes paramétriques de manière rapide.

Nous avons donc pris en compte ces travaux et les différentes techniques à notre disposition avant de nous embarquer dans l'aspect de la programmation pure, en faisant en sorte d'avoir un plan d'action et une démarche claire, dans les points principaux. Il convient maintenant de plonger au cœur de notre approche et de notre cheminement de pensée tout au long du projet.

[20] LU et al., 2020

II DEMARCHE MISE EN ŒUVRE

Le corps de nos travaux réside dans la création d'un plan d'intérieur 2D à partir de la précédente classification. Nous nous sommes appuyés pour cela sur les recherches menées dans ce domaine, même si ces dernières sont rares ou confidentielles. En effet, la plupart des travaux sur le sujet se sont orientés vers la réalisation d'une maquette BIM et non la production d'un plan d'intérieur en 2D, notamment les travaux de Maxime Renaud [11] ou encore Hélène Macher [5] ou Tania Landes [1]. Plusieurs approches peuvent être développées dans notre cas.

La première solution consiste à détecter les plans des différents éléments issus du nuage de points classifié (murs, fenêtres, portes..) et convertir les plans en droite, en ignorant la composante verticale. Cette solution a été mise en place dans les travaux de [1]. Cependant, cette option est chronophage car elle nécessite l'ensemble du nuage de points ; ce qui sous-entend des millions de points.

La seconde possibilité est de passer par une coupe du nuage de points 3d et d'utiliser cette dernière pour détecter les lignes correspondant à tous les éléments et notamment les murs dans un premier temps.

Par ailleurs, on pourrait également envisager passer par une maquette numérique, support à partir duquel la création du plan d'intérieur par une coupe est facilitée. Cette piste a été explorée car elle est présente dans certaines recherches universitaires comme celle de Camille BARBIN, et autres mais n'a pas été retenue car le travail a paru conséquent, avec de nombreuses étapes à achever en amont et l'incertitude d'arriver à réaliser le projet prévu, la création d'un processus automatique de réalisation de plan 2D.

Dans les travaux réalisés par Victorien Mallebay, une coupe de 1m d'amplitude verticale est réalisée dans le nuage de points, de manière automatique. Il faut ensuite calculer une approximation des équations de plan des différents murs avec RANSAC, puis passer en 2D et établir la régression linéaire des droites obtenues. Après avoir exploré cette piste, j'ai pu me rendre compte de la lourdeur des calculs réalisés.

[11]Maxime RENAUD, 2017

[5]Hélène MACHER, 2017

[1]Tania LANDES, 2020

II.1 Contexte de l'étude

II.1.1 Choix de l'environnement de travail

Le poste informatique sur lequel j'ai réalisé mes travaux de programmation était adapté et efficace. En effet, il a été important d'avoir un processeur performant et beaucoup de mémoire vive car les traitements sur nuages de points sont très gourmands en ressource.

Système	modèle
processeur	Intel(R) Xeon(R) CPU E3-1220 v5
Mémoire vive (RAM)	32,0 Go
Système d'exploitation	Windows 11 - 64 bits

Tableau 1 Caractéristiques du poste de travail utilisé

II.1.2 Langage de programmation

La grande majorité des recherches universitaires utilisent la programmation Python. Nous avons donc choisi ce langage de programmation pour nos travaux.

Une nouvelle piste a toutefois été envisagée : celle de la programmation sous C++, un langage de programmation bas niveau, qui sous-entend une efficacité et une rapidité de calcul impressionnante, comparé à Python, qui est un langage de haut niveau . Etant donné la quantité importante de données et la charge des calculs, cela semblait être une bonne option. Cependant, compte tenu de la durée de ce Projet de Fin d'Etudes et du manque de temps pour se consacrer à l'apprentissage d'un tel langage, je suis finalement resté en Python. Néanmoins, il existe un module nommé Cython, mélange entre le C++ et le Python, qui permet de compiler le code python en C++ et favorise une accélération des calculs de 2 fois, notamment au niveau des boucles et à la fin des recherches, il sera optimal de l'utiliser.

II.1.2.1 Bibliothèques et modules utilisés

modules Python	usage
Ezdx	création d'un fichier dxf
Pyautocad	dessin automatique dans Autocad
Open3d / Matplotlib	visualisation de données 3d /2d
numpy /math/numba/sickit...	calcul et traitements de données

Tableau 2 Bibliothèques utilisées dans notre projet

Il existe des modules sous Python qui permettent la création d'un dessin de manière automatique sous Autocad. La première option se nomme *Pyautocad*. Elle nécessite l'ouverture d'Autocad pour pouvoir fonctionner. Nous avons également la bibliothèque *Ezdx*, une bibliothèque sous Python qui permet la création de fichier .dxf par le biais de la programmation. Cette dernière n'implique pas forcément l'utilisation du logiciel Autocad et a pour but la conception d'un fichier d'échange universel, afin de pouvoir être éditable depuis la plupart des logiciels de modélisation et de conception 3D. C'est donc ce dernier module que j'ai choisi pour le projet, car il offre le plus d'ouverture et de liberté avec la création d'un fichier d'échange standard entre les différentes plateformes de DAO et de CAO.

Open3d est un module de Python optimisé pour la visualisation et le traitement de données 2D/3D. Nous avons également utilisé *Matplotlib* de manière simple, ces deux bibliothèques ont favorisé une meilleure évaluation des effets de chaque expérimentation .

II.1.3 Mise en place d'une méthode de modélisation des éléments

Afin de modéliser les différents éléments, nous nous sommes inspirés d'un plan d'intérieur réalisé par l'entreprise Siragusa, le but étant d'importer des blocs existants sous Autocad ou Covadis ou de créer des blocs le plus proche possible de ces derniers dans le programme Python. Nous avons donc tout d'abord, tenté d'importer des blocs d'Autocad sur Python, avec les bibliothèques *ezdx* ou *pyautocad*.

Ezdx permet de créer ou modifier un fichier DXF (DrawingXchange Format) sans utilisation d'Autocad. *Pyautocad* est une autre solution pour programmer sous Python, de manière à créer un dessin automatique en 2 dimensions lisible par Autocad. Cependant, ce dernier module nécessite le lancement du logiciel Autocad au préalable ; ce qui peut se révéler être un frein pour les utilisateurs n'ayant pas ce logiciel payant, mais un autre logiciel de DAO gratuit comme par exemple *Librecad* ou *Freecad*.

II.1.4 Données fournies en amont

La phase expérimentale qui concerne la mise en place d'un processus automatisé de création de plan d'intérieur 2D à partir d'un nuage de points classifié représente le cœur du projet réalisé.

Tout d'abord, malgré le fait que l'objectif était de se baser sur les travaux précédents de Victorien MALLEBAY, ceux-ci n'étaient pas fonctionnels en pratique. Il a cependant créé une ébauche de code concernant le dessin automatique des murs à partir d'un nuage de points classifié. Mais, même dans le cas où son programme avait marché, les calculs liés à son programme auraient été beaucoup trop longs. Cela est dû au fait que les différents traitements étaient directement effectués dans le nuage de points complet ; ce qui suppose un nombre très important de points à traiter, qui plus est, en 3 dimensions. La première partie reposait sur le principe de l'algorithme de RANSAC et la détection des plans de murs sous la forme d'équations du type :

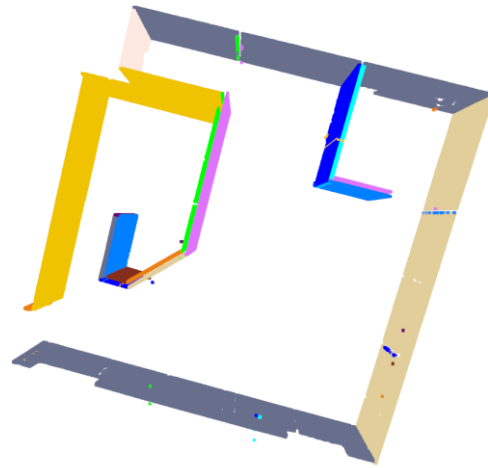
$$ax + by + cz + d = 0$$

J'ai essayé dans un premier temps de faire tourner tous les calculs mais me suis rapidement rendu compte de la limite temporelle engendrée par une telle approche. J'ai donc exploré les différentes options qui s'offraient à moi, pour réaliser le même travail sans avoir à attendre des heures pour un simple traitement sur le nuage de points 3d. Dans un premier temps, ma tâche a été d'essayer de comprendre le programme afin de le rendre apte à tourner mais il manquait apparemment des parties essentielles. Cela a néanmoins été l'occasion, pour moi, de me lancer en m'inspirant de certaines parties de code.

Le jeu de données issu du TFE précédents'appuie sur un nuage de points de 1.79 millions de points, labellisés à la main par ses soins, avec une segmentation et une classification manuelle. Il s'agit d'un travail qui peut se révéler fastidieux et chronophage. Pour cela, à l'aide du logiciel Cloud Compare, il a tout d'abord segmenté les différents éléments tels que le plafond, le sol, les murs afin d'associer respectivement une classe à chaque cluster (région de points 3D). Après, avoir classifié de manière manuelle tous les objets, il a réassemblé le nuage. La classification a pour but d'associer chaque point du nuage à une classe, comme par exemple la classe mur, fenêtre, porte, plafond, colonne, chaise, table, canapé...Le fichier est au format .ply et les points sont intégrés comme dans la figure ci-dessous

Label	Classe
1	Plafond
2	Sol
3	Mur
4	Porte
5	Cadre de porte
6	Fenêtre
8	Façade
9	WC
10	Evier
11	Table
12	Chaise
18	Bruit

(a)



(b)

Figure 7 Listing des classes et labels associés (a) et visualisation des murs dans le nuage de points classifié (b)

A travers nos travaux, nous avons tout d’abord utilisé le nuage de points classifié des travaux de Victorien (voir figure 7). Il sera profitable de le tester sur un nuage de points plus dense par la suite.

