Object detection in unstructured 3D data sets using explicit semantics

Jean-Jacques Ponciano

N d'ordre NNT : 2019LYSES059

Ecole Doctorale 488 : Sciences, Ingénierie, Santé

THESE de DOCTORAT DE L'UNIVERSITE DE LYON

opérée au sein de l'université Jean Monnet

Discipline de doctorat : Informatique

Soutenue publiquement le 14/11/2019, par :

Jean-Jacques PONCIANO

Object detection in unstructured 3D data sets using explicit semantics.

Détection d'objet dans des ensembles de données 3D non-structurés utilisant une sémantique explicite.

Devant le jury composé de :

Anita SELLENT, Professeur, Université des Sciences appliquées Mainz	Présidente
Pierre GRUSSENMEYER, Professeur, INSA de Strasbourg	Rapporteur
José-Fernán MARTÍNEZ-ORTEGA, Professeur, Université de Madrid	Rapporteur
Roland BILLEN, Professeur, Université de Liège	Examinateur
Alain TRÉMEAU, Professeur, Université de Saint-Étienne	Directeur de thèse
Frank BOOCHS, Professeur, Université des Sciences appliquées Mainz	Co-directeur de thèse

Dedicated to Denise GAVEAU

Object detection in unstructured 3D data sets using explicit semantics.

Jean-Jacques PONCIANO

14 Novembre 2019

Contents

Abstract

With the evolution of technologies and robotics, the possibilities offered by 3D acquisition systems have increased. Nowadays, these systems are used in different domains as for autonomous vehicles, rescue robots, cultural heritage, for example. These application fields often require to perform object recognition from acquired data. Therefore, various methodologies have been investigated to automatically process 3D point cloud data in order to detect contained objects. The best methodologies depend on the context, that means they are specific to the data to be processed and the objects to be recognized. They produce efficient recognition, which is essential whatever the application field. However, adapting methodologies to a particular application field or use case limits the flexibility to extend the use of a method to other fields. These observations highlight the importance of developing object recognition methodologies specific to a detection context, but also the limitation of existing methods to preserve their capacity within changing detection contexts. An excellent example of a high degree of flexibility to changing contexts is human intelligence and human's ability to design ad hoc methodologies. Humans can analyze the context according to their knowledge and combine different characteristics or strategies according to the objective to be achieved. It would, therefore, be helpful for Computer Vision tools to integrate elements of artificial intelligence, allowing to adapt to the context of an application fields and to guide the detection process in this respect. This Ph.D. thesis presents a knowledge-based approach for object recognition that can be used whatever the application field. Its architecture is based on semantic technologies to allow a knowledge management module to guide the objects detection process through a step by step procedure performing the selection, parameterization, and execution of algorithms. The detection process is performed thanks to an artificial intelligence approach that uses explicit knowledge to design a context-dependent object recognition solution. Its strength is its adaptability to the context, but also its capability to analyze and understand a scene and contained objects and the specificities of the data to be processed. This understanding capability is realized through a self-learning process able to define and validate hypotheses concerning the context, also enabling to enrich the knowledge base and to improve the objects recognition process. The efficiency of this adaptation capability will be demonstrated in four use cases from different application fields. The first use case is an indoor of a building. It is used for a monitoring purpose. The second use case is located in the field of Archaeology represented by ancient ruins containing a terrace house with a watermill. The third use case is an outdoor representing a part of the city of Freiburg in Germany.

It is used for an industrial purpose. Finally, the last use case is an indoor acquired by Microsoft's Kinect. It is used for a robotic purpose.

Résumé

Avec l'évolution des technologies et de la robotique, les possibilités offertes par les systèmes d'acquisition 3D ont augmenté. Aujourd'hui, ces systèmes sont utilisés dans différents domaines comme par exemple pour les véhicules autonomes, les robots de sauvetage, le patrimoine culturel. Ces champs d'application nécessitent souvent la reconnaissance d'objets à partir de données acquises. C'est pourquoi diverses méthodologies ont été étudiées pour traiter automatiquement les données 3D des nuages de points afin de détecter les objets contenus. Les meilleures méthodologies dépendent du contexte, c'est-à-dire qu'elles sont spécifiques aux données à traiter et aux objets à reconnaître. Elles produisent une reconnaissance performante, ce qui est essentiel quel que soit le domaine d'application. Toutefois, l'adaptation des méthodologies à un domaine d'application ou à un cas d'utilisation particulier limite la possibilité d'étendre l'utilisation d'une méthode à d'autres domaines. Ces observations soulignent l'importance de développer des méthodologies de reconnaissance d'objets spécifiques à un contexte de détection, mais aussi la limitation des méthodes existantes pour préserver leur capacité dans des contextes de détection changeants. Un excellent exemple d'un degré élevé de flexibilité face à l'évolution des contextes est l'intelligence humaine et la capacité de l'homme à concevoir des méthodologies ad hoc. L'homme peut analyser le contexte en fonction de ses connaissances et combiner différentes caractéristiques ou stratégies en fonction de l'objectif à atteindre. Il serait donc utile que les outils de vision par ordinateur intègrent des éléments d'intelligence artificielle permettant de s'adapter au contexte d'un domaine d'application et de guider le processus de détection à cet égard. Cette thèse de doctorat présente une approche de la reconnaissance d'objets basée sur la connaissance qui peut être utilisée dans tous les domaines d'application. Son architecture est basée sur des technologies sémantiques pour permettre à un module de gestion des connaissances de guider le processus de détection d'objets à travers une procédure étape par étape effectuant la sélection, le paramétrage et l'exécution des algorithmes. Le processus de détection est réalisé grâce à une approche d'intelligence artificielle qui utilise des connaissances explicites pour concevoir une solution de reconnaissance d'objets en fonction du contexte. Sa force réside dans son adaptabilité au contexte, mais aussi dans sa capacité d'analyse et de compréhension d'une scène et d'objets contenus ainsi que

dans les spécificités des données à traiter. Cette capacité de compréhension est réalisée par un processus d'auto-apprentissage capable de définir et de valider des hypothèses concernant le contexte, permettant ainsi d'enrichir la base de connaissances et d'améliorer le processus de reconnaissance des objets. L'efficacité de cette capacité d'adaptation sera démontrée dans quatre cas d'utilisation de différents domaines d'application. Le premier cas d'utilisation est l'intérieur d'un bâtiment. Il est utilisé à des fins de surveillance. Le second cas d'utilisation se situe dans le domaine de l'archéologie représenté par des ruines anciennes contenant une maison en terrasse avec un moulin à eau. Le troisième cas d'utilisation est un extérieur représentant une partie de la ville de Fribourg en Allemagne. Il est utilisé à des fins industrielles. Enfin, le dernier cas d'utilisation est un intérieur acquis par Kinect de Microsoft. Il est utilisé à des fins robotiques.

Acknowledgment

The Ph.D. thesis was carried out and financed at the institute i3mainz, Institute for Spatial Information and Surveying Technology of the University of Applied Sciences Mainz. The competences of this institute are in the scope of Spatial Information Technology in the Humanities with application fields such as cultural heritage, geology, environment (e.g. disaster management, bioaerosol), and energy. Its three main domains of competence are the geoinformation systems, the optical 3D metrology, and the information technology. Its works in geoinformation systems consist of designing and developing applications in business and administration. The optical 3D metrology works correspond to precise spatial detection of objects for documentation, control quality, and control process. Finally, the works in information technologies are composed of developments for Internet, multimedia, 3D visualization, and eLearning.

This thesis gave rise to the institute's project called "Knowledge-based Object Detection in Images and Point Clouds" (KnowDip). The work developed in this thesis has led to the development of an object detection framework capable of automatically understanding and structuring unstructured 3D data from various application fields. I want to thank my supervisor Professor Alain Trémeau and my co-supervisor Professor Frank Boochs, who supported and wisely advised me throughout my thesis. It was an honor for me to work under their supervision. I also want to thank Dr. Ashish Karmacharya with whom I worked closely in the KnowDIP project and Songül Polat for her help and her advice. Finally, I would like to thank all the members of the institute i3mainz for their working ethic that has allowed me to thrive within this institute.

Introduction (en Français)

Contexte et motivation

Grâce à la puissance des nouvelles technologies de détection, nous numérisons de plus en plus le monde réel. Cependant, les instruments produisent des données non structurées, principalement sous forme de nuages de points pour les données 3D et d'images pour les données 2D. Néanmoins, de nombreuses applications (p. ex. la navigation ou la documentation de conformité) nécessitent des données structurées contenant des objets et leur géométrie. La structuration des données nécessite l'interprétation de leur contenu. Cette interprétation correspond à l'identification des objets et de leurs géométries contenus dans les données. L'application détermine les objets et leurs géométries.

De plus, une application nécessite une qualité de données qui dépend des besoins et des contraintes de cette application. La qualité des données est déterminée par les caractéristiques des données (p. ex. densité, texture, couleur, bruit, résolution, régularité, complétude) qui répondent aux exigences de l'application. Les caractéristiques des données dépendent du processus d'acquisition. L'application guide le choix du processus d'acquisition correspondant au choix des méthodologies (p. ex. stratégie de mise en place et de mesure), des technologies d'acquisition (comme les instruments) et du contexte d'acquisition (p. ex. objet, scène, conditions environnementales). Par exemple, en archéologie, l'acquisition d'une copie virtuelle d'un objet peut nécessiter une résolution de données de 0,01 mm, alors que pour les fouilles, quelques millimètres peuvent être suffisants. Dans cet exemple, la technologie d'acquisition est choisie en fonction de sa capacité à fournir une résolution de données appropriée. D'autres applications nécessitent la préservation de la couleur et de la texture de la scène acquise. Dans ces cas, l'acquisition 2D combinée à la photogrammétrie, qui construit un nuage de points 3D exclusivement à partir de plusieurs images 2D, est généralement préférée aux technologies laser-scanner en raison de sa performance à préserver les textures. Dans le domaine industriel, les voitures autonomes ou la planification d'infrastructures urbaines, nécessitent l'acquisition de grandes scènes. C'est pourquoi un scanner laser (comme le scanner Lidar) est principalement utilisé pour cette application pour acquérir des scènes 3D sur de très longues distances.

Cependant, le processus de détection produit souvent une qualité de données inférieure à celle requise par l'application. Les compromis entre le choix des méthodologies et des technologies, d'une part, et les contraintes du contexte d'acquisition, d'autre part, produisent cette différence. Il est donc nécessaire de compenser la différence entre la qualité des données obtenue par le processus de détection et la qualité attendue. Cette compensation peut être obtenue en intégrant des informations sur le processus de détection (technologies d'acquisition, méthodologie d'acquisition et contexte d'acquisition) pour adapter l'interprétation du contenu des données à leur qualité.

Plusieurs approches de vision par ordinateur visent à interpréter le contenu des données et à essayer de compenser la qualité inférieure des données. Ces approches sont principalement contextuelles, c'est-à-dire spécifiques aux données à traiter et aux objets à détecter. Certaines approches (appelées approches axées sur les modèles) considèrent principalement une partie de l'information sur le contexte de la scène et sur les objets contenus dans les données. Ces approches utilisent des informations sur la géométrie et la forme des objets pour construire des modèles correspondants. Ces approches examinent les données pour identifier les sections qui sont similaires aux modèles construits. Les sections identifiées permettent l'interprétation des objets et de leur géométrie pour structurer les données. Toutefois, ces approches n'utilisent qu'une partie de l'information sur le contexte de l'acquisition et n'exploitent pas l'information sur les méthodes ou les technologies d'acquisition. Ces approches ne sont donc pas en mesure de compenser les faiblesses de la qualité des données. Les résultats de ces approches dépendent directement de la qualité des données. D'autres approches (appelées approches axées sur les données) n'exploitent pas directement l'information sur les méthodes d'acquisition et les technologies d'acquisition, mais se concentrent sur la qualité des données obtenues. Ces approches sont principalement composées de trois étapes principales. La première étape consiste à préparer les données pour compenser les faiblesses de la qualité des données (comme le bruit) en supprimant les éléments dont les informations ne sont pas pertinentes ou qui peuvent induire en erreur le traitement. La deuxième étape consiste à segmenter les données en sous-régions par des algorithmes ad hoc ou des combinaisons d'algorithmes. Cette étape vise à faciliter l'interprétation des données en divisant leur complexité. Chaque sous-région fournit des indices pour interpréter le contenu des données. La dernière étape consiste à classer chaque sous-région pour interpréter le contenu des données. Toutefois, ces approches n'utilisent pas directement l'information sur le contexte d'acquisition, les méthodes d'acquisition ou les technologies d'acquisition. Ils ne peuvent donc pas comprendre les principes fondamentaux de la qualité des données et doivent, par conséquent, compenser toute faiblesse dans la qualité des données. Compenser toutes les faiblesses de qualité des données est souvent trop fastidieux dans certaines applications pour réussir.

Par conséquent, il est logique d'explorer les possibilités d'utiliser l'information sur les méthodes et les technologies d'acquisition pour adapter l'interprétation des données à leur qualité. Par conséquent, il est nécessaire de connaître la qualité des données pour guider l'interprétation des données. Il est donc nécessaire d'utiliser l'information sur les méthodologies et les technologies d'acquisition pour estimer la qualité des données. Enfin, il faut utiliser l'information sur le contexte d'acquisition (principalement les scènes et leurs composantes) pour interpréter le contenu des données. Cette interprétation s'effectue en classant les sections de données. La prise en compte de toutes ces informations permet une interprétation plus robuste du contenu des données que d'autres approches (axées sur le modèle et sur les données). De plus, la compréhension de ces informations permet de s'adapter à différents types de contextes de données. Toutefois, la qualité estimée peut différer de la qualité obtenue. Cette différence crée des problèmes imprévisibles pour l'interprétation du contenu des données. Il est nécessaire de résoudre ces problèmes causés par la différence entre la qualité des données estimées et la qualité des données obtenues par le processus de détection. Une approche pour compenser cette différence de qualité consiste à comprendre la dérivation entre la qualité des données estimées et la qualité des données obtenues. Cette compréhension permet d'adapter l'estimation de la qualité des données. Cette adaptation permet donc d'adapter l'interprétation du contenu des données. De plus, il est difficile d'anticiper les écarts entre la qualité estimée et la qualité obtenue. Par conséquent, ces écarts doivent être observés et analysés pour l'ensemble des données. Ce traitement nécessite une évaluation locale de la qualité des données obtenues. De plus, les résultats produits par l'interprétation du contenu des données fournissent des indices sur la qualité des données obtenues. Ainsi, l'analyse de ces résultats permet de déduire de nouvelles informations sur la qualité des données obtenues. Ces nouvelles informations permettent d'adapter l'interprétation du contenu des données. Cette adaptation permet une meilleure structuration des données en fonction de leur qualité.

Portée de la thèse

Cas d'utilisation

Dans cette thèse, nous nous sommes concentrés sur quatre cas d'application différents pour illustrer les différentes complexités qui peuvent exister dans les principaux domaines de la détection d'objets. Ces cas d'application proviennent de différents domaines d'application (p. ex. patrimoine culturel, documentation conforme à l'exécution) dans différents contextes (p. ex. excavation de ruines, intérieur de bâtiment, extérieur dans un milieu urbain). Ils disposent de données 3D non structurées, sous la forme d'un nuage de points et ont besoin de la détection de divers objets (p. ex. mur, plancher, plafond, voiture, chaise, moulin à eau, arbre) pour structurer leurs données. Dans cette thèse, nous nous concentrons sur la structuration des données. C'est pourquoi nous nous concentrons sur la détection des éléments qui constituent les données. Cette tâche génère d'importantes difficultés scientifiques.

Cas d'utilisation en intérieur

Le premier cas d'application consiste à documenter un bâtiment en reconstruisant un modèle 3D de celui-ci. Pour ce faire, il est nécessaire d'identifier chacune des pièces qui composent le bâtiment. L'identification des pièces nécessite l'identification des murs, des planchers et des plafonds. Ces éléments sont de grands objets dont les couleurs peuvent être très variables. Dans les bâtiments modernes, ces éléments ont principalement une forme régulière et rectangulaire.

Le bâtiment choisi pour ce cas d'application est numérisé par scanner laser en intérieur, produisant un nuage de points 3D non structuré [*Armeni et al.*, 2017]. Ainsi, plusieurs facteurs influencent la représentation des planchers, des murs et des plafonds.

L'utilisation de scanners laser ne permet pas d'acquérir des objets en verre ou ayant une surface très réfléchissante. Ainsi, certaines parties des murs numérisés sont absentes lorsque les murs sont composés de fenêtres ou de surfaces réfléchissantes. De plus, la méthode d'acquisition consiste à déplacer le scanner à l'intérieur du bâtiment en suivant un chemin qui génère des occlusions. Ainsi, cet ensemble de données contient de nombreuses discontinuités, principalement pour les murs et le sol en raison de leur occlusion par des éléments à l'intérieur des pièces. De plus, parmi les principaux facteurs externes, la couleur artificielle de la lumière et les variations de lumière d'une pièce à l'autre influencent le processus d'acquisition en générant du bruit et influencent respectivement les couleurs des éléments numérisés. La figure 1 illustre l'ensemble de données de ce cas d'application.

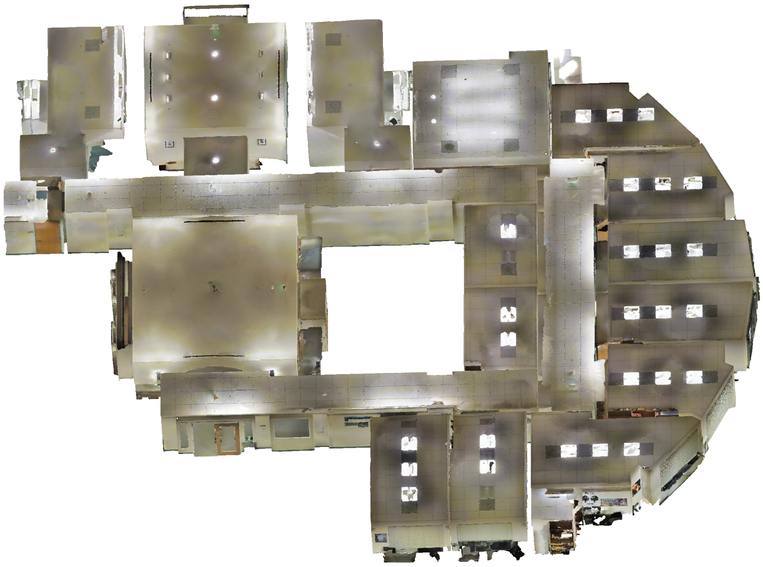


Figure 1: Présentation de l'ensemble de données sur les espaces intérieurs de "Stanford 3D à grande échelle" [*Armeni et al.*, 2017].

Le principal défi des approches de vision par ordinateur dans ce cas d'application est la segmentation des pièces qui ont des caractéristiques et des éléments communs. Par exemple, deux pièces différentes peuvent partager un plafond, un plancher ou des murs.

Cet ensemble de données est en libre accès et permet de comparer l'approche proposée dans cette thèse avec d'autres approches existantes dans la littérature.

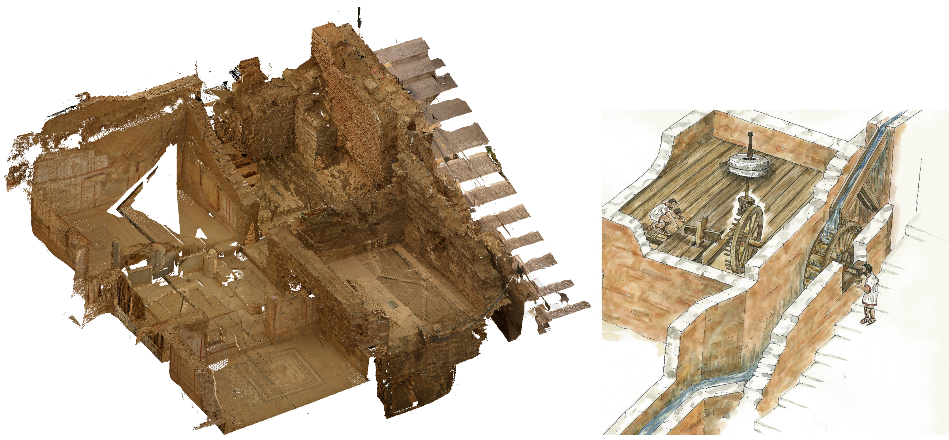
Cas d'utilisation pour le patrimoine culturel

Le deuxième cas d'application provient du domaine du patrimoine culturel et représente une maison en terrasse partiellement détruite en Turquie. Le but de cette application est de soutenir l'interprétation archéologique par la détection d'objets basée sur la connaissance.

L'objet principal recherché dans cette application est un moulin à eau. Cet objet est décrit comme étant composé de deux parties principales (une grande pièce et une pièce étroite), dont l'une contenait la roue du moulin et l'autre le mécanisme.

Dans ce cas d'application, une scène d'excavation de ruines est numérisée par scanner laser terrestre (TLS) [*Lemmens*, 2011], produisant un nuage de points 3D non structuré [*Wefers*, 2015]. Cependant, dans le contexte des fouilles de ruines, le moulin à eau n'a pas résisté au temps et seule la structure de certaines de ses pièces a été conservée. Le plafond et une grande partie des murs manquent, ce qui augmente la difficulté de détecter le moulin à eau. De plus, l'utilisation du laser-scanner sur ce nuage de points produit des formes irrégulières liées à la structure du sol et aux murs qui ne sont pas planaires, rendant l'utilisation et les configurations des algorithmes plus complexes.

La figure 2 montre le nuage de points acquis et un modèle du moulin à eau.



(a) Une portion de la maison en terrasse acquise en nuage de points.

(b) Dessin du moulin à eau prévu.

Figure 2: Présentation de la maison en terrasse avec le moulin à eau.

Le principal défi dans ce cas d'application est d'adapter le processus de détection au contexte de fouille de ruines qui transforme considérablement les caractéristiques de l'objet.

Cas d'utilisation en milieu extérieur urbain

Le troisième cas d'application est l'acquisition d'une scène dans un environnement extérieur urbain acquise par un laser-scanner. Cette technologie d'acquisition produit un nuage de points 3D non structuré. Le nuage de points est obtenu en balayant séquentiellement la scène avec le laser se déplaçant sur une voiture le long de la route. Dans cette application, de nombreux facteurs (variation de luminosité, vibration de l'instrument de mesure, conditions météorologiques) influencent le processus d'acquisition. Ainsi, ils influencent les caractéristiques (p. ex. bruit, densité, régularité) du nuage de points généré. En outre, la méthode d'acquisition séquentielle par balayage provoque d'une part l'acquisition de données sous forme "d'étapes" et d'autre part la génération de zones d'occlusion (zones sans information). De plus, les différents matériaux (p. ex. métal, verre, pierre) et la distance entre les objets et le scanner laser au moment de l'acquisition influencent grandement le processus d'acquisition. Ainsi, certaines parties des données peuvent être denses et continues, tandis que d'autres peuvent être discontinues avec une faible densité. De plus, la complexité des différents facteurs influençant le processus d'acquisition conduit à la génération de situations et de problèmes imprévisibles où les caractéristiques des données obtenues diffèrent sensiblement des caractéristiques attendues.

Le principal défi dans ce cas d'application est de résoudre les problèmes découlant de changements imprévisibles dans les caractéristiques des données. Deux nuages de points intéressants traitent de ce cas d'application.

Le premier nuage de points provient du domaine industriel et vise à documenter des parties de la ville de Freiburg en Allemagne. Certaines parties de la ville sont numérisées par Fraunhofer IPM[1] en utilisant un scanner laser Lidar.

La figure 3 illustre une partie de cet ensemble de données.

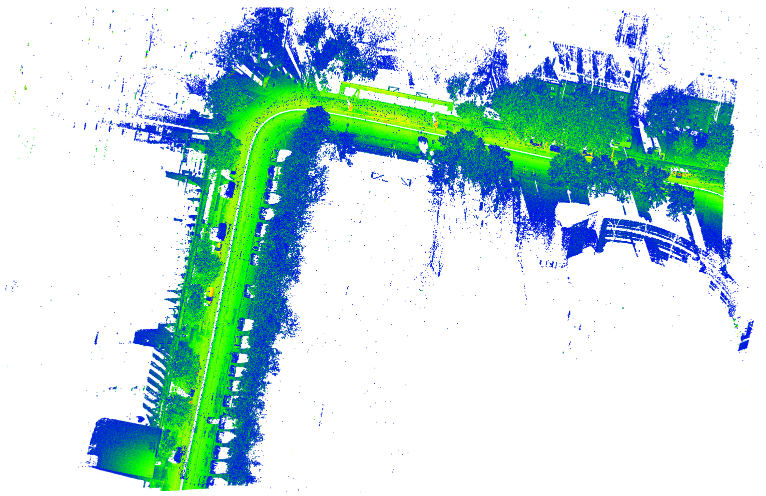


Figure 3: Présentation de la partie de Freiburg en nuage de points (fictivement coloré pour augmenter la visibilité).

L'autre nuage de points correspond à l'ensemble de données "Paris-rue-Madame : MINES ParisTech jeu de données de la rue Madame à Paris obtenu par scanner

[1] Fraunhofer IPM: https://www.ipm.fraunhofer.de/de/presse_publikationen/Presseinformationen/messfahrzeug-3D-Daten-breitbandausbau.html

laser mobile 3D" [2] [*Serna et al.*, 2014].

Il représente la rue Madame, une rue du 6^{eme} arrondissement parisien en France. Cette rue est numérisée par un système MLS (Mobile Laser Scanning) provenant du laboratoire de robotique (CAOR) de MINES ParisTech (Paris, France). La figure 4 illustre cet ensemble de données.

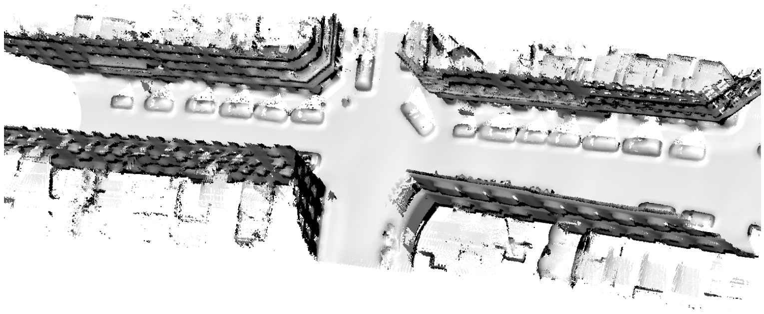


Figure 4: Illustration de la base de données "Paris-rue-Madame : MINES ParisTech jeu de données de la rue Madame à Paris obtenu par scanner laser mobile 3D".

Les principaux objets à détecter dans ce cas d'application en milieu urbain sont les voitures, les panneaux de signalisation, les murs (parfois aussi appelés façade dans ce contexte), le sol et les motos. Dans le contexte des scènes urbaines extérieures, ces objets ont des formes et des caractéristiques diverses.

Cas d'utilisation "Time-of-flight"

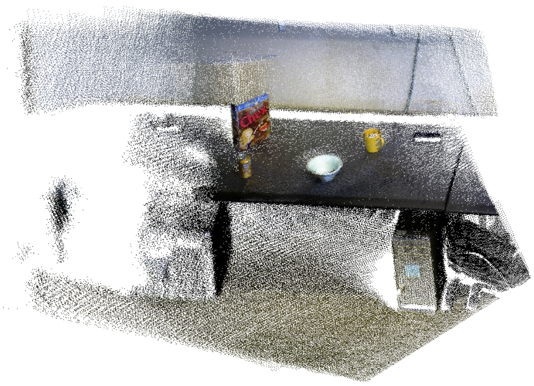
Le quatrième cas d'application vise à localiser des objets spécifiques dans différentes scènes intérieures pour la surveillance. Dans ce cas d'utilisation, les objets recherchés sont des meubles tels que des chaises et des tables, dont la forme et la couleur diffèrent d'une scène à l'autre. Les caractéristiques géométriques et physiques des objets sont donc diverses.

Les scènes choisies pour ce cas d'application sont numérisées par [*Lai et al.*, 2011] à l'aide d'une caméra de type Kinect de Microsoft qui produit des nuages de points 3D, comme le montre la Figure 5. Chaque nuage de points est composé de plus d'un million de points.

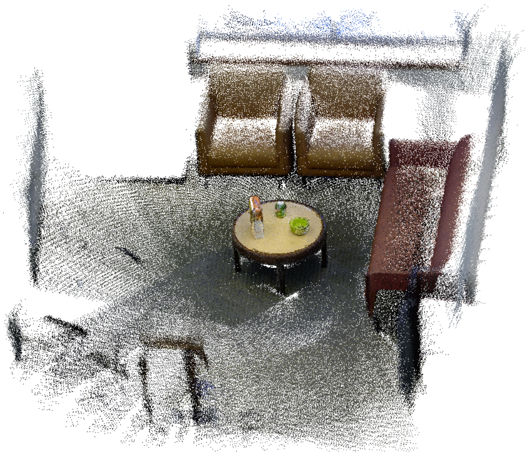
L'utilisation d'une caméra de type Kinect de Microsoft produit un ensemble de données de faible qualité avec des surfaces principalement irrégulières. De plus, le processus d'acquisition de ces nuages de points produit de nombreuses parties

[2] MINES ParisTech© copyright. MINES ParisTech a créé cet ensemble spécial de données MLS 3D pour les activités de recherche en détection-segmentation-classification, mais ne cautionne pas la manière dont elles sont utilisées dans ce projet ni les conclusions avancées.

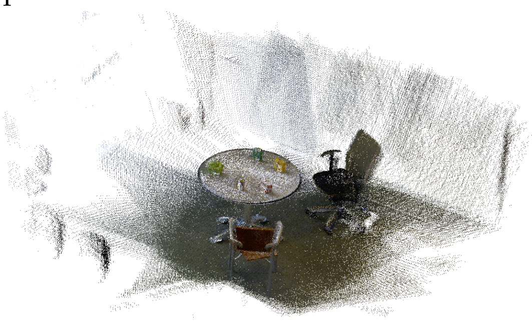
manquantes en raison des occlusions et de la faible fréquence d'acquisition. Ces caractéristiques augmentent la difficulté de détection des objets.



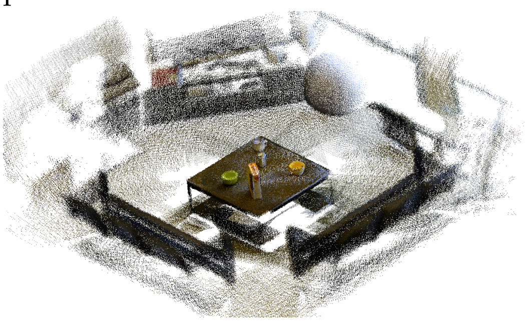
(a) Première scène dans le nuage de points.



(b) Seconde scène dans le nuage de points.



(c) Troisième scène dans le nuage de points.



(d) Quatrième scène dans le nuage de points.

Figure 5: Présentation des quatre scènes acquises [*Lai et al.*, 2011].

Le principal défi, dans ce cas d'application, est de détecter des objets même si leurs caractéristiques et leurs formes changent considérablement (p. ex. table ronde, table carrée, chaise de bureau, chaise en bois).

Énoncé du problème

Comprendre les données non structurées est un véritable défi. Cela nécessite de détecter les représentations dans les données des objets et géométries de la scène numérisée. La détection d'objets et de géométries dépend d'une part de leurs caractéristiques (ex. : taille, forme) et d'autre part des caractéristiques des données (ex. : densité, bruit, occlusion, rugosité). Les caractéristiques des données dépendent du processus d'acquisition (p. ex. technologie et méthodologie utilisées) qui les génère. Les caractéristiques des objets numérisés (p. ex. matériau, réflectance, rugosité, taille), le contexte de la scène (p. ex. extérieur en milieu urbain, bâtiment intérieur, excavation de ruines) et divers autres facteurs externes au processus d'acquisition (p. ex. lumière ambiante, intensité lumineuse, conditions météorologiques, mouvement de l'instrument de mesure ou encore des objets numérisés) ont un effet sur ce processus d'acquisition. De petites variations dans l'un de ces facteurs peuvent fortement influencer les caractéristiques des données. Ces variations de caractéristiques génèrent des divergences entre les caractéristiques attendues et les caractéristiques obtenues.

Les approches fondées sur l'apprentissage machine (comme l'apprentissage approfondi et l'apprentissage continu) remplacent le manque de compréhension de ces facteurs par l'identification de schémas fiables. Ces schémas doivent être appris au cours d'une phase d'entraînement. C'est pourquoi l'apprentissage machine "fiable" nécessite une grande quantité de données. Cependant, dans certains domaines d'application, tels que le patrimoine culturel (voir le cas d'application 2.1.2), les données annotées ne sont pas disponibles ou sont insuffisantes. Ainsi, les approches basées sur l'apprentissage machine ne sont pas pertinentes pour détecter des objets dans un tel ensemble de données. De plus, les approches d'apprentissage machine fonctionnent tant que les données considérées sont représentatives du contenu à comprendre. Plus l'objet ou l'apparence varie, plus il faut de données pour l'entraînement. Les méthodes d'apprentissage machine restent rigides et ne permettent pas de détecter l'objet ou la géométrie pour lesquels ils ne sont pas entraînés. De plus, une petite modification du processus d'acquisition ou des facteurs externes qui influencent les données peuvent engendrer un changement pour lequel les approches d'apprentissage machine ne sont pas entraînées.

Au contraire, d'autres approches basées sur les technologies sémantiques tentent d'intégrer des connaissances sur l'objet ou le processus de détection afin d'alimenter le processus de compréhension des données. Néanmoins, ces approches sont incapables de formuler la connaissance d'une manière qui représente tous les cas.

Par conséquent, les approches actuelles ne sont pas assez robustes, souples et généralisables pour surmonter ces différences. Leur manque de robustesse limite la compréhension des données (p. ex. parties de données mal identifiées ou non identifiées). De plus, ces approches sont contextuelles, c'est-à-dire spécifiques aux données à traiter et aux objets à détecter. Par conséquent, la question de cette thèse est de savoir comment fournir un processus de détection d'objets robuste face aux variations des caractéristiques des données et sans dépendre du contexte.

Pour composer avec les variations des caractéristiques des données, il faut comprendre leur origine et leur incidence sur le processus de compréhension des données. La compréhension est basée sur la connaissance, elle-même basée sur la connexion de l'information. Par conséquent, une approche fondée sur la connaissance de la source et de l'impact des variations des caractéristiques des données est nécessaire pour guider le processus de compréhension des données. Pour résoudre ce problème à l'aide d'une approche fondée sur les connaissances, il faut résoudre trois sous-problèmes. Le premier sous-problème est de savoir comment représenter les connaissances sur la source et l'impact des variations des données. Le deuxième sous-problème est de savoir comment fournir la flexibilité manquante dans d'autres approches fondées sur les connaissances pour fournir un processus de détection adapté aux variations. Le troisième sous-problème, également lié à la flexibilité, est de savoir comment apporter un enrichissement dynamique et adapté des connaissances pour surmonter le manque de connaissances sur la diversité des cas représentés dans les données.

Solution proposée

C'est pourquoi l'approche proposée dans cette thèse aborde d'abord le premier sous-problème en considérant le processus d'acquisition des données et les différents facteurs qui l'influencent. Cette approche tente de comprendre l'origine des caractéristiques des données en comprenant les influences des caractéristiques du processus d'acquisition et les différents facteurs qui l'influencent. Une telle compréhension permet d'anticiper les caractéristiques des données. Cette compréhension exige une connaissance explicite des domaines des données, de la scène et du traitement des données, ainsi qu'une connaissance de l'influence entre ces domaines.

Domaine de données: Le domaine des données se compose de la connaissance des caractéristiques des données et de la connaissance du processus d'acquisition (p. ex. méthodologie d'acquisition, technologie d'acquisition, instrument d'acquisition).

Domaine de scène: Le domaine de la scène numérisée est composé de la connaissance des objets, de leur répartition dans la scène, du contexte de la scène, et des facteurs externes.

Domaine de traitement des données: Enfin, la connaissance des algorithmes (p. ex. conditions d'utilisation, objectif de l'algorithme, configuration, conditions d'utilisation privilégiées) à utiliser pour traiter les données constitue le domaine de traitement des données.

La connaissance de ces domaines et de leurs influences permet d'établir une base de raisonnement efficace.