Il est intéressant de noter également la présence de jeux de données labellisés et en open source présents comme notamment S3DIS[32], qui a été développé conjointement par les universités de Stanford, Princeton et Munich, qui comporte 1500 scans ou encore ShapeNet Part et Stanford 2D-3D-S, eux aussi développés, dans un même cadre de collaboration.

II.2 Développement de la solution

Dans la perspective de la création d’un outil de création de dessin automatique à partir de nuage de points classifié, le premier objectif a été la détection des murs, véritable socle du plan d’intérieur. Une majeure partie de mes travaux a été consacrée à cela car il s’agit de la partie la plus importante dans un plan d’intérieur. L’approche mise en place est composée, dans les grandes lignes, de la création d’une coupe du nuage de points 3D, du traitement de cette dernière, de la détection des droites et des coins, de la création de

[32]Jeu de données S3DIS, jeu de nuage de points 3D classifié en open source

segments et ensuite, de l'application des contraintes de parallélisme et de perpendicularité. Ensuite, la deuxième partie concerne la détection et le dessin des éléments d'ouvertures (portes, fenêtres...etc) et le calcul de hauteurs (hauteur sous plafond). La figure ci-dessous(figure 8) présente brièvement les grandes étapes de l'approche voulue.

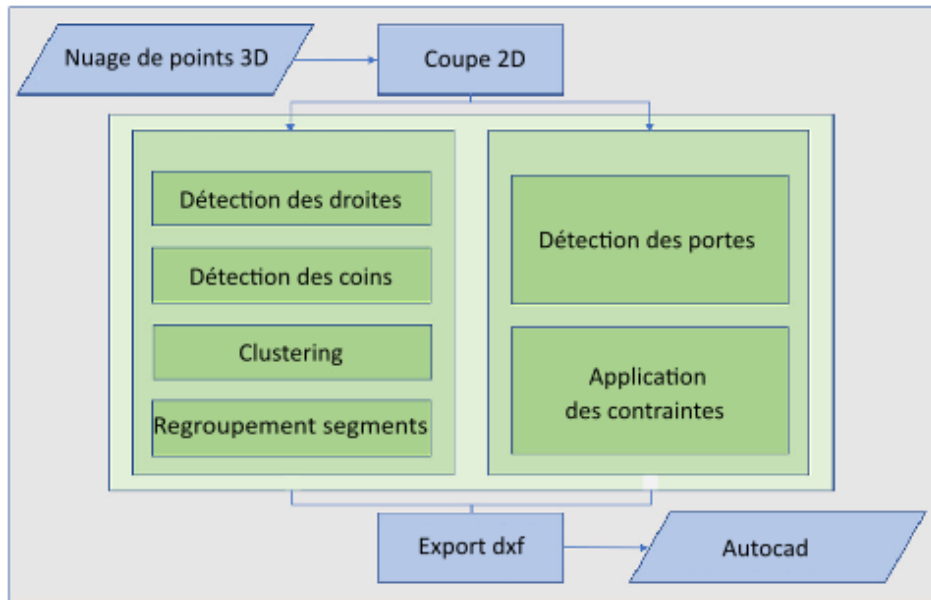


Figure 8 Solution proposée de processus de dessin automatique de plan d'intérieur à partir d'un nuage de points 3D classifié

II.2.1 Réalisation de la coupe du nuage de points

Les points issus du nuage sont entrés au format.ply (polygone). Il existe néanmoins des bibliothèques pour importer des fichiers e57 ou au format .las, fichiers bruts issus du scan, directement sous Python également.

L'idée d'une coupe pourrait sembler, intuitivement, constituer une perte d'information. Cependant, dans le cas d'un nuage de points, la quantité de données est tellement importante qu'une coupe est déjà suffisante pour détecter au mieux des droites 2d associées aux murs, et non des plans. Je me suis donc orienté vers cette solution dans le but d'une optimisation des calculs. Il fallait ensuite trouver un algorithme qui me permette d'approcher au mieux les différentes lignes de murs. Parmi ces approches, on retrouve la Transformée de Hough, la croissance de régions ou encore l'algorithme de détection RANSAC.

Tout d'abord, le nuage de points étant classifié, nous avons récupéré les éléments des différentes classes dont les murs, portes, fenêtres. Ensuite, nous avons effectué une

coupe du nuage de points, en sélectionnant les points dont la composante verticale est comprise entre 1m et 1,01m au-dessus de la hauteur du sol. Cette dernière est calculée à l'aide d'une détection du plan du sol à l'aide de l'algorithme de RANSAC. La coupe est donc l'élément de base sur lequel nous allons effectuer les différents traitements pour parvenir à la détection et au dessin automatique des lignes de murs.

La coupe du nuage de points (visible à la figure 9) nous permet d'une part de nous affranchir d'une grande partie des données du nuage, mais aussi de sélectionner les éléments qui nous intéressent, sans qu'il n'y ait d'éléments qui nous gênent pour la modélisation. Cependant, le rendu graphique n'est pas à confondre avec un plan 2D vectorisé ; la coupe du nuage de points 3D est composée uniquement de points et non de segments.

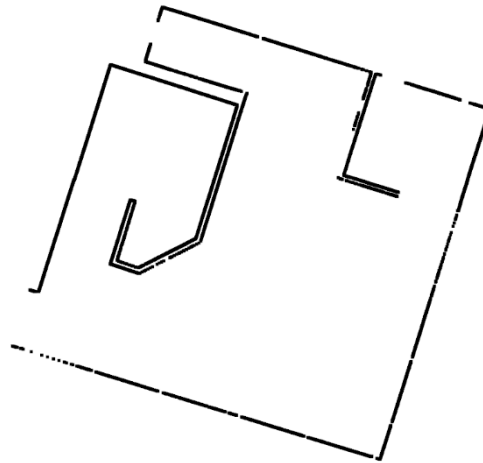


Figure 9 Coupe du nuage de points 3D réalisée sous Python

Il est important de noter une densité de points hétérogène dans la coupe du nuage. Cela provient de plusieurs facteurs dont [4] :

- la distance et l'orientation de la surface par rapport au scanner (dans chaque point de vue),
- les parties scannées dans plusieurs points de vue et d'autres par un seul,
- les occlusions
- la présence de scans spécifiques pour certains détails, éventuellement

[4]Thomas CHAPERON, 2002

II.2.2 Opérations de prétraitement de la coupe

A cette étape du programme, avant la détection des droites, le prétraitement des données a semblé opportun, pour 2 raisons, selon Tania Landes[1]. Son premier rôle est de réduire le bruit et le second est de restreindre le volume de données..

Il a ainsi paru évident d'appliquer un sous-échantillonnage au nuage de points. Cela a nécessité l'approche des k-plus-proches voisins avec la condition que, dans un rayon donné, il n'y ait aucun voisin autour d'un point. Cela nous a permis d'éliminer des points très proches les uns des autres, les doublons et de garder une quantité réduite, mais suffisante, de points,. Cela nous a ainsi permis de passer d'un nuage de points de 1.7M de points à un peu plus de 379000 points.

Le bruit peut se définir comme l'ensemble des points erronés, qui ne "représentent pas la surface de l'objet scanné" [3] pour plusieurs raisons et notamment la précision du scanner. J'ai appliqué différents algorithmes sur la coupe, afin de supprimer le bruit du nuage, dans l'optique de favoriser une bonne détection des modèles de droites. Pour cela, Adrien Hamelin [3] présente différents traitements de réduction de bruit comme le filtre gaussien ou encore l'algorithme de Hou et Al [25] (voir figure 10)

Cette dernière méthode consiste simplement par lisser le nuage de points, de manière à filtrer les hautes fréquences (qui correspondent aux plus grands écarts et donc au bruit). Nous l'avons donc appliquée à notre coupe 2D, dans l'espoir de « supprimer le bruit ».

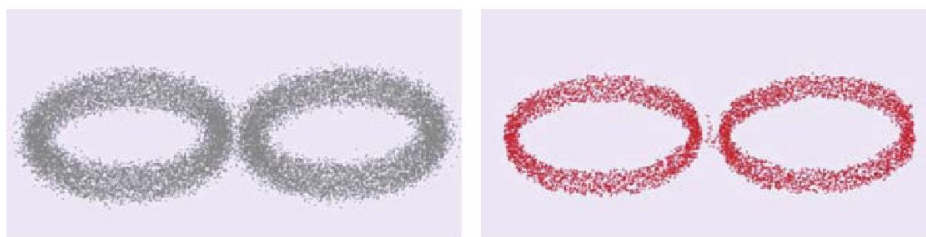


Figure 10 Suppression du bruit après une itération du programme de [25]

[1]Tanya LANDES, 2020
[3] Adrien HAMELIN, 2015
[25] Hou et al., 2012

II.2.3 Reconnaissance de la géométrie des murs

La détection des murs représente ainsi le cœur de mon travail et les différentes étapes que j'ai mises en place pour parvenir au dessin des murs dans un plan en 2 dimensions sont détaillées ci-dessous.

II.2.3.1 Test de l'algorithme de RANSAC

Afin de permettre l'extraction des plans de murs, l'algorithme de RANSAC a été utilisé à partir du nuage de points classifié lors du TFE de V. MALLEBAY. Cette méthode robuste nous a paru être une des meilleures options, et également celle des plus utilisées.

Nous l'avons ainsi appliquée également pour la détection droites dans nos premières expérimentations. Les paramètres de RANSAC sont le nombre d'itérations, le nombre minimal d'inliers, un inlier étant un point qui vérifie la condition de distance par rapport au modèle d'entrée (ici, le modèle défini est une droite). Pour détecter toutes les droites dans un nuage de points 2D, un RANSAC itératif est nécessaire. Ce dernier fonctionne en créant une copie du nuage de points pris en entrée. En détectant les droites au fur et à mesure, il supprime les inliers et s'arrête lorsqu'il n'y a plus de point présent dans le nuage.

RANSAC (nuage de départ ,largeur seuil)

1. La ligne= \emptyset
2. lignedéterminée= \emptyset
3. Inlier minimum=0
4. *Pour* itérations
 - Selection* de 2 points(nuagededépart)
 - Equation* de la ligne (deuxpoints)
 - Chercher*Inliers(plancalculé,largeurseuil)
 - Si*NombreInlier>InlierMaximum
 - InlierMaximum*=NombreInlier
 - Ligne déterminée*=plancalculé
 - Fin*Si
5. *Fin*Pour
6. RéajustementduPlanpar régression linéaire

Afin de vectoriser la coupe des murs, la solution privilégiée au départ était donc RANSAC (RANdomSample Consensus), dont l'algorithme pour détecter les droites est présenté dans la figure ci-dessus. Cependant, le nombre de lignes détectées était beaucoup trop important et les droites n'étaient pas toutes détectées. J'ai passé du temps à réfléchir à l'utilisation des paramètres : nombre d'itérations, nombre minimum d'inliers. Cependant,

par manque de temps, je me suis réorienté vers une nouvelle technique, ayant essayé multiples revers avec RANSAC et dans l'optique de trouver une solution.

Maintenant, après réflexion et avec davantage de recul sur le travail réalisé, je me suis rendu compte que le programme de RANSAC devait être encore plus adapté à la géométrie des murs et qu'il fallait déterminer des seuils minimaux d'inliers différents pour permettre la détection des petites lignes de murs, autant que les grandes, en trouvant tout d'abord les murs les plus grands, supprimant les inliers leur étant associés et en continuant avec une détection des murs plus petits. Cependant, il semble que le seuil minimal d'inliers est une donnée propre à chaque ligne de mur et donc à chaque nuage de points et un problème se poserait, a priori, pour une automatisation, si l'on a des pans de murs très courts.

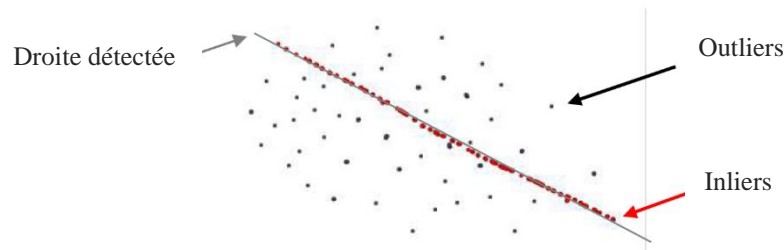


Figure 11 Résultat de l'algorithme de RANSAC, vue de dessus (Source : Revue XYZ n°141, modifié) [26]

Pour Poux et al. [26] cette méthode est « conceptuellement simple, robuste face au bruit et facilement extensible ». La figure 11 issue de ces mêmes travaux démontre ces propos. Cependant, l'algorithme peut conduire à une sur/sous-segmentation, il possède des difficultés d'isolement des détails fin, et le temps de traitement des données est conséquent[7].

II.2.3.2 Mise en place de la Transformée de Hough

Ainsi, après des tests infructueux avec RANSAC, dont la cause a été identifiée a posteriori comme vraisemblablement liée au processus itératif avec des paramètres d'inliers minimaux fixés au départ, qui rendaient la détection incomplète, je me suis donc attelé à la détection de droites avec la transformée de Hough 2d. De manière

[26] Florent POUX, 2014

[7] Benjamin DELSART, 2020

traditionnelle, cette technique est appliquée aux images, mais elle s'applique ici directement sur la coupe du nuage de points 3d. L'idée principale était d'obtenir des droites relativement fiables qui soient par la suite traitées pour obtenir les droites correspondant aux murs présents dans la coupe.

Le fonctionnement de la transformée de Hough peut être expliqué en se concentrant sur la détection de lignes dans une image. Le processus comprend plusieurs étapes. Tout d'abord, l'image est prétraitée en convertissant généralement en niveaux de gris et en détectant les contours.

Ensuite, un espace Hough paramétrique est construit pour représenter toutes les combinaisons possibles de paramètres de la forme recherchée, en l'occurrence des lignes dans notre cas.

Les points d'origine et la transformée de Hough sont tracés (dans la figure 12) à l'aide de la bibliothèque Matplotlib. La figure contient deux sous-graphiques. Le premier sous-graphique affiche les points d'origine, tandis que le deuxième sous-graphique affiche la transformée de Hough sous forme d'image. Les points correspondant aux valeurs maximales dans le tableau d'accumulateur Hough sont également tracés en rouge.

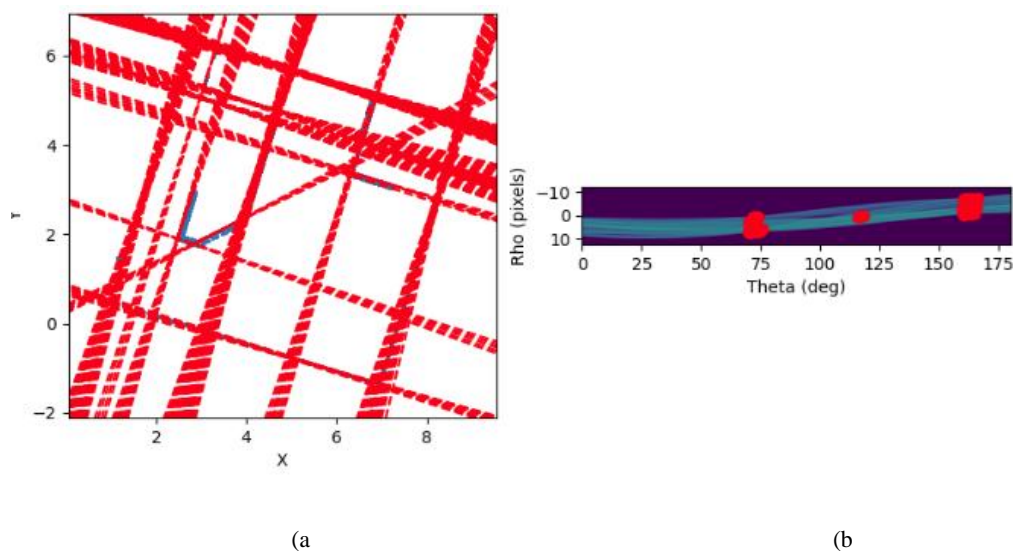


Figure 12 Application de la Transformée de Hough à notre coupe avec visualisation de la coupe et des droites détectées (a) et graphique des maxima dans le tableau d'accumulateur Hough (b)

La transformée de Hough est appréciée car elle permet de détecter des formes géométriques même dans des conditions difficiles. Également, le temps de calcul est assez restreint et a nécessité 2-3 minutes dans notre cas avec une coupe sous-échantillonnée de 47000 points. Cependant, afin de détecter la majeure partie des droites, tout en étant

cohérent avec la réalité, il a été nécessaire de baisser les seuils de détection. En effet, les maximums locaux sont gardés lorsqu'ils correspondent à au moins 10% du maximum dans Hough. De plus, les résolutions de Θ et ρ ont été diminuées.

L'inconvénient de cette méthode est que l'on obtient beaucoup de droites au même endroit en ajustant les paramètres de la sorte. On produit ainsi des conglomérats de droites au même endroit et cela pose une limite quant au dessin final, mais nous avons toutefois une première détection de droites. Également, on remarque aussi la faiblesse de la transformée de Hough qui n'a pas détecté toutes les droites (ici après avoir créé des segments).

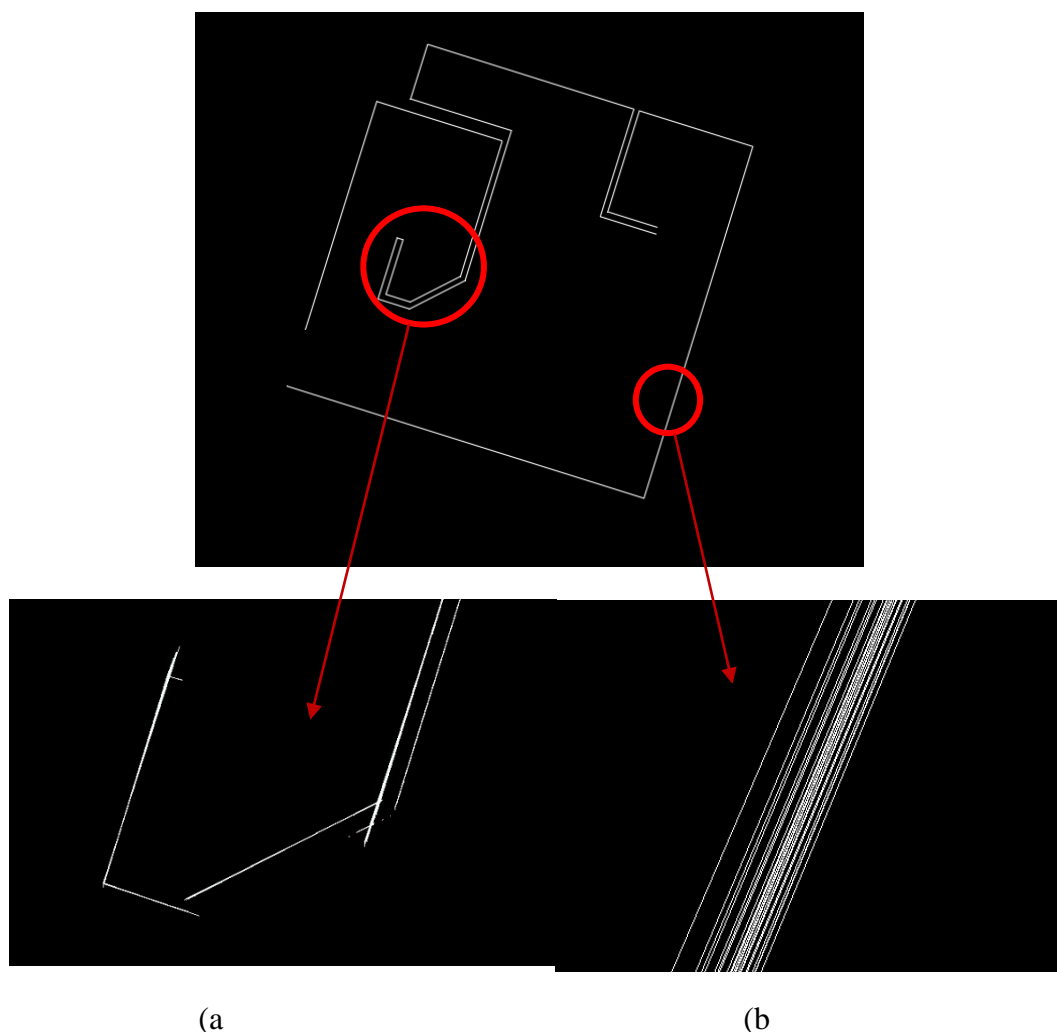


Figure 13 manque de détection de droite (a) et détection de droites parallèles (b)