Par conséquent, l'approche présentée utilise cette connaissance explicite pour guider le traitement des données. Les algorithmes de vision par ordinateur de diverses bibliothèques (telles que *PCL* [*Rusu and Cousins*, 2011], *OpenCV*[3]) sont utilisés pour traiter les données. De même, les technologies du Web sémantique permettent la gestion des connaissances. L'approche présentée répond ensuite au deuxième sous-problème en combinant ces deux paradigmes à travers un pont à la fois technique et conceptuel. Ce pont permet aux connaissances de piloter pleinement le processus de traitement des données étape par étape, grâce à un échange continu entre les deux paradigmes. Cet échange continu permet d'adapter le processus de détection d'objets et de géométrie en fonction des nouvelles connaissances acquises à chaque étape du traitement des données.

L'approche proposée commence par la combinaison des données et de la connaissance des scènes pour déduire et anticiper les caractéristiques des données (telles que la rugosité, la densité, les occlusions) et ainsi enrichir la connaissance des données. Cette combinaison de connaissances augmente également les connaissances sur les représentations possibles des objets dans les données. Par exemple, la technologie du scanner laser n'acquiert pas correctement le verre. Ainsi, une table avec pieds en bois et plateau en verre ne sera représentée dans les données que par ces pieds.

Ensuite, l'approche combine les connaissances sur les trois domaines pour sélectionner et configurer automatiquement et adéquatement les algorithmes en fonction de ces connaissances, et donc en fonction du cas d'application considéré. Les algorithmes doivent être sélectionnés et configurés en fonction des résultats déjà obtenus par les algorithmes précédemment exécutés pour obtenir un traitement efficace des données. C'est pourquoi l'approche proposée interprète automatiquement les données issues de l'exécution des algorithmes pour enrichir les connaissances. Ainsi, les algorithmes sont dynamiquement sélectionnés et configurés en fonction de l'évolution du traitement des données.

L'approche combine ensuite la connaissance des résultats des algorithmes avec la

[3] Laganière, R. (2014). OpenCV Computer Vision Application Programming Programming Cookbook Deuxième édition. Packt Publishing Ltd.

connaissance de la représentation des objets dans les données pour les comprendre et les structurer.

Bien qu'une telle compréhension puisse suffire à structurer entièrement les données dans certains cas, elle peut ne pas suffire dans d'autres. Dans ce cas, la structuration des données peut être de qualité insuffisante pour l'application en question. Ce cas se produit lorsque la connaissance des représentations d'objets diffère des représentations réelles des objets dans les données. Dans ce cas, il est nécessaire d'augmenter les connaissances sur la représentation des objets en fonction des caractéristiques des données.

C'est pourquoi nous proposons d'intégrer une nouvelle méthode de génération automatique de connaissances, répondant au troisième sous-problème. Cette méthode consiste à utiliser les connaissances existantes comme base d'apprentissage pour construire et tester de nouvelles connaissances. Ce processus d'auto-apprentissage consiste à rassembler et à combiner les informations contenues dans les connaissances pour formuler de nouvelles hypothèses. Ces hypothèses sont ensuite testées en vérifiant que leur intégration dans les connaissances initiales ne crée pas d'incohérence. En d'autres termes, cela consiste à vérifier que l'ajout de l'hypothèse n'engendre pas de contre-exemple dans la base de connaissance. Si une hypothèse ne crée pas d'incohérence, elle est considérée comme cohérente et intégrée aux connaissances existantes. L'intégration de nouvelles connaissances enrichit les connaissances sur les représentations des objets et modifie le comportement du processus de détection.

Prenons, par exemple, une partie des données représentant une table. Si cette partie a été mal acquise et qu'aucune connaissance n'était disponible pour anticiper cette mauvaise acquisition, alors la connaissance des représentations de la table peut ne pas correspondre à celle de la table dans les données. Ainsi, la table ne serait pas détectée. Dans ce cas, le processus d'auto-apprentissage fondé sur les connaissances recueillerait toute l'information dont il dispose sur les autres représentations des tables qui ont été détectées. Supposons que les chaises entourent toutes les autres tables détectées. Alors le processus formulerait l'hypothèse que "si des chaises entourent un objet, alors cet objet est une table". Il vérifiera ensuite que la connaissance ne contient pas de contre-exemple de cette hypothèse. Dans ce cas, la vérification consiste à vérifier que tous les objets entourés de chaises sont des tables. Si l'hypothèse est validée, ces nouvelles connaissances sont intégrées aux connaissances existantes. Supposons maintenant que les chaises détectées entourent la partie des données correspondant à la table non détectée. Alors, les nouvelles connaissances ajoutées permettraient d'identifier cette partie des données en tant que table.

Le processus d'auto-apprentissage vise à adapter les connaissances sur les représentations des objets pour qu'elles correspondent aux représentations réelles des objets dans les données. De cette façon, il améliore la compréhension des données non structurées.

Aperçu des contributions et de la thèse

Les travaux présentés dans cette thèse visent à produire un processus de détection d'objets robuste à la variation de la qualité des données 3D et utilisable quel que soit le domaine d'application. En d'autres termes, ce processus doit rester efficace, quelles que soient les données ou les spécificités de l'objet. La réalisation de cet objectif se fait en utilisant la connaissance humaine et en l'adaptant dynamiquement au cours du processus de détection de l'objet. Nos principales contributions se situent donc dans le domaine de l'intelligence artificielle avec la création d'un processus d'auto-apprentissage basé sur la connaissance, ainsi que dans le domaine du Web sémantique et de la vision par ordinateur.

Contributions

Web sémantique

Nous avons trois contributions principales dans le domaine du Web sémantique. La première contribution est l'intégration automatique d'informations (telles que des métadonnées dans des fichiers de données ou des informations géographiques) dans le Web sémantique. Ces travaux ont été publiés dans [*Prudhomme et al.*, 2017] et dans [*Prudhomme et al.*, 2019].

La deuxième contribution est une extension de *SPARQL*, qui est une technologie standard du Web sémantique. Cette contribution est un ensemble d'extensions intégrées à *SPARQL* (built-ins) pour le traitement des données pour la vision par ordinateur.

La troisième contribution est l'interprétation automatique de *OWL-restriction* et *OWL2-restriction* en requêtes *SPARQL*.

Ces trois contributions sont combinées dans un framework qui exécute automatiquement des algorithmes de vision par ordinateur uniquement à travers des requêtes *SPARQL* à partir de processus de raisonnement. Ce processus de raisonnement est appliqué à l'ontologie qui contient toutes les descriptions nécessaires. Ainsi, nous créons un pont entre l'ingénierie des connaissances du Web sémantique et les algorithmes de vision par ordinateur.

Vision par ordinateur

En plus de proposer un état de l'art compréhensif des différentes approches de vision par ordinateur pour détecter des objets dans un nuage de points 3D, nous avons trois contributions principales dans le domaine de la vision numérique.

Tout d'abord, nous proposons une adaptation automatique du processus de détection des objets en fonction du contexte (cas d'application, processus d'acquisition et caractéristiques des données) et des objets recherchés. Cette approche tient compte de l'acquisition des données et du contexte pour identifier les objets dont les caractéristiques géométriques ne sont pas suffisantes ou utilisables pour les reconnaître. Ce travail a été appliqué dans le domaine du patrimoine culturel et a abouti à la publication [*Ponciano et al.*, 2019*b*].

Deuxièmement, nous proposons un système qui sélectionne et configure automatiquement les algorithmes de détection d'objets 3D par l'utilisation d'une base de connaissances et d'un mécanisme de raisonnement. Ce système sélectionne et paramètre les algorithmes en fonction des objets recherchés, des données utilisées et des conditions préalables des algorithmes. Ce travail a été publié dans [*Ponciano et al.*, 2017].

Troisièmement, nous proposons un processus de détection des objets, qui est priorisé en fonction de la taille des objets et de leurs relations topologiques. La détection hiérarchique réduit la zone de recherche d'un objet aux endroits où l'objet peut se trouver. Cette stratégie de détection améliore la performance des algorithmes appliqués aux parties localisées des données plutôt qu'à l'ensemble des données. Ces travaux ont été publiés dans [*Ponciano et al.*, 2019*a*].

La combinaison de ces trois contributions produit un système de détection d'objets robuste et puissant. Ce système obtient d'excellents résultats pour différents domaines d'application et objets recherchés.

Intelligence artificielle

La principale contribution au domaine de l'intelligence artificielle est un processus d'auto-apprentissage basé sur la connaissance. Nous créons un système capable d'adapter ses connaissances en fonction de l'expérience acquise lors d'un premier processus de détection. Ce système formule des hypothèses puis les valide par une analyse des conséquences produites par leurs applications. Un tel comportement permet au système de se développer par lui-même sans aucune condition préalable, même si l'intégration des connaissances humaines améliore sa progression.

Aperçu de la thèse

Cette thèse commence par expliquer, au chapitre 4, toutes les connaissances nécessaires à la compréhension de la thèse. Ces connaissances couvrent le domaine de l'acquisition des données, le domaine du traitement des données et le domaine de l'ingénierie des connaissances. Les travaux de thèse sont ensuite présentés en quatre parties : la revue de littérature (partie II), la méthodologie (partie III), la mise en œuvre (partie IV) et la conclusion (partie V).

La partie II donne un aperçu de la littérature sur la détection d'objets 3D. Cette partie est composée de quatre chapitres. Chacun des trois premiers chapitres présente une catégorie de détection d'objets 3D. Le dernier discute de la comparaison des trois catégories.
Le chapitre 5 présente la première catégorie correspondant aux approches guidées par un modèle. La stratégie des approches dans cette catégorie consiste à créer des modèles pour chaque objet recherché et à les comparer à chaque région de données.
Le chapitre 6 présente la deuxième catégorie correspondant aux approches fondées sur des données. Contrairement aux approches axées sur les modèles, les approches de cette catégorie visent à caractériser les données afin d'en isoler des parties et de les classer selon les objets recherchés.
Le chapitre 7 présente la dernière catégorie d'approches, qui sont des approches fondées sur la connaissance. Les approches de cette catégorie utilisent la connaissance des objets et des données pour adapter le processus de détection.
Enfin, le chapitre 8 compare ces catégories d'approche selon les critères de qualité, d'ambiguïté, de robustesse, de flexibilité et de généralisabilité.

La partie III explique l'approche proposée par cette thèse. Cette partie commence par un aperçu du système. Ensuite, elle explique l'ingénierie des connaissances utilisée par le système. Enfin, elle présente l'approche axée sur la connaissance appliquée par le système pour la détection d'objets 3D.
Le chapitre 9 donne un aperçu du système. Il rappelle les problèmes liés à la détection des objets 3D. Puis il présente les composants du système et leurs interactions.

Le chapitre 10 explique l'ingénierie des connaissances. Celui-ci permet la modélisation de la connaissance des domaines des données, de la scène et du traitement des données. Le but de cette connaissance est de guider le processus de détection d'objet.
Le chapitre 11 présente la détection d'objets basée sur la connaissance. Cette détection est tout d'abord constituée d'une phase de gestion des algorithmes qui consiste à sélectionner, configurer et exécuter les algorithmes pertinents pour traiter le cas d'application. Une phase de classification suit cette gestion des algorithmes. Ces deux phases permettent d'effectuer la détection d'objets selon des connaissances explicitement définies. Cette détection est ensuite suivie d'une étape d'auto-apprentissage visant à enrichir la base de connaissances afin de réexécuter une détection d'objet plus précise.

La partie IV décrit la mise en œuvre de l'architecture de traitement et donne un aperçu de la modélisation et du traitement des cas d'utilisation. Il présente enfin les résultats obtenus par l'approche implémentée.
Le chapitre 12 présente l'architecture implémentée pour l'approche proposée.
Le chapitre 13 présente la modélisation des connaissances pour les cas d'application.
Le chapitre 14 décrit le processus de détection d'objet hiérarchique pour ces cas d'application.
Le chapitre 15 présente les résultats obtenus pour chacun des quatre cas d'utilisation étudiés. Il compare également les résultats obtenus par l'approche proposée avec les approches de la littérature.
Le chapitre 16 traite des choix de mise en œuvre et des résultats obtenus pour conclure sur l'efficacité de l'approche proposée.

Enfin, la partie V avec chapitre 17 résume les contributions apportées par cette thèse et discute des avantages et des limites de l'approche proposée. Il conclut en suggérant des travaux futurs.

Publications

Les travaux de cette thèse ont été diffusés dans les publications suivantes :

[Ponciano et al., 2017] Ponciano, J.-J., Boochs, F., and Trémeau, A. (2017). Knowledge-based object recognition in point clouds and image data sets. gis.Science - Die Zeitschrift für Geoinformatik.

[Ponciano et al., 2019a] Ponciano, J.-J., Boochs, F., and Tremeau, A. (2019a). Identification and classification of objects in 3d point clouds based on a semantic concept. In 3D-Tage, Oldenburger, Germany.

[Ponciano et al., 2019b] Ponciano, J.-J., Karmacharya, A., Wefers, S., Atorf, P., and Boochs, F. (2019b). Connected semantic concepts as a base for optimal recording and computer-based modelling of cultural heritage objects. In Aguilar, R., Torrealva, D., Moreira, S., Pando, M. A., and Ramos, L. F., editors, Structural Analysis of Historical Constructions, pages 297–304, Cham. Springer International Publishing.

[Ponciano et al., 2019c] Ponciano, Jean-Jacques, Trémeau, Alain, and Boochs, Frank. Automatic detection of objects in 3d point clouds based on exclusively semantic guided processes. ISPRS International Journal of Geo-Information, 8(10) (2019c). ISSN 2220-9964. URL http://dx.doi.org/10.3390/ijgi8100442.

[Prudhomme et al., 2017] Prudhomme, C., Homburg, T., Ponciano, J.-J., Boochs, F., Roxin, A., and Cruz, C. (2017). Automatic integration of spatial data into the semantic web. In WebIST 2017 , Porto, Portugal.

[Prudhomme et al., 2019] Prudhomme, C., Homburg, T., Ponciano, J.-J., Boochs, F., Cruz, C., and Roxin, A.-M. (2019). Interpretation and automatic integration of geospatial data into the semantic web. Computing, pages 1–27

[Ponciano et al., 2019a] concerne les chapitres 11, 14 et 15.

[Ponciano et al., 2019b] concerne les chapitres 10, 13 et 15.

[Ponciano et al., 2019c] concerne les chapitres 9, 10 et 11 de la partie III (Methodologie) et les chapitres 12, 13, 14 et 15 de la partie IV (Implémentation).

[Ponciano et al., 2017] concerne les chapitres 9 et 10 .

[Prudhomme et al., 2017] and **[Prudhomme et al., 2019]** concerne les chapitres 4 et 12.

Part I

Introduction

The introduction part is composed of four chapters that aim at providing an overview of the research work presented in this thesis.

Chapter 1 presents the context and motivations for this research. These works are involved in the detection of context-dependent objects. They are motivated by the engineering of a framework to perform this type of detection in any application case automatically.

Chapter 2 explains the scope of the thesis. First, it presents the four application cases considered to evaluate the proposed approach. Then, it gives the problem related to the difficulty of dealing with the detection of objects in very different application cases such as the four cases considered. Finally, it presents the solution proposed in this thesis.

Chapter 3 presents the contributions. First, it presents the contributions made in the different research disciplines. Then, it gives an overview of the thesis. Finally, it sets out the published research work.

Chapter 4 presents the background of this thesis through the following three domains: data acquisition, data processing, and knowledge engineering. Finally, this chapter discusses the usage of these three domains within the thesis.

1 Context and motivation

Through the power of new sensing technologies, we are increasingly digitizing the real world. However, instruments produce unstructured data, mainly in the form of point clouds for 3D data and images for 2D data. Nevertheless, many applications (e.g. navigation or as-built documentation) need structured data containing objects and their geometry. Structuring data requires to interpret its content. This interpretation corresponds to the identification of objects and their geometries contained in the data. The application determines the objects and their geometries.

Moreover, an application requires a data quality, which depends on the needs and constraints of the application. Data quality is determined by data characteristics (e. g. density, texture, color, noise, resolution, regularity, completeness) that satisfy the application's requirements. The data characteristics depend on the sensing process. Application guides the choice of sensing process corresponding to the choice of methodologies (e.g. set up and measurement strategy), technologies of acquisition (such as instruments) and the acquisition context (e.g. object, scene, environmental conditions). For example, in archaeology, the acquisition of a virtual copy of an object may require a data resolution of 0.01 mm, while for excavations, a few millimeters may be sufficient. In this example, the acquisition technology is chosen based on its ability to provide appropriate data resolution. Other applications require the preservation of the color and texture of the acquired scene. In these cases, 2D acquisition combined with photogrammetry, which builds a 3D point cloud exclusively from several 2D images, is generally preferred to laser-scanner technologies due to its performance in preserving textures. In the industrial field, autonomous cars or urban infrastructure planning, require the acquisition of large scenes. Therefore a laser scanner (as Lidar scanner) is mainly used for this application to acquire 3D scenes from very long distances.

However, the sensing process often produces lower data quality than required by the application. Compromises between the choice of methodologies and technologies on the one hand, and the constraints of the acquisition context, on the other hand, produces such difference. Therefore, it is necessary to compensate for the difference between the data quality obtained by the sensing process and the expected quality. This compensation can be achieved by integrating information on the sensing process (acquisition technologies, acquisition methodology, and acquisition context) to adapt the interpretation of the data content to its quality.

Several Computer Vision approaches aim to interpret the data content and try to compensate for the lower quality of the data. These approaches are mainly context-dependent, that means they are specific to the data to be processed and to the objects to the detected. Some approaches (called Model-Driven) mainly consider part of the information on the scene context and on the objects contained in the data. These approaches use information about the geometry and shape of objects to build corresponding models. These approaches examine the data to identify sections that are similar to the models constructed. The identified sections allow the interpretation of objects and their geometry to structure the data. However, these approaches use only part of the information on the acquisition context and do not exploit information on acquisition methodologies or acquisition technologies. These approaches are, therefore, not able to compensate for weaknesses in data quality. The results of these approaches depend directly on the quality of the data.

Other approaches (called Data-driven approaches) do not directly exploit information on acquisition methodologies and acquisition technologies but focus on the resulting data quality. These approaches are mainly composed of three main steps. The first step is to prepare the data to compensate for weaknesses in data quality (such as noise) by removing elements whose information is irrelevant or which may mislead processing. The second step consists in segmenting the data into sub-regions by ad hoc algorithms or combinations of algorithms. This step is intended to facilitate the interpretation of the data by dividing its complexity. Each sub-region provides clues to interpret the content of the data. The last step is to classify each sub-region to interpret the data content. However, these approaches do not directly use the information on the acquisition context, acquisition methodologies, or acquisition technologies. Thus, they cannot understand the fundamentals of data quality and must, therefore, compensate for any weaknesses in data quality. Compensating for all data quality weaknesses is often too tedious in some applications to be successful.

Therefore, it is logical to explore opportunities to use the information on acquisition methodologies and acquisition technologies to adapt data interpretation to the data quality. Therefore, it is necessary to know the quality of the data to guide the interpretation of the data. Thus, it is necessary to use the information on the methodologies and acquisition technologies to estimate data quality. Finally, it has to use the information on the acquisition context (mainly scenes and their components) to interpret the content of the data. This interpretation is carried out by classifying the data sections. Considering all of this information allows for a more robust interpretation of data content than other approaches (Model-driven and Data-driven). Moreover, understanding this information allows adaptation to different types of data contexts. However, the quality estimated may differ from the quality obtained. This difference creates unpredictable problems for the interpretation of data content. It is necessary to solve the unpredictable problems caused by the difference between the estimated data quality and the data quality obtained by the sensing process. One approach to compensate for this quality difference is to understand the derivation between the estimated data quality and the obtained data quality. This understanding allows for the adaptation of the data quality estimation. Thus, this adaptation allows for the interpretation of the data content to be adapted. Besides, it is difficult to anticipate the deviations between the estimated and the obtained quality. Therefore, these deviations need to be observed and then analyzed for the entire data. This treatment requires a local assessment of the data quality obtained. Moreover, the results produced by interpreting the data content provide clues to the obtained data quality. Thus, the analysis of these results allows deducing new information on the obtained data quality. This new information allows the interpretation of the data content to be adapted. This adaptation allows for better structuring of data according to their quality.

2 Scope of the thesis

2.1 Use cases

In this thesis, we focused on four different applications cases to illustrate the different complexities that can exist in the major fields of object detection. These application cases come from different application domains (e.g. cultural heritage, as-built documentation) in different contexts (e.g. ruins excavation, indoor building, urban outdoor, indoor room). They have unstructured 3D data, in the form of a point cloud and require the detection of various objects (e.g. wall, floor, ceiling, car, chair table, watermill, tree) to structure their data. In this thesis, we focus on data structuring. Therefore we focus on the detection of the elements that constitute the data. This task involves the main scientific difficulties.

2.1.1 Indoor use case

The first application case consists of documenting a building by reconstructing a 3D model of it. To this end, it is necessary to identify each of the rooms that make up the building. The identification of rooms requires the identification of walls, floors, and ceilings. These elements are large objects whose colors can be highly variable. In modern buildings, these elements have mainly a regular and rectangular shape.

The building chosen for this application case is digitized by laser-scanner in indoor context, producing an unstructured 3D point cloud [*Armeni et al.*, 2017]. Thus several factors influence the representation of floors, walls, and ceilings.

The use of laser scanners does not allow to acquire objects made of glass or having a highly reflective surface. Thus, some parts of the digitized walls are missing when the walls are composed of windows or reflective surfaces. Also, the acquisition method consists in moving the scanner inside the building following a path, which generates occlusions. Thus, this data set contains many discontinuities, mainly for walls and ground due to their occlusions by elements inside rooms. Moreover, among the main external factors, the artificial color of the light and the variations in light from one room to another influence the acquisition process by generating noise and influences the colors of the digitized elements respectively. Figure 2.1 illustrates the dataset of this application case.

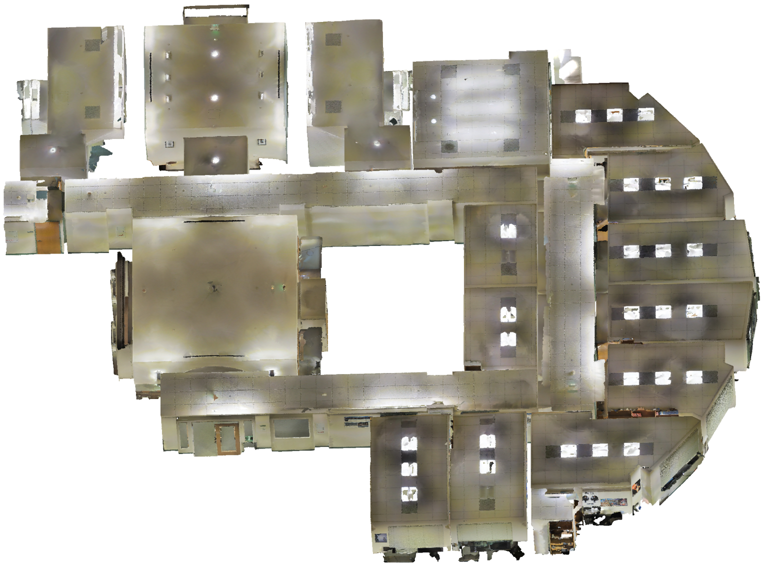


Figure 2.1: Presentation of the Stanford 3D Large-Scale Indoor Spaces Dataset in point cloud [*Armeni et al.*, 2017].

The main challenge for Computer Vision approaches in this application case is the segmentation of rooms that have common characteristics and elements. For example, two different rooms may share some ceiling, floor, or walls.

This data set is open access allowing the comparison of the proposed approach in this thesis with other approaches existing in the literature.

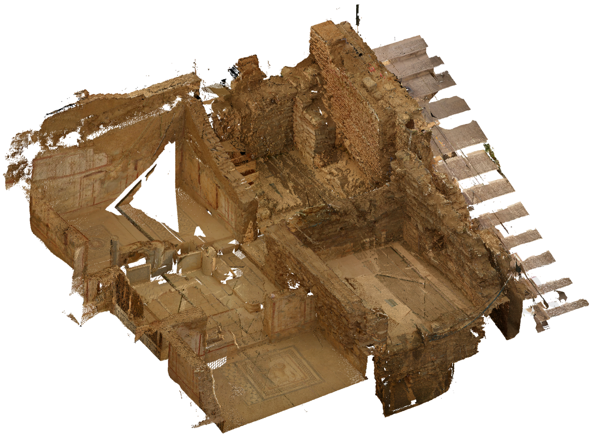
2.1.2 Cultural heritage use case

The second application case comes from the domain of cultural heritage and represents a terrace house partially destroyed in Turkey. The purpose of this application is to support archeological interpretation through knowledge-based object detection.

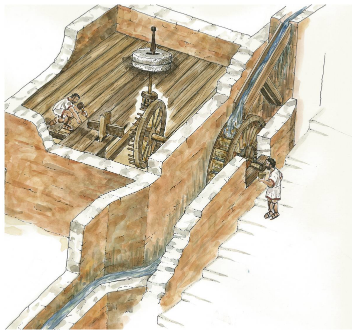
The main object sought in this application case is a watermill. This object is described as being composed of two main parts (a big room and a narrow room), one of which contained the mill wheel and the other the mechanism.

In this application case, a scene of ruins excavation is digitized by Terrestrial laser scanner (TLS) [*Lemmens*, 2011], producing an unstructured 3D point cloud [*Wefers*, 2015]. However, in the context of ruins excavations, the watermill did not endure the time, and only the structure of some of its rooms was preserved. The ceiling and a large part of the walls are missing, which increases the difficulty of detecting the watermill. Also, the use of laser-scanner on this point cloud produces some irregular shapes related to the ground structure and walls that are not planar, complicating the use and configurations of algorithms.

Figure 2.2 shows the acquired point cloud and a model of the watermill. The main



(a) The terrace house acquired in point cloud.



(b) Drawing of the watermill expected.

Figure 2.2: Presentation of the terrace house with watermill.

challenge in this application case is to adapt the detection process to the context of ruin excavation, which considerably transforms the characteristics of the objects.

2.1.3 Urban outdoor use cases

The third application case is the acquisition of a scene in a context of urban outdoor acquired by a laser-scanner. This acquisition technology produces an unstructured 3D point cloud. The point cloud is obtained by sequentially scanning the scene with the laser moving on a car along the road. In this application, many factors (variation in brightness, the vibration of the measuring instrument, meteorological condition) influence the acquisition process. Thus, they influence the characteristics (e.g. noise, density, regularity) of the generated point cloud. Besides, the sequential scan acquisition method causes data acquisition in the form of "steps" on the one hand and generates occlusion areas (without information) on the other hand. Moreover, the different materials (e.g. metal, glass, stone) and the distance between the objects and the laser-scanner at the time of acquisition greatly influence the acquisition process. Thus, portions of data can be dense and continuous, and others can be discontinuous with low density. Furthermore, the complexity of the different factors influencing the acquisition process leads to the generation of unpredictable situations and problems where the characteristics of the data obtained differ significantly from the expected characteristics.

The main challenge in this application case is to address problems arising from unpredictable changes in data characteristics. Two interesting point clouds address this application case.

The first point cloud comes from the industrial domain and aims at documenting parts of the city of Freiburg in Germany. Parts of the city are digitized by Fraunhofer IPM[1] using Lidar laser-scanner.

Figure 2.3 illustrates a part of this data set.

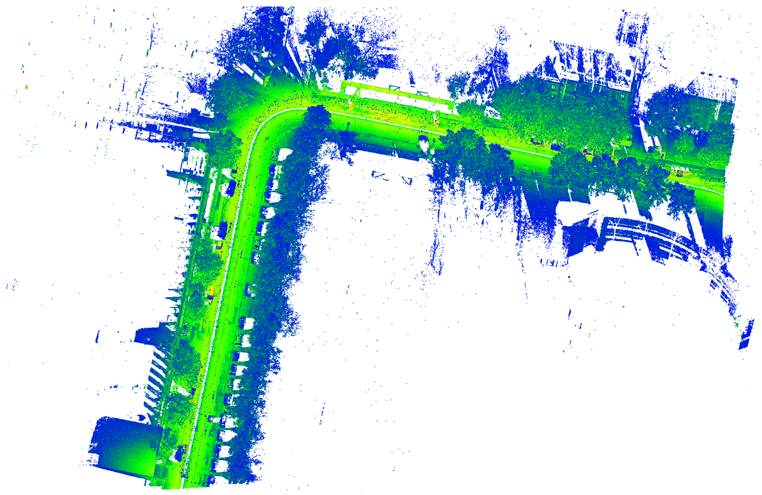


Figure 2.3: Presentation of the part of Freiburg in point cloud (fictively colored to increase visibility).

The other point cloud corresponds to the dataset "Paris-rue-Madame database: MINES ParisTech 3D mobile laser scanner dataset from Madame street in Paris" [2] [*Serna et al.*, 2014]. It represents rue Madame, a street in the 6th Parisian district in France. This street is digitized by a Mobile Laser Scanning (MLS) system from the Robotics laboratory (CAOR) at MINES ParisTech (Paris, France). Figure 2.4 illustrates this dataset.

The main objects to detect in this application case of urban outdoor are cars, traffic signs, walls (sometimes also called facade in this context), floor, and motorcycle. In the context of urban outdoor scenes, these objects have diverse shapes and characteristics.

[1] Fraunhofer IPM: https://www.ipm.fraunhofer.de/de/presse_publikationen/Presseinformationen/messfahrzeug-3D-Daten-breitbandausbau.html

[2] MINES ParisTech© copyright. MINES ParisTech created this special set of 3D MLS data for the purpose of detection-segmentation-classification research activities, but does not endorse the way they are used in this project or the conclusions put forward.

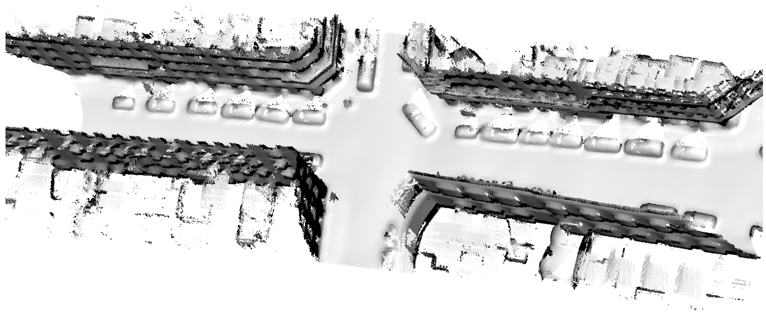


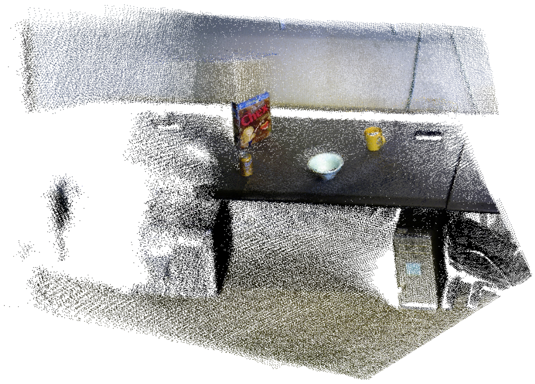
Figure 2.4: Illustration of "Paris-rue-Madame database: MINES ParisTech 3D mobile laser scanner dataset from Madame street in Paris".

2.1.4 Time-of-flight use case

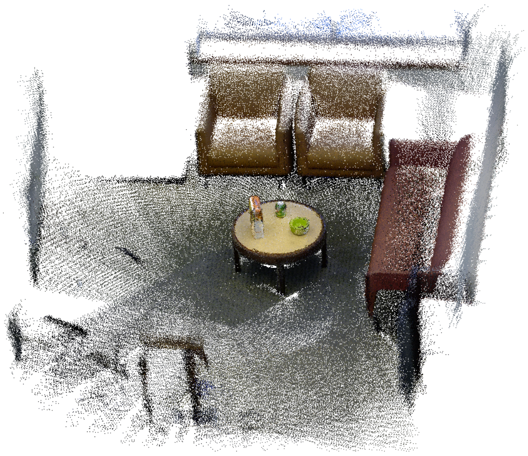
The fourth application case aims to locate specific objects in different indoor scenes for monitoring. In this use case, the objects sought are furniture such as chairs and tables, whose shape and color differ from one scene to another. Thus the geometric and physical characteristics of objects are diverse.

The scenes chosen for this application case are digitized by [*Lai et al.*, 2011] using Microsoft's Kinect that produces 3D point clouds, as shown in Figure 2.5. Each point cloud is composed of more than 1 million points.

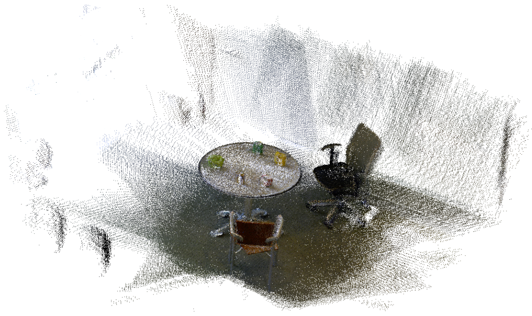
The use of Microsoft's Kinect produces low-quality of the data set with mostly irregular surfaces. Moreover, the acquisition process of these point clouds produces many missing parts due to occlusions and sparse acquisition. These characteristics increase the difficulty of object detection. The main challenge, in this application case, is to detect objects even if their characteristics and shapes change considerably (e.g. round table, square table, office chair, wooden chair).



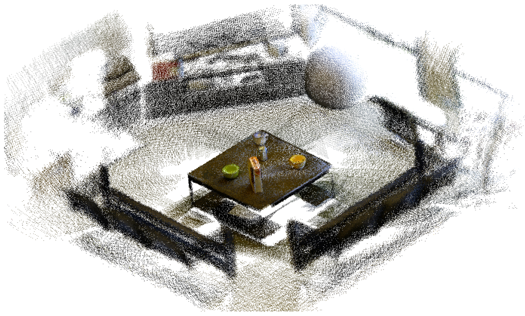
(a) First scene in point cloud.



(b) Second scene in point cloud.



(c) Third scene in point cloud.



(d) Fourth scene in point cloud.

Figure 2.5: Presentation of the four scenes [*Lai et al.*, 2011].

2.2 Problem statement

Understanding unstructured data is a real challenge. It requires detecting the representations in the data of the objects and geometries of the digitized scene. The detection of objects and geometries depends on their characteristics (e.g. size, shape) on the one hand, and the characteristics of the data on the other hand (e.g. density, noise, occlusion, roughness). The characteristics of the data depend on the acquisition process (e.g. technology and methodology used) that generates them. The characteristics of the digitized objects (e.g. material, reflectance, roughness, size), the context of the scene (e.g. urban outdoor, indoor building, ruin excavation), and various other factors external to the acquisition process (e.g. ambient light, light intensity, weather conditions, movement of the measuring instrument or digitized objects) influence the acquisition process. Small variations in one of these factors can strongly influence the characteristics of the data. These variations in characteristics generate divergences between the expected and obtained characteristics.

The approaches based on Machine Learning (such as deep-learning, continuous learning) substitutes the lack of understanding of these factors through the identification of reliable patterns. These patterns have to be learned in a training stage. That is why "reliable" Machine Learning requires a vast amount of data. However, in some application domains, such as cultural heritage (see application case 2.1.2), annotated data are not available or are not sufficient. Thus, approaches based on Machine Learning are irrelevant for detecting the object in such data set. Moreover, Machine Learning approaches work as long as the data considered is representative of the content to be understood. The more variation in object or appearance occur, the more data is required for training. Machine learning approaches remain unflexible and are unable to detect the object or the geometry for which they are not trained. Furthermore, just a small change of the acquisition process or external factors that influence the data can result in a change for which the Machine learning approaches are not trained.

On the contrary, other approaches based on semantic technologies try to integrate some knowledge on object or sensing process to drive the data understanding process. Nevertheless, these approaches are unable to formulate knowledge in a way that represents all cases.

Thus, current approaches are not robust, flexible, and generalizable enough to overcome these differences. Their lack of robustness limits the understanding of the data (e.g. inadequately identified or unidentified portions of data). Moreover, these approaches are context-dependent that means they are specific to the data to be processed and to the objects to the detected. Therefore, the issue of this thesis is how to provide a robust object detection process in the face of data characteristics variations and no context-dependent.

Dealing with variations in data characteristics requires an understanding of their origin and their impact on the data understanding process. Understanding is based on knowledge, itself based on connecting information. Therefore, an approach, based on knowledge about the source and impact of data characteristics variations, is needed to guide the process of data understanding. Solving this problem using a knowledge-based approach requires solving three sub-problems. The first sub-problem is how to represent knowledge about the source and impacts of data variations. The second sub-problem is how to provide the missing flexibility in other knowledge-based approaches to provide a detection process adapted to variations. The third sub-problem, also related to flexibility, is how to provide dynamic and adapted knowledge enrichment to overcome the lack of knowledge about the diversity of cases represented in the data.

2.3 Solution proposed

That is why the approach proposed in this thesis first addresses the first sub-problem by considering the data acquisition process and the different factors influencing it. This approach attempts to understand the origin of data characteristics by understanding the influences of the characteristics of the acquisition process and the different factors influencing it. Such an understanding allows for the anticipation of data characteristics. This understanding requires explicit knowledge of the domains of data, scene, and data processing, as well as knowledge of the influence between these domains.

Data domain: The data domain is composed of knowledge about the characteristics of the data and knowledge about the acquisition process (e.g. acquisition methodology, acquisition technology, acquisition instrument).

Scene domain: The domain of the digitalized scene is composed of knowledge about objects, their distribution in the scene, the scene context, and external factors.

Data processing domain: Finally, knowledge of algorithms (e.g. conditions of use, purpose of the algorithm, configuration, preferred usage conditions) to be used to process the data constitutes the data processing domain.

The knowledge about these domains and their influences allows for establishing an efficient reasoning base.

Therefore the presented approach uses this explicit knowledge to guide data processing. Computer Vision algorithms from various libraries (such as *PCL* [*Rusu and Cousins*, 2011], *OpenCV*[3]) are used to process the data. Similarly, Semantic Web technologies allow knowledge management. The approach presented then responds to the second sub-problem by combining these two paradigms through both a technical and a conceptual bridge. This bridge allows knowledge to fully drive the data processing process step by step, through a continuous exchange between the two paradigms. This continuous exchange allows the object and geometry detection process to be adapted according to the newly acquired knowledge at each step of the data processing.

The proposed approach begins by combining data and scene knowledge to infer and anticipate data characteristics (such as roughness, density, occlusion) and thus enrich data knowledge. This combination of knowledge also increases knowledge about the possible representations of objects in the data. For example, laser scanner technology does not acquire glass material correctly. Thus, a table with wooden legs and glass tray will only be represented in the data by these legs.

Then, the approach combines knowledge on the three domains to automatically and adequately select and configure algorithms according to this knowledge, and thus according to the application case under consideration. Algorithms must be selected and configured according to the results already obtained by previously executed algorithms to obtain efficient data processing. That is why the proposed approach automatically interprets the data resulting from the execution of algorithms to enrich the knowledge. Thus the algorithms are dynamically selected and configured according to the evolution of data processing.

The approach then combines the knowledge from the algorithms results with knowledge about how objects are represented in the data to understand and structure it.

Although such an understanding may be sufficient to structure the data in some cases entirely, it may not be sufficient in others. Thus, the data structuring may be of insufficient quality for the application in question. This case occurs when knowledge about object representations differs from the real representations of objects in the data. In this case, it is necessary to increase knowledge about the

[3] Laganière, R. (2014). OpenCV Computer Vision Application Programming Programming Cookbook Second Edition. Packt Publishing Ltd.

representation of objects according to the data characteristics.

That is why we propose to integrate a new method of automatic knowledge generation, answering the third sub-problem. This method consists of using existing knowledge as a learning base to build and test new knowledge. This self-learning process consists of gathering and combining the information contained in the knowledge to formulate new hypotheses. These hypotheses are then tested by verifying that their integration into the original knowledge does not create inconsistency. In other words, the knowledge does not include a counter-example to this hypothesis. If a hypothesis does not create inconsistency, then it is considered coherent and integrated with existing knowledge. The integration of new knowledge enriches knowledge about object representations and changes the behavior of the detection process.

Consider, for example, a part of the data representing a table. If this part was badly acquired, and no knowledge was available to anticipate this bad acquisition, then the knowledge about the table representations may not correspond to the table representation in the data. In this case, the table would not be detected. In this case, the knowledge-based self-Learning process would collect all the information it has about the other table representations that have been detected. Suppose that chairs surround all the other tables detected. Then the process would formulate the hypothesis that "if chairs surround an object, then that object is a table." He will then verify that the knowledge does not contain a counter-examination to this hypothesis. In this case, the verification consists of verifying; there is no object surrounded by a chair that is not a table in the knowledge. If the hypothesis is validated, then this new knowledge is integrated with existing knowledge. Now suppose that detected chairs surround the portion of the data corresponding to the undetected table. Then the new knowledge added would identify this portion of data as a table.

The self-learning process aims to adapt knowledge about object representations to match the real representations of objects in the data. In this way, it improves the understanding of unstructured data.

3 Contributions and thesis overview

The work presented in this thesis aims at producing an object detection process that is robust to the variation of 3D data quality and can be used whatever the application field. In other words, this process has to stay efficient, regardless of the data or object specificities. The achievement of this purpose is done by using human knowledge and by dynamically adapting this knowledge during the object detection process. Therefore our main contributions are in the Artificial Intelligence domain with the creation of a knowledge-based self-learning process, and also in the Semantic Web domain and Computer Vision domain.

3.1 Contributions

3.1.1 Semantic Web

We have three main contributions in the Semantic Web domain. The first contribution is an automatic integration of information (such as meta-data within datafile or geographic information) into the Semantic Web. This work has been published in [*Prudhomme et al.*, 2017] and in [*Prudhomme et al.*, 2019].

The second contribution is an extension of *SPARQL*, which is a standard technology in the Semantic Web. This extension consists of *SPARQL* built-ins for data processing in Computer Vision.

The third contribution is the automatic interpretation of *OWL-restriction* and *OWL2-restriction* into *SPARQL* queries.

These three contributions are combined into a framework that executes Computer Vision algorithms through only *SPARQL* queries automatically from reasoning processes. This reasoning process is applied to the ontology that contains all the necessary descriptions. Thus, we create a bridge between Semantic Web knowledge engineering and Computer Vision algorithms.

3.1.2 Computer Vision

Beyond proposing a comprehensive state of the art of the different Computer Vision approaches to detect objects in a 3D point cloud, we have three main contributions in the Computer Vision domain.

Firstly, we propose an automatic adaptation of the objects detection process according to the context (i.e., application case, acquisition process, and data characteristics) and objects sought. This approach considers the data acquisition and context to identify objects whose geometric characteristics are not sufficient or usable to recognize them. This work has been applied in the field of cultural heritage and has resulted in the publication [*Ponciano et al.*, 2019*b*].

Secondly, we propose a system that automatically selects and configures algorithms for 3D object detection through the use of a knowledge base and a reasoning mechanism. This system selects and parameterizes algorithms according to the objects sought, the data used, and the algorithms prerequisites. This work has been published in [*Ponciano et al.*, 2017].

Thirdly, we propose a process for detecting objects, which is prioritized according to the size of the objects and their topological relationships. The hierarchical detection reduces the area of an object search to locations where the object can be. This detection strategy improves the performance of algorithms applied to localized portions of the data rather than to the entire data. These works have been published in [*Ponciano et al.*, 2019*a*].

The combination of these three contributions produces a robust and powerful object detection system. This system obtains excellent results for different application domains and objects sought.

3.1.3 Artificial Intelligence

The main contribution to the Artificial intelligence domain is a knowledge-based self-learning process. We create a system able to adapt its knowledge according to the experience obtained from a first detection process. This system formulates assumptions and then validates these assumptions by an analysis of the consequences produced by their applications. Such behavior allows the system for growing by itself without any prerequisites, even if the integration of human knowledge improves its progression. These works have been published in [*Ponciano et al.*, 2019*c*].

3.2 Thesis overview

This thesis begins by explaining, in Chapter 4, all the necessary knowledge for the understanding of the thesis. This knowledge covers the Data Acquisition domain, the Data Processing domain, and the Knowledge Engineering domain. The thesis works are then, presented through four parts: the literature review (Part II), the methodology (Part III), the implementation (Part IV), and the conclusion (Part V).

Part II provides an overview of the literature on 3D Object Detection. This part is composed of four chapters. Each of the three first chapters presents a category of 3D object detection. The last one discusses the comparison of the three categories.
Chapter 5 presents the first category corresponding to model-driven approaches. The approaches strategy in this category is to create models for each object sought and compare them to each data region.
Chapter 6 sets out the second category corresponding to data-driven approaches. Unlike model-driven approaches, approaches in this category aim at characterizing data to isolate portions of it and classify these portions according to the objects sought.
Chapter 7 introduces the last category of approaches, which are Knowledge-driven approaches. Approaches in this category use knowledge about objects and data to adapt to the detection process.
Finally, Chapter 8 compares these categories of approach according to the criteria of quality, ambiguity, robustness, flexibility, and generalizability.

Part III explains the approach proposed by this thesis. This part begins with an overview of the system. Then, it explains the knowledge engineering used by the system. Finally, it presents the Knowledge-driven approach applied by the system for the detection of 3D objects.
Chapter 9 provides an overview of the system. It recalls the problems related to the detection of 3D objects. Then it presents the system components and their interactions.
Chapter 10 explains knowledge engineering. This one allows the modeling of the knowledge of the domains of data, the scene, and data processing. The purpose of this knowledge is to guide the object detection process.
Chapter 11 presents knowledge-driven object detection. This detection is, first of all, made up of an algorithm management phase which consists in selecting, configuring, and executing the relevant algorithms for processing the application case. A classification phase follows this management of algorithms. These two phases

allow object detection to be performed according to explicitly defined knowledge. This detection is then followed by a self-learning step aimed at enriching the knowledge base in order to re-execute a more accurate object detection.

Part IV describes the implementation of the processing architecture and provides an overview of use cases modeling and processing. It finally presents the results obtained by the implemented approach.
Chapter 12 presents the architecture implemented for the proposed approach.
Chapter 13 presents knowledge modeling for application cases.
Chapter 14 describes the hierarchical object detection process for these application cases.
Chapter 15 presents the results obtained for each of the four studied use cases. It also compares the results obtained by the proposed approach with approaches from the literature.
Chapter 16 discusses the implementation choices and the results obtained to conclude on the efficiency of the proposed approach.

Finally, Part V with Chapter 17 summarizes the contributions brought by this thesis and discusses the advantages and limits of the proposed approach. It concludes by suggesting future works.

3.3 Publications

The works of this thesis have been published through the following publications:

[Ponciano et al., 2017] Ponciano, J.-J., Boochs, F., and Trémeau, A. (2017). Knowledge-based object recognition in point clouds and image data sets. gis.Science - Die Zeitschrift für Geoinformatik.

[Ponciano et al., 2019a] Ponciano, J.-J., Boochs, F., and Tremeau, A. (2019a). Identification and classification of objects in 3d point clouds based on a semantic concept. In 3D-Tage, Oldenburger, Germany.

[Ponciano et al., 2019b] Ponciano, J.-J., Karmacharya, A., Wefers, S., Atorf, P., and Boochs, F. (2019b). Connected semantic concepts as a base for optimal recording and computer-based modelling of cultural heritage objects. In Aguilar, R., Torrealva, D., Moreira, S., Pando, M. A., and Ramos, L. F., editors, Structural Analysis of Historical Constructions, pages 297–304, Cham. Springer International Publishing.

[Ponciano et al., 2019c] Ponciano, Jean-Jacques, Trémeau, Alain, and Boochs, Frank.

Automatic detection of objects in 3d point clouds based on exclusively semantic guided processes. ISPRS International Journal of Geo-Information, 8(10) (2019c). ISSN 2220-9964. URL http://dx.doi.org/10.3390/ijgi8100442.

[Prudhomme et al., 2017] Prudhomme, C., Homburg, T., Ponciano, J.-J., Boochs, F., Roxin, A., and Cruz, C. (2017). Automatic integration of spatial data into the semantic web. In WebIST 2017 , Porto, Portugal.

[Prudhomme et al., 2019] Prudhomme, C., Homburg, T., Ponciano, J.-J., Boochs, F., Cruz, C., and Roxin, A.-M. (2019). Interpretation and automatic integration of geospatial data into the semantic web. Computing, pages 1–27

[Ponciano et al., 2019a] concerns chapters 11, 14, and 15.

[Ponciano et al., 2019b] concerns chapters 10, 13, and 15.

[Ponciano et al., 2019c] concerns chapters 9, 10 and 11 of Part III (Methodology) and chapters 12, 13, 14 and 15 of Part IV (Implementation).

[Ponciano et al., 2017] concerns chapters 9 and 10 .

[Prudhomme et al., 2017] and **[Prudhomme et al., 2019]** concern chapters 4 and 12.

4 Background

This chapter presents the main aspects constituting the background of this thesis. Section 4.1 explains the data acquisition process and the different factors that influence the data characteristics. Section 4.2 presents data processing through the use of algorithms and explains the general knowledge that defines each algorithm. Finally, Section 4.3 presents how the information relevant to understand unstructured data is modeled as formal knowledge. It presents the leading technologies that allow the exploitation of this knowledge.

4.1 Data acquisition

Different applications such as robotic control [*Liu*, 2015], as-built documentation [*Giel and Issa*, 2011], site modelling [*Farjas et al.*, 2003], quality control [*Habib et al.*, 2008], payload monitoring [*Bewley et al.*, 2011] need to acquire scenes as 3D data and structure this data to exploit them. The needs of the application determine the choice of technologies and acquisition methods that generate the data. However, the acquired scene, as well as external factors, influence the acquisition process. Therefore, each of these aspects influences the characteristics of the data. This section presents the influence of these different aspects on data characteristics through a sample of data characteristics. This sample contains data characteristics that frequently impact the understanding of the data: density, noise, occlusion, roughness, color. Density and noise characteristics cause many problems in understanding the content of the data as highlighted by [*Velizhev et al.*, 2012], and [*Hackel et al.*, 2016*a*]. In addition to problems due to density and noise, the work [*Nguyen and Le*, 2013] also highlights problems caused by occlusions. Meanwhile, the work [*Lague et al.*, 2013] highlights problems linked to the roughness. Another characteristic of data that is important in the data understanding is the color as shown by works of [*Zhan et al.*, 2009] and [*Strom et al.*, 2010].

4.1.1 Acquisition process

The evolution of applications requiring the acquisition of 3D data has led to the development of various acquisition techniques. The works [*Aboali et al.*, 2017] presents a review of the different acquisition techniques. Among these acquisition techniques, the 3D Laser Scanner [*Jecić and Drvar*, 2003] and Light Detection and Ranging (LIDAR) [*Reutebuch et al.*, 2005], the photogrammetry [*Remondino et al.*, 2008] [*Hartley and Mundy*, 1993], and Time-of-flight camera [*Fürsattel et al.*, 2015] are the most commonly used.

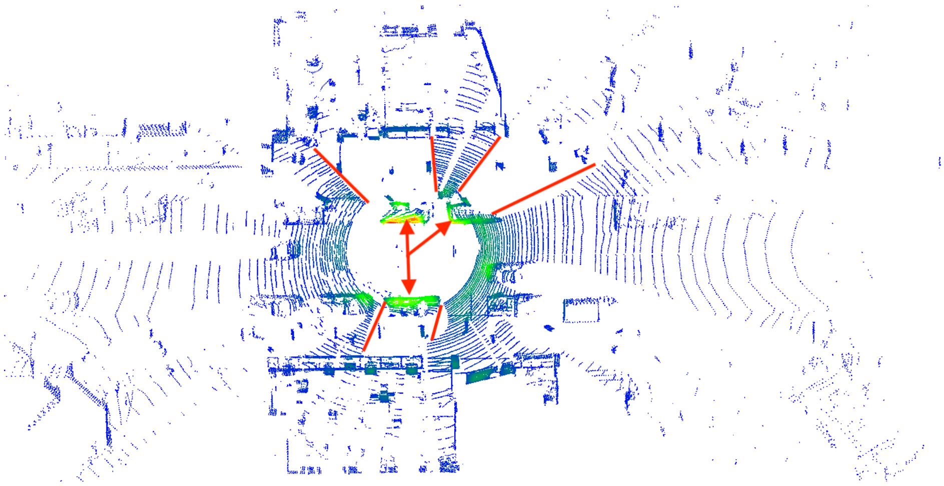
The use of these techniques depends on the purpose and the needs of each application. Let us take, for example, two main methods of remote sensing: the "active" method and the "passive" method. The "active" acquisition methods consist of projecting light onto the area to be digitized. These methods are mainly used with laser scanner technologies such as Lidar. "Passive" acquisition methods use photogrammetry. They consist of calculating a 3D representation in the form of a point cloud of an area to be digitized, based on other data such as images. Photogrammetry methods require the use of algorithms to match the pixels of the stereo images used to produce the point cloud. The algorithms that perform this matching are based on the comparison of the texture of stereo images. Thus, when scenes have few or uniform textures (such as dense drills, blacktop parking lot), algorithms are not able to match the textures. Thus, these techniques cannot correctly digitize the elements. On the other hand, using "active" techniques are less affected by the textures of the elements to be digitized. Thus, such techniques as scanning by Lidar scanner are more efficient than photogrammetry techniques for digitizing scenes with low or uniform texture.

One of the primary data characteristics that depend on the acquisition process is color. In photogrammetry, the information on the color of the elements comes from the images and can be directly calculated for the point cloud obtained. On the contrary, acquisition techniques using laser scanners (such as Lidar) first provide a colorless point cloud. Then, these techniques add color to the point cloud if necessary, using information from images. The colors and textures calculated in this way may be less accurate than those obtained by photogrammetry.

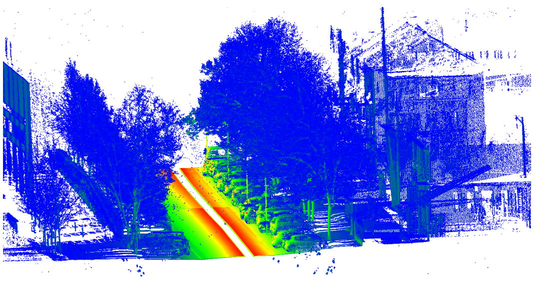
The pipeline used in the acquisition process also plays an essential role in the characterization of the generated data. The works [*Bernardini and Rushmeier*, 2002], [*Son et al.*, 2015], [*Pauly et al.*, 2005] and [*Vrubel et al.*, 2009] present the pipeline of 3D data acquisition in different fields.

One data characteristic that is highly dependent on the pipeline of the acquisition process is the data density. The data density is mainly defined according to the resolution provided by the acquisition technology and decreases with distance of the elements from the acquisition instrument. Thus, the acquisition made by a single scan of an area will cause discrepancies in density, while an acquisition made by multiple scans will provide a more homogeneous density. Therefore, the acquisition pipeline used greatly influences the characteristics of the data obtained. Besides, some acquisition technologies such as 3D Lidar scanners produce discrete "foot pulse" data depending on the type of scanning performed by the scanner (linear or circular). This discretization also affects the density of the data.

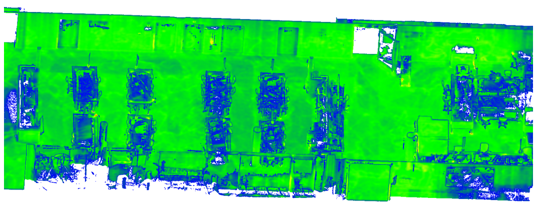
To illustrate these influences, let us take the acquisition examples of two urban scenes and an interior scene. A 3D laser scanner acquires these three scenes. Figure 4.1 shows the influence of the acquisition process on the density and occlusions of the data. These point clouds are colored according to the density of the points going from red for the densest areas, to blue for the less dense areas.



(a) Point cloud acquired by laser-scanner with circular scanning (provided by [*De Deuge et al.*, 2013]).



(b) Point cloud acquired by laser-scanner with linear scanning.



(c) Point cloud acquired by recursive scanning (provided by NavVis company [*Wu et al.*, 2018]).

Figure 4.1: Comparison of the density of point clouds acquired by 3D laser scanner with three different acquisition methods.

The urban scenes in Figure 4.1(a) and in Figure 4.1(b) are digitized by circular and sequential scanning, respectively,. They follow a linear path. This method of movement combined with the acquisition technique causes a lot of occlusions (illustrated by red arrows and lines in Figure 4.1(a)). Besides, the data density decreases gradually with the distance of the digitized elements from the measuring instrument. Thus the data have a high variation in density.

On the contrary, the indoor scene (Figure 4.1(c)) is acquired by recursive scanning. In other words, the measuring instrument is moved inside the room, scanning the same areas several times. This method reduces the occlusion occurrence on the one hand, and unifies the data density, on the other hand, as shown in Figure 4.1.

Moreover, the data may be more or less noisy, depending on the technologies used. Generally, data from Time-of-flight technology produces more noise than data from other technologies such as 3D laser scanners.

In 3D point clouds, noise corresponds to points that do not coincide with any digitized element. Therefore, they do not provide any information on the data but can lead to misinterpretation.

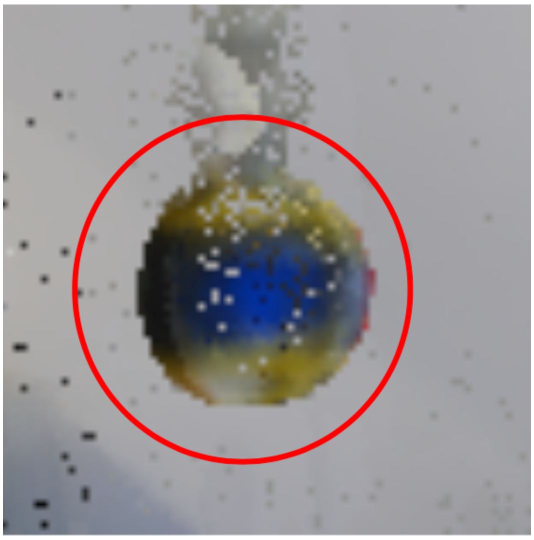
4.1.2 Factors of influence

Several factors influence the acquisition process. These factors are grouped into two categories: internal and external factors.

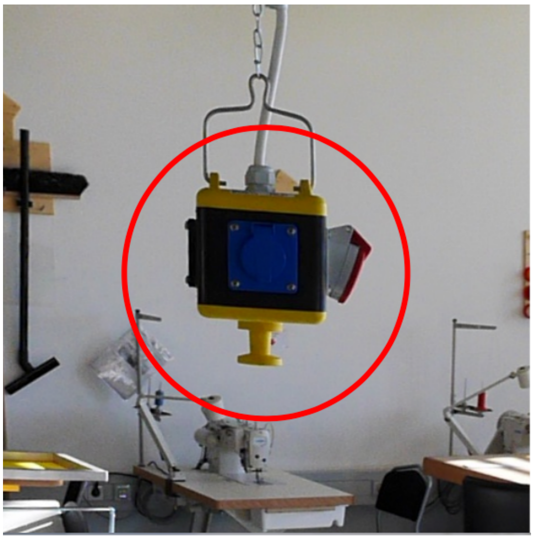
Internal factors

The internal factors are related to the digitized scene, the objects it contains, their distributions, the geometries, and the materials of these objects. The distribution and geometry of the objects influence the acquisition process by affecting the difficulty of to digitize a scene. When the scene to digitize is composed of a single object, the acquisition process is not congested, and the generated data does not include occlusions. On the contrary, when the scene is composed of several objects, it becomes more complex to digitize the scene, which leads to more occlusions. The more objects the scene is composed of, the more difficult it is to digitize completely, and the higher the risk that the generated data will have occlusions. Moreover, the more complex the geometry of an object is, the more difficult it is to digitize it accurately. This difficulty can lead to a wide variety of data characteristics. Figure 4.2 shows the deformation of an object whose original geometry is mainly cubic while the geometry obtained in the point cloud is spherical.

The characteristics of the objects belonging to the digitized scene, such as color, tex-



(a) Data acquired by a 3D laser scanner.



(b) Picture of the digitized object.

Figure 4.2: Illustration of the geometric deformation of an object during its digitization by a 3D laser scanner.

ture, and material, can influence the acquisition process. These influences depend on the type of technique used by the detection process.

For example, the texture of objects (see Section 4.1.1) strongly influences the acquisition techniques based on photogrammetry. While the materials of objects profoundly influence technologies using "active" acquisition methods such as laser scanners. More precisely, the reflectivity and transparency of the objects impact the light projected on them by these acquisition techniques. Thus they affect the objects digitization. Figure 4.3 shows the digitization by laser scanner techniques of a glass table surrounded by chairs. In this case, the data generated for the digitization of the table has a very low density compared to the density obtained for the chairs (in black in Figure 4.3).

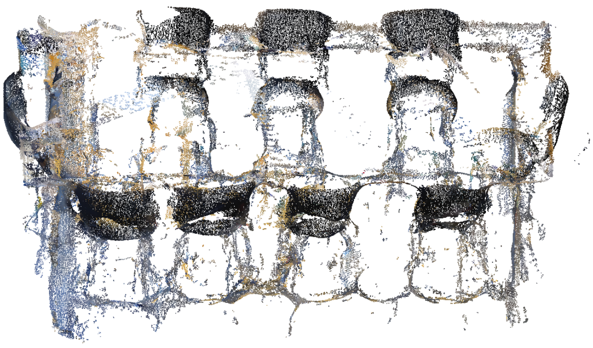
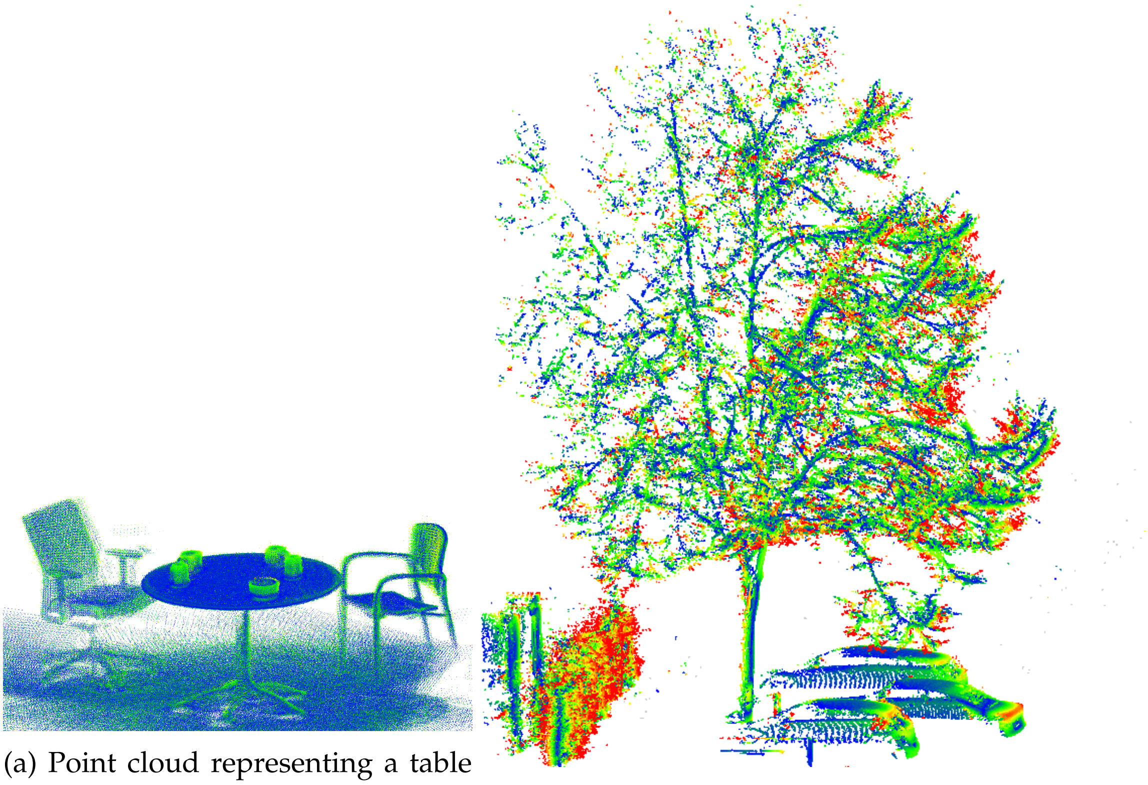


Figure 4.3: Point cloud of a glass table and chairs, acquired by a laser scanner.

The object type contained in the scene also influences the acquisition process. Indeed, digitized objects can have smooth or rough surfaces, which will influence the acquisition process and change the roughness characteristics of the generated data. For example, Figure 4.4 illustrates the difference in roughness between a table and a tree. In this example, a table has a smoother surface than a tree. Thus the table in Figure 4.4 (a) has a low roughness (represented in blue). On the contrary, in Figure 4.4) (b), the tree and the bushes are a high roughness (represented in green and red).



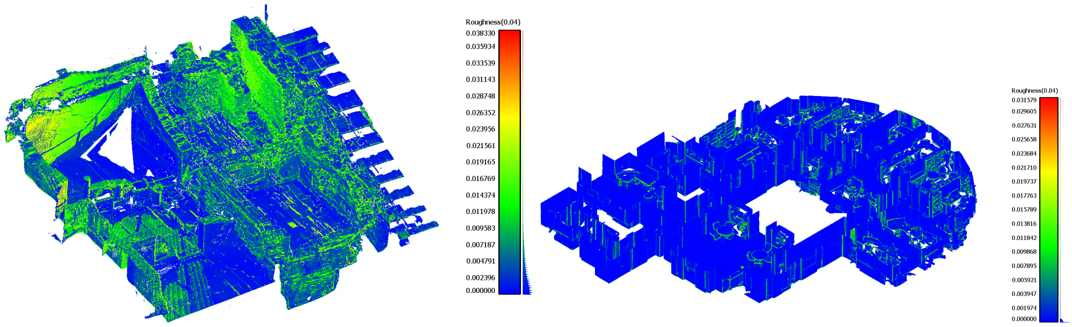
(a) Point cloud representing a table and chairs.

(b) Point cloud reprenting a tree and brushes.

Figure 4.4: Comparison of the roughness of two point clouds. Point clouds are colored according to the roughness computed (from blue to low roughness to red for hight roughness).

Finally, the context of the scene acquisition has a significant influence on the roughness characteristic of the data. For example, a wall digitized by a 3D laser scanner, in the context of the 3D reconstruction of a modern building, does not have the same roughness as a digitized wall in the context of an archaeological ruin excavation.

Figure 4.5 illustrates the difference in roughness between digitized point clouds in the context of archaeological ruin excavation and the context of the 3D reconstruction of a modern building.



(a) Roughness of the point cloud in the context of archaeological ruin excavation.

(b) Roughness of the point cloud in the context of the 3D reconstruction of a modern building.

Figure 4.5: Comparison of the roughness of two point clouds acquiring by a laser scanner.

External factors

External factors influencing the acquisition process can be related to diverse elements such as the weather condition, the vibrations of the measuring instrument, or the movement of the elements being digitized, the light intensity, or the color of the light.

These factors can sufficiently influence the acquisition process to transform the characteristics of the data completely.

Figure 4.6 shows the digitization by laser scanner techniques of an interior part. In this case, a large part of the data (surrounded in red in Figure 4.6) is missing. This problem is probably due to the influence of light, which must have been too strong and which no longer allowed the digitization of this area by laser scanner technologies.

Figure 4.6: Point cloud of a room with a missing part due to the influences of external factors.

Similarly, weather conditions can influence the acquisition process when it is carried out in an external context. These influences result in more noise and lower density in the acquired data. Figure 4.7 shows the noise produced by 3D laser scanner techniques when digitizing an urban scene outdoors.

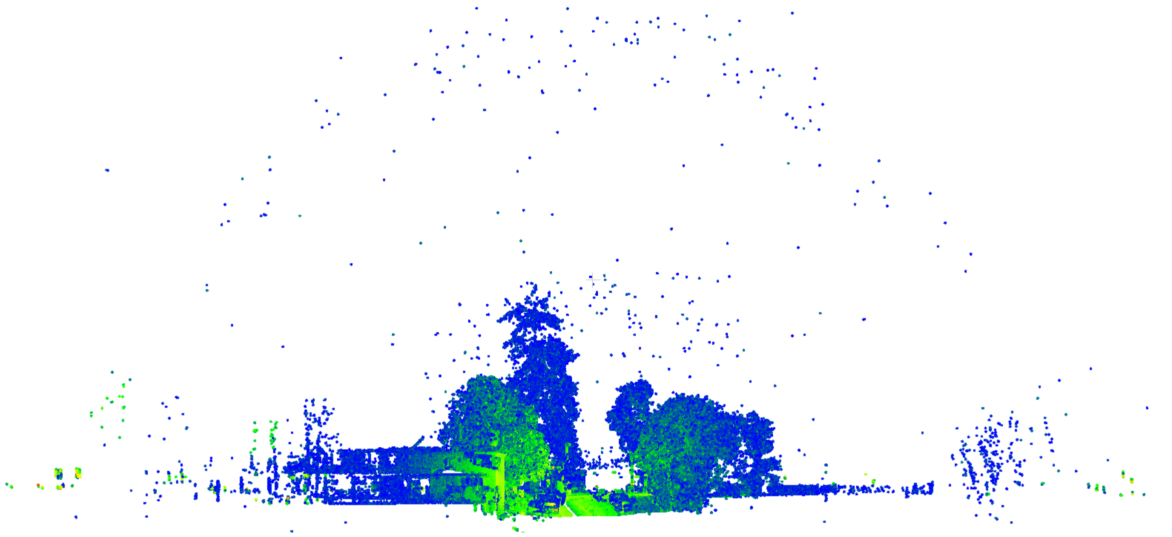


Figure 4.7: Point cloud of an urban scene acquired by a laser scanner technique.

The acquisition of 3D data can produce unstructured data with a wide variety of characteristics. The data acquisition process is responsible for data characteristics. However, the acquisition process is influenced by technique, methodology, and various internal (such as the scene and the objects that compose it) and external (such as light, weather conditions) factors. Thus these factors influence the characteristics of the data indirectly. These influences can lead to very different scene representations that can be not expected.

Table 4.1 summarizes the influence of acquisition techniques, acquisition methodology, internal and external factors in acquisition processes according to density, noise, occlusion, roughness, and color characteristics.

Characteristiques	Techniques	Methodology	Internal factor	External factor
Noise	high	low	medium	high
Occlusion	low	high	medium	low
Density	high	high	high	medium
Roughness	medium	low	high	low
Color	high	low	medium	medium

Table 4.1: Table of characteristics influences.

Many applications need structured data. Therefore, the data must be structured. It is necessary to understand the data in order to structure 3D data.

4.2 Data processing

Understanding the content of the data requires identifying the objects that constitute the data. The identification of objects requires processing data through algorithms, as explained in the work [*Walsh et al.*, 2013].

The scene characteristics, the objects it contains, and the geometry of these objects, guide the choice of algorithms. For example, data representing a scene composed of planar objects, such as tables, guides the choice of algorithms towards plane detection algorithms. These algorithms allow detecting the planar geometry of each table. Thus, algorithms allow the detection of each table among the planar geometries having the size and other characteristics of a table.

4.2.1 Algorithms

Algorithms are used to perform various tasks such as noise cancelling ([*Zeng et al.*, 2018]), filtering the data ([*Han et al.*, 2017]), segmenting the data ([*Vosselman*, 2013]), describing the data ([*Hana et al.*, 2018]), and classifying the data ([*Griffiths and Boehm*, 2019]) .

These tasks are determined according to the needs of the application (quality of data structuring), the data and objects characteristics, acquisition process, and data processing used.

Several libraries dedicated to data processing in Computer Vision (such as *PCL* [*Rusu and Cousins*, 2011], *OpenCV*[1]) provide algorithms to perform multiple tasks essential to data processing.

Generally, each algorithm is defined by these inputs, outputs, parameters, and prerequisites.

The inputs of the algorithms correspond to the data that the algorithms have to process. The outputs of the algorithms correspond to the types of results they produce. For example, a plane detection algorithm (such as [*Deschaud and Goulette*, 2010], [*Hulik et al.*, 2014], [*Oehler et al.*, 2011], and [*Limberger and Oliveira*, 2015]) can detect planes in a 3D point cloud. It, therefore, has a point cloud as its input, and planes (which can be represented by an equation or a set of points) as its output. Similarly, sphere detection algorithms (such as [*Abuzaina et al.*, 2013], and [*Camurri et al.*, 2014]) can detect spheres in a 3D point cloud. It has a point cloud as its input, and spheres as its output (which can be represented by an equation or

[1] Laganière, R. (2014). OpenCV Computer Vision Application Programming Programming Cookbook Second Edition. Packt Publishing Ltd.

a set of points).