Avec l'obtention de ces droites, il fallait, dans un premier temps, effectuer un filtrage des lignes, puis calculer les extrémités des segments de murs et les extraire. Pour cela, et après recherches, j'ai opté pour un algorithme qui puisse détecter les coins pour implémenter ces derniers, dans mon programme en vue d'obtenir les segments de murs.

II.2.3.3 Reconnaissance des coins et des points de contour

Les travaux de Hakim Boulaassal [2], dont la thèse porte sur la vectorisation automatique de la façade d'un bâtiment, se sont appuyés sur une extraction des contours, avec une enveloppe convexe de la coupe du nuage de points, puis sur une triangulation de Delaunay. Les résultats obtenus étant satisfaisants, cela m'a poussé à m'intéresser à cette approche partant des points de contour de la coupe du nuage de points.

Surtout, il était nécessaire d'avoir une donnée qui nous permette de contraindre les droites obtenues avec la Transformée de Hough. Les points de contour sont une donnée fiable et qui allège les calculs, car il y a beaucoup moins de points d'intérêt ; ce qui favorise une certaine optimisation de calcul. Cependant, les points de contour seuls ne représentent pas assez la coupe du nuage et ne sont pas susceptibles de nous intéresser réellement.

J'ai donc adopté une nouvelle approche en détectant à la fois les coins et les points de contour dans le nuage de points, les points significatifs qui permettraient de simplifier la coupe et de se baser sur eux pour avoir la version la plus simplifiée de la coupe.

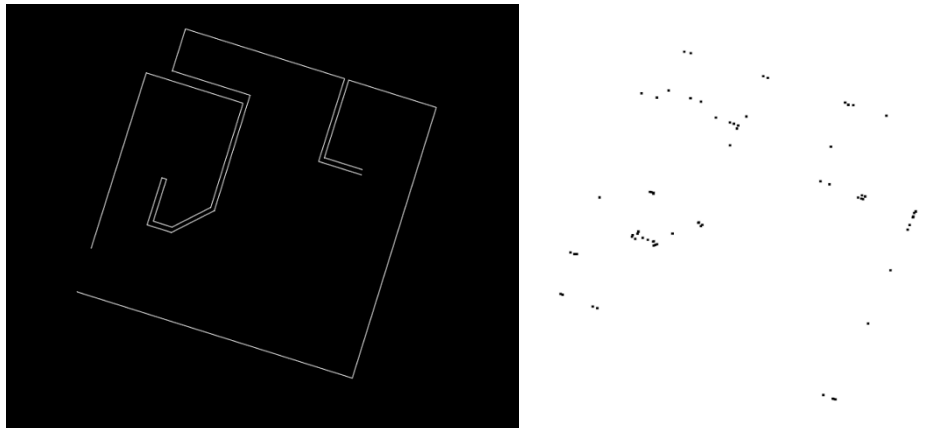


Figure 14 Plan manuel et coins et points de contour détectés avec l'algorithme de Ramer-Douglas-Peucker (RDP)

Pour cela, je me suis servi de l'algorithme de Ramer-Douglas-Peucker (RDP)[13] . On a recours à ce dernier pour simplifier une courbe en réduisant le nombre de points nécessaires pour la représenter et en éliminant les points redondants, tout en préservant les coins ou les caractéristiques importantes de la courbe. L'algorithme RDP est couramment

[2]Hakim BOULAASSAL, 2010
[13]David H DOUGLAS et al, 1973

utilisé dans le traitement d'images, la reconnaissance de formes et d'autres domaines où la simplification des données est nécessaire.

La fonction implémentée prend en entrée une liste de points représentant une courbe et un seuil d'erreur ϵ . La fonction commence par vérifier si le nombre de points est inférieur ou égal à 2. Si c'est le cas, elle renvoie simplement les points d'origine. Sinon, elle recherche le point le plus éloigné de la ligne reliant le premier et le dernier point de la liste. Une distance maximale **dm_{max}** est mise à jour à chaque itération. Une fois que le point le plus éloigné est trouvé, la fonction divise la liste de points en deux parties et réapplique récursivement l'algorithme RDP à ces parties. Les résultats sont ensuite concaténés pour former la liste finale des points simplifiés. Si la distance maximale **dm_{max}** dépasse le seuil ϵ , la fonction continue à subdiviser et simplifier les sous-courbes. Sinon, elle renvoie simplement les points de début et de fin de la courbe.

On obtient ainsi une version simplifiée de la courbe (visible à la figure 14). Ensuite, le programme parcourt les points simplifiés et vérifie si la distance entre chaque point et la ligne formée par le point précédent et le point suivant dépasse le seuil ϵ . Si la distance dépasse le seuil, le point est considéré comme un coin et ajouté à la liste des coins.

II.2.3.4 Implémentation d'un algorithme de regroupement (Clustering)

Une fois les lignes détectées grâce à la Transformée de Hough et l'implémentation des points caractéristiques à l'aide de l'algorithme de Ramer Douglas-Pecker, l'objectif était de déterminer les extrémités des différents segments correspondant aux murs, afin de pouvoir tracer les différentes lignes de murs.

L'idée était d'unir les données que sont les droites et les coins détectés avec le nuage de points afin de pouvoir obtenir les segments des murs. Pour cela, des algorithmes de regroupement existent. On retrouve notamment la méthode des k-plus-proches-voisins ou encore l'algorithme de clustering DBSCAN [14] (Density-Based Spatial Clustering of Applications with Noise). Comme indiqué dans l'acronyme de la méthode, le regroupement des points est réalisé en fonction d'un critère de densité de points et

[14] Ester et al., 1996.

s'applique sur des données bruitées ; ce qui correspond à notre cas, malgré les prétraitements effectués afin de supprimer le bruit de la coupe du nuage.

Un des avantages de DBSCAN est qu'il n'est pas nécessaire de prédéfinir un nombre de regroupements (clusters). Il recherche de manière autonome les échantillons ayant une grande densité de points et les étend en sélectionnant les voisins.

Nous l'avons ainsi utilisé pour la recherche des extrémités des segments, et j'ai obtenu le résultat de la figure 15 ci-dessous.

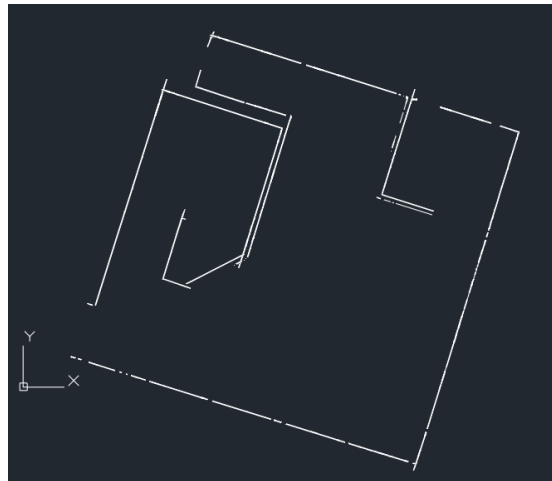


Figure 15 Dessin des droites issues de Hough auxquelles est appliqué le Clustering DBSCAN

Ensuite, nous avons intégré une contrainte qui garde les droites issues de la Transformée de Hough qui sont situées à moins de 2cm des points de contour. Ces droites subissent ensuite une régression linéaire avec les points de la coupe situés au même seuil de distance.

Nous avons obtenu grâce à ce clustering DBSCAN des résultats plus satisfaisants qu'au préalable, notamment en terme de cohérence par rapport à la coupe du nuage de points. En effet, certaines droites non détectées par la Transformée de Hough sont ici prises en compte et on est en présence de résultats plus exhaustifs (voir figure 16). Cependant, de nombreux segments ont été créés le long des lignes de murs. Cela est directement dû à l'application de l'algorithme DBSCAN qui admet de nombreuses régions de points différentes, en s'appuyant sur les densités importantes de points.



Figure 16 Dessin des segments obtenus grâce aux droites issues de Hough avec application de DBSCAN, avec contrainte de se raccrocher sur les points de contour

II.2.3.5 Formation des lignes de murs à partir des segments alignés

Il convient donc de regrouper tous les segments détectés en fonction de critères de cohérence, afin de créer les segments des murs avec les points extrêmes issus des segments alignés. Nous avons testé plusieurs critères comme l'angle entre les segments, des tolérances entre le coefficient directeur et l'ordonnée à l'origine.