Algorithm parameters are mostly values that influence the behavior of the algorithm. Let us take, for example, the region growing algorithms (as used in the approaches [*Vo et al.*, 2015], [*Ackermann and Troisi*, 2010], and [*Klasing et al.*, 2009]). These algorithms segment the data and produce homogeneous regions according to a criterion (such as color, roughness, or orientation, and proximity). These algorithms require two main parameters: a "tolerance threshold" and a "distance threshold". The "tolerance threshold" is a value used to determine whether two points belong to the same region. In other words, if points are a similarity value under this threshold value, then they can belong to the same region. This similarity value is evaluated according to criteria (such as color, roughness, orientation). The "distance threshold" is a value used to determine whether two points are adjacent. In other words, if the distance (often Euclidean) between two points is less than this threshold, then the points are adjacent. Parameter values of algorithms are often defined according to the characteristics of the data. For example, the value of the "distance threshold" can be chosen according to the point cloud density on which the algorithm is executed. Furthermore, the parameters of the algorithms can have interdependent interactions. In this example, the two threshold values are used to determine which points belong to the same regions. The interactions between the parameters of an algorithm can be complicated. It is necessary to understand their interactions in order to choose their values according to the application case, as explained in [*Deb and Agrawal*, 1998].

The prerequisites of algorithms are the "sine qua non" conditions for the execution of algorithms. These conditions mainly concern data characteristics of the algorithms input data, must have. For example, color segmentation algorithms (as used in the [*Sareen et al.*, 2010] approach) require the data to be colored. Similarly, normal segmentation algorithms (as used in the approaches [*Woo et al.*, 2002]) require the estimation the normal of each point constituting the point cloud. These normals can be estimated before by algorithms as proposed by the approaches [*Zhang et al.*, 2013] and [*OuYang and Feng*, 2005].

4.2.2 Processing

It is necessary to choose the best algorithms to process data efficiently. To this end, it is necessary to understand in which situations (data characteristics, task performed) each algorithm is the most effective. Similarly, it is necessary to understand in which situations the algorithms are not usable.

For example, segmentation algorithms based on normals estimation, are useful for segmenting point clouds, when they are composed of regular shapes objects (such as tables, walls, floor, ceilings, boxes). On the contrary, such algorithms are not very useful in segmenting data to detect elements (such as vegetation) with non-regularly shape.

Such an understanding of algorithms requires understanding the data characteristics and therefore understanding the different factors influencing these characteristics (see Section 4.1).

Algorithms sequence

Depending on the tasks to perform and the elements to detect, it may be necessary to combine several algorithms. For example, to detect cars in data representing an urban scene, it is interesting first to use algorithms to detect the floor. Then, to use other algorithms to detect cars (as proposed in the approach [*Hernandez and Marcotegui*, 2009]). These algorithms constitute a sequence to detect the elements.

The combination of algorithms is also used to change (transform or add) data characteristics when they do not satisfy the requirements of the algorithms. For example, it may be necessary to use a sampling algorithm (as proposed in the [*Puttonen et al.*, 2013] approach), or a simplifying algorithm (as proposed in the [*Pauly et al.*, 2002] approach) to reduce the size of a point cloud. Such needs appear if an algorithm requires small data sizes to work on it (as in the case of the [*Cao et al.*, 2010] approach).

The algorithms may be combined according to their inputs and outputs. In other words, the results produced at the output of one algorithm can be used at the input of another algorithm (if the prerequisites of the latter are satisfied).

For example, a "normal estimation" algorithm takes a point cloud as input and provides as output, a point cloud for which, the "normal" of each point is estimated. This last point cloud can be taken as the input of a "normal segmentation" algorithm that provides output segments (data portion). Then, such segments can be analyzed to understand the data content.

The combination of algorithms should be guided by the objects and geometry contained in the digitized scene, and which must be detected in the data.

The configuration of each algorithm should depend on the characteristics of the data that are impacted by multiple factors (as explained in Section 4.1). Therefore it is necessary to link the knowledge from the digitized scene, the data domain whose acquisition process, with the knowledge from the algorithm domain to well process the data. The following section 4.3 presents the technologies for managing such knowledge.

4.3 Knowledge engineering

Knowledge enables humans to understand the world around them as a coherent and structured whole. Therefore, it is interesting to look for ways to integrate and use knowledge through computer processes to understand unstructured data.

This section first shows how knowledge can be expressed. Then it shows the technologies for modeling and using knowledge.

4.3.1 Knowledge design

The works of [*Davenport et al.*, 1998] and [*Ahsan and Shah*, 2006] define data, information, knowledge, and their interconnection.

Data are raw values that can be of various types (such as numeric, textual, boolean, symbols). The data has no meaning in itself and is not organized. For example, the values "128", "54.4", "14.4", "0", "-89.4", "0" are data. On their own, it only represents values.

However, when organized according to an objective, the data can be interpreted as information.

According to the authors of [*Davenport et al.*, 1998], converting data into information requires several steps. First, it is necessary to collect the data for a known purpose. Second, the data should be organized into categories. Third, errors in the data should be deleted. For example, noise in point clouds is an error to delete. Fourth, the data should be summarized and should not contain any duplication. Finally, data must be aggregated and unified to form information.

For example, if the values in the previous example are aggregated into two ordered groups such as the group "128,0,0" and the group "54.4,14.4,-89.4" then they can be interpreted as information. In this example, the first group may be interpreted as the brown color in RGB format and the second group may be interpreted as a position in a 3D Cartesian coordinate system (X, Y, Z). Thus, such data may be converted as the information of a localized brown point ("54.4,14.4,-89.4" in the Cartesian coordinate).

When information is grouped and described as a logical and coherent whole related to a domain, then it becomes knowledge. For example, the information that forms the knowledge of a 3D digitized table tray is as follows: a set of brown points aligned in the same horizontal plane. They are located at the height of *70 cm*, and their total area is at least 0.3 m^2. The grouping of this information is logical, coherent, and related to the Object domain.

Similarly, information on the inputs, outputs, prerequisites, and parameters of an algorithm (as explained in Section 4.2) forms knowledge of this algorithm, in the Data processing domain.

4.3.2 Ontology

The domain needs to be represented through concepts understandable both by human and machine. An ontology is "a formal, explicit specification of a shared conceptualization " [*Studer et al.*, 1998]. That is why this thesis uses an ontology to represent the knowledge used to guide the process of data understanding.

The ontology provides several advantages. Ontologies allow defining a precise vocabulary for the domain they describe through a set of concepts and properties. These properties describe relationships between concepts by an interlinking. This interlinking allows a vast and complete description of a domain.

Different applications can share this vocabulary. This sharing allows a better extension of knowledge. Indeed, the extension of ontologies is possible through the linking of different ontologies.

The emergence of the Semantic Web [*Berners-Lee et al.*, 2001] has led to a significant evolution of semantic technologies. Among these developments, there are two standards to model an ontology, which are the Resource Description Framework (*RDF*)[*Miller*, 1998] and the Web ontology language (*OWL*) [*McGuinness et al.*, 2004]. *RDF* allows the creation of knowledge graph based on concepts and properties, whereas *OWL* allows the logic description of the knowledge, thanks to restrictions on concepts and properties.

This language has then been improved to form an *OWL2* [*Hitzler et al.*, 2009] increasing the variety of description to describe concepts more precisely.

The knowledge that guides the process presented in this thesis is represented in an ontology through the *OWL2* language. This knowledge can be described in *OWL2* by tools such as "*PROTEGE*" [*Musen et al.*, 2015] through the Manchester syntax [*Horridge and Patel-Schneider*, 2009].

Table 4.2 highlights the conversion of the main owl-restriction of property to Manchester syntax.

OWL-restriction	Manchester syntax
owl:allValuesFrom	only
owl:someValuesFrom	some
owl:hasValue	value
owl:maxCardinality	max
owl:minCardinality	min
owl:Q cardinality	exactly
owl:complementOf	not
owl:intersectionOf	and
owl:unionOf	or

Table 4.2: Conversion table of OWL-restriction to Manchester syntax.

In order to illustrate the description logic through the Manchester syntax, let us consider a description of a floor which is a simple and common element to textit-many application cases. Its description is as follows: *The floor is a "Segment" that has a planar geometry, which is horizontal. It can be composed of various colors, can be rough or smooth, and is not above any other object (e.g. wall or roof).*

Code 4.1 shows the description of the floor in *OWL2*, according to this example and through the Manchester syntax.

```
Floor:
Segment
and (hasGeometry some (PlanarSurface and hasOrientation only Horizontal))
and hasColor some Color
and hasShape some (Rough or Smooth)
and not (isAbove some Object)
```

Code 4.1: Description logic of a Floor through Manchester syntax

The ontology allows for designing a knowledge base. A knowledge-base is composed of a set of terminologies (representing the concepts and properties) and a set of assertions (representing the aggregated information according to the terminology). Information is called represented by individuals. This information becomes knowledge when it is linked to other information and concepts. For example, *C* (an individual) is a *Coordinate* (a concept) and *has value* (a property) "54.4, 14.4, -89.4" (data).

4.3. KNOWLEDGE ENGINEERING

4.3.3 Knowledge management through SPARQL

The *SPARQL* Protocol And RDF Query Language *(SPARQL)* [*Prud'hommeaux and Seaborne*, 2008] has become a standard for accessing and modifying knowledge. Various approaches of knowledge-based segmentation such as [*Meditskos et al.*, 2014] and [*Triboan et al.*, 2017], or semantic annotation such as [*Kanimozhi and Christy*, 2013] uses *SPARQL*.

Three main types of queries compose the language *SPARQL* : "Select", "Update", and "Construct". The "select" query allows for retrieving information. The "update" query allows for adding, deleting, and modifying information. Finally, the "construct" query allows for adding new relationships between the information contained in the triplestore. These queries can be used to retrieve, update, add, or delete knowledge. For example, the [*Kumar et al.*, 2008] approach uses *SPARQL* queries to analyze knowledge on large biomedical image datasets.

Select query

"Select" queries are used to select individuals from an ontology. These individuals are selected based on the properties specifying the relationships they have or not (e.g. the property "is not connected to"). These queries allow for the selection of individuals with great flexibility and efficiency.

Code 4.2 presents the *SPARQL* query to select all segments (elements resulting from a segmentation process) that have planar and horizontal geometry, and are below all other individuals (i.e. the floor).

```

SELECT ?seg WHERE{
 ?seg rdf:type ComputerVision:Segment.
 ?seg ComputerVision:hasGeometry ?plane.
 ?plane rdf:type ComputerVision:Planar.
 ?plane ComputerVision:hasOrientation ?hori.
 ?hori rdf:type ComputerVision:Horizontal.
 FILTER NOT EXIST(
                  ?object rdf:type ComputerVision:Object.
                  ?seg ComputerVision:isAbove ?object.
              ).
}
```

Code 4.2: Example of *SPARQL* select query

Update query

"Update" queries allow to add, delete, or modify knowledge about individuals in ontologies.

Code 4.3 presents the *SPARQL* update query that allows adding an individual called "pointcloud-c7" representing an RGB colored point cloud and loaded from a file called "cloudFile".

```
INSERT DATA { ComputerVision:pointCloud-c7 rdf:type PointCloud.
ComputerVision:pointCloud-c7 ComputerVision:readFrom ComputerVision:File.
ComputerVision:pointCloud-c7 ComputerVision:hasColor ComputerVision:RGB.
  }
```

Code 4.3: Example of *SPARQL* update query

The adding of knowledge is done through the keyword *INSERT* following by the assertion to add, whereas, deleting knowledge used the keyword *DELETE* followed by assertion to delete. The update of knowledge consists in deleting the knowledge to update and add the new knowledge corresponding to the updated knowledge. It combines thus, the two keywords *INSERT* and *DELETE*.

Construct query

"Construct" queries are a kind of combination between the "insert" and "select" queries. They thus allow for modifying the properties of selected individuals. Code 4.4 presents a *SPARQL* construct query that classifies the segments that have planar and horizontal geometry, and are below all other individuals, as being a floor.

```

CONSTRUCT ?seg rdf:type ComputerVision:Floor WHERE{
 ?seg rdf:type ComputerVision:Segment.
 ?seg ComputerVision:hasGeometry ?plane.
 ?plane rdf:type ComputerVision:Planar.
 ?plane ComputerVision:hasOrientation ?hori.
 ?hori rdf:type ComputerVision:Horizontal.
 FILTER NOT EXIST(
                 ?object rdf:type ComputerVision:Object.
                 ?seg ComputerVision:isAbove ?object.
              ).
}
```

Code 4.4: Example of *SPARQL* construct query

4.3.4 Inference

The expression of knowledge in the form of a description logic in an ontology allows reasoning mechanisms for inferring new knowledge on the one hand, and for verifying the coherence of the set of knowledge on the other hand. These inference processes are performed by reasoners such as "Pellet" [*Sirin et al.*, 2007], "HermiT" [*Shearer et al.*, 2008], "FaCt++" [*Tsarkov and Horrocks*, 2006], *, Karlsruhe ontology 2* (*KAON2*) [*Motik and Studer*, 2005], or "Jena reasoner" [*Carroll et al.*, 2004]. The authors of [*Bock et al.*, 2008] presents a benchmark the main reasoners used. They conclude that the reasoner should be chosen according to the type of task they have to perform. In the context of this thesis, the ontology involves a high level of language complexity and a large number of assertions. According to[*Bock et al.*, 2008], *KAON2* is the most adapted reasoner for such ontology.

Two main methods are used to infer knowledge: inference through reasoning on description logics of concepts and inference through the application of logic rules.

Inference through reasoning on description logics

The reasoning on description logics ensures the consistency of the ontology by checking it on the definition of the concepts (as explained in the approach [*Wang et al.*, 2007]). Let us take the example of the conceptual description of the floor; this one is described as being connected to walls.

Besides, the reasoner allows inferring new knowledge from reasoning on the description logic of each concept. Let us take the example of an element belonging to the floor concept as described in the previous example. The description logic of the floor concept allows inferring that it is below all other elements described in the ontology. Thus a property "isBelow" is added to link the element belonging to the floor concept to all other elements. Such reasoning is use in various approaches of computer vision such as [*Hwang et al.*, 2006], [*Tongphu et al.*, 2012], [*Choi et al.*, 2008], [*Günther et al.*, 2011], and [*Johnston et al.*, 2008] to recognized objects.

Rule of inference

The rule of inference allows for describing the influences that concepts can have on each other on the one hand, and for gathering information and knowledge to form new knowledge on the other hand. Rules of inference are used in various approaches of Computer Vision, mainly to classify elements and identify data content. For this purpose, approaches such as [*Marroquin et al.*, 2018], [*Gómez-Romero et al.*, 2016] and [*Othmani et al.*, 2010] collect and organize information on data segments and then use rules of inference that describe concepts to classify these elements and understand the content of the data.

The primary language used to formalize rules of inference is "Semantic Web Rule Language (SWRL)" presented in the work [*Horrocks et al.*, 2004]. *SWRL* is a language that combines *OWL* and *RuleML* [*Boley et al.*, 2010]. Let us take a simple and famous example of rule illustrated by Equation 4.1 to assert that the combination of the hasParent and hasBrother properties implies the hasUncle property.

$$hasParent(?x1, ?x2) \wedge hasBrother(?x2, ?x3) \Rightarrow hasUncle(?x1, ?x3) \tag{4.1}$$

Code 4.5 shows the abstract syntax corresponding to the *SWRL* rule.

```
Implies(Antecedent(hasParent(I-variable(x1) I-variable(x2))
           hasBrother(I-variable(x2) I-variable(x3)))
    Consequent(hasUncle(I-variable(x1) I-variable(x3))))
}
```

Code 4.5: Example of the abstract syntax of a *SWRL* rule

From this rule, if Alan (x1) has Julius (x2) as a parent and Julius is the brother of Harvey(x3) then Alan has Harvey as an uncle.

The work [*Cregan et al.*, 2005] studies the limits of the rules *SWRL*. They highlight the difficulty related to the expressiveness of the rules *SWRL*, which can become very complicated and lead to inappropriate reasoning. Furthermore, the rules *SWRL* require a complete ontology analysis for each rule execution, which can become a problem if the ontologies have many concepts.

These limitations have led to the development of alternative technologies to use rules of inference across *SPARQL*, which provides greater efficiency. Among these technologies, *SPARQL* Inferencing Notation (*SPIN*) [*Knublauch et al.*, 2011] is increasingly used thanks to its effectiveness (such as in the works [*Järvenpää et al.*, 2018], [*Fürber and Hepp*, 2010], [*Tomaszuk*, 2016], and [*O'Riain et al.*, 2015]). The works [*Bassiliades*, 2018] shows how to convert logic rules *SWRL* to *SPIN*.

An alternative to the use of *SPIN* is the use of "Shapes Constraint Language" (SHACL) [*Knublauch and Kontokostas*, 2017] which is a World Wide Web Consortium (W3C) recommendation. The authors of [*Corman et al.*, 2018] describe SHACL as "One of the most promising schema languages". It also exists other promising approaches such as "Alternative Shapes Constraint Language" (ASHACL)

[*Patel-Schneider*, 2017]. All of these approaches are too recent, and it is not yet possible to estimate which ones are most relevant or efficient.

4.4 Discussion

The characteristics of 3D data depend on their acquisition process. The acquisition process is a combination of techniques, methodology, and is influenced by many internal and external factors. More precisely, the acquisition technique has a high influence on density, noise, and color, but also a medium influence on roughness. The methodology highly influences the density and occlusions in the data. The internal factors impact the data characteristics intensively since they have a high impact on density and roughness, as well as a medium impact on noise, color, and occlusion. Finally, external factors have a high impact on noise and medium impact on color and density. Besides, these individual impacts are combined through the acquisition process. Thus, the unstructured data resulting from the acquisition process can have a large variety of characteristics. Some applications (such as robotics, navigation, monument documentation) require this data to be structured. Structuring data requires processing and understanding the data through the use of algorithms. Algorithms are mainly chosen according to the characteristics of the digitized scene (such as object distribution, context), as well as the characteristics of the objects (material, texture, roughness, size). However, these algorithms are combined and configured according to the data characteristics. Thus, it is necessary to know and understand the data characteristics in order to configure and combine algorithms properly.

The understanding of data characteristics can be directly provided by the knowledge about data or indirectly from the knowledge of the acquisition process by deducing data characteristics from this knowledge. It depends on the knowledge provided by the user.

Such an understanding requires to collect and organize information on different domains in order to build knowledge. Then this knowledge should be used to drive the data processing. This knowledge constitutes concepts within three domains: Data processing domain, Data domain (including acquisition process), and Scene domain. The exploitation of this knowledge should drive the data processing in order to obtain better data understanding, and thus, obtain better data structuring.

Technologies from the Semantic Web allow for explaining, modeling, and using such knowledge. The language *OWL2* allows specifying the knowledge in the form of a description logic whose consistency can be checked automatically. Knowledge can be managed and used through the query language *SPARQL*. Finally, the use of reasoning and rules of inference enriches explicit knowledge by inferring new knowledge.

Therefore, the approach proposed in this thesis is based on technologies from the Semantic Web. These technologies are first, used to model the knowledge on data, scene, and data processing domain. They are then used to guide the data processing and enrich the knowledge, according to the specificities of the application case. This specific enrichment of knowledge allows for adapting the data processing specifically to the application case. This adaptation leads to a better understanding of the data.

Part II

Literature review

This part presents a review of the different 3D data processing techniques classified into three main categories according to their paradigms. The review [*Grilli et al.*, 2017] presents the most popular methodologies for object detection in 3D point clouds.

Chapter 5 sets out the first category corresponding to model-driven approaches. The approaches strategy in this category is to create models for each object sought and compare them to each data region.
Chapter 6 presents the second category corresponding to data-driven approaches. Unlike model-driven approaches, approaches in this category aim at characterizing data to isolate portions of it and classify these portions according to the objects sought.
Chapter 7 introduces the last category of approaches, which are Knowledge-driven approaches. Approaches in this category use knowledge about objects and data to adapt to the detection process.
Finally, Chapter 8 discusses the evaluation of these different approach categories according to the following criteria:

Quality: The quality of object detection depends on correctly detected objects and forgotten objects, as well as on the accuracy of their detection.

Ambiguity: The ambiguity depends on correctly detected objects and incorrectly classified (classify into separate categories).

Robustness: The robustness refers to the effectiveness of approaches when the context of object detection is challenging. Context is considered challenging when the data have a wide variation in density, are noisy and composed of occlusions. This type of context requires the detection of objects when their representation in the data differs from their expected form, which means solving unpredictable problems.

Flexibility: The flexibility refers to the ability to adapt to the detection of different objects in different data with different characteristics.

Generalizability: Generalizability refers to the ability to improve the results obtained.

5 Model-driven approaches

Model-driven approaches are based entirely on the characteristics of the objects. Indeed, model-driven approaches consist in creating a model for each object based on their geometric characteristics such as their shape. Then, these approaches scan the data sequentially and compare each scanned portion to the models created for each object to detect them. Figure 5.1 presents the workflow of these approaches that are mainly used to solve specific object detection problems.

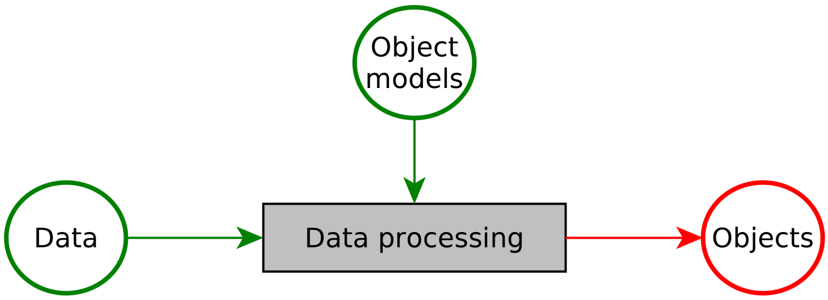


Figure 5.1: Common workflow of model-driven approaches.

This chapter details the most used model-driven approaches divided into three sub-categories according to their general methodology. The first sub-category is projection detection approaches that reduce the complexity of 3D data by projecting points onto a defined plane. The second sub-category is the approximation of the 3D mathematical model that allows effective detection of primary shapes. Finally, the last sub-category is a sophisticated and free 3D shape detection approaches, which allows detecting objects of complex and free shapes. The two previous sub-categories can not detect these objects.

5.1 Projection detection approaches

Widespread model-driven approaches, in the field of detection of large 3D objects (such as building elements), consist in making a two-dimensional projection of the points of the cloud. Such projection is on a plane (mainly the XY plane) or in voxels as performed by [*Qian et al.*, 1992]. Such projection aims to simplify object detection. The [*Diaz-Vilarino et al.*, 2015] approach proposes the development of two simple two-dimensional mathematical models (a circular model, and a rectangular model) to detect two different types of columns. He then proposes to project the points of the cloud on the XY plane and to perform shape recognition based on the Hough transform [*Ballard*, 1981]. Similarly, the approach [*Adan and Huber*, 2011] proposes a detection of walls by projecting the point cloud on the XY plane and then uses the Hough transform to detect the walls. Figure 5.2 illustrates the common workflow of projection detection approach. This type of model-driven approach is suitable for detecting objects that are regular and large enough to cause negligible variation in noise or data quality.

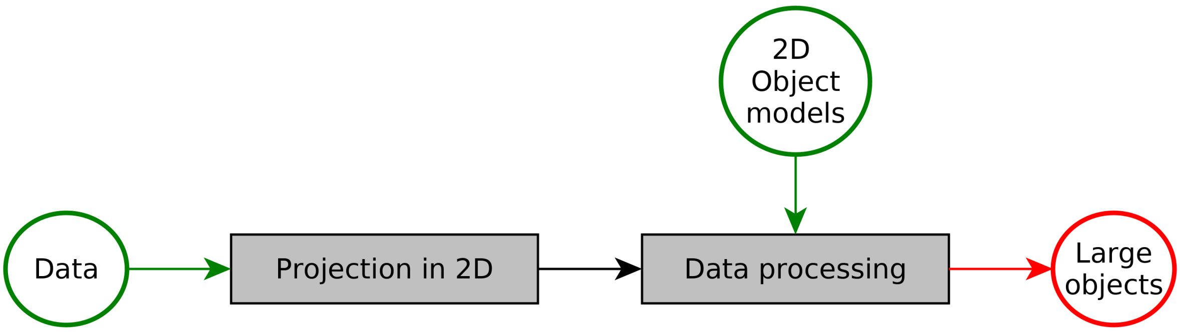


Figure 5.2: Common workflow of projection detection approaches.

However, these approaches do not exploit the three-dimensional spatial characteristics of the data. This lack does not allow the detection of complex objects (such as a car) nor the dissociation of objects with a similar surface, nor the detection of small objects that are not composed of a significant number of points (such as a traffic sign). Moreover, these approaches do not allow for dissociating overlapping objects (such as an object on a table) which decreases the detection quality of objects.

5.2 3D mathematical model approximation

In order to exploit three-dimensional spatial information and improve detection quality (separation of superposed elements), approaches such as [*Vosselman et al.*, 2001] and [*Overby et al.*, 2004] use a Hough transform adapted to 3D [*Borrmann et al.*, 2011] to detect planes. Such detection allows for identifying the ground and buildings through their different facades and then, extracting the 3D model of the identified object. The disadvantage of these approaches is the sensitivity to noise, outliers, and variations in data density.

In order to obtain a more noise-resistant and outlier-resistant detection of objects, approaches such as [*Anagnostopoulos et al.*, 2016] use RANdom SAmple Consensus (*RANSAC*) [*Schnabel et al.*, 2007] to detect shapes (such as planes, spheres, and cylinders) that are more resistant to noise and outliers than methods based on Hough's transforms.

However, these methods present much erroneous detection due to the wrong separation of the shapes detected by *RANSAC*. Indeed, if two objects are close and coplanar, *RANSAC* detects only one shape encompassing the two objects. The approaches [*Tarsha-Kurdi et al.*, 2008] and [*Xu et al.*, 2016] overcome this issue by improving *RANSAC* to increase the accuracy of detection. However, despite these improvements, data pre-processing steps must be performed and supervised by the users. Although the approaches of this sub-category are efficient for detecting large plans, they do not detect small elements or buildings. Besides, their efficiency decreases significantly as a function of the ratio between data density and object size. Indeed, lower the data density is, and more significant the object is; worse is the detection. These approaches have thus also difficulties in detecting large objects in data with a low density.

These different approaches are based in general on the thresholding technique to determine whether or not the studied portion corresponds to the geometric model of the object. The user chooses the threshold value for the overall process that makes the threshold setting static and does not adapt to variations in data characteristics. Figure 5.3 shows the common workflow of these approaches.

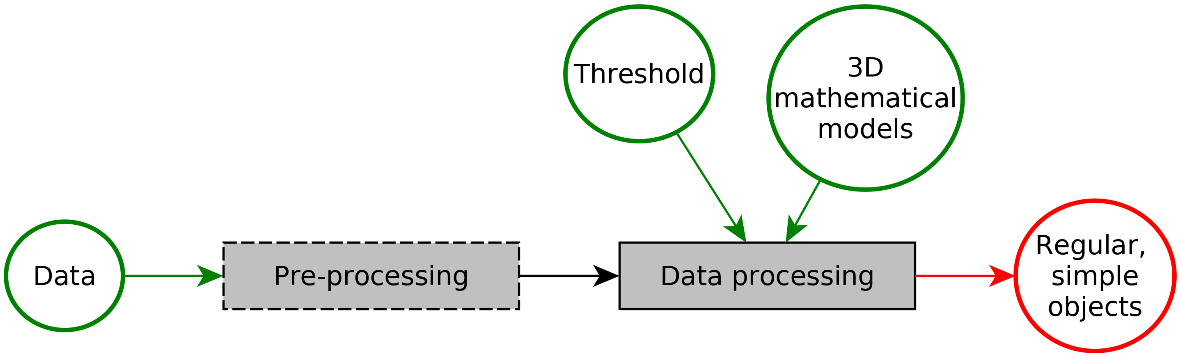


Figure 5.3: Common workflow of 3D mathematical model approximation approaches.

This system is also sensitive, and source of many classification errors. In order to solve these classification errors, model-driven approaches such as [*Henn et al.*, 2013] combine *RANSAC* shape detection with supervised machine learning methods, thus giving more classification flexibility.

Although the addition of machine learning solves classification problems, detectable objects are limited by the detectable geometric shapes (e.g. plane, cylinder, sphere, line) by algorithms like *RANSAC* or transformed Hough. However, some objects are composed of free or very complex shapes. When this type of object is isolated and surrounded only by objects in regular and straightforward shape, approaches such as [*Hu et al.*, 2018], combine simple representations and complex meshes to isolate and describe this kind of objects. However, when different complex objects are part of the same scene, their dissociation and detection are more complicated.

5.3 3D complex and free shape detection approaches

In order to be able to detect complex or free shapes of objects, some ad hoc describe the shape of the objects to be detected from arbitrarily chosen key points and seek to match these key points with the data. Figure 5.4 presents the common workflow of 3D complex and free shape detection approaches.

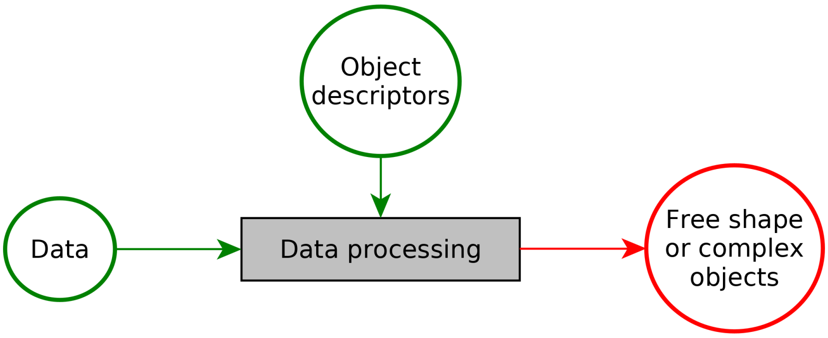


Figure 5.4: Common workflow of 3D complex and free shape detection approaches.

In the field of railway track detection in point clouds acquired by LiDar scanner, the [*Ponciano et al.*, 2015] approach uses a model of railway track representation by key points to automatically detect them. This approach linearly scans the data to identify railway track, portion by portion. Each portion is then compared to the railway track model, and the detection process computes a correspondence between the points in the model and the point cloud to detect the tracks by thresholding.

Other, more sophisticated approaches use 3D object descriptors to describe complex or freely shaped objects automatically. The comparative study [*Alexandre*, 2012] compares the most famous 3D object descriptors. Among the different approaches based on 3D object descriptors, the approach [*Tombari et al.*, 2011] proposes to describe an object both by its shape and by its texture, which allows for better detection in the context of colored objects. However, this approach is very sensitive to data variations and noise that compromises the matching of key points chosen with the data and therefore, compromises object detection. Also, these approaches have the disadvantage of not being able to detect partially occluded objects and require a local model of each portion of the objects.

The [*Drost et al.*, 2010] approach proposes to overcome this issue by detecting occluded objects with a single description of the objects. Its quality of detection depends on the complexity and uniqueness of each object, according to others. This quality also depends on the density of the data. However, the description of these objects does not take into account variations in data quality. Thus, these approaches remain very sensitive to density variation in the data and noise. They are suitable for object recognition in clouds composed of few objects but are ineffective in detecting objects in large point clouds composed of various objects and scenes.

5.4 Discussion

Model-driven approaches based on projection are adapted to detect large objects in dense data. However, they are limited by problems of objects overlap related to the projection. These overlaps imply firstly, a weak detection quality due to the impossibility to detect some overlapped objects. It implies then, an ambiguity due to the loss of spatial information (e.g. two superposed objects considered as one object). They are robust to noise in dense data context but are very sensitive to decreased data density. They do also not detect unexpected cases. Furthermore, these approaches do not detect objects with small or complex shapes and are not effective in clouds with low-density of points. They are, therefore, not flexible enough. Finally, the results obtained by these approaches can only be slightly improved by the parametric modification of the algorithms used, but cannot be significantly extended.

Mathematical model approximation approaches allow the detection of regular and straightforward objects that can be approximated by a mathematical model (mainly spheric, linear, and planar objects). The detection quality of these approaches is weak due to similar shape separation errors. Despite the use of 3D spatial data, two objects side by side composed of a similar planar surface are often detected as one planar surface. These approaches generally produce many ambiguities, mainly when objects are decomposed as a result of occlusion. They are sensitive to variations in data density and do not allow the detection of unexpected cases. These approaches do not detect objects with complex or free shapes and do not automatically adapt to variations in data quality. They are, therefore, not very flexible. Finally, the improvement of the results obtained is only possible by the adaptation of the parametric values of the algorithms and the addition of pre-processing algorithms. The extension of these approaches is thus, limited. These approaches are insufficient in most application cases using large datasets.

Finally, the last sub-category of Model-driven approaches developed to detect complex, or free 3D shapes of objects are efficient for detecting objects in scenes composed of few objects. The quality of detection of these approaches is very high when the execution context is adapted. However, they have detection ambiguities when objects do not have sufficiently complex and unique shapes. Some of these approaches (such as [*Drost et al.*, 2010]) can quickly and efficiently detect complex objects, even if these objects are partially occluded. However, they require dense data and the compound of only a few objects (of the order of ten). However, they remain sensitive to noise and density variations within the data. Besides, they cannot deal with unforeseen cases. They, therefore, have rather poor robustness. Moreover, these approaches do not adapt to the detection of objects with different possible representations and require a specific model for each of the possible representations. Furthermore, they cannot adapt to large data sets composed of various objects and scenes. They are, therefore, not flexible enough. Finally, the improvement in detection is limited to refining the description of the object model, which may not be sufficient to detect some object representations.

Table 5.1 summarizes the advantages and disadvantages of the different types of model-driven approach.

Approach	Quality	Ambiguity	Robustness	Flexibility	Generalizability
Projection	- -	- -	=	- -	- -
Mathematical model	-	- -	- -	- -	- -
Complexe/ free shape	+	-	-	- -	-

Table 5.1: Comparative table of the different types of model-driven approaches.

Model-driven approaches are not adapted to the use cases presented in chapter 2.1. Indeed, their lack of flexibility, robustness and low improvement capacity make them unsuitable for detecting different objects in different point clouds, especially those with a high variation in quality (such as use case 2.1.3 and 2.1.4). However, these approaches can be combined with other approaches for solving specific problems (such as detecting a specific traffic sign).

5.4. DISCUSSION

6 Data-driven approaches

Data-driven approaches have the particularity of detecting objects only by processing data without any prior knowledge. The approaches in this category extract characteristics from the data. The characteristics can be geometric (such as orientation, size, shape) or physical characteristics (such as color). These characteristics are then grouped to form different non-predefined models of elements in the data. These approaches then classify these models according to their characteristics, in different classes representing the objects to detect. Figure 6.1 shows the workflow of these approaches.

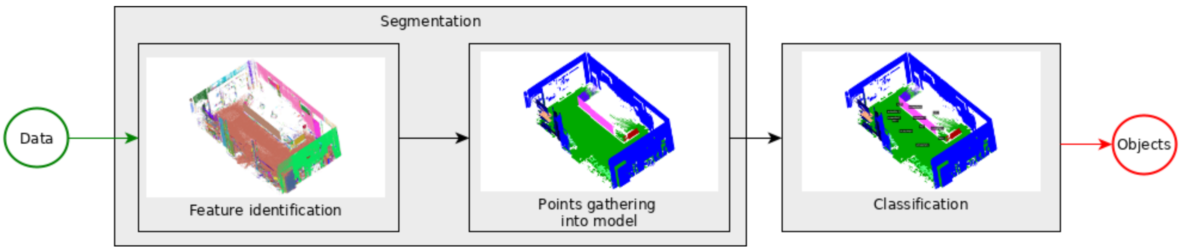


Figure 6.1: Common workflow of data-driven approaches.

These approaches are conceptually more flexible than model-based approaches.

This section presents the three most used sub-categories of data-driven approaches gathered according to their principal methodology.

The first sub-category is feature-based object recognition approaches. These approaches extract characteristics from the data by a segmentation process. This process takes into account one or more characteristics of the objects to be detected. Then, they classify the segments into different elements and finally group them and identify the objects. The second sub-category is Machine Learning approaches. These approaches use large annotated datasets containing the objects to be detected, to automatically learn their characteristics and recognize them in the desired data. Finally, the last sub-category is data-driven approaches using semantic that improve the classification process by using knowledge representations.

6.1 Features-based object recognition approaches

The approaches based on feature-based object recognition often use model-driven approaches to extract geometric features such as plans or lines. Unlike model-driven approaches, this sub-category of data-driven does not use predefined knowledge about the objects to detect. They do not seek to detect objects directly. First, they apply a segmentation process to the data. Then, they characterize each segment to group them into entities according to criteria. Entities are finally classified to recognize objects. Figure 6.2 presents this workflow.

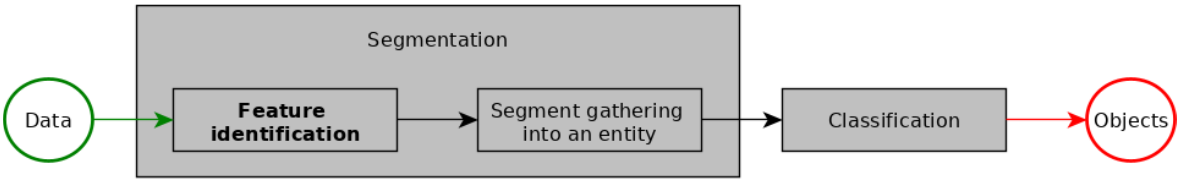


Figure 6.2: Common workflow of features-based object recognition approaches.

Step 1: Segmentation Segmentation aims to divide data into sub-element to facilitate the extraction of their features. This process is especially necessary for object detection in large data sets representing a scene composed of several elements, for example, a building composed of walls, ground, and a roof. In this case, one of the most common approaches of segmentation consists of segmenting the ground to extract and remove it from the remaining data. The ground extraction is intended to facilitate the segmentation of other elements.

The authors of [*Himmelsbach et al.*, 2010] present a segmentation approach in two steps based on model-driven approaches. First, they perform ground extraction to facilitate the segmentation of other elements. The seek of the most extensive horizontal plane allows the authors for ground segmentation. Then, they reduce the data complexity by rasterizing the point cloud with a 3D voxel grid. This rasterization corresponds to the projection of the data without the ground on a 2D plane inside voxels. The advantage of this approach is its speed of processing. Fast processing is essential in the context of extensive data. The use of model-driven approaches also has similar disadvantages. However, this approach is limited to objects with a simple shape. The shapes of objects also need to be radically different from one another to avoid the ambiguity of classification. Besides, this approach is very sensitive to noise and occlusion in 3D point clouds. This approach is designed for a specific application to detect ground and cars. It works only on flat grounds.

Unlike this first approach, the authors of [*Moosmann et al.*, 2009] focus their research on segmentation of no-flat ground. This approach studies the local convexity of elements to segment them. The authors propose to project and transform 3D data into a 2D graph to segment elements according to the computation of their local convexity.

However, these two approaches ([*Moosmann et al.*, 2009], [*Himmelsbach et al.*, 2010]) do not use a priori information about the shape and surface of objects to identify them. This lack produces a loss of accuracy. Besides, they are not usable to segment a large variety of different object types. They also depend on the detection context (here, object detection in an urban environment) and the data quality (e.g. noise density, occlusion).

In the domain of building information modeling (BIM), the approaches presented in [*Macher et al.*, 2015] and [*Macher et al.*, 2015] present segmentation approaches based on model-driven approaches. These approaches are designed only to detect walls, floors, ceilings, and rooms on 3D point clouds. They are mainly based on the definition of thresholds that determine the results obtained. Thus their adaptability is relatively limited.

The detection of objects not specific to an application context is a challenge. The authors of [*Khaloo and Lattanzi*, 2017] address this challenge. They propose a strategy based on a robust normal estimation next, segment the data through an algorithm of region growing [*Pauling et al.*, 2009]. The region growing process uses the distance of Mahalanobis [*De Maesschalck et al.*, 2000] for each point rather than the Euclidean distance. This adaptation of region growing is more accurate than region-growing based on "Minimum Volume Ellipsoid" [*Van Aelst and Rousseeuw*, 2009], "Minimum Covariance Determinant" [*Rousseeuw and Driessen*, 1999], or based on "Maximum Likelihood Sample Consensus" [*Torr and Zisserman*, 2000]. The normal estimation used in this strategy is based on the estimation of a plane tangent to the point. Although the computation of this distance takes time due to the computation of the covariance matrix, the estimation allows for obtaining more accurate values. Thanks to this accuracy, this approach allows for detecting various curves of, both large and small amplitude. Objects composed of sharp features are thus easily detectable. This approach estimates the points outliers through an adaptation of MM-estimators of regression [*Yohai et al.*, 1987]. The authors call this adapted version "Deterministic MM-estimator". However, this approach has two main disadvantage: long processing time and accuracy of segmentation. The high degree of derivation used by this approach produces an over-segmentation. Due to this over-segmentation, objects are divided into several parts that complicate the use of a model to identify them.

The over-segmentation is the main problem in the segmentation process of 3D point clouds. The authors [*Rabbani et al.*, 2006] present a methodology to avoid over-segmentation. The methodology proposed is based on a segmentation related to smoothness constraints. They study the surfaces locally inside the data through normal estimation. Points are then, gathered into areas according to their normal and their connectivity. Finally, areas being smoothly connected between them, are gathered to constitute a model that can be classified to identify objects. Although this methodology solves the over-segmentation problem in the majority of cases, it is not adapted in the case of objects to detect are mainly planar. Besides, this methodology has long time processing, which is a problem for the processing of extensive data.

Step 2: Classification The most trivial methodology to classify elements extracted from data is the use of a decision tree containing all the predefined possible classes of objects. The set of objects features constitutes the branches of the decision tree. The authors of [*Aijazi et al.*, 2013] present an approach based on this methodology. This approach consists of segmenting data into super-voxels [*Lin et al.*, 2018] and then grouping super-voxels to create object models. Algorithms of feature extraction characterize then these models. This approach considers characteristics such as surface normals, barycentre, geometric center, color, intensity, and overall shape of the model. The classification by decision-tree uses the feature description of the sought objects to classify each model according to their features.

Another approach presented in [*Ochmann et al.*, 2016] segments the data by detecting planes through the RANSAC algorithm. Then it classifies the segments obtained according to the characteristics defined for walls and floors. For example, a segment that is a vertical plane and has an area greater than $1.5\,m^2$ is classified as a wall. This classification is based entirely on the characteristics of the data. However, many factors (e.g. sensing process, measuring instrument, acquisition context, external factors) influences the characteristics of the data. Small variations in one of these factors can lead to significant variations in the characteristics of the data. Thus the characteristics of the segments to be classified may differ from the characteristics expected to identify objects. This divergence (e.g. area less than $1.5\,m^2$, non-planar surface) leads to the failure of segment identification.

This classification strategy is very rigid and does not allow for adaptation to a wide variety of objects. It also requires that the objects have significantly different characteristics. This strategy uses implicit knowledge provided by experts that designs decisions. However, the lack of flexibility in the decision-making made by tree generates ambiguity of classification. Moreover, the decision-tree must cover all the possible models to obtain a proper classification. The description of all models in the tree is unadapted for the detection of various objects. It is also not adapted to process data with many variations. These variations impact object representations, creating thus, a lot of different models for the same object. Therefore, this strategy is not adapted to process large point clouds. That is why this strategy is more and more replaced by classification strategies based on machine learning or semantic approach.

6.2 Machine Learning approaches

Machine Learning is a field of research that in general, allow obtaining more effective results than feature-based approaches using artificial intelligence. That is why it is more and more used. Indeed, Machine Learning approaches improve the detection process through learning approaches specialized in the sought object. This learning system requires a so-called "annotated" data set where each object in this set is annotated through a label that characterizes the object type (e.g. car, tree, or chair). These labels allow the learning mechanism to identify objects associated with their representation in the data. Among approaches to machine learning, there are two sub-categories to process object detection. The first sub-category aims at improving feature-based object recognition approaches through a more flexible and robust classification based on the learning. The second sub-category represents the object recognition approaches entirely based on machine learning technologies.

6.2.1 Machine Learning classifiers

Among the first sub-category of Machine Learning, called Machine Learning classifier, three main types of approaches emerge. The first type of approach is based on "Markov random fields" [*Rue and Held*, 2005]. The second type is based on "Random Forest" [*Breiman*, 2001]. Finally, the last and the most used type is based on "Support-Vector Machine" [*Cortes and Vapnik*, 1995]. These three types of Machine Learning classifier follow the workflow presented in Figure 6.3.

Markov random fields The approaches based on Markov random fields aim at creating a no-oriented graph between a set of random field verifying the Markov property through active learning. Different approaches such as [*Niemeyer et al.*, 2011], and [*Lu et al.*, 2009], use then, the Markov random fields to

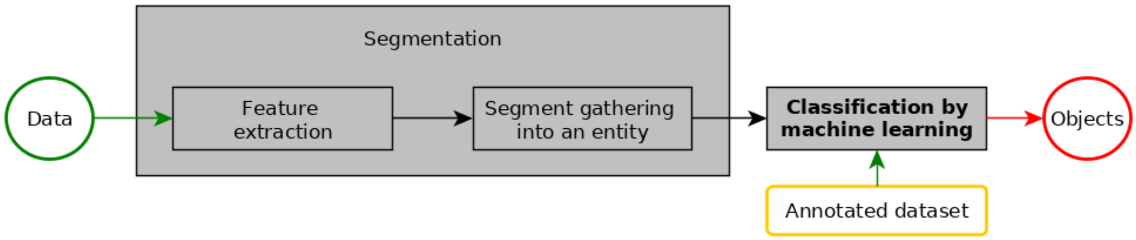


Figure 6.3: Common workflow of Machine Learning classifiers.

classify objects in point clouds acquired by Lidar scanner in an urban context. The authors of [*Husain et al.*, 2014] use this strategy to classify objects like books, tables, chairs, and walls in point clouds representing an indoor context. These two different contexts of application illustrate the flexibility of this strategy.

The authors of [*Luo et al.*, 2018] present another approach using this strategy combined with human supervision and classification. This approach begins with segmentation using super-voxels. User labeling follows the segmentation. This labeling provides stable and reliable support for the use of Markov random fields to label the rest of the elements efficiently. Although its efficiency, this type of Markov random fields requires much human supervision, moreover it has a high cost to process an extensive set of data. That is why this strategy is qualified as lazy.

Random Forest Random Forest-based approaches consist of creating a multitude of decision trees. Each decision tree is trained on a portion of data that differs slightly from the others. This training allows for improving the efficiency and adaptability of the classification used in feature-based object recognition approaches. Among the different approaches based on Random forest, the authors of [*Sun et al.*, 2018] present an efficient approach, but this one requires human intervention. This approach makes a segmentation by voxel, then groups the voxels according to their local convexity. When data allows it, the extraction of geometric features [*Weinmann et al.*, 2015*a*], height-features [*Maas*, 1999], spacial features [*Rabbani et al.*, 2006], and radiometric features [*Aijazi et al.*, 2013] enrich each set. A user must perform and parameterize this extraction of features, but also the selection of neighborhood distance. This neighborhood distance allows for the study of local convexity and the creation of a voxel adjacency graph. This human supervision is static and cannot be adapted to the variations of the data quality (e.g. variations of noise and density). However, the classification is automatic thanks to the creation of decision-tree forest during the training phase. Nevertheless, this classification does not allow for solving ambiguity problems. This type of approach has difficulty in differentiating the small objects having similar geometric features.

Support-vector machines (SVM) The classifiers based on Support-vector machines (SVM) are designed to solve classification and regression problems. These classifiers are generic that allows a more flexible application in different contexts.

The authors of [*Serna and Marcotegui*, 2014] use such classifier after a segmentation. Their segmentation consists of generating a 3D data elevation image to segment the ground using a 2D model-driven approach. Points that do not belong to the ground are then gathered into segments according to the Watershed approach [*Couprie et al.*, 2011]. The authors extract then, the contextual features of every segment. Morphological filtering allows for removing small and isolated segments. Finally, an SVM classifier is applied to the segments keeping by the filtering. The disadvantage of this approach is to consider only spacial information and not information about the texture or shape of the elements. Besides, the used segmentation is sensitive to occlusions that conduct to an over-segmentation for close objects, and a sub-segmentation for objects with a complex shape.

The authors of [*Lehtomäki et al.*, 2016] generate a histogram based on local descriptors and point repartition of every segment from the data segmentation. An SVM classifier uses then, the histogram to classify the models and identify objects. However, this approach fails to differentiate close objects, which are considered as one object. This failure is due to the limit of segmentation. The sub-segmentation leads to obtain one histogram for a set of elements that have been considered as one element. Therefore, SVM classifier cannot detect these elements. This problem shows that a proper segmentation is crucial to obtain proper histograms for the learning phase.

Therefore, the authors of [*Yang et al.*, 2017] propose to improve the approach of [*Lehtomäki et al.*, 2016] through finer segmentation. These authors propose a segmentation in four steps to segment elements of an urban scene. Firstly, the segmentation aims at extracting the ground by classifying points between two categories: "in-ground" and "out-ground". Secondly, the approach gathers points (from the category "out ground") into elements based on their proximity. Thirdly, it extracts features of every segment by using geometrical information and the redundant information (e.g. a mast that appears every 100m). Finally, it gathers all segments according to their similitude to build a model and compute a histogram of these objects. Nevertheless, this approach can only detect object defined by an accurate histogram. The approach requires to have one histogram for every different representation of an object in data, but also every variation of object feature. Therefore it leads to an exhaustive set of histograms for proper detection.

Although SVM classifier can be adapted to different approaches of segmentation, they are based on statistical information to perform the classification. That is why they cannot integrate the complexity of shapes or the object texture to improve the efficiency of the classification.

Combining classifiers The authors of [*Li et al.*, 2016] propose to face the limits of the different classifiers based on machine learning by combining them to improve the quality of results. The authors combine an SVM classifier with a multi-label graph cut ([*Sedlacek and Zara*, 2009]) to firstly segment elements having a simple texture and shape. They classify then the other elements by using a decision tree defined for sought objects. Results obtained by classifiers combination have better quality than results obtained by a single classifier. However, approaches that combine classifier have a lengthy processing time.

The approach [*Xiang et al.*, 2018] presents an object detection in three steps in urban scenes. This approach applies firstly, a P-linkage segmentation ([*Lu et al.*, 2016]). This segmentation performs first a region growing according to a hierarchical structure. It is an approximative segmentation, but fast. The approach [*Xiang et al.*, 2018] characterizes and classifies then, the obtained segments through the use of three different classifiers: Support Vector Machine, Random Forest, and Extreme Learning Machine. Finally, the approach refines the first classification through an energy minimization via graph cuts [*Boykov et al.*, 1999]. It uses the context around the classified objects to correct the first classification and improve the accuracy of classification. This approach illustrates the importance of the context for proper object recognition. Despite this improvement, this approach has difficulties in recognizing objects with similar geometric features (e.g. distinguish fences and buildings that have both a planar and vertical geometry, which is perpendicular to the ground).

6.2.2 Convolutional neuronal network (CNN)

The second category of data-driven approaches using machine-learning follows the workflow presented in Figure 6.4.

Among the Machine Learning approaches, the convolutional neuronal network (CNN) allows to excellent results. It experiences a great effervescence in computer vision, especially for 2D images as shown by the approach [*Krizhevsky et al.*, 2012*a*] that uses high definition images. Nevertheless, all approaches based CNN require a

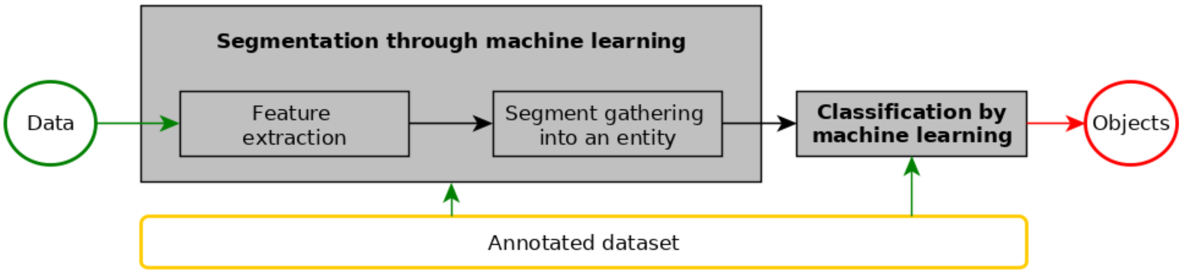


Figure 6.4: Common workflow of Machine Learning detection approaches.

broad set of training data. The approach of deep learning presented by the authors of [*Dong et al.*, 2014] is based on CNN and allows for obtaining excellent results on high-resolution images. It applies the training on a large dataset of high-resolution images. However, the training is long and challenging to parameterize to obtain proper results. "Resnets" [*He et al.*, 2016] improves the learning by allowing a more straightforward and faster training.

For object detection in 3D point clouds, the approach [*Du et al.*, 2018] presents a general pipeline to extract the features of 3D data through the use of model-fitting. This approach reduces then, the complexity of data by a 2D projection that allows for using CNN directly. The authors of [*Meyer et al.*, 2019*b*] presents an approach using CNN to predict the location of objects in point clouds by transposing the point cloud into images. If the point cloud is designed from 2D images, this conversion from point cloud into images is not necessary, and the approach [*Meyer et al.*, 2019*a*] can use the 2D images directly to improve the detection process. These approaches require complete data (without occlusion) and objects with very different features to obtain excellent results.

Several approaches use mechanisms of 3D data projection into 2D images to apply then a CNN approach. Due to the long processing time, some approaches focus their research on the speed improvement of CNN processing. Indeed the authors of [*He et al.*, 2015] improve CNN with an approach called "SPPnet". This approach uses a spatial pyramid to reduce the complexity of images and increase the processing time. The approach, called "Fast R-CNN" [*Girshick*, 2015], adapts CNN, by proposing several improvements that reduce its processing time significantly. The authors of [*Ren et al.*, 2015] also propose improvements to process data in real-time through CNN. Although these approaches improve the CNN processing time, they execute more than 100 test iterations to compute each region of the sub-networks. That is why the authors of [*Dai et al.*, 2016] presents the "R-FCN" approach that shares the computation on the whole image through a fully convolutional neuronal network. The authors explain they integrate "ResNet" and claim their approach is faster than the fast R-CNN.

Among approaches working on 3D data, the approach [*Ku et al.*, 2018] proposes an aggregate view object detection based on region proposal network. This approach builds a 3D oriented bounding box to wrap objects after a segmentation. This approach aims at solving problems linked to the object orientation. The geometrical features of an object change according to its orientation creating classification ambiguities. The objects contained in this oriented bounding box are then characterized by feature descriptor from model-driven approaches. The extracted features are then used by CNN. Although this approach is efficient on the specific use case (self-driving car), it is inefficient on images with deformation. The authors of [*Ouyang et al.*, 2015] address the problem of object detection in images with deformation. They increase the robustness of detection in this context through a deep convolutional neural network adapted to images with deformation.

These different approaches based on CNN require that data are organized in a tensor structure. As 3D point clouds are not intrinsically structured in that way, the use of spacial and geometric object features for their identification is limited. Due to this limit, the authors of [*Li*, 2017] focus their research on the direct use of 3D data to detect vehicles in point clouds. They address this issue through an adaptation of fully convolutional network (FCN). This adaptation is based on FCN used in 2D like the approach [*Long et al.*, 2015] that trains a CNN pixels-to-pixels to obtain better results both for segmentation and classification. This 2D FCN approach is itself an adaptation of the approaches [*Szegedy et al.*, 2015], [*Simonyan and Zisserman*, 2014] and [*Krizhevsky et al.*, 2012*b*].

CNN applied voxel-grid The object detection in variable 3D data is challenging and hardly manageable by CNN approach. Process irregular data through CNN requires to take into account all possible representations of an object that can appear in these data, to train the system. In this case, it requires a dataset containing the different shapes and representations of an object exhaustively. Such intensive training has the risk to lead the CNN to overlearning that reduce its global efficiency among others by increasing problems of detection ambiguity. That is why a pre-processing step is applied to use approaches based on Region Proposal Network (RPN). The most widespread is the transformation of 3D point clouds into a voxel-grid.

The approach [*Tchapmi et al.*, 2017] uses an FCN combined with a Conditional random field (explained in [*Dobruschin*, 1968]) to increase the efficiency of object detection. This approach simplifies the point cloud through a voxel-grid pre-processing. It uses then the FCN adapted to 3D to classify each voxel and projects the result of

FCN in the original point cloud through a trilinear interpolation. This approach allows for obtaining proper results both in indoor and outdoor scenes. These results are significantly better than CNN applied on 2D images or the different classifiers of machine learning (SVN, random forest).

The approach called Voxelnet [*Zhou and Tuzel*, 2018] segments the point cloud in voxels having the same size. It characterizes then each voxel to group them according to their shared features. The authors call this voxels gathering according to their features a "voxel feature encoding" (VFE). The VFE is then connected to RPN to predict the bounding box that locates the sought objects.

This approach obtains proper results to detect a set of objects that have not significant variations of size. Similarly, the authors of [*Maturana and Scherer*, 2015] use a "3D convolutional neural network" such as the approach [*Ji et al.*, 2012] to detect objects based on the relative location of elements between them and their different sizes. They realize such a process through the study of repartition in voxel-grid. This approach highlights the importance of topological relationships between the different elements. It also highlights the importance of the size in the identification process. However, it has difficulty to differentiate objects with a similar size. This limit is a critical limit because many scenes contain objects with similar size.

CNN applied to points The transformation of a point cloud into a voxel-grid simplifies the point cloud. Nevertheless, it limits the functionalities of the point cloud usage and also limits the quality of object detection. That is why the approach "Pointnet" [*Qi et al.*, 2017*a*] allows an object detection without steps of point cloud simplification by working directly on the points. The authors of this approach present a deep-learning framework that computes normals of point, and points according to their neighbors (the closest points). This approach aggregates then, points information locally and globally, to gather points into elements that can be identified as an object.

The authors of [*Qi et al.*, 2017*b*] improve this approach through an approach called "Pointnet++". This improved approach uses a hierarchical structure of neuronal networks, which is applied to each required network parts recursively. The measure of space distance is the base to learn the local object features and improve the contextual information of models that represent objects to detect.

Finally, the approach called "IPOD" of [*Yang et al.*, 2018] improves "Pointnet++" by using a CNN directly on each point of the cloud. This methodology keeps information about the precise location of points and reduces the ambiguities of detection. It uses the backbone network of "Pointnet ++" and predicts the box that bounds the sought or located objects.

Despite the improvements, "Pointnet", "Pointnet++" and "IPOD" cannot recognize elements having a rather subtle and precise pattern. They do not work well on complex scenes composed of many different objects. These approaches are sensitive to noise and variations of data density, like the others CNN based approaches presented in [*Ouyang et al.*, 2015] and [*Wirges et al.*, 2018].

6.3 Semantic data-driven approaches

From the evolution of Web semantic and its technologies, many knowledge-based approaches emerged, which also constitute a significative branch of artificial intelligence. These approaches use Semantic Web technologies to represent explicitly and process human knowledge. It allows inference of new facts from the defined knowledge and a base of fact. In computer vision, the integration of knowledge about objects allows for obtaining a robust classification based on logical reasoning verifiable by humans. The knowledge is modeled by description logic, mainly through the language *OWL2* and by logical rules, mainly through the language *SWRL*. Thanks to knowledge representation, these approaches do not require training data.

The first step of these approaches consists of extracting information from sought objects to generate knowledge. The type of extracted knowledge varies according to the used approach. The authors of [*Anguelov et al.*, 2005] use a description of the object detection context corresponding to the scene that contains them. They use Markov Random Fields to extract this description. The approach [*Triebel et al.*, 2007] extracts the object geometry in 2D or in 3D through various techniques, to facilitate object recognition.

The semantic technologies are mainly used in data-driven approaches to extract features and to build an object representation or to improve the process of machine learning-based classification.

Semantic in features extraction and object building model Figure 6.5 presents the workflow of semantic feature extraction and object building model. The approach [*Pu et al.*, 2011] uses the semantic to improve the extraction process of data features. This approach performs first segmentation of the ground, in an urban context. It organizes then elements hierarchically and classifies them into two categories: "on the ground" and off the ground". It applies a second segmen-

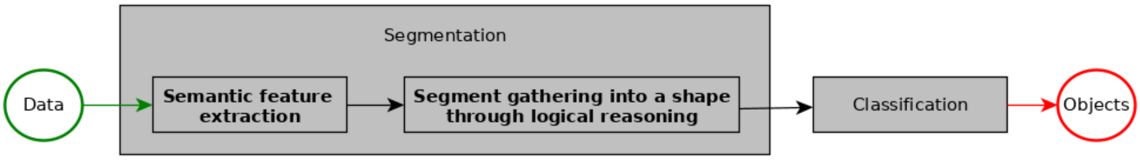


Figure 6.5: Common workflow of approaches using semantic feature extraction and object building model.

tation phase on the classified elements. These segmentations use model-driven approaches of geometry fitting. They aim at extracting features and detecting planar surfaces with Hough transform, allowing the detection of the ground as the largest horizontal plane. This approach uses a rule-based system to deduce knowledge from the segmented elements. This rule-based system uses information about point coordinates and their bounding structure (e.g. plane, line) to gather elements into shapes. The authors exploit then, the spatial relationship between the different created shapes. The disadvantage of this approach is its sensitivity to data variations and the choice of the threshold(s) used by the different algorithms. For example, this approach does not allow the detection of an object with low density. This lack is a problem with data acquired by a laser scanner and containing objects with high reflectance. The high reflectance of an object produces an object representation with a low density.

Semantic in classification process Figure 6.6 presents the workflow of approaches using a semantic classification to solve problems of machine learning ambiguities. The approach [*Torralba et al.*, 2010] presents a shape-based recognition

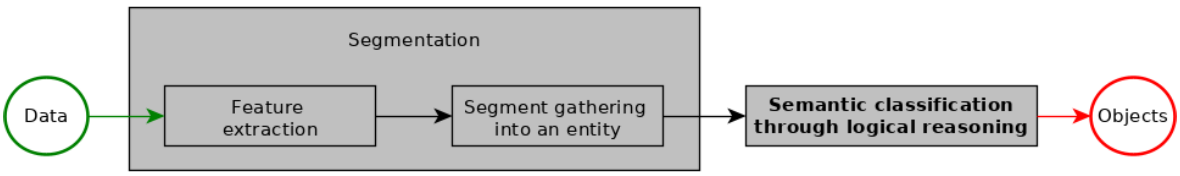


Figure 6.6: Common workflow of approaches using a semantic classification.

using a prestructured knowledge to identify both semantic and geometric classes for objects. In this approach, the semantic techniques are used to represent the content of the 3D model resulting from the identification system of classes.

The authors of [*Yang et al.*, 2015] use the semantic technologies to identify objects after the segmentation. The segmentation used is based on attributes of each point to group them into super-voxel. The authors create then a graph of super-voxels and compute the saliency of each super-voxel. Next, they merge the adjacent segments through the use of the rule-based semantic system (like SWRL). These rules build a logical representation of the merging results in the knowledge base. This logical representation in the knowledge base allows then for detecting objects without ambiguity.

Similarly, the authors of [*Babahajiani et al.*, 2017] apply a segmentation by supervoxels and use a hybrid strategy to detect the objects. This hybrid strategy is based on a rule-based semantic detection for the detection of objects having a simple shape or representing a large structure like facades or roads. It also contains an enriched methodology for the detection of objects with a more complex shape like trees, cars, or pedestrians. This enriched methodology begins with a boosted decision tree to classify the first elements in different models. It continues by a template matching to identify objects. Voxels are then classified in a supervised manner. This approach is high-speed compared to other approaches using semantic technologies. However, this approach requires many templates that must be adapted to the variations of data. The disadvantage of this approach is its sensitivity to variations of data quality and its difficulty to detect small objects or building parts when the data density decreases.

The data-driven approaches using the semantic apply the segmentation steps with threshold values chosen arbitrarily. These approaches have difficulties in detecting objects with free shapes. Some of these approaches have the advantage to solve the problem of detection ambiguity between objects that have slightly different geometrical features through reasoning. For example, they can apply reasoning on topological relationship to differentiate similar objects.

6.4 Discussion

The feature-based approaches for object recognition allow for solving problems in a precise context. They have the advantage to obtain a high quality of detection and precision due to a selection of ad hoc algorithms adapted to the data. However, the configuration of each algorithm must be done manually and empirically for each type of data. Therefore, this parameterization limits their flexibility. Thus these approaches are not generalizable. Moreover, the used algorithms do not adapt automatically to the variations of data quality (e.g. a variation of noise, a variation of density). These approaches depend entirely on object representation in data. The final result of detection depends on the quality of the algorithm output results. Thus, errors or default of results produced by an algorithm accumulates when several algorithms are combined. Moreover, they do not take into account the different characteristics of the data. Therefore, the quality of their results decreases proportionally with the decreasing data quality (e.g. an increase in noise, a decrease in density).

These approaches are not very robust, and their classification system is rigid. Therefore, the classification step is often replaced by the use of classifiers based on machine learning or semantic approaches. The relevance of these approaches depends mainly on the object's characteristics. Furthermore, such objects must be different enough to be well classified and to decrease the ambiguity, which is not managed.

Finally, the potential improvement of the results obtained is limited to an empirical improvement of the combination of algorithms and an empirical parameterization of each algorithm. Besides, the detection strategy must be adapted empirically to each specific context of detection. This adaptation capability is thus limited and insufficient to detect objects in heterogeneous data with wide variations of density and noise.

On the other hand, the machine learning approaches are efficient to solve problems of classification and regression but require a large dataset of training. They allow for obtaining proper results in various types of data as long as a sufficient dataset similar to the data to process is available for the training. However, they have a low precision of detection. Indeed, the different machine learning approaches cannot intrinsically adapt their approach to the sought objects. This lack of adaptation leads to problems of over-segmentation or sub-segmentation. Besides, the approaches based CNN do not provide a precise detection (point by point), but rather an extraction and location of objects through a bounding box.
Moreover, they do not solve the ambiguity problem of object classification generated by the variations in the data.

The machine learning approaches lose efficiency when data are noisy or incomplete. Various factors cause this incompleteness: for example, occlusions and reflectance decrease the density. Furthermore, the results of these approaches are entirely dependant on the annotated dataset used during the training. Thus, the approaches lose in quality and robustness when the representation of objects in the data to process, is not or few present in the training dataset.
The main disadvantage of machine learning approaches is their flexibility.

Indeed, machine learning approaches require adapting all training data to provide relevant results. The training data must contain the objects sought with characteristics that are similar to the characteristics of the data to be processed. However, multiple factors influence data characteristics. These approaches do not consider these factor. Thus, they cannot compensate for data characteristics variations that are too large for these approaches. Moreover, the improvement of their results is hardly manageable due to their dependence on the training phase. The increase in training data leads to overlearning that reduces the quality of results by increasing ambiguity of detection. That is why the training phase is hardly changeable.

The main difficulty of data-driven approaches using semantic is to structure well the relationships that objects have between them. The semantic approaches require an accurate knowledge representation of objects in the knowledge base. This representation solves the ambiguity problems of classification and provides a decent quality only if all algorithms are correctly parameterized. However, all possible representations of an object in the data must be described exhaustively by experts, and that for each type of data used. That is why these approaches are not adapted for the process of different types of data or data with a variety of quality (e.g. a variation of noise or density).

These approaches use semantic technologies only to address the part of the recognition process. Therefore, the obtained results depend entirely from intermediate results obtained by the segmentation phase and the feature extraction phase. These two intermediate phases are not guided by semantic; therefore they are conducive to error propagation.

Finally, the flexibility of the semantic approaches depends on the choice and the parameterization of the algorithm. This dependence limits the improvement of results because the algorithms are not adapted automatically to the variations of data quality.

Table 6.1 summarizes the advantages (+ or ++) and disadvantages (- or –) of the different types of data-driven approaches.

Approach	Quality	Ambiguity	Robustness	Flexibility	Generalizability
Features-based object	++	- -	- -	-	-
Machine learning	=	- -	+	- -	- -
Semantic data-driven	+	++	- -	-	=

Table 6.1: Comparative table of the different types of model-driven approaches.

The approaches based on feature-based object recognition are not robust enough and cannot be improved enough to work efficiently on point clouds with a high variation of quality (such as use case 2.1.3 and 2.1.4).

The approaches based on machine learning have higher robustness that allow them to work on use cases 2.1.3 and 2.1.4. However, they have not sufficient quality and cannot solve the problem of ambiguity to work on the use cases 2.1.2 and 2.1.1. Moreover, these approaches cannot be generalized.

Machine Learning approaches remains unflexible and are unable to detect the object or the geometry for which they are not trained. Moreover, some data characteristics variation can jeopardize the results of Machine Learning approaches. Furthermore, just a small change of the acquisition process or extern factor may produce such data characteristics variation. The approaches based on semantic data-driven apply logical reasoning on knowledge representation to recognize objects. Thanks to this advantage, these approaches are efficient on use cases 2.1.2 et 2.1.1. Nevertheless, they are not robust enough to work on use cases 2.1.3 and 2.1.4.

7 Knowledge-based approaches

As explained in chapter 6, data-driven approaches fail in their detection when the data are incomplete (occlusion) or have too much discontinuity. Indeed, data-driven approaches do not use prior knowledge about objects or data. These approaches cannot, therefore, be adapted to the multitude of configurations corresponding to different variation combinations of data characteristics and objects features. Therefore knowledge-driven approaches are more adapted to perform object detection from data in accordance with human knowledge about both objects and data characteristics. To this end, these approaches structure and represent knowledge about objects and data into ontologies, mainly through languages (OWL2). These approaches then use rule-based systems such as (SWRL) to support decision-making in the object detection process.

The success of these approaches depends on understanding the relationship between different objects and between objects and data. These approaches require the implementation of a strategy to understand the scenes which compose the data and in which the objects are represented, to achieve a more robust, generalizable, and flexible object detection.

These approaches fall into two subcategories. The first subcategory includes approaches using a detection process and a predefined knowledge about objects and data adapted to each application case. They use the predefined knowledge to identify objects through the use of logical reasoning. The second subcategory includes approaches that use predefined knowledge to adapt to the object detection process, both in the choice of algorithms to be used and in the identification of objects.

7.1 Knowledge-based object detection approaches

The knowledge-based object detection approaches design a detection system based on a predefined knowledge of objects and data. Figure 7.1 presents the workflow of approaches belonging to this category.

In the field of object detection in data from archaeological sites, the book

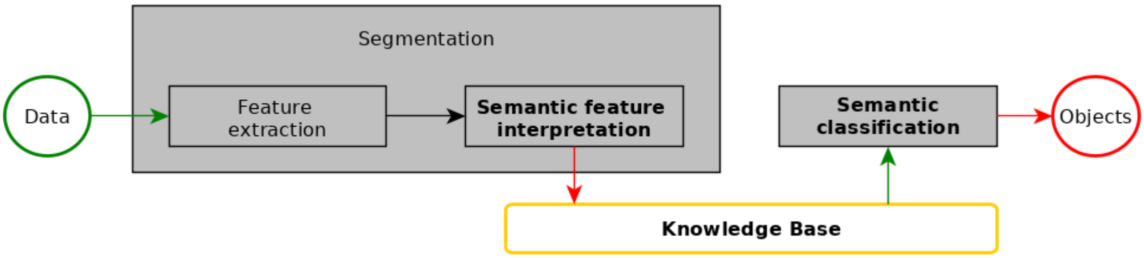


Figure 7.1: Common workflow of knowledge-based object detection approaches.