Enfin, la solution développée, à la fois plus rapide et efficace, a été de concevoir un algorithme basé sur RANSAC avec les segments en entrée. Voici le pseudo-code :

1. -prendre un segment aléatoire parmi les segments détectés
 - calculer la droite passant par ce segment
 - stocker les segments situés dans une distance à la droite (les deux points du segment sont compris dans la distance seuil)
 - garder les points extrêmes et créer un segment avec ces deux points.
 - les intersections se font si les extrémités de deux segments sont à une distance seuil (de 50 cm par exemple)
2. -supprimer les segments concernés
3. -Répéter les étapes précédentes jusqu'à ce qu'il n'y ait plus de segments.

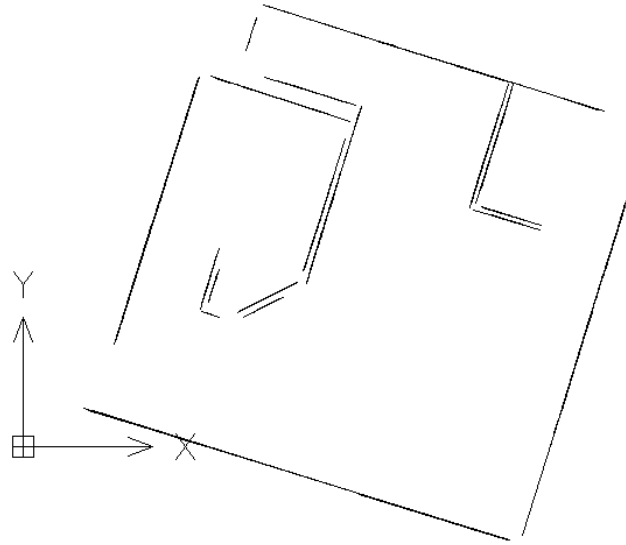


Figure 17 Plan avec fusion des segments alignés générés

Un programme a été réalisé dans cet objectif et le dernier résultat est visible en figure 17. De manière empirique, je me suis aperçu que le seuil de distance pour obtenir les meilleurs résultats était de 5mm. Cela correspond directement à l'écart entre les différents segments. Une perspective sera de réaliser les intersections de segments par la suite et d'implémenter les contraintes de perpendicularité et de parallélisme aux murs dessinés.

Nous avons ainsi choisi, à travers notre méthode, de générer une coupe du nuage de points automatiquement, à partir de laquelle reconnaître les formes de murs et des ouvertures. Pour cela, nous avons tout d'abord implémenté l'algorithme de RANSAC afin de permettre la détection de droites dans la coupe. Ce dernier a été complexe à mettre en place et après de nombreux tests infructueux, l'idée a été d'inclure la Transformée de Hough pour la détection des droites. Les résultats ont dû être améliorés grâce à deux procédés : la détection des coins et points de contour et un algorithme de clustering nommé DBSCAN. Ces techniques fusionnées, nous ont permis d'obtenir des résultats satisfaisants, qui semblent cohérents, de manière macroscopique, comme le montre la figure 17.

III ANALYSE DES RESULTATS

Il convient maintenant d'analyser et d'évaluer la méthode que nous avons mise en place, à travers une partie dédiée à l'analyse des résultats.

III.1 Analyse temporelle

Premièrement, l'un des objectifs de ce programme était l'optimisation des travaux précédents de Victorien Mallebayet un gain de temps, avec la création d'un outil de dessin simplifié et efficace. Cette perspective de développement a été remplie, étant donné que les temps de calculs prennent 15 minutes pour un nuage de 1.79M de points en entrée, au lieu de plusieurs heures précédemment. Cela s'explique par la quantité moindre de points utilisée, en passant du nuage de points entier et des calculs sur les plans de murs à une coupe et des calculs sur les droites correspondant aux murs.

méthodes de dessin du plan d'intérieur	temps passé
dessin manuel	10 min
dessin automatique avec le programme précédent (V.MALLEBAY)	estimé à 3-4h (programme non fonctionnel actuellement)
dessin automatique avec mon programme	15 minutes
dessin automatique modifié	25 minutes

Tableau 3 Comparaison des temps passés pour le dessin du plan d'intérieur

Cependant, d'un point de vue pratique, il faudrait être en mesure de générer les segments de murs de manière complète à terme. A ce stade, les travaux entrepris peuvent néanmoins se révéler être une aide efficace au dessinateur comme j'ai pu le constater. Les segments étant fiables en majorité, nous pouvons réaliser de longs segments dans les directions visibles et réaliser les intersections avec la commande de prolongement de ligne/polyligne sous Autocad. Une approche a été développée afin de créer des segments globaux et sera bientôt efficace.

III.2 Analyse des précisions au long du processus

Dans cette partie dédiée à l'analyse des résultats, il m'est paru intéressant de mettre en avant les précisions obtenues, en partant de la détection avec la méthode de la Transformée de Hough à la génération du plan au format .dxf, en passant par la précision de la détection des coins et points de contour dans la coupe du nuage de points. Il est intéressant, de voir

l'évolution et le gain de précision tout au long du processus. Tout d'abord, il s'agit de constater la justesse des droites détectées.

III.2.1 Détection des droites à l'aide de la Transformée de Hough

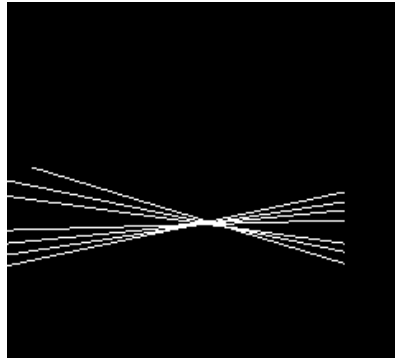


Figure 18 Faisceaux de lignes issues de la Transformée de Hough

En effet, la Transformée de Hough détecte beaucoup trop de droites en parallèles et quasiment confondues ; ce qui nécessite un post-traitement. On a pu constater des écarts de l'ordre de plusieurs cm entre les différentes droites. Afin de pouvoir valider l'exactitude et la justesse du plan au format .dxf généré, nous avons également contrôlé une série d'écarts, que nous allons présenter par la suite. Comme nous l'observons dans la figure 19, les écarts entre le plan créé avec les droites issues de la Transformée de Hough et le plan manuel sont plutôt faibles. Nous obtenons une moyenne de 4mm d'écart en ayant pris un échantillon aléatoire dans la coupe (voir figure 18) et un emq de 4.9 mm.

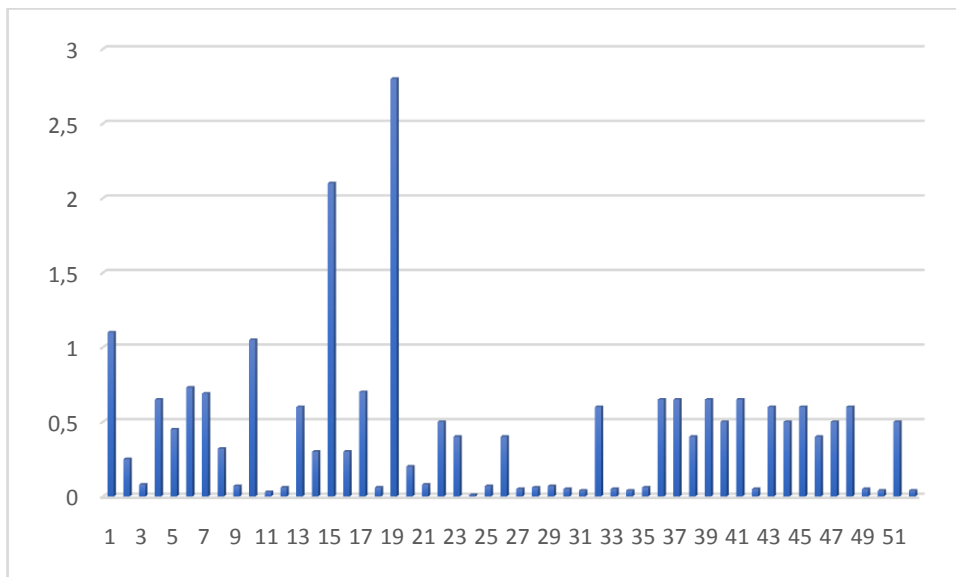


Figure 19 Ecart entre les lignes issues de Hough (en cm)

On remarque quelques écarts importants entre le plan généré et le plan manuel (figure 19) ; ces derniers sont principalement dus à l'algorithme de DBSCAN que nous avons implémenté en prenant comme contrainte la présence de coins et points de contour. Les droites se croisent ainsi aux coins et forment des « double-faisceaux », comme cela est visible à la figure 18.

III.2.2 Extraction des coins et points de contour

Concernant les coins et les points de contour, nous les avons implémentés dans le fichier DXF manuel afin de déterminer s'ils sont précis ou non, par rapport à un plan 2D de référence.

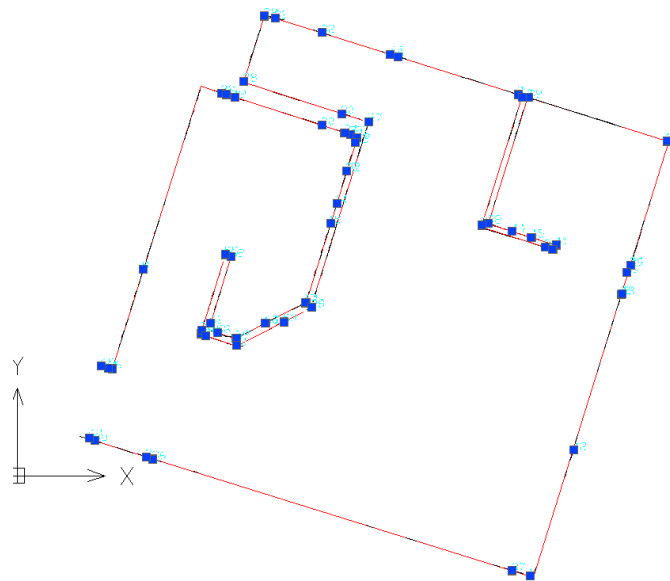


Figure 20 Superposition des coins et points de contour et du plan manuel

Nous remarquons globalement que les coins et points de contour sont quasiment confondus avec le plan manuel réalisé (figure 20), les points en bleu étant les points d'intérêt et le plan manuel étant dessiné en rouge. Les écarts obtenus sont tous inférieurs à 2cm, étant donné la contrainte donnée dans notre programme. En pratique, les écarts sont de précision millimétrique.