[*Poux et al.*, 2017] presents a detection process entirely designed to perform a classification of elements that is knowledge-based and uses semantic techniques. In the study cases reported in this book, the objects to be detected are "quasi-planar". The knowledge of these objects is expressed in OWL in an ontology. The detection process segments the point cloud then estimates the properties of each segment and stores these properties in an ontology. These properties relate to the color of each element, the material of the element, and the "light property". The description of the element links the material property to the "property of light", mainly concerning reflectivity. For example, a "matt rock" is related to the "non-reflective" "light property", and a "matt silver" is related to the "reflective" "light property". The combination of this information allows a better understanding of colors. Indeed, the color of an element is influenced by the light and material of the element. Thus, an element could have the same RGB color value than another element even if it should not be the case due to the light and material. The detection process can better understand the color of an element and therefore better detect objects, through knowledge of these characteristics and their relationships. The detection process classifies the elements by reasoning on the description logics expressed OWL2 within the ontology. This reasoning is performed by OWL reasoner such as Hermit [*Shearer et al.*, 2008].

The previous example highlights the power of knowledge management to improve the object detection process by better-exploiting feature information. This knowledge of the characteristics and their relationships can be extended to other approaches. However, this approach is limited to each application case and does not allow the classification of objects with complex shapes. In addition, this approach requires much-predefined information about light and materials, which is not generally known for many applications.

The work presented in [*Hmida et al.*, 2013] proposes a detection system using predefined knowledge of objects (e.g. mast, signals) and data (e.g. Lidar point cloud) in the railway context. The detection process, presented in this work, consists in detecting mainly linear elements (mast, signals). To do this, the authors use different combinations of model-driven approaches, such as RANSAC to extract geometric shapes from the data (line). They then group the lines close to each other in a box to form a "domain concept" to be classified. The classification is based on the definition of an ontology and the application of the SWRL rule. Knowledge of algorithms, geometries, characteristics, domain concepts, and scenes make up this ontology. Properties are used to link this knowledge together to form a consistent set that is conducive to the application of logical reasoning. Indeed, algorithms are defined as "designed for" a geometry. Geometry is defined as "has topological relation" with itself and "has characteristics" according to characteristics. The characteristics are used with the geometries to classify the "domain concept" via SWRL rule. Indeed, the "domain concept" "has topological relation" with itself, "has geometry" and "has characteristics". Scenes are defined as containing "domain concept".

The conceptual relationships defined in the ontology then allow the approach to defining objects to be detected through these relationships. One of the most commonly used topological relationships in this work is the spacing distance between two "domain concepts" and especially the recurrence of this spacing. Indeed, this work uses the recurrence of spacing between the "mast" sought in a rail point cloud, which remains constant. Thus, the detection of an object (mast) using these geometric criteria allows the detection of other objects that would not have been identified by their geometric characteristic.

The use of knowledge and more particularly, the exploitation of the topological and geometric characteristics of objects by semantic technologies significantly improve the robustness of approaches and solve ambiguity problems. The structuring of knowledge in the form of ontology also facilitates the adaptation of the approach to other fields. However, this approach does not allow to detect complex objects when there is not enough topological link between them (no recurrence). Besides, the quality of detection is quite poor due to the use of inclusive boxes to identify objects rather than the use of precise segmentation of objects.

The work presented in [*Poux et al.*, 2018] uses semantic technologies and knowledge management to generate a "knowledge-based point cloud infrastructure with multiple Levelling Of Details" that significantly improves detection quality. They are applied for the detection of chairs of various types (garden, living room, office). The knowledge of geometry and the topological relationship of each of these types is predefined in a knowledge-based. The authors propose to enrich the point cloud by characterizing it with semantic characteristics. To begin, they apply a point cloud voxelization to describe each voxel using a distribution histogram and shape descriptor. SWRL rules then extract the layout from the voxels to form elements. A representation in muti level of detail is then generated for each element. Element recognition is carried out by models-fitting created from predefined knowledge about each element constituting the objects to be detected (foot, site, and back are elements of chair). The detection process studies the relationships between elements to group/aggregate them and form an object. This work uses SPARQL queries to retrieve information about the detected objects.

This approach requires creating a model knowledge of each possible representation of each object to be detected. However, the representation of an object is impacted by data quality. Moreover, this approach does not provide management of data quality variation. Therefore this approach cannot be used effectively on heterogeneous (variable quality) data.

7.2 Knowledge-driven approaches

Knowledge-driven approaches use the knowledge to guide the detection process by selecting sequences of algorithms adapted to the characteristics of data and objects. Figure 7.2 presents the workflow of approaches belonging to this category.

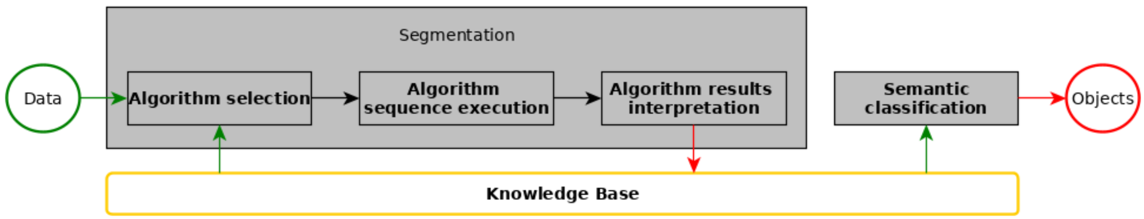


Figure 7.2: Common workflow of knowledge-driven approaches.

The work [*Dietenbeck et al.*, 2017] creates an expert system using a decision tree to select the best algorithm for the segmentation process. This decision applies only once to the overall segmentation process. For such a selection, the reasoner searches for compatible algorithms through a process of reasoning applied to the ontology that contains knowledge about algorithms and objects. All possible executable algorithms are instantiated. This work provides rules designed to remove incompatible algorithms by linking the output of each algorithm with the characteristics of each object (geometry, color, and topology) expressed in the ontology. Then reasoning on these rules removes all incompatible algorithms. Thanks to this process, the segmentation is adapted to the characteristics of the object. Algorithms can be selected and executed to perform object detection. The algorithms used in this work are mainly algorithmic models. The results of the execution of the algorithms are hand-parsed into semantic information to be incorporated into the ontology. Then, the application of rules and reasoning classify the segments provided by the segmentation process into the object. This work produces an adapted detection process that is flexible and capable of resolving ambiguity through a knowledge-managed classification process. However, the results obtained depend on the first decision taken and the result of each algorithm executed. Thus, the resulting detection cannot be adapted to variations in data quality that have an impact on algorithm results. Only objects composed of a primitive shape can be detected. Besides, a single decision made for the segmentation process does not take into account the experience gained from the results of the algorithms on the data. Thus, the detection process has low robustness.

Robustness to noisy data or variations in data density is one of the most challenging aspects in the Computer Vision domain, especially in object detection domain. The work [*Karmacharya et al.*, 2015] selects algorithms based on noise or data density and object characteristics. This selection is based on an Expert knowledge framework consisting of knowledge representation and an algorithm selection module. The approach [*Hmida et al.*, 2013] adds characteristics that influence the algorithms in order to improve knowledge about the algorithms, the "domain concept", the data, and the scene. These improvements allow the creation of a graph of algorithms' dependency and input-output dependency. The selection of algorithms is made by creating a graph of all possible combinations of algorithms. These combinations are created by connecting the input and output of the algorithm. Thus the graph of the possibilities start of the data and ends with the characteristics of the objects. The graph is composed of several possible uses of each algorithm. Indeed, the ontology used to construct the graph contains many combinations of parameter values for each algorithm. For a single algorithm, many sub algorithms are created with different parameter values, and each of them is linked to data characteristics. For example, for the RANSAC line detection algorithm, two sub algorithms are created. One is intended for the detection of irregular lines in low-density data and the other for the detection of high-density lines. Both algorithms are described in the ontology and are located inside the graph. Then, the algorithm called "Dijkstra" [*Dijkstra*, 1959] is used to navigate through the graph and find the best path in the graph. The path is interpreted as a sequence of algorithms that are executed. Then, reasoning on the SWRL rules classifies the results of the sequence of algorithms. These rules use topological characteristics to improve the classification of objects and solve many ambiguity problems. Besides, the approach of [*Karmacharya et al.*, 2015] allows the detection of multiple objects ordered by their size. Thus, the detection process first detects the most massive object described, and finally, the smallest object described. Finally, this work has demonstrated its flexibility by detecting objects in the indoor and outdoor point cloud. Indeed, this work detects noise-canceling walls in the context of railway point clouds, even if the walls have curves. Moreover, it also detects chairs in interior-point clouds, even if some chairs have a low density.

However, the algorithm parameters are fixed and are not dynamically adapted to the data characteristics or object characteristics. Furthermore, creating sub algorithms with different parameter values increases the number of algorithms, and the parameters are often imprecise. Besides, the selection of the algorithm sequence is applied only once. Meanwhile, the algorithm sequences are fully executed, even if some algorithms in the sequence fail or produce unexpected results. Thus, the detection process does not take into account the results of the intermediate algorithms. Similarly, the classification process is not iterated with the segmentation process. Thus, the detection process may fail when the geometric and topological characteristics are not sufficient to detect an object. Finally, this approach requires modeling of all possible objects and geometric characteristics according to the variation in data quality (e. g. lowercase line modeling, high density). Otherwise, the detection process fails to detect an object if its representation within the data differs from its expected form, modeled in the knowledge-based.

7.3 Discussion

Knowledge-based approaches mainly use knowledge about objects, data, and algorithms to strengthen the detection process. These approaches may solve many ambiguity problems by using topological knowledge between objects to improve the classification process.

Knowledge-based object detection approaches to design their detection process to use knowledge through semantic techniques. This knowledge is modeled within the ontology by the OWL2 language, which increases the flexibility of these approaches. However, this strategy used for the detection process is not dynamically adapted to the characteristics of the data. Thus, flexibility is limited by this strategy. The result obtained by these approaches can be improved by adding some knowledge within the limits of the results of the algorithms that have an impact on the final result. Indeed, these approaches do not manage variations in data quality and are mainly sensitive to noise. Thus, they are of moderate robustness. The accuracy of objects detection in these approaches depends on the strategy used. If the strategy is based on the location of boundary boxes, the quality of the detection is less accurate than the strategy based on a multi-resolution detection.

Knowledge-driven approaches also use knowledge to drive the detection process by selecting the algorithms that make up the detection processing. These approaches are, therefore, flexible. Indeed, the use of semantic technologies to select algorithms and perform classification allows these approaches to work on data that may have different characteristics. The quality of the results obtained by these approaches suffers mainly from a lack of parameterization of the algorithms. Indeed, algorithms have predefined parameter values that limit the quality of their results.

Table 7.1 summarizes the advantages and disadvantages of the both types of knowledge-based approaches.

Approach	Quality	Ambiguity	Robustness	Flexibility	Generalizability
Based on Knowledge	=	++	-	=	=
Driven by Knowledge	=	++	=	+	+

Table 7.1: Comparative table of the both types of knowledge-based approaches.

Knowledge-based object detection approaches are not adapted to use cases 2.1.3 and 2.1.4 because of their lack of robustness. Knowledge-driven approaches are more robust and can be applied to all cases presented in Section 2.1. However, these approaches are limited in quality by the parametrization of their algorithms. Moreover, these approaches do not consider the quality of the results obtained by the algorithms at each step of their execution, to adapt the detection process. Similarly, the process of segmentation and characterization of the data does not take into account the results obtained by the classification. Therefore the results are determined by the quality of the execution of each algorithms composing the sequence. Thus, these approaches fail to detect many objects representations, especially when this representation in the data differs from the model described in the knowledge-based.

8 Discussion

The literature review for object detection highlights several advantages of combining different approaches. Table 8.1 summarizes the advantages and disadvantages of the different approaches reviewed in the previous sections and compared to the approach presented in this thesis.

Method	Quality	Ambiguity	Robustness	Flexibility	Generalizability
Driven by Model	+	-	-	- -	-
Driven by Data	+	++	- -	-	-
Driven by Knowledge	=	++	=	+	+
Presented Approach	+	++	++	++	++

Table 8.1: Comparative table of the different types of approaches surveyed.

Model-driven approaches are not usable in the application cases presented in Section 2.1. Their real interest lies in their integration into other data-driven or knowledge-based approaches.

Indeed, among the data-driven approaches, the most effective features based object recognition approaches integrate model-driven approaches. However, this type of approaches does not automatically adapt the detection process to the data used or to the different types of objects sought. The user chooses and configures the algorithm sequences, and each individual algorithm empirically. Besides, each of these elements must be readjusted for any different use case. On the other hand, these approaches use in general classification system, which is static and often quite not efficient. Most recent approaches use a classification based on machine learning. Machine learning is increasingly used because of its autonomy to detect objects in data. However, they require annotated data for their training stage. This training stage must be focused on the detection of specific objects present in both the data to be processed and the training data. Moreover, each object must have similar representations in these two data sets.

Although more and more annotated data are available in different sectors of activity (e.g. autonomous car, BIM), the fast evolution of 3D acquisition technologies causes significant changes in data characteristics (e.g. resolution, noise, density, roughness). The evolution of these characteristics often compromises the relevance of old training data sets, requiring the creation of new sets. Indeed, the training data sets must have similar characteristics to the data used. The main limitations of machine learning approaches are their flexibility and their dependence on their learning phase, which can lead to overtraining if the learning phase is too extensive. The objective of this phase is to generate implicit knowledge from experience acquired through training. Indeed, machine learning approaches are intrinsically limited both for the point of view of object modeling (detection limited to representations sufficiently present in training sets) and data evolution (new acquisition technology). Finally, Machine Learning approaches are not able to detect objects or geometries for which they have not been trained. Therefore, they cannot understand the data, when the data characteristics differ from those on which these approaches were trained. For these reasons, other approaches propose to exploit some knowledge about objects and data to eliminate learning needs thus and considers more situations.

These approaches use technologies from the Semantic Web to manage human knowledge. This human knowledge can be used to improve existing data-driven approaches but is much more powerful and effective when used to drive the detection process. In knowledge-based approaches, knowledge is used to guide the selection of algorithms and the classification of objects. These two knowledge-based processes allow for having a detection system adapted to both the characteristics of the data and the characteristics of the objects sought. These approaches highlight the importance of using semantic technologies to manage knowledge to drive the detection process. Indeed, these approaches use the power of semantic technologies to be able to adapt the detection process to the object sought and global data characteristics (such as the average noise and average density in the overall data). These approaches use knowledge to generate a sequence of algorithms to segment and characterize each segment extracted from the data. The results of the execution of this sequence are then analyzed by logic reasoning to identify the objects. These approaches have the most considerable advantages for object detection in application cases such as the ones presented in Section 2.1.

The approaches studied show that the combination of geometric and topological characteristics on objects, allows ambiguity problems to be solved. Thus it allows for better dissociates objects. Primarily when used through OWL reasoner semantic classification technology, which allows geometric and topological characteristics to be linked while ensuring overall consistency through consistency checking. The combination of these characteristics allows for classifying an object through its topological relationship when the geometric information did not allow alone the classification. Thus, such a combination strengthens the classification process. However, it requires that the objects be strongly related to each other. These approaches have also shown that the use of knowledge to link objects, data, and algorithms together allows for the efficient creation of algorithm segmentation. Besides, the use of semantic technologies allows for managing vast knowledge about algorithms, objects, and data. This use allows for great flexibility and scalability of these approaches.

However, several capacities are lacking in these approaches to allow them to be more generic and effective, particularly on all the application cases presented in Section 2.1. Indeed, these approaches do not take into account the local characteristics of the data, especially when part of the data is already segmented. More precisely, the segmentation process may divide the data into unexpected segments if the data has different characteristics (such as high-density segments with low noise, or low-density segments with high noise). Besides, the execution of algorithms may run outside the control of semantic technologies. The detection process cannot be adapted to the results of each algorithm (and intermediate algorithms in an algorithms chain) that cannot be dynamically integrated. Therefore, segmentation and feature detection are performed only once, regardless of the needs of the classification process (more precise or specific features). Indeed, after the selection of algorithms and execution, the data are no longer enriched by other processes. As a result, the classification process cannot have more precise or context-specific information to identify objects. This is a limitation of the classification process. Finally, the main limitation of these approaches is their inability to adapt predefined knowledge about objects and data to specific application cases. Indeed, the description of the data provides useful information for the selection of algorithms. However, in the context of object detection in 3D point clouds, characteristics such as the resolution of the acquisition system, provide information on the potential density of the data. However, this expected density may be very different from the density found in some parts of the data. Similarly, predefined knowledge about objects allows the creation of a conceptual model of these objects, to guide their detection. Nevertheless, such a conceptual model can strongly differ from the observed representation of objects in the data, especially when the quality of the data is variable. When these discrepancies occur, the approaches previously studied fail in their detection process. This problem is called "indirect realism problem" in the rest of this manuscript.

Therefore we propose in this thesis a system of automatic detection of objects in 3D point clouds addressing these limits. Unlike knowledge-based approaches, the new solution we propose uses semantic techniques to drive processes and not just a few processing steps thoroughly. It integrates model-driven and data-driven approaches to characterize object geometry and topology through the use of algorithms, which significantly facilitates system scalability. Besides, this system also integrates semantic classification mechanisms. Such mechanisms aim at avoiding ambiguities and allowing fast and secure control and improvement through reasoning and consistency verification. The quality of the results obtained allows identifying objects in point clouds and those for different types of objects in heterogeneous data with variations in quality (variation in noise, occlusion, and density).

Besides, this system overcomes the problem of robustness of literature approaches and more particularly, the "indirect realism problem" by creating a self-learning process module based on semantic analysis and statistics. This semantic self-learning process allows to directly adapt the knowledge by integrating the experience acquired by the object detection process.

Part III

Methodology

The approaches reported in the literature part only use data characteristics or objects information that constitutes the data. Unlike these approaches, the proposed approach, presented in this methodology part, also uses information from the sensing process, which generates data. The characteristics of the data depend on the sensing process (e.g. instrument, acquisition method) and many external factors (e.g. ambient brightness, acquisition condition, motion); Small variations incurring during the sensing process by external factors can completely change the characteristics of the data. This methodology presents a knowledge-driven object detection that adapts step-by-step the detection process according to the processed data, the objects to detect and the performed processing. It also describes a self-learning process that improves object detection through the knowledge base enrichment by knowledge more specific to the processed application case.

Chapter 9 provides an overview of the system. It recalls the problems related to the detection of 3D objects. Then it presents the system components and their interactions.
Chapter 10 explains knowledge engineering. This one allows the modeling of the knowledge of the domains of data, the scene, and data processing. The purpose of this knowledge is to guide the object detection process.
Chapter 11 presents knowledge-driven object detection. This detection is, first of all, made up of an algorithm management phase which consists in selecting, configuring, and executing the relevant algorithms for processing the application case. A classification phase follows this management of algorithms. These two phases allow object detection to be performed according to explicitly defined knowledge. This detection is then followed by a self-learning process step aimed at enriching the knowledge base in order to re-execute a more accurate object detection.

9 System overview

This chapter explains the structure, the components, and the operation of the artificial intelligence system proposed, dedicated to object detection in unstructured 3D data.

9.1 Limits and problems of the 3D object detection

The digitization of data mainly produces 3D point clouds. The representation of objects to be detected is thus a point cloud. Acquisition systems discretize the information of the digitized scene, reducing then the information available to detect objects. Besides, different factors influence the acquisition conditions such as light, sensor sensitivity, object materials, acquisition context, and the variation in distance between sensors and digitized objects. These factors influence the representation of objects by transforming their geometric characteristics such as shape, size, orientation, and color. Thus, for a single object, there may be a multitude of different representations in the data. Moreover, more the scene to acquire is extensive, in the sense of composed of different object types, more its acquisition is complicated. More a scene acquisition is complicated, and higher is the probability of having characteristics variation inside the resulting data. This difference of characteristics corresponds mainly to a variation in noise, regularity, occlusion, and density. Such variations also amplify the diversity of representation for the same object in the same data.

The literature review shows that knowledge-driven approaches are the most appropriate solution to manage the different representations of objects. These approaches use expert knowledge to adapt the detection process according to the type of object sought and the quality of the data representing these objects. However, these existing approaches only take into account a few characteristics of the data. They are not able to adapt the detection process to some variations produce by the sensing process and external factors. Moreover, they combine several algorithms to create a sequence that extracts information from the data to identify objects. Besides the algorithms are greatly influenced by the data characteristics. Thus their effectiveness decreases sharply when the characteristics of the data vary. Furthermore, these approaches cannot take into account each of the possible representations of objects. Although these approaches use predefined knowledge of the different possible representations of objects, they fail to detect objects when the representation of an object, observed in the data, differs from its semantic descriptions. This "indirect realism problem " is common in the 3D point cloud.

Solving these problems and limitations requires the creation of a detection system that incorporates a form of "intelligence". This "intelligence" must allow the adaptation to the different object representations and the variation of data quality. Such adaptations require the management of several aspects. An intelligent process must firstly use knowledge to drive the entire detection process. Leading the knowledge-based detection process requires techniques to formulate, integrate, and use this knowledge. Then, the intelligent process needs algorithms to perform the processing. It must automatically evaluate the relevance of each algorithm. This relevance estimation depends on the processed data features and the sought objects features. The study of the significance of the algorithms should allow selecting the most appropriate algorithms to execute to detect the desired objects. A knowledge-driven detection process that executes algorithms step by step requires an understanding of each parameter of the algorithms to adapt their parameterization to the detection context. Therefore, the knowledge base must contain a link between the value of algorithm parameters and features values of the sought objects or the processed data. Then, the process must use these defined relationships to configure the algorithms. Furthermore, the detection process must dynamically adapt to the intermediate results of the executed algorithms. To do this, it needs to execute the algorithms dynamically and transform their results into knowledge. Such a result is a fundamental value (e. g. String, Integer, Boolean) or memory pointer, representing an object. The integration of memory pointer inside the knowledge base allows the algorithm execution on results of other algorithms previously executed. Then, the analysis of the results is necessary to update the knowledge of the data. Such interactions between knowledge representation and algorithms execution require a technical and conceptual connection between the knowledge management paradigm and the algorithmic paradigm. Finally, the results obtained must allow concluding. These conclusions should not only serve to identify the objects but also to guide the detection process to increase its effectiveness. Indeed, the analysis of these conclusions must be carried out to draw lessons from the experiments from which they come. Learning from unsupervised experiments requires the ability to formulate hypotheses to improve the detection process, to apply these hypotheses, and to draw new conclusions that will validate or invalidate these hypotheses.

9.2 Solution proposed

The system proposed in this thesis is based on the automatic detection of objects in 3D point clouds exclusively guided by semantic processes.

Figure 9.1 shows the main components of the system that addresses these requirements.

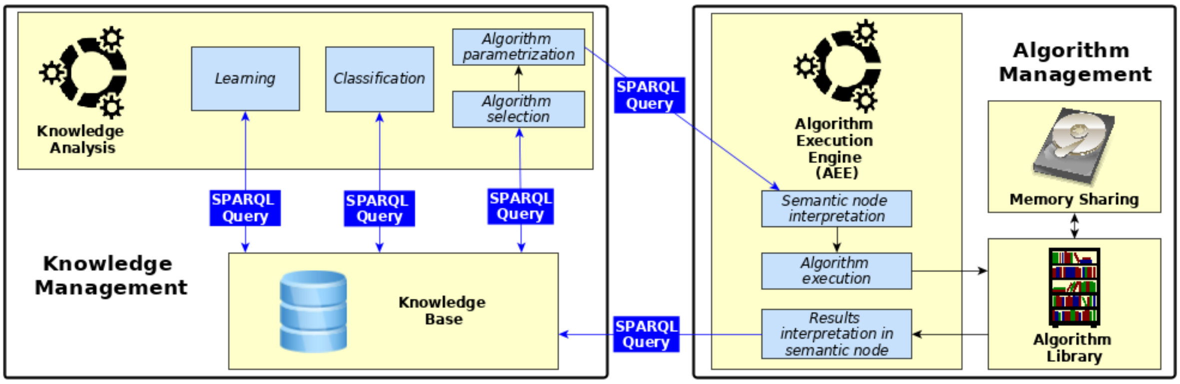


Figure 9.1: System Overview.

The system presented is composed of two main modules, a module for knowledge management and a module for algorithmic management. These two modules communicate using queries.

Knowledge is organized in a knowledge-base using a triplestore to store knowledge provided by experts, and knowledge dynamically created during the detection process. Chapter 10 explains this organization. The knowledge base contains knowledge about objects, data, and algorithms. The modeling of these three components is detailed in chapter 10. This knowledge is used and managed to guide the entire object detection process and adapt it to the data provided and the objects sought (explained in Chapter 11). The algorithmic management module is composed of three distinct components. These components are a library of algorithms, an Algorithm Execution Engine (AEE), and a submodule of memory sharing. The library of algorithms allows for covering the processing of different 3D data. The main component of this module is the Algorithm Execution Engine (AEE). The AEE uses the information contained in the received query to execute the algorithm related to this one. It then interprets the result to be returned by the query to enrich the knowledge base. The execution of algorithms by the AEE is under the control of a memory sharing submodule. This submodule allows algorithms to share their inputs and results while allowing access through knowledge management tools. The functioning of the algorithmic management module is detailed in Section 11.1.

The adaptation of the detection process begins with the selection of the most relevant algorithms. It selects these algorithms from among all the algorithms available in the algorithm library. The estimation of this relevance depends on the characteristics of the objects sought and the data considered. Figure 9.2 highlights the algorithm selection process in red. The selected algorithms are then automatically

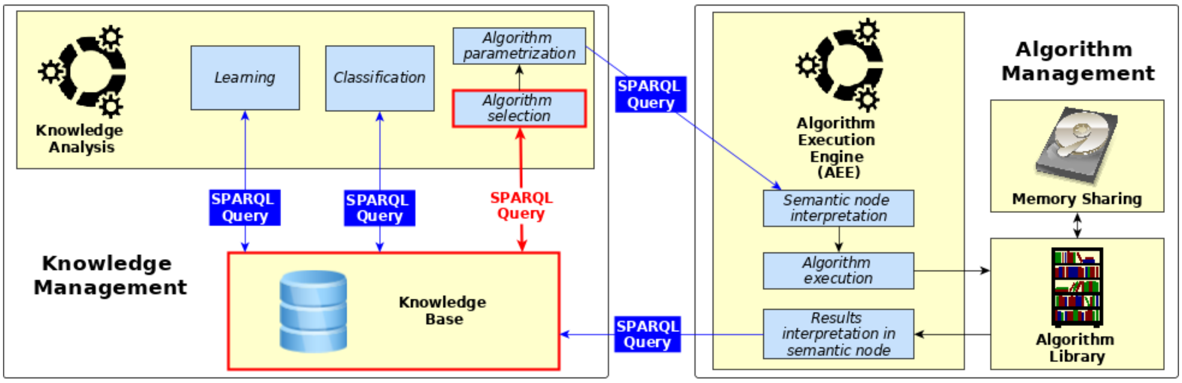


Figure 9.2: System Overview.

configured according to their execution context. Figure 9.3 highlights in red this step of algorithm configuration. In addition to the characteristics of sought objects and the processed data, the context of algorithms execution depends on the results obtained from already executed algorithms. The enrichment of the knowledge base

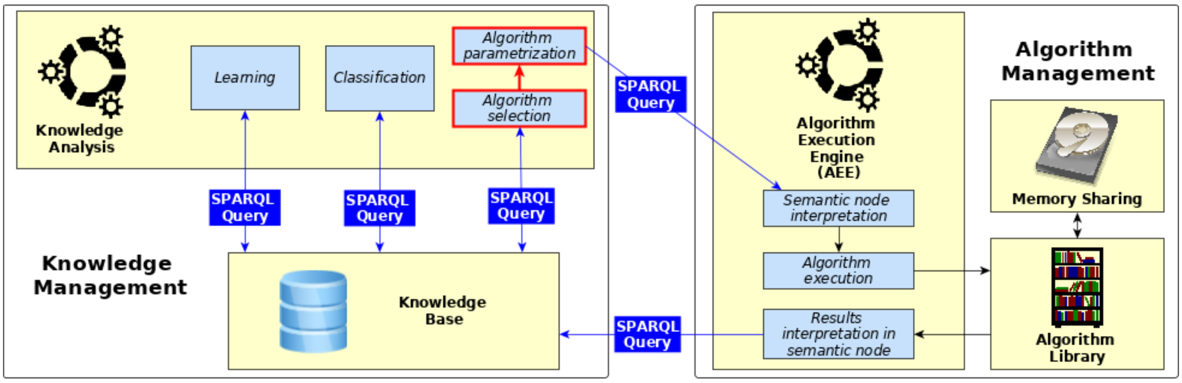


Figure 9.3: System Overview.

(in green in Figure 9.4), provided after the algorithms execution (process in red in Figure 9.4), allows to draw conclusions and identify objects by classification (process in orange in Figure 9.4). This step is followed by an analysis of the results to learn dynamically from experience, which they provide. This learning improves knowledge explicitly about objects and data, depending on the specific context of the application. Figure 9.5 highlights in red the learning process. The repetition of

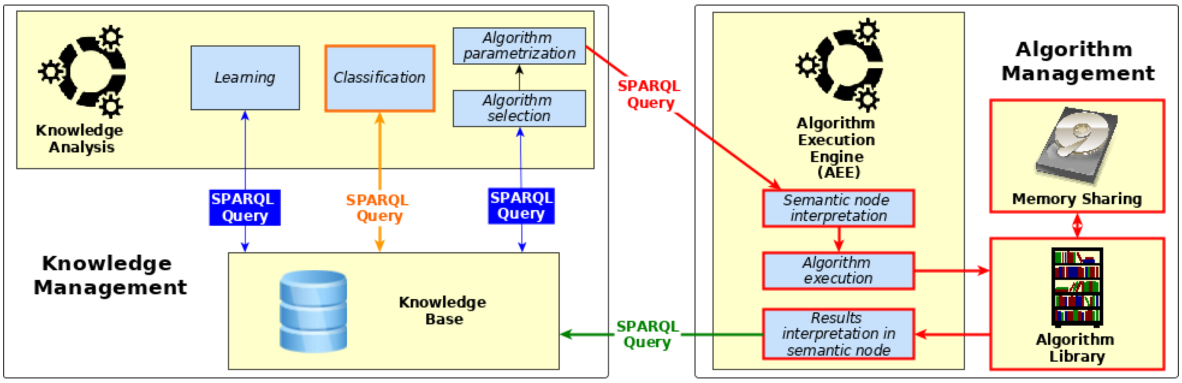


Figure 9.4: System Overview.

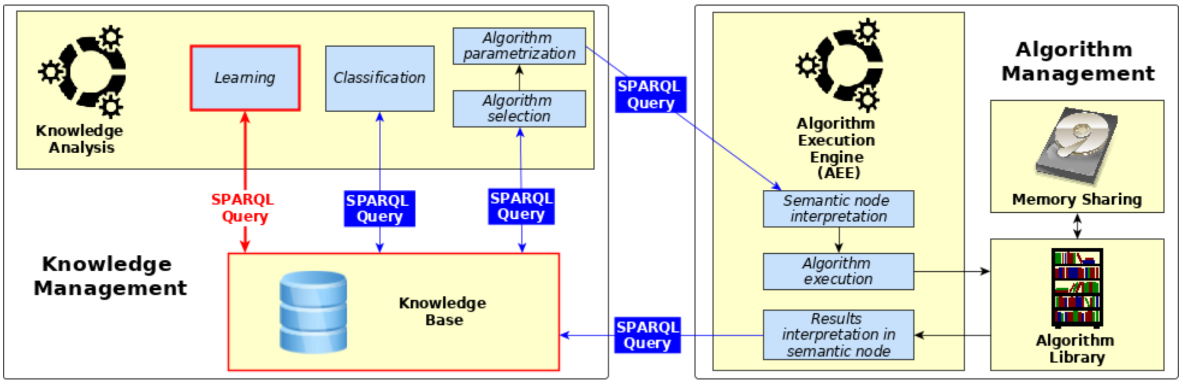


Figure 9.5: System Overview.

this strategy continues throughout the detection process until there is no more inference of new information. These iterations allow the system to become more and more efficient as object detection progresses. The adaptation and dynamic enrichment of knowledge through self-learning process from experience not only solves the "indirect realism problem" but also extends its application to a wide range of data sets and different application cases.

10 Knowledge engineering

This chapter explains how knowledge is structured and modeled to analyze and understand data content. Section 10.1 provides an overview of knowledge modeling to understand data content.

Section 10.2 explains data modeling and the impact of the Scene and various external factors on the acquisition process that generates the data and determines the characteristics that the data has.

Section 10.3 explains how to model objects, the geometry they have, and the Scene that contains them. These elements allow for adapting the process of data understanding, according to the geometric complexity of the objects, their topological relationships, their specific characteristics (e. g. colour, material, size) and the context in which they are located (e. g. ruin excavation, city street, modern building).

Finally, Section 10.4 explains the modeling of algorithms to detect objects and geometry in the data. Algorithm modeling allows algorithms to be selected, combined, and configured according to each application case. The algorithms are adapted according to the representation of objects in the data. The representation of an object depends on its geometry, its arrangement in the Scene, and also on the characteristics of the data. Moreover, data characteristics are impacted by the acquisition process, which is impacted by other external factors. The adaptation of algorithms considers all these factors.

10.1 Knowledge modeling overview

Knowledge is structured and modeled to guide the understanding of data content. Figure 10.1 shows the organization of knowledge.

It is necessary for understanding the data content to locate and identify the objects which constitute the data. These objects can be diverse (e.g. car, floor, wall, water mill, tree, table) and are characterized by their geometry (e.g. shape, surface,) and their topological links (e.g. the distance between objects, parallel, perpendicular).

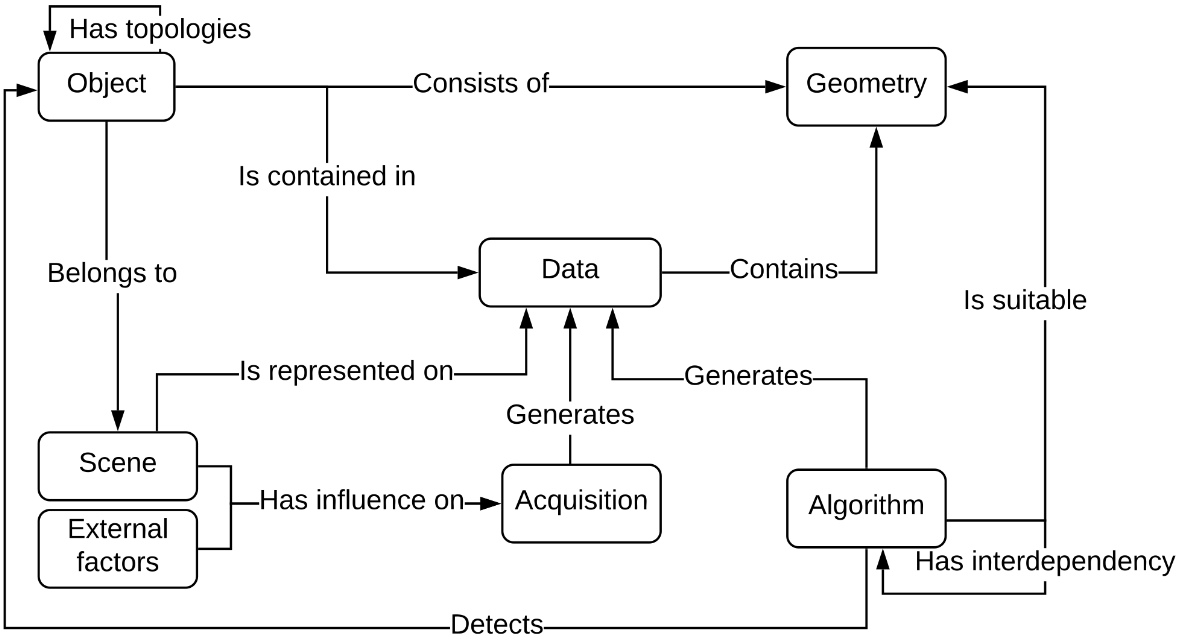


Figure 10.1: Overview of the knowledge structure.

Locating and identifying objects requires the use of algorithms. Algorithms are designed to identify geometries (e.g. plane, line, sphere, segments, orientation) in data to allow the detection of objects. They produce new data (e.g. normalized data, sampled data, filtered data) or new data characteristics (e.g. segments, local density, homogeneous regions). As an illustration, consider an algorithm designed to detect plans (such as RANSAC or Hough's transform). It generates segments (point groups) from data where each segment represents a plane. Thus, it allows the detection of different objects whose geometry is composed of planes (e.g. table, desk, chair, wall, ceiling, floor, wardrobe, door). The behavior of algorithms depends mainly on the characteristics of the data (e. g. size, density). Furthermore, since they produce data characteristics, and their behavior depends on these characteristics, the behavior of algorithms depends on the results produced by other algorithms. Therefore, they are interdependent.