On constate par ailleurs que la mauvaise orientation des segments au niveau de la figure 20 est due au manque de coins à ce niveau-là et à la contrainte que nous avons implémentée, qui garde les droites issues de la Transformée de Hough qui sont situées à moins de 2cm des points de contour. Ces droites subissent ensuite une régression linéaire avec les points de la coupe situés au même seuil de distance. Il est donc intéressant de réfléchir sur la valeur du seuil epsilon dans l'algorithme de Ramer Douglas Pecker, afin de

ne pas ignorer de coins dans certaines parties de la coupe, où les densités de points sont moins importantes notamment.

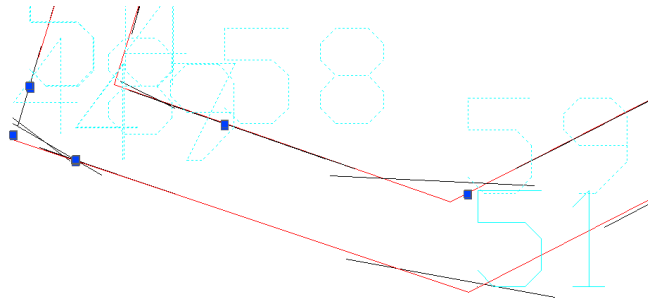


Figure 21 erreur d'orientation des segments (en rouge, le plan manuel de référence)

III.2.3 Segments issus des résultats précédents et du Clustering DBSCAN

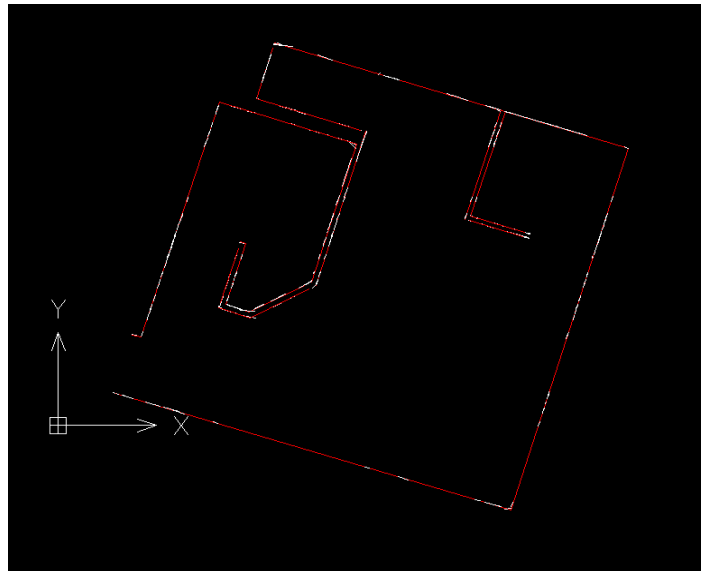


Figure 22 Superposition de la coupe automatique (blanc) et de la coupe manuelle (rouge)

En superposant le dessin automatique créé par notre programme et le dessin réalisé manuellement, nous constatons que les murs que nous avons estimé sont très proches, voire confondus avec le plan manuel. Nous avons mesuré un ensemble d'écart afin d'évaluer la précision de notre méthode et la valider, à cette étape. Pour cela, nous avons pris un échantillon de 70 distances d'écart aléatoirement et régulièrement le long des murs dans la coupe du nuage de points.

Le graphique ci-dessous (figure 20) nous indique les écarts en question. La valeur moyenne des écarts est de 1.1 mm(figure 22) avec une erreur moyenne quadratique de 0.9 mm. Cet écart millimétrique est très faible et montre à quel point la méthode est précise. Cela est très encourageant pour la suite.

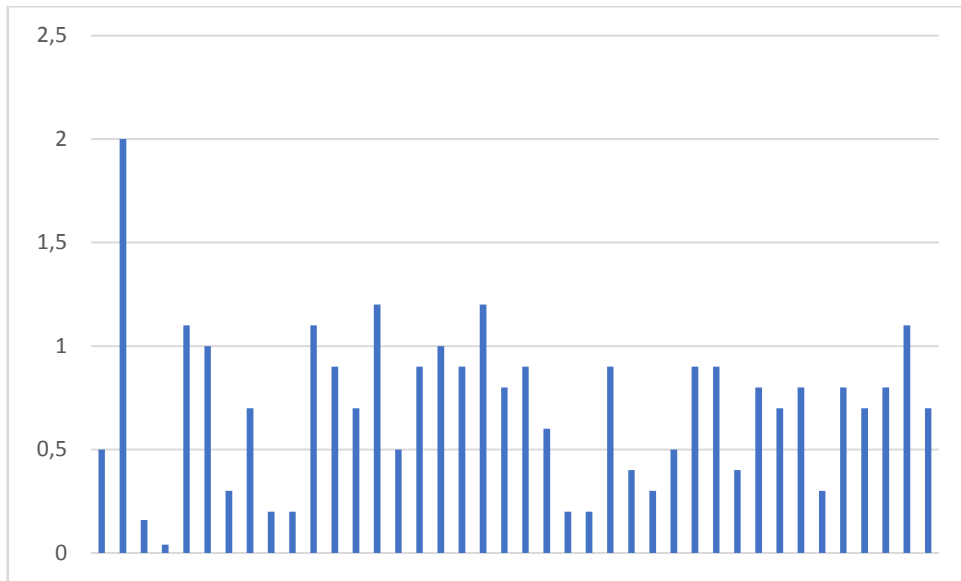


Figure 23 Ecarts entre le dessin automatique et le plan manuel (en mm)

III.2.4 Fusion des segments alignés

Un programme basé sur l’algorithme que nous avons écrit à la partie II.2.3.5 a été développé. Le résultat obtenu est présenté ci-dessous (figure 23). On remarque une avancée par rapport aux résultats précédents avec la réalisation de segments plus représentatifs des murs, en terme de longueur. Il conviendra par la suite d’effectuer les intersections entre les différentes lignes de mur. Et nous pourrons ainsi appliquer les contraintes de perpendicularité et de parallélisme.

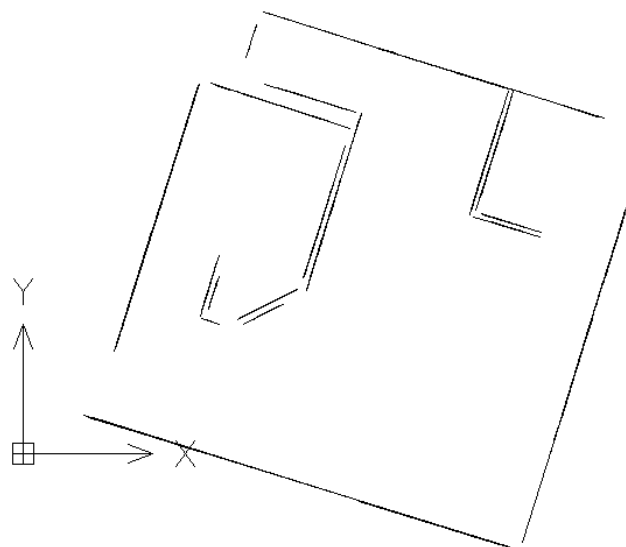


Figure 24 Plan généré avec fusion des segments alignés

On perçoit nettement la cloison en haut à gauche. Cependant, certaines parties incertaines dont celle visée par la figure 20 ont été supprimées. La contrainte de trouver au moins 2 segments dans le seuil de distance par rapport à la droite issu du segment aléatoire a permis de s'affranchir des segments erronés qui étaient présents dans notre dessin.

III.3 Améliorations futures possibles

Dans les perspectives d'évolution du programme, il serait intéressant de créer un partitionnement automatique des pièces en entrée. Cela pourrait nous permettre de réellement générer un plan d'intérieur automatique à terme, avec utilisation de hachures et calcul de superficie, de hauteurs sous plafond pour chaque pièce, comme décrit dans les travaux "Floor plan generation from 3D point clouds: A space partitioning approach"[17]

De même, pour ce qui est des portes, l'insertion de symboles de portes aux dimensions universelles constituerait une avancée, même si cette étape ne prend pas beaucoup de temps manuellement en pratique pour quelques éléments.

III.3.1 Intégration des contraintes associées au plan d'intérieur

Dans les points à développer, il est primordial d'indiquer l'application des contraintes associées au plan d'intérieur. Elles sont de 2 types ; nous sommes en présence de contraintes géométriques :de perpendicularité / de parallélisme et également de contraintes de dimensions standards.

III.3.1.1 Perpendicularité et parallélisme des murs

Tout d'abord, concernant la perpendicularité et le parallélisme des différents murs, de ces deux contraintes dépend toute la géométrie générale du dessin d'intérieur. En effet, après différents ajustements, les murs et cloisons changent faiblement de direction ; ce qui est aussi le cas des objets se situant dans le prolongement des murs tels que les ouvertures (portes, fenêtres).

Pour appliquer une contrainte de perpendicularité des murs, il faudrait modifier l'orientation des droites associées aux murs en fixant l'angle à 90 degrés. Cela permettrait de réajuster les contraintes. En partant du principe qu'à un angle proche de 90 degrés, il

[17] FANG et al., 2021

s'agit d'un angle droit, nous avons ainsi réajusté la plupart des droites. Il faut néanmoins veiller à effectuer les liaisons entre les différents segments dessinés car le but est d'automatiser le processus. Concernant le dessin des liaisons manquantes, Léa TALEC [12] , a procédé à une méthode avec le plus proche voisin. Le coefficient de la droite doit être proche de l'un des coefficients directeurs modifiés.

III.3.1.2 Contraintes de construction standard

Il existe également des exigences de standardisation dans le dessin des différents éléments de construction. En effet, ces derniers ont des dimensions normalisées et cela doit être intégré dans le dessin, même si les cotes mesurées sont parfois légèrement différentes.

Tout d'abord, dans une maison, les murs sont larges de 20cm au moins et de 7cm pour les cloisons. De manière générale, il existe deux types de portes dans la construction : les portes d'entrée et les portes d'intérieur. Pour les portes d'entrée, les largeurs standards possibles sont de 63,73,83 ou encore 93cm. En ce qui concerne les portes d'intérieur, on distingue seulement 2 largeurs : 80 ou 90cm. Pour ce qui est des fenêtres, les dimensions standards d'une fenêtre classique sont de 115, 125 et 135cm dans le cas d'une fenêtre classique.

Notre approche et le programme réalisés constituent un réel progrès par rapport au programme précédent de Victorien Mallebay. En effet, il y a, tout d'abord, un gain de temps important réalisé en passant d'environ 3-4h à 20 minutes de traitement ; ce qui représente une réelle optimisation. De plus, le plan généré est très fidèle à la réalité avec un écart moyen de 1mm par rapport au plan manuel et les derniers résultats laissent présager de belles perspectives futures. Pour parvenir à ces nouveaux objectifs qui s'ouvrent à nous, il serait intéressant de pouvoir segmenter les pièces de manière automatique dès le début afin de pouvoir incorporer de nouveaux éléments dans le plan tel que le calcul de hauteur sous plafond pour chaque pièce ou encore le calcul de surface. Par ailleurs, l'intégration des contraintes de perpendicularité et de parallélisme et des contraintes de construction standards est l'un des points visés à terme par notre processus de génération automatique de plan 2D.

[12] Léa TALEC, 2018

Conclusion

Le plan d'intérieur est indispensable à la réalisation de plusieurs missions que le géomètre-expert est amené à réaliser, et notamment, pour les prestations afférentes à la copropriété, dont le géomètre est le professionnel missionné pour la réalisation des plans.

Notre travail de recherche a constitué en la réalisation d'un processus automatisé destiné à générer un plan d'intérieur 2D à partir d'un nuage de points 3D classifié. Ce travail avait pour objectif de simplifier le traitement d'un nuage de points et de générer un gain de temps en automatisant le processus de conception de plan d'intérieur à partir d'un nuage de points 3D classifié et ce, en s'appuyant sur les recherches de Victorien Mallebay.

Notre étude a porté sur une coupe de nuage de points 3D et a permis de combiner diverses méthodes de détection de lignes, de coins et points de contour et algorithme de regroupement de points, afin de pouvoir envisager une nouvelle approche de reconnaissance et de dessin de lignes de murs avec la création d'un fichier DXF, directement visualisable depuis la plupart des logiciels de DAO ou de CAO. Dans cette perspective, nous avons utilisé la Transformée de Hough afin de détecter les lignes de murs. Cette technique a été couplée à un algorithme de Clustering, DBSCAN, avec la contrainte de créer des segments avec les points de contour lorsque ces derniers sont à un seuil centimétrique de la ligne. Les résultats obtenus sont plutôt exhaustifs même s'ils ne sont pas complètement finaux. Il serait opportun de regrouper tous les segments alignés afin de créer des segments de murs entiers.

Dans la continuité de ce projet, trois points pourront nous permettre d'affiner le projet et de développer davantage le processus déjà bien abouti. Tout d'abord, il est important de créer des segments globaux liés aux lignes de murs. il serait également nécessaire de s'intéresser à l'application des contraintes liées au dessin des différents éléments dans le plan d'intérieur, et notamment les contraintes de perpendicularité, parallélisme et les contraintes de dimensions standards.

Ensuite, il conviendra de réfléchir sur les approches techniques qui nous permettraient à terme de générer un véritable plan d'intérieur directement. Une première perspective est une segmentation automatique des différentes pièces du bâtiment, en entrée, afin de pouvoir calculer les différentes hauteurs sous plafond notamment,

Pour conclure, ce travail m'a permis de m'investir complètement dans un projet, de développer des réflexions sur les paramètres utilisés et les possibles limites à chaque partie de programmation réalisée, d'entrevoir enfin des perspectives d'évolution en s'intéressant aux évolutions techniques dans le domaine de l'exploitation des données issues du nuage de points 3D.

Références bibliographiques

Travaux Universitaires

- **Habilitation à Diriger des Recherches (HDR)**

[1] Tania LANDES, 2020. Contribution à la segmentation et à la modélisation 3D du milieu urbain à partir de nuages de points, [en ligne]. Université de Strasbourg, Laboratoire des Sciences de l'Ingénieur, de l'Informatique et de l'Imagerie, 146p. Disponible sur : [Tania_landes_these](#), consulté le 12/07/2023.

- **Thèse**

[2] BOULAASSAL Hakim, Segmentation et modelisation geometriques de facades de batiments a partir de releves laser terrestres, Memoire de these. Soutenu a Strasbourg : INSA, 2010, 214p, [En ligne] ; Disponible sur [thèse Hakim Boulaassal](#) (consulté le 03/02/2023)

[3] HAMELIN Adrien, Sélection et amélioration de nuages de points 3D, Mémoire de Thèse, Soutenu à Toulouse, INP Toulouse, 2015, 144p, [En ligne] ; Disponible sur <https://oatao.univ-toulouse.fr/14707/1/hamelin.pdf>(consulté le 04/04/2023)

[4] Thomas CHAPERON ; 2002. Segmentation de nuage de points 3D pour la modélisation automatique d'environnements industriels numérisés, [en ligne]. ECOLE DES MINES DE PARIS Collège doctoral, 251p. Disponible sur : [Thèse CHAPERON](#)(consulté le 12/05/23)

[5] Hélène MACHER, 2017. Du nuage de points à la maquette numérique de bâtiment : Reconstruction 3D semi-automatique de bâtiments existants, [en ligne]. Université de Strasbourg, Discipline : Sciences de l'Ingénieur – Topographie, Géomatique, 166p. Disponible sur : [Helene_macher_these](#), (consulté le 17/05/2023)

- **Mémoire TFE**

[6] Gabriel BERGE, 2020. Analyse de la dérive d'un scanner dynamique, le Zeb-Revo RT, [en ligne]. Géomatique et foncier, École Supérieure d'ingénieurs Géomètres et Topographes, 64p. Disponible sur : [Mémoire BERGE](#), (consulté le 12/03/23)

[7] Benjamin DELSART, 2020. Segmentation manuelle de nuage de points au sein d'un système d'information géographique web 3D, [en ligne]. Faculté des Sciences, Liège Université, 109p. Disponible sur : [Mémoire DELSART](#), (consulté le 02/03/23)

[8] Nicolas DESCHATRETTE, 2017. Etude d'un système d'acquisition dynamique mobile et mise en place de procédures permettant de gérer, suivre et valoriser de multiples données, [en ligne]. Géomatique et foncier, École Supérieure d'ingénieurs Géomètres et Topographes, 77p. Disponible sur : [Mémoire DESCHATRETTE](#)(consulté le 05/03/23)

[9] Stéphane FERNANDEZ-CUIRET, 2013. *Étude comparative des différentes méthodes du levé d'intérieur d'un bâtiment en vue de réhabilitation et analyse des besoins et des*

moyens d'un architecte dans le but de produire une modélisation 3D optimale, [en ligne]. Géomatique et foncier, École Supérieure d'ingénieurs Géomètres et Topographes, 60p. Disponible sur : [Memoire FERNANDEZ-CUIREZ](#), (consulté le 06/03/23)

[10] Thibault NOELLAT, 2019. *Mise en place d'une chaîne de traitement semi-automatisée au sein d'une cellule de dessin sur nuage de points*, [en ligne]. Géomatique et foncier, École Supérieure d'ingénieurs Géomètres et Topographes, 75p. Disponible sur : [Mémoire NOELLAT](#), (consulté le 05/04/23)

[11] Maxime RENAUD, 2016. *Création automatique d'une maquette numérique 3D d'un bâtiment à partir d'un relevé scanner laser 3D*, [en ligne]. Géomatique et foncier, École Supérieure d'ingénieurs Géomètres et Topographes, 73p. Disponible sur <https://dumas.ccsd.cnrs.fr/dumas-01811031>, (consulté le 04/03/23)

[12] Léa TALEC, 2018. *Mise en place d'un processus de dessin automatisé de plans d'intérieurs à partir de nuages de points acquis par LIDAR*, [en ligne]. Géomatique et foncier, École Supérieure d'ingénieurs Géomètres et Topographes, 68p. Disponible sur : [Léa TALEC](#), (consulté le 02/03/23)

- **Ouvrage imprimé**

[13] David H Douglas et Thomas K. Peucker, « Algorithms for the reduction of the number of points required to represent a digitized line or its caricature », *Cartographica: The International Journal for Geographic Information and Geovisualization*, University of Toronto Press, vol. 10, n° 2, 1973, p. 112-122

[14] Martin Ester, Hans-Peter Kriegel, Jörg Sander, and Xiaowei Xu. 1996. A density-based algorithm for discovering clusters in large spatial databases with noise. In Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining,

[15] : François CHOLLET, 2017, *L'apprentissage profond avec Python*, [en ligne]. Les Essentiels de l'IA, Manning Publications 39p. Disponible sur : [ouvrage CHOLLET](#),

[16] HOUGH, P. V. (1962). *Method and means for recognizing complex patterns*. US Patent 3,069,654.