The characteristics of the data depend on the acquisition process that generates the data. However, the acquisition process is influenced by the Scene which contains the objects to digitize (e.g. one object may occlude another), and by various external factors (e.g. the light intensity, the light color, the vibration of the measuring instrument, the artificial or natural light).

Besides, knowledge is organized hierarchically. In other words, one element can be a subset of another. For example, a horizontal plane is a kind of plane that is a kind of geometry. So a plane can be linked to an object since an object can be linked to geometries.

10.2 Data domain

The data have many characteristics (density, noise, resolution, dimension, size, occlusion), which impacts the process of understanding the data content. These characteristics must be considered to guide the process of understanding. The acquisition process generates data and significantly influences the data characteristics. This influence is mainly related to the instruments used and the methods used for the acquisition. Furthermore, the acquisition process itself is influenced by external factors and by the Scene that is represented in the data. Figure 10.2 shows the generic semantic description of any data and any acquisition process.

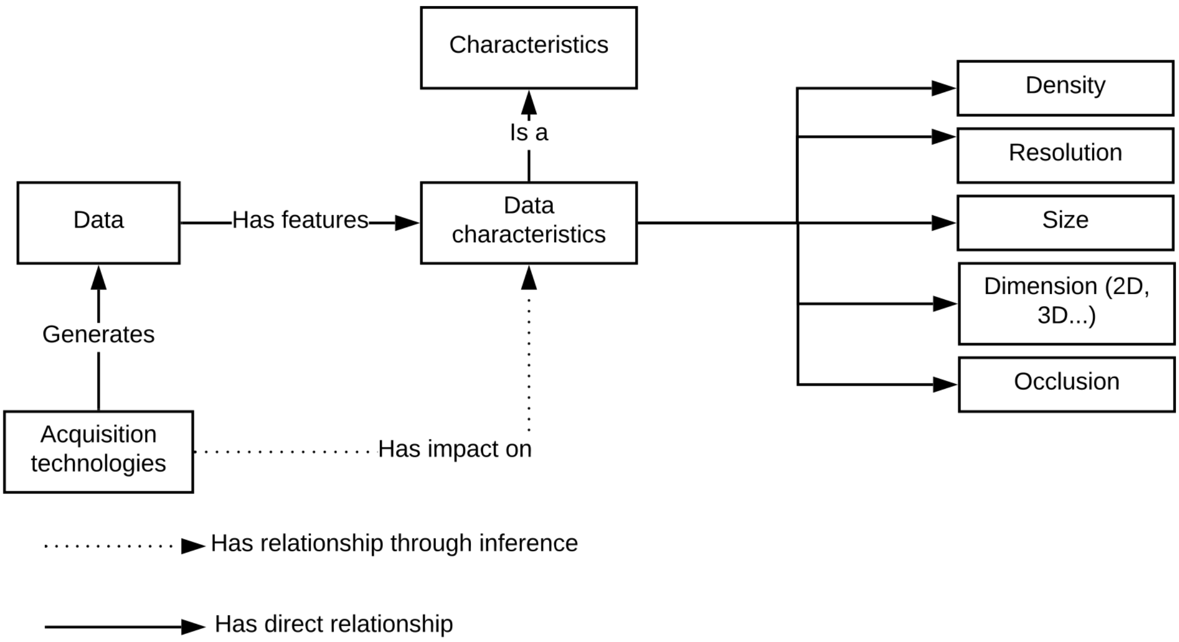


Figure 10.2: Semantic description of the data domain.

The semantic description of the data and the acquisition process, allows information to be correlated with each other. Such a correlation aims to infer data characteristics (e.g. each part of an object which was separated due to occlusion must be associated with the same object). Let us take as an example an external scene acquired by laser scanner Lidar, illustrated by Figure 10.3.

In this example, the data is acquired by scanning horizontally (shown by the brown arrows in Figure 10.3) along a linear path (shown by the red arrow in Figure 10.3). This acquisition methodology, combined with the instrument used (Lidar laser scanner) systematically generates a lack of local information caused by the occlusion of parts of the Scene by objects. This lack of information often leads to an over-segmentation of objects (the same object cut into several parts) or to a failure of the object's detection.

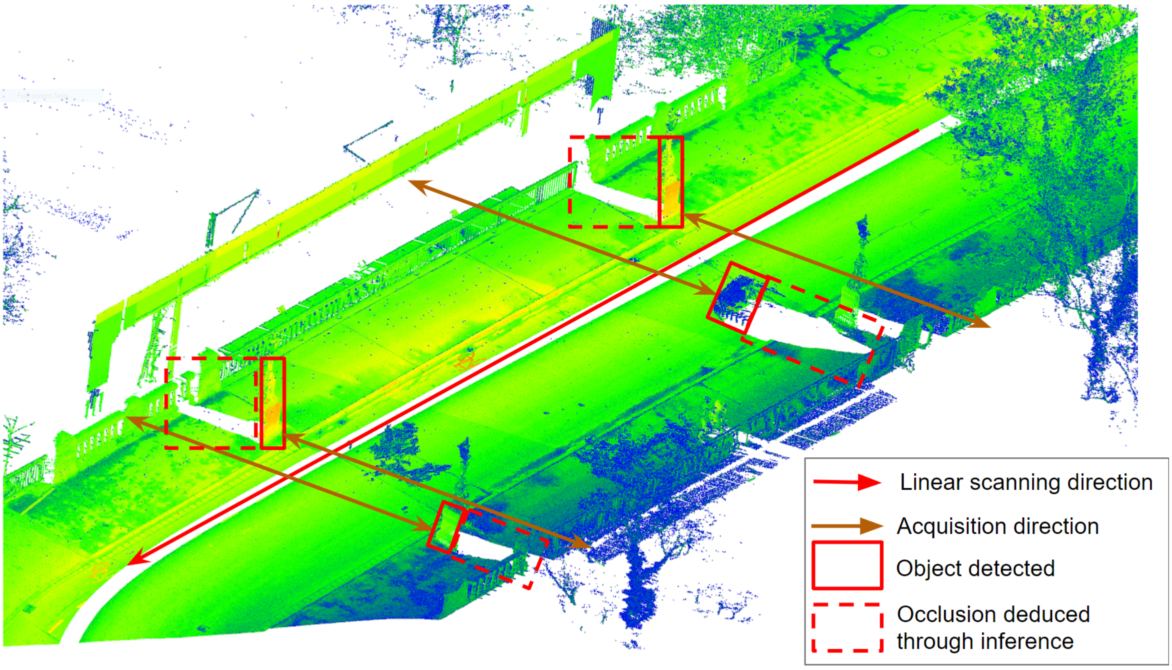


Figure 10.3: Illustration of occlusions inference.

However, the lack of local information can be anticipated and compensated by reasoning. In this example, the detection of an object (shown by the red rectangle in Figure 10.3) in the data can be correlated with the position of the measuring instrument to infer the location of the area possibly occluded by the object (shown by the red dotted rectangle in Figure 10.3). Thus, reasoning on the acquisition method, the measuring instrument used, and the information of the Scene represented in the data, allow the anticipation and localization of the occluded areas. The location of these areas allows, for example, the reconstruction of an object segmented into several sections. Moreover, it constitutes essential information to guide the choice and adaptation of algorithms.

10.3 Scene domain

The semantic description of the objects present in the digitized Scene is essential to understand the data content effectively. The object detection strategy must be adapted to the different characteristics of the objects. Indeed, the simpler an object is, the simpler the strategy used to detect it can be. On the contrary, objects with complex characteristics require more elaborate detection strategies.

The semantic description of objects is separated into three main parts to facilitate the adaptation of detection strategies. Figure 10.4 shows the generic semantic description of an object.

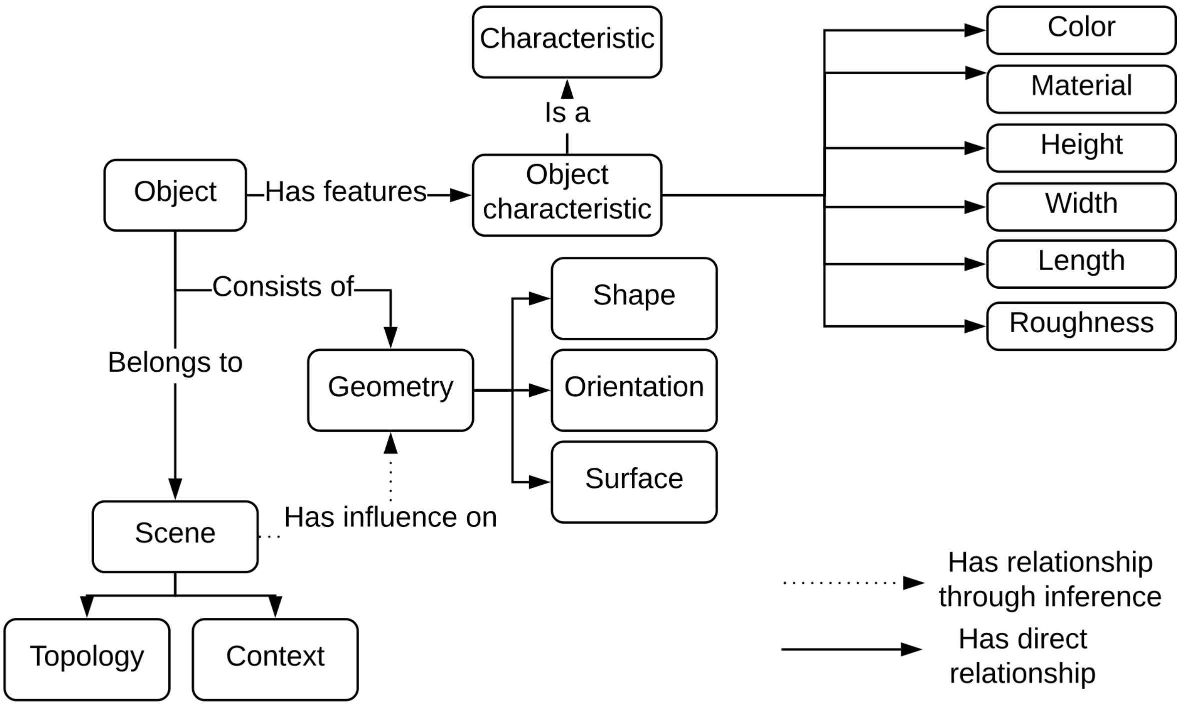


Figure 10.4: Semantic description of any object.

10.3.1 Object characteristics

The first part of the object description is composed of the characteristics specific to the objects that influence the acquisition process (e.g. color, material). Let us take as an example a highly reflective metal object such as a traffic sign. The acquisition of such an object by laser scanner technologies generates insufficient density data, as shown in Figure 10.5. On the contrary, a matt object such as an asphalt sidewalk acquired by this technology generates data of proper density.

The formal rules of inference shows in Equation 10.1 automatically infers the variation in density of a data (*?d*) by comparing the characteristics (*?c*) of the object (*?o*) with the acquisition technologies (*?t*).

$$\begin{aligned} &Data(?d) \wedge Object(?o) \wedge LaserScanner(?t) \wedge MetalMaterial(?c) \\ &\wedge isContainedIn(?o, ?d) \wedge hasCharacteristics(?o, ?c) \\ &\wedge generates(?t, ?d)) \Rightarrow hasLowDensityFor(?d, ?o) \end{aligned} \tag{10.1}$$

This deduction allows adaptation of the algorithms used to detect objects. Similarly, the color of an object can be related to external factors impacting the acquisition process, such as ambient light, to automatically infer the color acquired in the data. For example, a blue pen acquired under yellow light generates black data. The deduction of such characteristics is essential for the detection of objects.

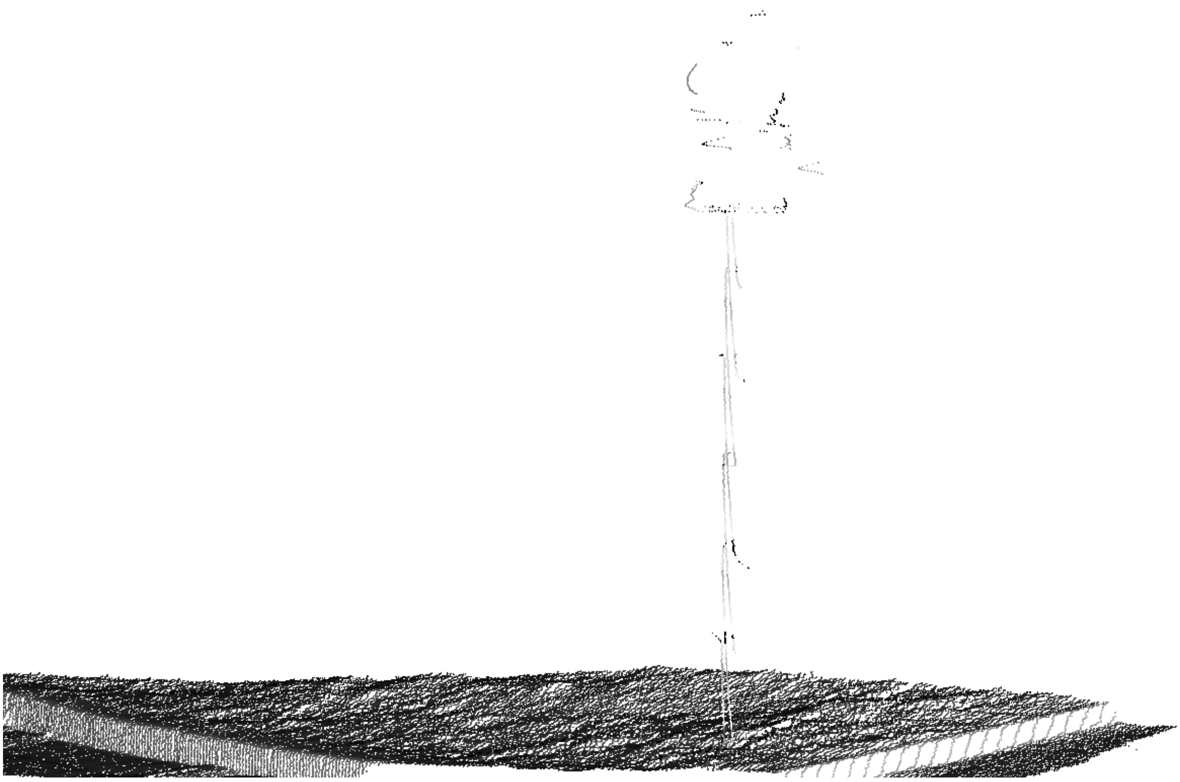


Figure 10.5: Illustration of the influence of the object material on the acquisition (asphalt sidewalk in the lower part of the image and a metal traffic sign in the upper part of the image).

10.3.2 Geometry

The second part of the object description is the description of the geometries representing the objects in the data. The shape defines the geometry of an object (e. g. rectangular, triangular, cubic, cylindrical, spherical, free), an orientation (e. g. vertical, horizontal, oblique) and a surface (e. g. regular, irregular, planar, linear). Besides, objects can be represented as a compound of other objects.

Let us take as an example of the semantic description of a room. A room is composed (mainly) of at least three walls, a floor, and a ceiling. The geometry of a wall is commonly defined as a planar surface, vertical and of rectangular parallelepipedic shape. The floor and ceiling are commonly defined as a planar surface, horizontal, with a rectangular parallelepipedic shape. This information helps to guide the choice of algorithms and to detect geometry and allow the identification of objects.

However, the geometrical object's description may not be sufficient to differentiate objects (the ceiling and floor have the same geometrical definition) or to detect objects in complex contexts (ruin excavation). That is why descriptions of the Scene are necessary.

10.3.3 Scene

The Scene is the third part of the description of the objects. The Scene describes the topological links of the objects (in contact, parallel, perpendicular, above, below, on, inside, next to), and the context in which objects are found (archaeological excavation, modern building, street, outside, inside).

Topological link

Topological links allow objects to be more easily detected in the data or to deduce the position of an object by detecting another. Let us take as an example of the street light detection illustrated by Figure 10.6. Street lights are described as being spaced about 30 meters apart in a city. If the acquisition context is a city having this property, then this information can be used to infer the position of street lights automatically. In Figure 10.6, the green street light is detected by this geometric description, while the position of the red street light is deduced by reasoning on the position of the green street light, and the distance of 30 meters separating them.

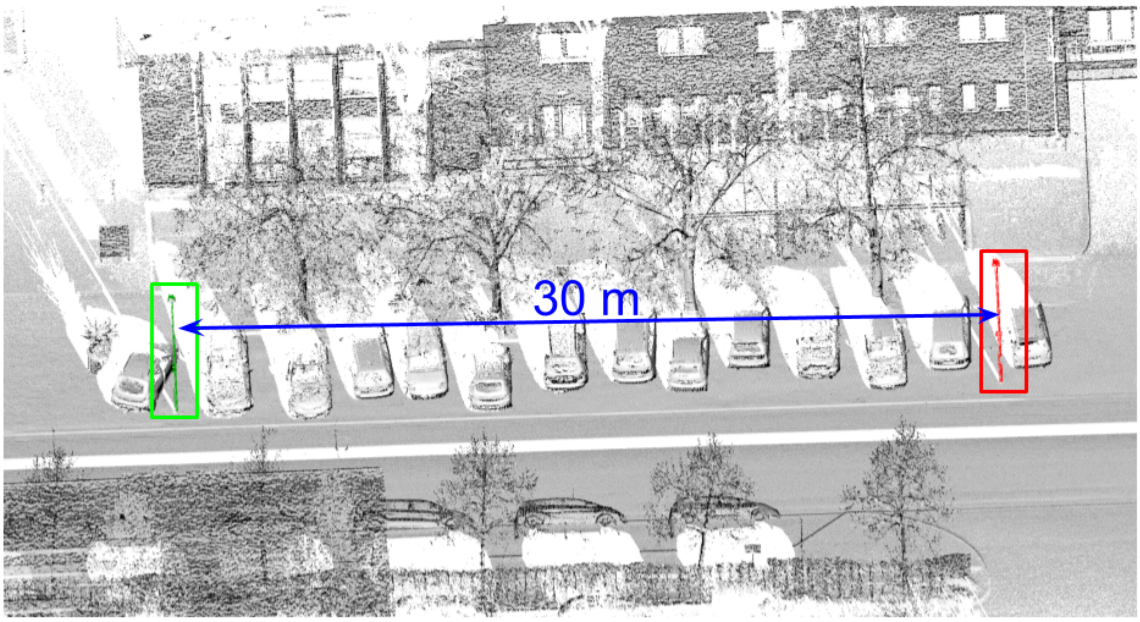


Figure 10.6: Illustration of the inference of the position of a street light (in red) thanks to the detection of another street light (in green) and its topological link (in blue).

The description of the topological links of the objects allows facilitating the detection of the objects. It also allows objects to be located when the acquisition conditions are not optimal, and thus, when the data generated are not sufficiently characterized to detect the object (e. g. lack of density, occlusion, shape distortion).

Scene context

The context of the scene dramatically influences the geometric characteristics of the objects. Therefore the semantic description of the context of the Scene is essential.

Let us take as an example, the following geometric description of a wall: A wall has a rectangular shape and a height greater than two meters (the description is simplified for the example). The context in which the wall is located will strongly influence these characteristics. Let us take two walls illustrated by Figure 10.7, from two different scene contexts.

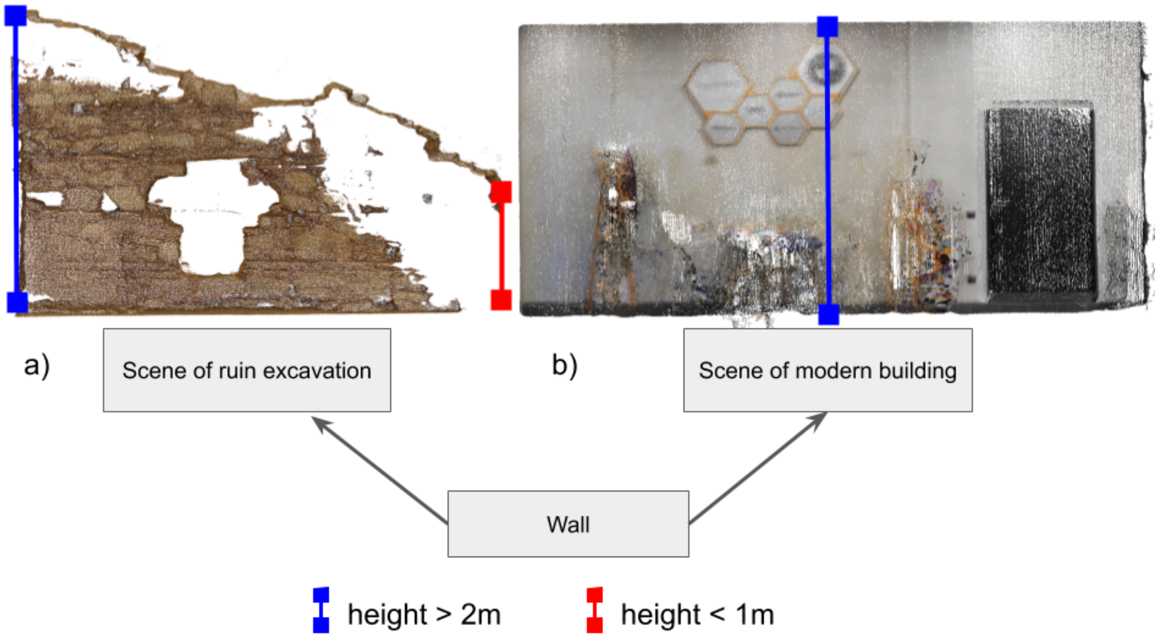


Figure 10.7: Illustration of two walls from two different scene contexts: a) wall in ruin excavation context, b) wall in modern building context.

The wall shown in Figure 10.7(a) is in the context of a ruined excavation and the wall shown in Figure 10.7(b) is in the context of a modern building. These walls theoretically have the same semantic definition. However, the walls located in the ruin excavation are here partially destroyed, and their geometries can be significantly altered by time and excavation methods used. Thus, the wall located in a modern building (Figure 10.7(b)) has a uniform shape and height, while the wall located in the ruin excavation (Figure 10.7(a)) has a variable height (possibly less than one meter) and an irregular shape.

The geometry obtained for the wall located in a ruin excavation (Figure 10.7(a)) differs significantly from the geometric description of a wall. This difference is too significant to allow the wall to be detected. The geometric description of the wall must, therefore, be adapted to the context. This adaptation is mainly carried out by the application of rules of inference that materialize the influence of the context on the geometry of objects.

Let us take as an example, the rule shown by Equation 10.2.

$$\begin{aligned} &Object(?o) \wedge Scene(?s) \wedge belongsTo(?o, ?s) \\ &\wedge DestroyedObject(?d) \wedge hasCharacteristics(?s, ?d) \\ &\Rightarrow IrregularShape(?i) \wedge hasShape(?o, ?i) \wedge hasHeigh(?o, > 0m) \end{aligned} \tag{10.2}$$

This rule formalizes that if an object (*?o*) belongs to a scene (*?s*), and the scene has objects destroyed (*?d*), then the shape of the object can be irregular (*?i*), and its height (*?h*) can be uncertain. Such rules allow the geometric and object-specific characteristics to be adapted to the context of the Scene.

10.4 Data processing domain

The purpose of algorithms is to allow the detection of objects in the data. They generate data or data characteristics to give clues about the geometries and objects that the data contains. They are therefore semantically described as generating data, detecting objects and adapted to geometries. Figure 10.8 shows the generic semantic description of any algorithm.

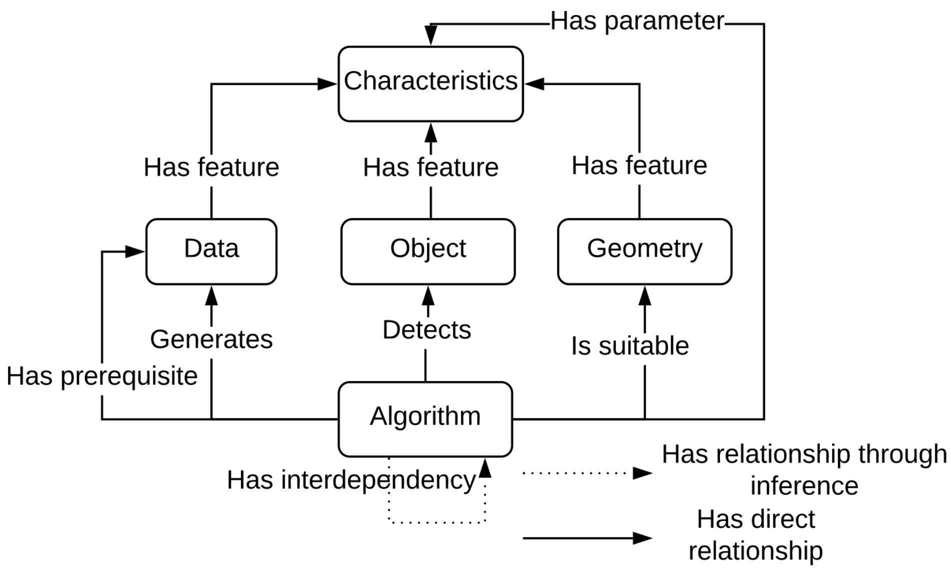


Figure 10.8: Semantic description of any algorithm.

The semantic description of each algorithm allows for selecting algorithms according to their relevance to detect objects. Their relevance is estimated through their "is suitable for" link with the geometries of the objects.

Some algorithms also have data prerequisites to be taken into account. Moreover, algorithms produce data characteristics. Thus, algorithms can produce data characteristics that satisfy the prerequisites of other algorithms. These algorithms are interdependent. These interdependence relationships are established by automatic inference (illustrated by the dotted arrow in Figure 10.8). Let us take as an example region growing algorithms. These algorithms require that the data is not noisy, so as not to cause over-segmentation, and that the data is small in size, as these algorithms are mostly exorbitantly in time execution. Beside, denoising algorithms can satisfy the prerequisite on data noise, meanwhile filtering or sampling algorithms can reduce the size of the data. Therefore, by inference, it is possible to deduce interdependences relationships between region growing algorithms and denoising, filtering, and sampling algorithms. These interdependencies allow algorithms to be efficiently combined.

Most algorithms require parameters that value influences its behavior. These parameters are generally primitive values (integer, double) and are very often used to perform basic task such as thresholding. Experts traditionally choose these values based on the characteristics of data and objects, and optimized, through the use of Machine Learning, for specific cases. However, the choice of parameters set by experts i is not adapted to the object or data characteristics variations. Moreover, the more variation there is in the application case, the more they need for training data increases for the use of Machine Learning. Also, the use of Machine learning does not allow adapting parameters to situations for which they have not been trained. However, the variations are multiple and can occur during the acquisition process (change of measurement instrument or methodology), or produced by external factors, or by the arrangement of objects in the Scene, or by changes in the geometry or material of the objects. Small variations can thus jeopardize the detection of objects by algorithms.

Therefore, we propose to adapt the algorithm parameters according to the characteristics of the objects and their geometry, as well as to the characteristics of the data, including the different factors impacting the data (acquisition process, external factors, Scene). This adaptation is made by using equations that relate different characteristics to calculate an element. For example, the calculation of a parameter used to threshold a minimum distance between two neighbors can be defined according to the object size and the density of the point cloud. Appendix A illustrates the modelling of various algorithms.

10.5 Discussion

The structuring and modeling of knowledge allow the analysis and understanding of data content to be adapted according to multiple factors. These factors are mainly the acquisition technology used (e.g. laser scanner, photogrammetry, Kinect), external factors impacting the acquisition process (e.g. intensity of ambient light, color of light, vibration of the measuring instrument), the structure of the Scene (e.g. occlusion of objects), the type of context (e.g. ruins excavation, modern building), as well as the geometry (e. g, surface, shape) and the topological links of the objects contained (e.g. traffic sign, car, water mill, street light). Adapting data understanding requires to consider the variations of all these factors to provide appropriate and robust solutions. The adaptation of the content understanding is achieved by adapting the algorithms used. The following chapter explains how these algorithms are automatically adapted.

11 Knowledge-driven object detection

Understanding the data content requires analyzing the data. The purpose of data analysis is to identify objects and geometry. It requires the use of algorithms to obtain clues about the data content. Then reasoning processes use these clues to identify the objects and geometries that constitute the data. The reasoning is based on knowledge. The previous chapter (Chapter 10), has presented the modeling of the primary knowledge needed to guide the detection of objects and geometry in the data. The application of reasoning on this knowledge enables not only to manage the algorithms to use but also to identify the elements that constitute the data. This chapter shows how this knowledge is used to detect objects and geometries in data. Section 11.1 presents the management of algorithms by the different reasoning processes. Section 11.2 presents how the application of rules and reasoning enable to identify objects and geometries in data. Finally, Section 11.3 presents how knowledge is automatically adapted to the application case to compensate for unexpected variations and give more flexibility to the detection approach.

Appendix B provides an overview of the main mechanisms essential for detection processes.

11.1 Algorithms management

Algorithm management requires a high degree of flexibility to effectively adapt each algorithm to the application case (e.g. data, object, acquisition context) as well as to the results obtained by the other algorithms. Algorithms must also be "intelligently" selected and configured. To provide such "intelligence", we propose to fully manage the algorithms by applying reasoning on the global modeled knowledge to understand the content of the data. The purpose of this reasoning is first to automatically select the "best" algorithms among the set of algorithms available, for a given application case, secondly to configure these algorithms according to the situation and finally to execute them and integrate their results.

11.1.1 Algorithms selection

The selection of algorithms is carried out by reasoning on the entire knowledge modeled for an application case. The reasoning identifies the most appropriate algorithms to detect objects and geometry in the data.

This identification is carried out in three steps. First, the reasoning process evaluates the relevance of each algorithm to detect the objects and geometries defined. Second, it assesses the relevance of each algorithm specifically to the data containing the objects and geometries being searched. Finally, it assesses the relevance of each algorithm to algorithms that have already been found to be relevant.

Each algorithm is considered relevant firstly to the data, which it produces or on which it is applicable, and secondly to the objects, which it detects and the geometries for which it is suitable.

Let us take as an example the case of the detection of a green object in a colored point cloud. Region growing algorithms with color as an aggregation criterion are adapted to segment data according to color. They are therefore considered relevant for detecting the object being sought. However, among the different algorithms for region growing, only algorithms designed for 3D data will be considered relevant for processing a colored point cloud. Thus a limited set of algorithms is relevant both to detect such object and to work on such data.

Among this set of algorithms, the algorithms constitute a list of "candidates", if all of their prerequisites are satisfied. It is necessary to have a set of pre-processing algorithms to satisfy the prerequisites of algorithms. Thus, each algorithm that satisfies one of the preconditions of an algorithm is part of this set of pre-processing algorithms.

Let us take again the previous example where some region growing algorithms were selected as "candidates". These algorithms require that the data are not noised and small. Denoising and sampling algorithms can satisfy the requirements of the region growing algorithms. Thus, these algorithms are considered relevant for region growing algorithms and integrated into the set of pre-processing algorithms.

Since the reasoning mechanism is continuous (repeated until nothing changes), the relevance of the algorithms is evaluated recursively.

Finally, the last selection is carried out on all the "candidates". This selection consists of retaining among the "candidates" only those whose all prerequisites are satisfied. Thus all algorithms relevant and adapted to the application case are selected to be configured and used.

This selection is concretized in the knowledge by linking the algorithms to the objects, geometry, and data defined for the application case, by the property "is relevant for". This selection allows the configuration and execution of only those algorithms that may be relevant to the application case. Moreover, it allows linking the application case data and objects with the algorithms. These links are essential to configure the algorithms.

11.1.2 Algorithms configuration

Each algorithm that is selected for execution is automatically configured according to the application case. This configuration is carried out directly in the knowledge base by adding properties such as the parametric values necessary to run the algorithm. It is necessary to understand the needs of the algorithm in order to configure each algorithm. Moreover, the configuration of the algorithms determines their execution. It is therefore essential to configure the algorithms according to the results already obtained, in order to optimize their use. Considering the results already obtained requires a link between the inputs and outputs of algorithms. Therefore, a property defining by the algorithm links its inputs and outputs (e.g. property "comes from", "is a segment of"). The configuration of each algorithm is done first by setting up its inputs, then by setting its parameters.

Algorithms inputs

In the knowledge modeling presented in the previous chapter (Chapter 10), the "works on" property links the inputs of each algorithm to the data. Besides, each algorithm can have prerequisites on the data (expressed by the "has prerequisite" property). Finally, the "produces" property links the algorithm to its output, which is data or characteristics.

A rule of inference structuring these properties configures the algorithm inputs. Figure 11.1 illustrates this rule.

This rule stipulates that a data is considered as an input of an algorithm (represented by the "has input" link in Figure 11.1) if this data is of the same type (represented by the "rdf:type" link in Figure 11.1) than the one on which the algorithm can work (represented by the "works on" link in Figure 11.1) and that it is not already linked by the algorithm to an data or a characteristic (represented by

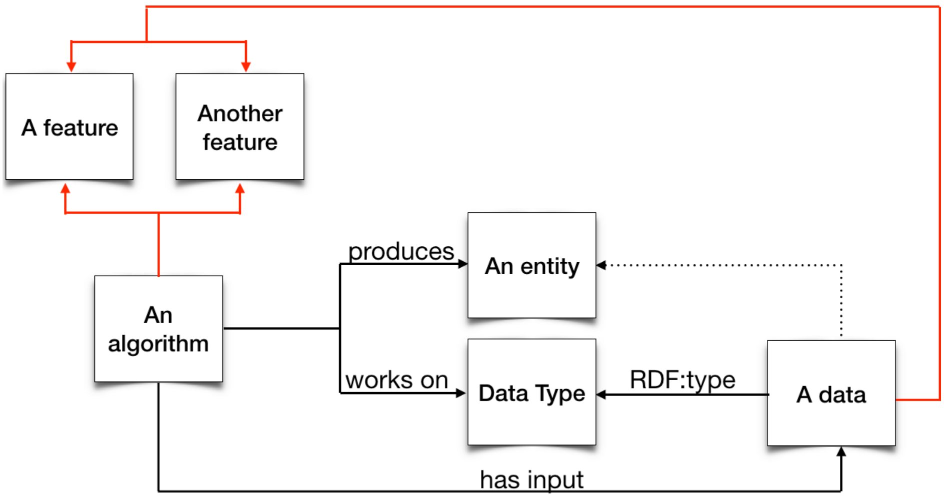


Figure 11.1: Illustration of the process that sets up algorithm inputs.

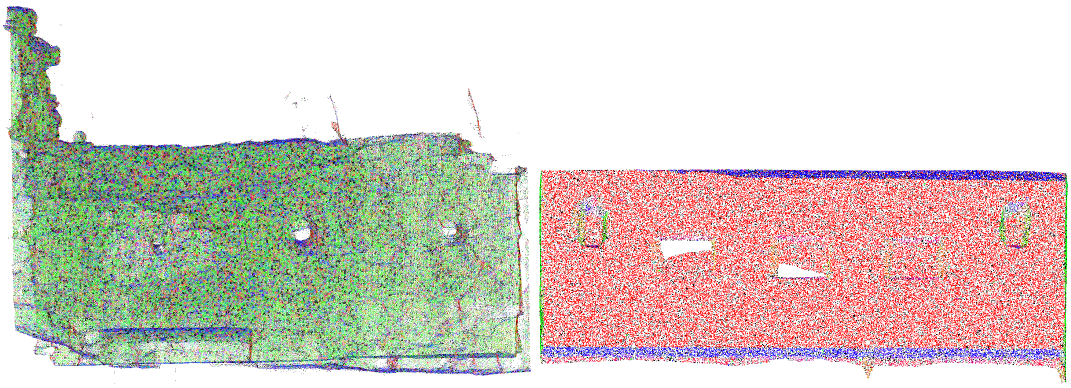
the dotted link in Figure 11.1). This ensures that the same algorithm is not configured several times. Besides, this data must satisfy all the characteristics defined as prerequisites for the algorithm (represented by the red links in Figure 11.1).