- **Articles scientifiques électroniques**

[17] Hao FANG et al., 2021. *Floor plan generation from 3D pointclouds :A space partitioning approach*, [enligne]. Shenzhen University. Disponible sur : [Hao FANG et al](#), (consulté le 15/03/23)

[18] Martin A. FISCHLER, Robert C. BOLLES, 1981. *Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography*, [enligne]. Communications of the ACM. Disponible sur [FISCHLER, BOLLES](#), (consulté le 15/06/23)

[19] T. Landes, P. G. (2011). Les principes fondamentaux de la lasergrammétrie terrestre 2 : acquisition, traitement des données et applications. XYZ n°129, Association Française de Topographie, pp. 25-38.

[20] Xiaohu LU, Yahui LIU et Kai LI, 2020. *Fast 3D Line Segment Detection From Unorganized Point Cloud*, [enligne]. Department of Civil, Environmental and Geodetic Engineering, Ohio State University, Columbus, USA. Disponible sur : [Fast 3D Line Segment Detection From Unorganized Point Cloud](#), (consulté le 15/03/23)

[21] Jakub stefan MARKIEWICZ, Piotr PODLASIAK, Dorota ZAWIESKA, 2015. *A New Approach to the Generation of Orthoimages of Cultural Heritage Objects—Integrating TLS and Image Data*, [enligne]. Department of Photogrammetry, Remote Sensing and Spatial Information Systems, Faculty of Geodesy and Cartography, Warsaw University of Technology, Poland. Disponible sur : [Integrating TLS and Image Data](#), (consulté le 12/03/23)

[22] Qingzhao et al, 2023, Rapid generation of editable engineering drawings from 3D point cloud reconstruction for large-scale buildings, *Journal of Building Engineering*, [enligne], Department of Disaster Mitigation for Structures, Tongji University, Shanghai, China. Disponible sur : [rapid generation of drawing from 3d point cloud](#) (consulté le 27/04/23)

[23] Blanca QUINTANA GALERA et al, 2018. *Door Detection in 3d coloured point clouds of indoor environments*, [enligne]. *Automation in Construction*. Disponible sur : [doordetection](#), (consulté le 06/04/23)

[24] TARSHA-KURDI et al. (2007). Hough-Transform and Extended Ransac Algorithms for Automatic Detection of 3D Building Roof Planes From Lidar Data. *In ISPRS Workshop on Laser Scanning*, volume XXXVI, pages 407–412. (consulté le 29/06/23)

[25] Hou WENGUANG, Chan TAIWAN et Ding MINGYUE, 2012. *Denoising point cloud*, [en ligne]. *Inverse Problems in Science and Engineering*, pages 297-298. Disponible sur : [Denoising point cloud](#), (consulté le 06/04/23)

- **Articles de périodiques électroniques**

[26] Florent POUX et al., 4^e trimestre 2014. *Segmentation semi-automatique pour le traitement de données 3D denses : application au patrimoine architectural*. *Revue XYZ* [en ligne]. Disponible sur : [Revue XYZ n°141](#), (consulté le 08/03/23)

- **Cours électroniques**

[27] Ransac algorithm (2D). Slides from University of Illinois Lectures by D. Hoiem. First Accessed: 2020-05-15. [en ligne]. Disponible sur : [cours RANSAC-Hough](#)(consulté le 02/03/23)

- **Article de Loi**

[28] Loi n°65-557 du 10 juillet 1965, *fixant le statut de la copropriété des immeubles bâtis*, [en ligne]. Disponible sur [legifrance](#), (consulté le 14/06/23)

- **Site web**

^[29] Aftopo, *Lexique : 3.2 Calage, mise en station* [en ligne]. Disponible sur : [aftopo](#), (consulté le 04/03/23)

^[30] *Jeu de données S3DIS, nuage de points 3D classifié*, [en ligne] : [S3DIS](#) (consulté le 18/03/23)

^[31] Leica-geosystems, *Leica RTC360 Solution de capture de la réalité 3D*[en ligne]. Disponible sur : [leica RTC360](#), (consulté le 19/07/23)

^[32] My digital building, *Générer et exploiter le nuage de points 3D d'un bâtiment*, [en ligne].Disponible sur : [my digital building](#), (consulté le 05/07/23)

- **manuel**

^[33] : Jsinfo, 2019. *Plan d'intérieur compensé (PIC), version 19*, [en ligne].Top station, 111p. Disponible sur : [Plan d'Interieur Compensé](#), (consulté le 17/03/23)

Liste des figures

Figure 1 Scanner fixe RTC360 [31]	8
Figure 2 Geoslam Zeb-REVO RT [6]	9
Figure 3 Coordonnées sphériques d'un scanner laser terrestre	10
Figure 4 Nuage de points 3d brut [22]	11
Figure 5 Algorithme de RANSAC pour trouver la meilleure droite dans un nuage de points bruité [30]	15
Figure 6 Transformation de Hough et changement de l'espace cartésien à l'espace paramétrique[21]	16
Figure 7 Listing des classes et labels associés (a) et visualisation des murs dans le nuage de points classifié (b)	22
Figure 8 solution proposée de processus de dessin automatique de plan d'intérieur à partir d'un nuage de points 3D classifié	23
Figure 9 Coupe du nuage de points 3D réalisée sous Python	24
Figure 10 Suppression du bruit par [25].....	25
Figure 11 Résultat de l'algorithme de RANSAC, vue de dessus (Source : Revue XYZ n°141, modifié) [26].....	27
Figure 12 Application de la Transformée de Hough à notre coupe avec visualisation de la coupe et des droites détectées (a) et graphique des maxima dans le tableau d'accumulateur Hough (b)	28
Figure 13 manque de détection de droite (a) et détection de droites parallèles (b).....	29
Figure 14 coins et points de contour détectés avec l'algorithme de Ramer-Douglas-Peucker (RDP)	30
Figure 15 Dessin des droites issues de Hough auxquelles est appliqué le Clustering DBSCAN	32
Figure 16 dessin des segments obtenus grâce aux droites issues de Hough avec application de DBSCAN, avec contrainte de se raccrocher sur les points de contour.....	33
Figure 17 Plan avec fusion des segments alignés généré.....	34
Figure 18 Faisceaux de lignes issues de la Transformée de Hough	36
Figure 19 Ecarts entre les lignes issues de Hough (en cm)	36
Figure 20 Superposition des coins et points de contour et du plan manuel	37
Figure 21 erreur d'orientation des segments (en rouge, le plan manuel de référence).....	38
Figure 22 Superposition de la coupe automatique (blanc) et de la coupe manuelle (rouge).....	38
Figure 23 Ecarts entre le dessin automatique et le plan manuel (en mm).....	39
Figure 24 Plan généré avec fusion des segments alignés.....	39

Liste des tableaux

Tableau 1 Caractéristiques du poste de travail utilisé	19
Tableau 2 bibliothèques utilisées dans notre projet.....	19
Tableau 3 Comparaison des temps passés pour le dessin du plan d'intérieur	35

Optimisation et développement d'un processus automatisé de dessin d'un plan d'intérieur, à partir d'un nuage de points 3D classifié.

Mémoire d'Ingénieur C.N.A.M., Montpellier2022

RESUME

Le plan d'intérieur est une tâche souvent réalisée par les cabinets de Géomètres Experts, notamment car depuis le 29 Juillet 2022, les plans annexés à la Copropriété relèvent du monopole du Géomètre-Expert. Lorsqu'elle est manuelle, la création du plan d'intérieur est un mission qui peut se révéler chronophage. De nos jours, afin de permettre une optimisation et un gain de temps, les cabinets de géomètre utilisent souvent des scanners laser 3D pour le levé d'intérieur. Est ainsi obtenu un nuage de points brut qu'il convient de traiter afin d'obtenir un plan d'intérieur. Dans cette perspective de productivité, j'ai été missionné pour produire un programme de dessin automatisé de plan 2D à partir d'un nuage de points classifié.

Mon étude a mené à l'utilisation conjointe de différentes méthodes de reconnaissance de forme comme RANSAC ou la Transformée de Hough, d'un algorithme de clustering tel que DBSCAN ou encore à la détection des coins et points de contour avec le procédé de Ramer-Douglas-Peucker.

Les résultats obtenus sont cohérents et décrivent bien la réalité des lieux, avec des écarts millimétriques entre le plan manuel et le plan automatique généré par nos soins.

Mots clés : Plan d'intérieur, Scanner, nuage de points 3D, RANSAC, Transformée de Hough 2D, détection de lignes, Ramer-Douglas-Peucker, DBSCAN, Python

SUMMARY

The interior plan is a task often carried out by Géomètres Experts firms, particularly since since July 29, 2022, plans appended to condominiums have come under the monopoly of Géomètre-Expert(Chartered Land surveyors in France). When done manually, creating a floor plan can be a time-consuming task. Nowadays, to optimize and save time, surveyors often use 3D laser scanners for interior surveys. The result of the measurements is a raw point cloud that needs to be processed to produce an interior plan. With this productivity in mind, I was asked to produce an automated 2D drawing program from a classified point cloud.

My study led to the joint use of various shape recognition methods such as RANSAC or the Hough Transform, a clustering algorithm such as DBSCAN, and the detection of corners and contour points using the Ramer-Douglas-Peucker process.

The results obtained are coherent and accurately describe the reality of the site, with millimetric differences between the manual plan and the automatic plan generated by us.

Key words :Floor Plan, interior plan, Scanner, 3D point cloud, RANSAC, 2D Hough Transform, line detection, Ramer-Douglas-PeuckerDBSCAN, Python