Let us take as an example a segmentation algorithm defined as working on 3D point clouds, producing homogeneous colored segments, and having as a prerequisite that the data have a color. Thus this algorithm produces segments from a point cloud. This information allows for considering that the segments "come from" the point cloud. The knowledge on the segment origin allows knowing if a point cloud has already been processed or not by the algorithm considered. In this example, if some homogeneous colored segments already come from the considered point cloud, then the algorithm will not take this point cloud as an input. Thus, the algorithm takes data as input if and only if the data is a point cloud with colors (i.e. satisfy the two algorithm prerequisites) and it does not exits homogeneous colored segment that comes from this data.

Algorithms parameters

The parameters of algorithms determine their behavior. In general, the value of each parameter of an algorithm is assigned according to the application case (data characteristics) and the objective that the algorithm must achieve (detecting an object, characteristic or geometry).

For example, a parameter used as a threshold to determine the maximum distance between two entities so that they can be considered as neighboring is determined according to the density of the data. Similarly, an algorithm for detection of planes is influenced by the occlusion or non-occlusion in the data that will determine the type of plane sought (e.g. tiny plane, full plane). Finally, let us take as an example the case of a stone wall which is a rather rough object, and the case of a modern wall which is a smooth object, to illustrate the influence of the choice of parameters on the execution of the task that the algorithm must accomplish. The estimation of the normals of these two objects, therefore, differs significantly. Figure 11.2 shows the difference between the estimation of the normals on a stone wall (a) and a smooth wall (b), performed by the same algorithm without parameter variations.



(a) Stone wall. (b) Modern wall.

Figure 11.2: Illustration of the normal estimation. The value of the normal is converted in RGB format for the visualisation.

The stone wall has highly heterogeneous normals (displayed in green) while the smooth wall is composed of homogeneous normals (displayed in red). The segmentation of these objects based on the estimation of normals requires a so-called tolerance parameter to determine if the difference between two neighboring normals is small enough to belong to the same segment. In this case, the tolerance parameter must be adapted to the roughness. Thus its value must be higher in the case of the stone wall than in the case of the smooth wall. Otherwise, it would cause an over-segmentation in the case of the stone wall, which would be detrimental to its detection.

The selection of algorithms explained in the previous section (Section 11.1.1) allows establishing links between the algorithms and the data and the objects they contain. These links allow the retrieval of the necessary information to configure each algorithm. This information can be used to set the value of a parameter (such as the density for a thresholding parameter) or to set a set of linked parameters. Therefore, the "depends on" property links each parameter of an algorithm to "types" of characteristics on which it depends. Similarly, when necessary, the "has equation" property links a parameter to an equation that formalizes the dependencies of the parameter. This equation aims at combining different information to calculate the value of a parameter in a way that is adapted to the application case. The equation variables correspond to the type of characteristics on which the parameters are dependent.

For example, the "RANSAC" algorithm whose modeling is described in Appendix A.3, requires a number of iterations as a parameter to detect the best plane in the point cloud. This number of iterations depends on the size characteristic (number of points) of the data. Thus the number of iterations of this algorithm is defined as depending on the size of the data.

Thus the dependency relationship between the "Iteration Number" parameter of the algorithm and the "Data Size" type characteristics is defined as a constraint of the algorithm (see Equation 11.1). This constraint means that the algorithm parameter must have the "depends on" link with the "Data Size" type characteristics owned by a data. The dependency relationship between the parameter of an algorithm and a characteristic is satisfied by a rule of inference.

$$IterationNumber = 10\% \times DataSize \tag{11.1}$$

This rule stipulates that if a parameter of an algorithm depends on a type of characteristic of an object, data or geometry for which this algorithm is "is suitable", "detects" or "is relevant for" respectively, then the parameter dependency is assigned to the characteristic that the object, data or geometry has.

When all the dependencies of a parameter are satisfied, the algorithm parameterization process uses the equation assigned to the parameter to calculate the parameter value. This assignment is carried out through the "has value" property that links the value to the parameter.

11.1.3 Algorithms execution

Once the algorithms have been selected and configured, they can be executed. Information about the algorithm are stored in the semantic knowledge base. However, the algorithms are executed using an algorithmic paradigm that differs from the semantic paradigm used to model knowledge. Therefore, it is necessary to create a conceptual and technical bridge between the algorithmic paradigm and the semantic paradigm.

This conceptual bridge requires retrieving and transcribing the information contained in the knowledge base (in the semantic paradigm) into a value that can be used by algorithms (e.g. string, integer, double, float, point). It then requires to dynamically instantiate each algorithm to be executed and set them with these values and then execute them. Finally, the results produced by the algorithms must be interpreted in semantic form and integrated into the knowledge base to enrich it with new information.

Among the different technologies resulting from the Semantic Web, SPARQL stands out as a standard by the W3C to allow the selection and dynamic modification of knowledge through queries (see Section 4.3.3). Therefore, we propose a framework based on SPARQL to provide a bridge between the two paradigms. This framework allows algorithms to be automatically executed according to the information on their parameterization, contained in the knowledge base. It then allows the enrichment of the knowledge base by the integration of the results obtained by the execution of algorithms. Figure 11.3 shows the structure of this framework.

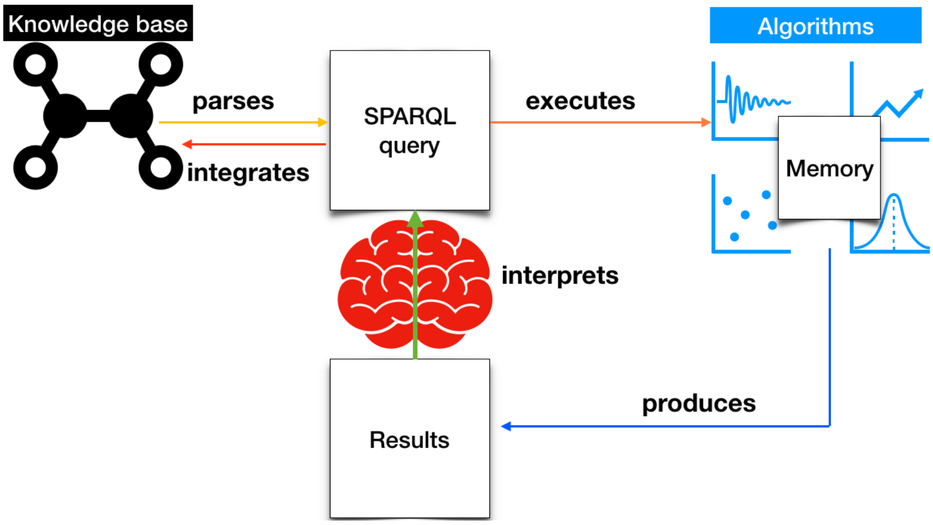


Figure 11.3: Overview of the algorithms execution framework.

This framework translates into a SPARQL query the information on the algorithms contained in the knowledge base (yellow arrow in Figure 11.3). The algorithms are then instantiated and executed using the SPARQL query (orange arrow in Figure 11.3). The execution of the algorithms (in blue in Figure 11.3) is performed through shared memory. This memory allows the sharing of information between the algorithms. Finally, the results of the algorithms are interpreted (green arrow in Figure 11.3) in the semantic paradigm and are integrated into the knowledge base (red arrow in Figure 11.3) through the SPARQL query.

Design of SPARQL query

We have extended the functionalities of SPARQL to the Computer Vision domain by creating SPARQL built-in from the most commonly used algorithms in this domain.

These built-ins have the objective of triggering the execution of each algorithm and integrating its result into the knowledge base.

The execution of the algorithms is performed using the call of a SPARQL built-in specific to each algorithm. It is necessary to collect all information firstly from each algorithm. The information required to create the SPARQL built-ins is the name of the algorithm class and a set of pairs consisting of a field name and a field value for each algorithm parameter.

The name of the algorithm class is assigned to the name of the SPARQL built-in to call the appropriate algorithm. The set of pairs is used to reconstruct the algorithm parameters. The name of the field is obtained by retrieving the type of the parameter. Similarly, the value of the field is obtained by retrieving the value of the parameter (linked by the "has value" property).

Let us take as an example the "RANSAC" algorithm applying to the point cloud "pc1" of type "Point cloud" and having a parameter of type "Tolerance" of a value of "0.2" and a parameter of type "Iteration Number" of a value of "10". The name of the first field for the SPARQL built-in is "pointcloud", the name of the second field is "tolerance" and the name of the last field is "iterationNumber". The value associated with the field name "tolerance" is "0.2", the value associated with the field name "iterationNumber" is "10" and the value associated with the field name "pointcloud" is "pc1". Finally, the name of the SPARQL built-in corresponds to the name of the type of the algorithm. In this example, the name of the built-in is "ransac".

The SPARQL built-ins are linked to a return variable to integrate the results provided by the execution of each algorithm. On the other hand, each result is firstly linked to the input of the algorithm. This link depends on the type of the result.

If the result is a value with a specific Datatype (e.g. "string", "integer", "double"), the result is linked to the input of the algorithm by property specific to the algorithm. For example, an algorithm to estimate the height of a segment generates a "double" value as a result and takes a segment as entered. Thus the input segment is defined to be linked to the double type value by the property corresponding to the algorithm, which in this case is "has a height". Thus the height of the segment is defined by the result of the algorithm.

If the result is a complex element like a "plane", the SPARQL query firstly, attributes it the adapted type thanks to the definition of the output constraint on the algorithm. For example, the "RANSAC "algorithm is defined as generating elements of "Plane" type. Thus the results of the algorithm can be automatically considered as plane.

Then, the query retrieves every restriction on this type, which is useful to ensure the consistency of the knowledge base. These restrictions define the characteristics that the results must have and allow for defining new information related to the result through the adding of new relationships. Moreover, these characteristics correspond to information coming from the inputs, parameter, or prerequisite of algorithms. For example, each plane is constrained to "come from" a point cloud and to have a "precision" of the same type as a tolerance parameter. Thus the planes generated by the RANSAC algorithms are linked by the "come from" property to the input data of the algorithm and these planes will have as "precision" characteristics the value of the "tolerance" parameter of the algorithm.

To summarize the conversion of "class construct" into SPARQL query, let us take the example of the RANSAC algorithm whose "class construct" is presented in Code 11.1 under the Manchester syntax (explained in Section 4.3.2).

```
RANSAC:
Algorithm
and (worksOn some (PointCloud) )
and (hasParameter exactly 1 Tolerance)
and (hasParameter exactly 1 IterationNumber)
and (generates some (Plane that comesfrom exactly 1 Pointcloud
           and hasFeature exactly 1 Precision)
```

Code 11.1: Example of RANSAC algorithm modeling

Thus in this simplified example, the "class construct" of the "RANSAC" algorithm is translated into the following *SPARQL* query illustrated by Code 11.2:

```
CONSTRUCT {
?planes rdf:type Plane.
?planes comesFrom ?input.
?planes hasPrecision ?p1
 } WHERE {
 ?input rdf:type PointCloud.
 ?p1  rdf:type Tolerance.
 ?p2  rdf:type IterationNumber.
 ?a rdf:type RANSAC.
```

```
 ?a hasParameter ?p1.
 ?a hasParameter ?p2.
 ?a worksOn ?input.
 ?plane ransac("pointcloud=?input" "tolerance=?p1" "iterationNumber=?pc2")
}
```

Code 11.2: Example of SPARQL query translated from RANSAC class construct

The execution of this query not only calls the execution of an algorithm and provides the information to define its result but also integrates the results of the algorithm and their definition (after they are translated into the semantic paradigm) into the knowledge base in a consistent way.

Execution though SPARQL

The different field names and their associated field values provided as parameters of the SPARQL built-in, allow for instantiating a programming instance of the corresponding algorithm class with the values of its variables.

For this purpose, the fundamental values (e.g. "xsd:string", "xsd:integer", "xsd:double") resulting from the semantic paradigm are translated into fundamental values in the programming paradigm (e.g. string, integer, double). When the parameters or input of an algorithm are compound elements (e.g. point cloud, image), it means that they are the result of another algorithm previously executed.

Even if the SPARQL functionalities can be extended through built-ins, these built-ins are executed independently without sharing memory. Thus algorithms cannot directly share their results. This implies that the results of the algorithms should be saved in files or databases and reloaded when an algorithm needs it. In the context of massive data processing, the lack of memory sharing drastically slows down the execution process. Therefore SPARQL built-ins are combined with an external execution engine that completely manages the execution of algorithms and shares the memory between them. The execution engine manages the memory between different algorithms by linking individuals inside the ontology with their corresponding pointers inside the memory. The execution engine executes the algorithm called in the SPARQL query and matches the URI of "owl:individual" with the address of the memory pointer.

This memory management provides us a flexible use of the algorithms alignment by linking data or the result of an algorithm, to other algorithms by their pointer inside the memory.

This allows the algorithms to be dynamically instantiated and configured. This dynamic instantiation is possible through the use of reflection APIs such as the "reflect" API in "Java". The algorithms can then be executed, and their results collected.

The results of an algorithm is a fundamental value (e.g. string, integer, double) or a compound element (e.g. point cloud, image). Whatever the result type, this result must be semantically interpreted to be returned to the SPARQL query that will add the semantic representation of the result into the knowledge base.

The results of algorithms that are fundamental value(s) aim at improving the knowledge about an object or data. This type of result is interpreted as a semantic node.

For example, an algorithm for detecting the height of an object produces a fundamental value in the form of a "double" that corresponds to the height of the data to which it has been applied.

The results of algorithms, which are compounds elements, aims at improving the knowledge base by adding new individuals representing the same information type. Therefore, a semantic node representing an "individual" is created for each result. The creation of an "individual" requires the creation of a URI specific to it. The URI created allows for accessing to its semantic representation into the knowledge base and for accessing its programming representation into the memory system. All semantic nodes representing the result of the algorithm execution is returned to the SPARQL query.

11.2 Classification

The structuring of the data is carried out by identifying objects or geometry. The execution of algorithms allows enriching the knowledge base by adding information about the data and their content. This information provides clues to identify objects and geometries.

For example, the execution of several segmentation and extraction algorithms provide information that a portion of the data (segment) is cylindrical, has a height of 180 cm, a width of 20 cm, and is vertical. Figure 11.4 shows an example of such a segment in the 3D point cloud considered.

This segment represents the trunk of a tree. The process of segment identification is based on the semantic descriptions of objects. It checks whether the characteristics of a segment corresponding to the geometric or topological description of an object. Semantics, in this case, is based on geometry and topology, it could also be based

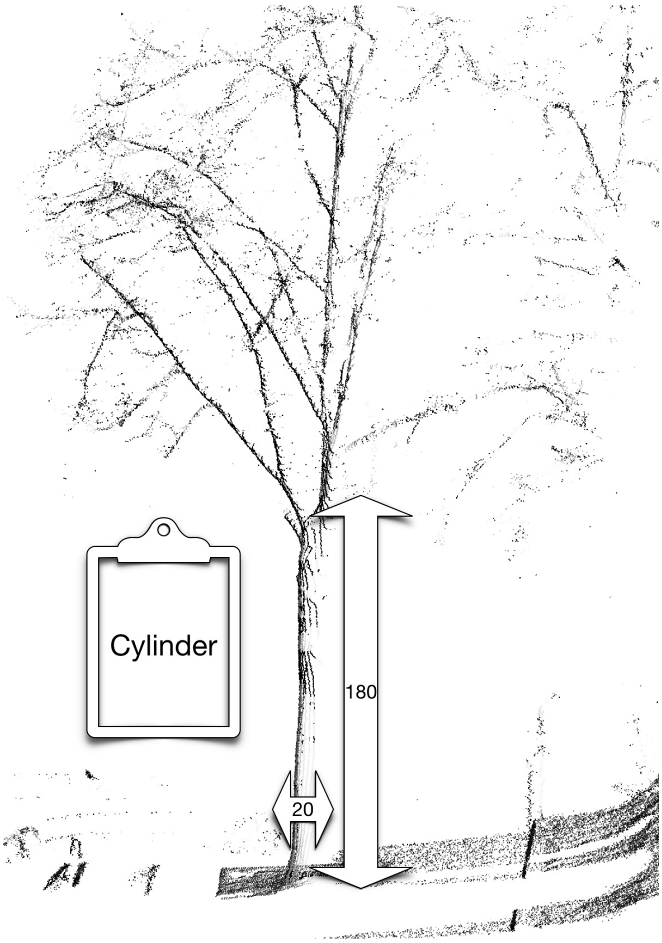


Figure 11.4: Example of a point cloud segment representing a trunk.

on other characteristics (e.g. color).

Our objective is not to be exhaustive as to the possible characteristics but to explain how the approach works through two of the most commonly used characteristics in Computer Vision.

11.2.1 Geometric classification

The classification, according to the object's geometry, is based on objects described in the knowledge base. The description of each object is automatically interpreted as a rule of inference to allow the classification of segments into objects.

In the example illustrated in Figure 11.4. Let us consider a class called "Trunk" defined as being composed of a vertical cylinder, with a height greater than 170 cm and a width greater than 15 cm. Code 11.3 shows its description in the Manchester syntax.

```
Trunk:
Object
and (isComposedOf exactly 1
       (Cylinder and (hasOrientation only Vertical))
and (hasWidth exactly 1 xsd:double[> "15"^^xsd:double])
and (hasHeight exactly 1 xsd:double[> "170"^^xsd:double])
```

Code 11.3: Example of Trunk modeling

Equation 11.2 shows the rule of inference for the description of this object.

$$\begin{gathered} Segment(?s) \wedge isComposedOf(?s,?c) \wedge Cylinder(?c) \wedge hasOrientation(?c,?o) \wedge \\ Vertical(?o) \wedge hasHeight(?c,?h) \wedge GreaterThan(?h,170) \wedge hasWidth(?c,?w) \wedge \\ GreaterThan(?w,15) \Rightarrow Trunk(?c) \end{gathered} \tag{11.2}$$

Thus the segment illustrated by Figure 11.4 checks if each condition of the rule of inference is satisfied and is therefore classified as a "Trunk".

11.2.2 Topological classification

The geometric description of an object is not always sufficient to identify objects, although it can be adapted to different contexts (as explained in Section 10.3.3).

For example, the geometry of the segment shown in red in Figure 11.5 differs radically from the geometry of a car.

This difference may be due to multiple unpredictable factors, such as acquisition errors or due to material reflectance.

Moreover, some objects are defined as the composition of other objects. For example, a tree is defined as composed of a trunk and leaves. The study of the topological relationships between objects is, therefore sometimes necessary to reconstruct these objects.

Classification based on the topological descriptions of objects consists in classifying a segment according to the topological links it has. As with classification based on geometry, the topological description of each object is interpreted into a rule of inference. Thus, if a segment satisfies all the topological links described for a given object, it is classified as belonging to this object class. Moreover, topological links can be described directly in the class of the object (e.g. composition, parallelism) or inferred by reasoning on knowledge.

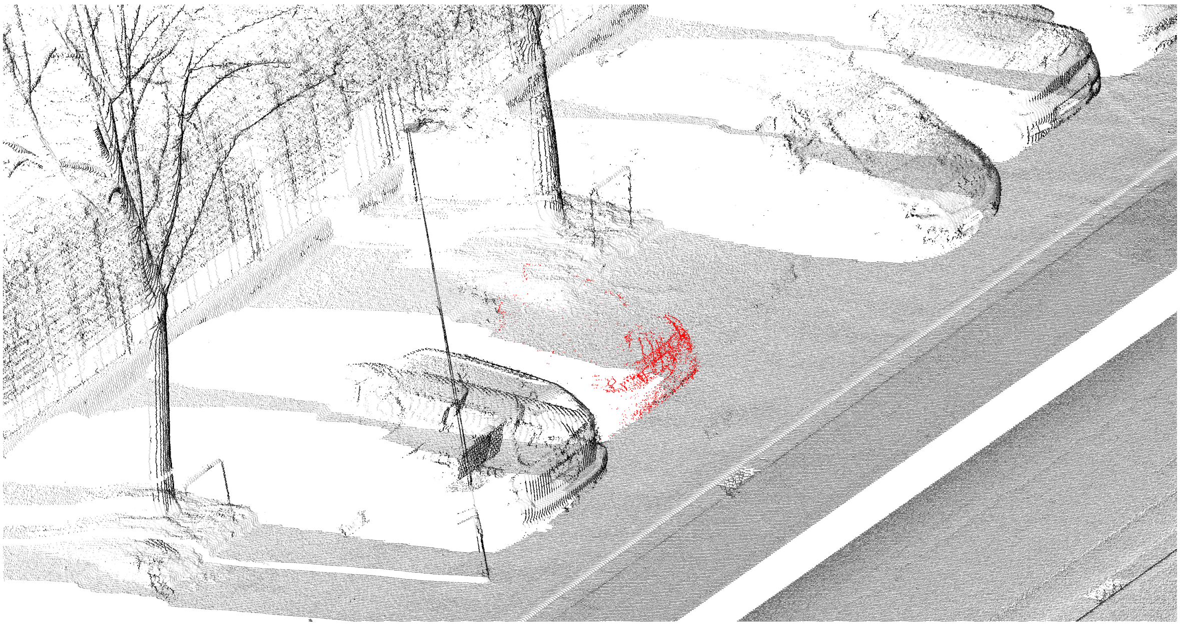


Figure 11.5: Example of a point cloud segment (in red) representing a car.

Let us take the example of detecting a tree. If the technology and acquisition process lead to occlusion, the presence of localized occlusion between the leaves and the trunk of the tree can be inferred (as explained in Section 10.2). These occlusions can cause the absence of connection information between the trunk and the leaves of the tree. However, the detected occlusion is considered as a topological connection link between the different segments located around the occlusion. This topological link allows considering the leaves and trunk as connected. Thus the tree can be identified by the composition of the trunk and leaves.

Similarly, in Figure 11.6, the wall is divided into several segments, which are all identified as independent walls.

The occlusions of trees causes this division. However, through the prediction of the occlusion presence (in red in Figure 11.6), walls are defined as having a topological "occlusion" relationship (in green in Figure 11.6). Let us consider that the description of a wall specifies that if two walls have the same geometry and are connected, then they form a single wall. This description, therefore, allows unifying the walls into a single wall.

However, some segments are not detected because their geometry differs from the description of the object on the one hand, and the description of topological links is not sufficient on the other. This is particularly the case for detecting the car shown in red in Figure 11.5.

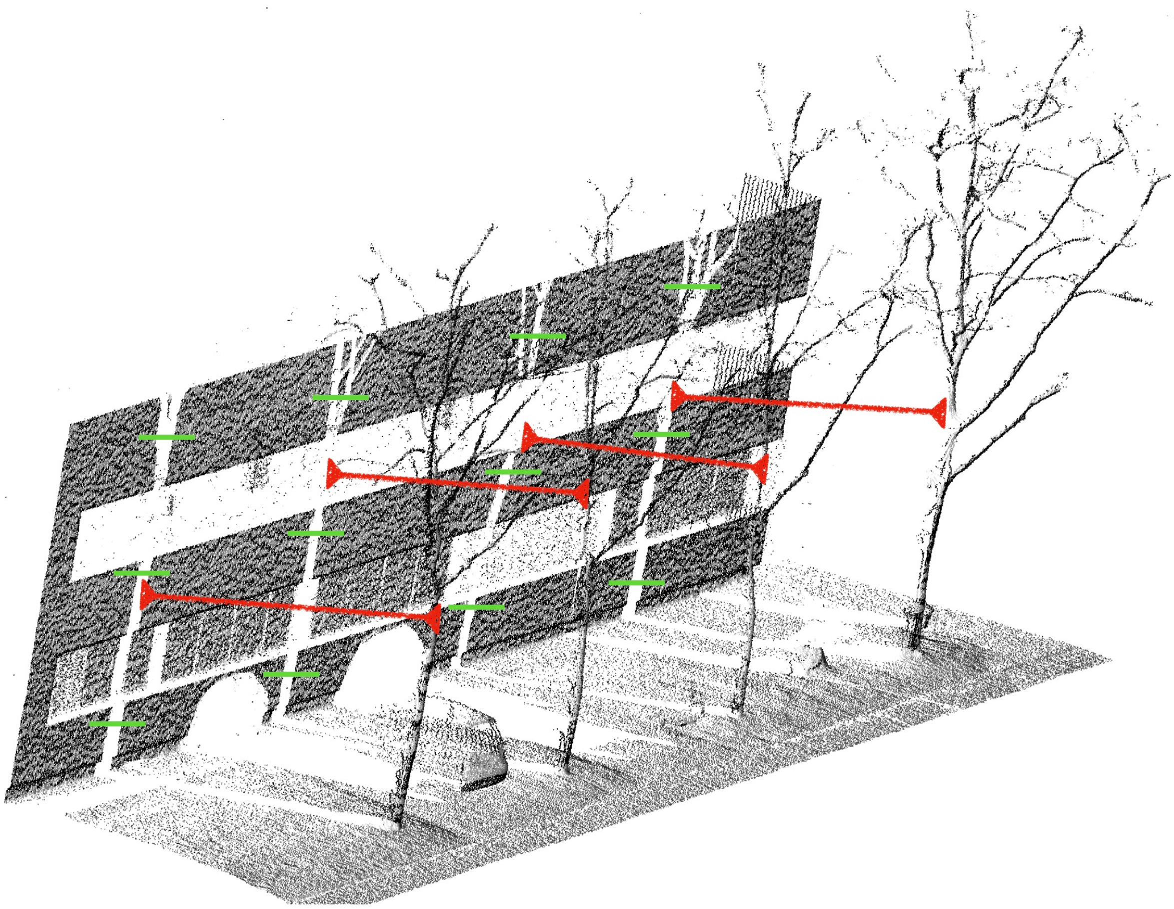


Figure 11.6: Recomposition of a wall using occlusions deduction (in red) and inferred topological link (in green).

11.3 Knowledge-based self-learning process

The objective of Knowledge-based self-learning process is to improve the understanding of data content by generating new knowledge or by compensating for any deviation between the knowledge and the information obtained. This learning is based on the experience gained during the detection process. Its generates or adapts global knowledge by studying the results obtained on each application case. This new knowledge is therefore specific to each application case, and thus, it is more precise than the general knowledge previously used. The addition of new and more precise knowledge on case studies makes it possible, on the one hand, to adapt the selection and configuration of algorithms better and, on the other hand, to make the classification more efficient and robust. In particular, it solves problems such as the one illustrated in Figure 11.5 where global knowledge of the objects is not sufficient to detect them. Indeed, the more precise the knowledge is, the more effective the object detection is.

It is necessary to understand how data is structured in order to infer new knowledge such as new relationships between objects or to modify knowledge such as object geometries. The information extracted by the algorithms and the results obtained after the classification step provides clues to understand and make assumptions about the structure of the data. The analysis of cues obtained must contain a form of "intelligence" that allows to make hypotheses about the structure of the data and draw conclusions that generate new knowledge.

That is why we propose a knowledge-based self-learning process system consisting of three steps. First, it enriches the knowledge base with new information on the topological relationships of objects and the geometry already inferred. Secondly, it analyses the information obtained so far to formulate new hypotheses on the structuring of the data. Third, it checks the consistency of the hypothesis from study-cases, and it integrates the validated hypotheses in the form of knowledge or reformulates the invalidated hypotheses.

11.3.1 Knowledge enrichment

The objective of enriching knowledge is to generate more clues on the topological links between the different objects detected and on their geometry. This enrichment depends on the diversity of algorithms available. Thus, the addition of algorithms improves the enrichment of knowledge when the application case requires it. Enrichment is carried out by queries requesting the execution of algorithms to add characteristics, on segments that do not yet have these characteristics.

Let us take as an example the scene illustrated by Figure 11.7

Cars have been identified (in yellow in Figure 11.7) but some segments also representing cars (in blue in Figure 11.7), have not been identified. The knowledge used to detect cars has no topological link. Thus no topological link has been studied to detect cars. The enrichment of knowledge through the execution of algorithms allows the establishment of topological links between objects by adding information. Among this information, the minimum distance between each car is added as well as the alignment link between the cars. Thus, these new links provide information relevant to the analysis.

11.3.2 Hypothesis formulation

The automatic formulation of hypotheses is based on the analysis of the characteristics common to the segments identified as objects of the same type (e.g. segments identified as cars). More precisely, hypothesis formulation consists of grouping the

Figure 11.7: Example of a point cloud containing cars that have been detected (yellow) and not detected (blue) before the learning process.

common points of segments of the same type, for each type of object.

Let us take the previous example illustrated in Figure 11.7. In this example, the detected cars are aligned and close to each other (no elements are between them). They also all have estimated dimensions (height, width, length). Each of these characteristics forms a group that characterizes the studied set (such as "object aligned with a car", "object close to a car").

Each characteristic grouping is a sub-set (or sub-class) of an object type. These groups are based on an aggregation rule related to the characteristics used to create it. This rule allows the assessment of the belonging of an element to a group. For groups formed from non-numerical characteristics (such as topological links), their aggregation rule is directly linked to their characteristics.

In the previous example, let us call the group based on the alignment criterion "AlignedWithCar". Equation 11.3 shows the aggregation rule.

$$Segment(?s) \wedge Car(?c) \wedge alignedWith(?s, ?c) \Rightarrow AlignedWithCar(?s) \quad (11.3)$$

It states that if a segment has a link "aligned with" to an object identified as a car, then it belongs to that group.

For groups formed from numerical characteristics, their aggregation rules are linked to an interval of values. Each value interval is intended to provide more flexibility than a thresholding method to assess the integration of a value. The calculation of a value interval requires a statistical study of the values of the characteristics that this interval represents. However, the characteristics studied are part of a set of objects that is identified as belonging to a given object type, but does not include all the objects of that type. Therefore, each value interval calculated from a confidence interval [*Kalinowski*, 2010], which allows calculating an interval from a set of value samples. The calculation of this interval provides a confidence level corresponding to the percentage of belonging to this interval, of the complete set of values. The confidence interval is defined by Equation 11.4 with $\bar{x}$ the mean of the values, δ the standard deviation, η the number of values and t_α the confidence coefficient.

$$I_c = \left[\bar{x} - t_\alpha \frac{\delta}{\sqrt{\eta}}; \bar{x} + t_\alpha \frac{\delta}{\sqrt{\eta}}\right] \quad (11.4)$$

In the previous example, let us call the group based on the proximity criterion "NearCar". This group is based on the proximity characteristic, which has a numerical value corresponding to the distance between two elementary elements. Thus the distances between the cars enables to compute a set of values used to calculate the confidence interval. In this example, the calculated confidence interval is $[1.2; 2.4]$. It means that for a segment to be in this group, it must have a distance value with a segment identified as a car between 1.2m and 2.4m. This process of grouping is similarly applied for the height, width, and length of cars.

These groups are combined to formulate hypotheses about the characteristics of an object. Thus if a segment belongs to every group composing the hypothesis, then it is classified as belonging to the set of objects corresponding to the hypothesis.

Let us reiterate the previous examples by considering the hypothesis that if a segment belongs to the "NearCar" and "AlignedWithCar" groups, then it is a car. In this example, the hypothesis is composed of two groups and is translated into the rule of inference presented in Equation 11.5.

$$Segment(?s) \wedge AlignedWithCar(?s) \wedge NearCar(?s) \Rightarrow Car(?s) \tag{11.5}$$

The formulation of hypotheses by combining groups allows a precise formulation of complex hypotheses. Besides, the more groups a hypothesis combines, the more specialized the hypothesis is on the whole studied. However, if a hypothesis is too specialized, it would not allow for identifying a new segment because the only segments that will validate this hypothesis are those used in the study and therefore those that are used to formulate the hypothesis. Conversely, if a hypothesis does not include enough groups, it may be too general and invalid.

Therefore, the formulation of hypotheses must be hierarchically ordered to improve identification as much as possible. It begins with the automatic formulation of a general hypothesis composed of a single group, and then. Then, if this hypothesis is invalidated, it is automatically combined with other groups to make a more complex hypothesis. Such a strategy is often providing more precise results.

11.3.3 Hypothesis verification

The object detection approach we propose is knowledge-driven. The hypotheses formulated produce new knowledge, which has the effect of changing the behavior of the detection approach for each application case considered. However, each hypothesis may be incorrect, and the knowledge it adds may lead to a regression in the quality of the detection obtained. Therefore it is necessary to verify the hypotheses.

Verification of a hypothesis requires measuring the consequences of changing or adding knowledge on the result of object detection to ensure that the detection is at least equivalent in quality. The hypothesis verification can, therefore, be performed by comparing the results obtained before the integration of hypothetic knowledge with the results obtained by integrating the new knowledge.

Thus, to validate a hypothesis, the results obtained with the addition of the new knowledge must not cause any inconsistency with the results previously obtained. Knowledge modeling is done in the form of constraints and rules of inference. Thus, checking the consistency of the knowledge base ensures the validation of hypotheses.

Let us take the example used in Section 11.3.2 and illustrated in Figure 11.7. In this example, several groups of characteristics are used to formulate hypotheses, including the "NearCar" and "AlignedWithCar" groups. Let us consider that the first hypothesis is based only on the "NearCar" group, which groups together the elements that must have a distance between 1.2m and 2.4m from a segment identified as a car. In this case, the hypothetic is that any element located at a distance between 1.2m and 2.4m from a car is a car. This hypothesis leads to the classification as a car of segments previously classified as trees. However, since these objects (such as trees) are defined as "disjoint" groups of cars, the result obtained by integrating this hypothesis is considered as being inconsistent. Consequently, the results obtained with the integration of this new knowledge are deleted, and the hypothesis is reformulated.

As explained in Section 11.3.2, the reformulation of a hypothesis consists in integrating more groups of characteristics.

In the previous example, the hypothesis can be reformulated by integrating the characteristic group called "AlignedWithCar". In this case, the assumption would be that any segment aligned with a segment identified as a car and located at a distance between 1.2m and 2.4m from that segment, is a car. This hypothesis is reformulated to become more and more specific, until the results induced by it are coherent, or it is no longer possible to enrich it. In the latter case, the reformulation process is over.

In this example, the reformulated assumption by integrating the two groups of characteristics allows the identification of segments that were not previously identified correctly (in blue in Figure 11.7).

The integration of new knowledge resulting from learning allows increasing the robustness and quality of the interpretation of data content by performing object detection that adapts to the specificity of each application case.

11.4 Discussion

The object detection presented in this chapter is fully driven by Knowledge. This process is based on a data processing step and a learning step.

Two phases compose the data processing: firstly a phase of algorithm management, secondly a phase of classification.

The algorithm management is itself divided into three steps. The first step is selecting algorithm relevant for the data and objects automatically. The second step is configuring the inputs and parameters of the algorithm automatically. These two first steps are applied by reasoning on the Knowledge-base. The last step is the automatic execution of algorithms through SPARQL queries that makes a conceptual and technical bridge between the semantic paradigm and the algorithmic paradigm. The SPARQL query parses information from the knowledge base to execute algorithms. Then, it interprets the algorithm results to integrate them into the knowledge base.

The phase of classification is based on the automatic integration of algorithm results inside the Knowledge-base. The information provided by these results allows for identifying objects and geometry through a geometric classification (e.g. the definition of shape, size) and topological classification (e.g. proximity, parallel, perpendicular).

Finally, the self-learning process step is composed of three phases to ensure its robustness: firstly, the knowledge enrichment, secondly the hypothesis formulation and thirdly, the hypothesis verification. The stage of knowledge enrichment aims at adding information about detected elements of the same type (e.g. cars) to identify further common characteristics between elements of this type. Based on the automatic identification of common characteristics, the process of learning creates new types defined from common specifical characteristics automatically. These new types are created through rules based on specific properties or values belonging to a confidence interval. These types are then used in a hierarchical order to formulate hypothesis aiming at improving the detection automatically. The step of hypothesis verification applies the hypothesis formulation automatically in the hierarchical order to ensure the improvement of the detection.

This process of self-learning process allows for adapting the detection process accurately and efficiently to the specificity of each application case automatically. Such adaptation is provided through the combination of knowledge-driven data processing and each self-learning process. This combination allows, for adapting the data processing according to the knowledge base, and for refining the knowledge according to each application case. Thus, this combination allows for more efficient data processing. The next chapter illustrates this automatic adaptation mechanism by presenting the implementation of some use cases .

Part IV

Implementation

The implementation part presents the choice made to implement the methodology presented in the previous part.
The methodology uses explicit knowledge to understand the data characteristics better and adapt the detection strategy to it and objects to detect. Furthermore, the proposed approach learns from the experience it acquires during the detection of objects, thanks to its ability to both formulate and validate hypotheses designed to understand the data characteristics better. This capability allows for compensating the lack of knowledge or precision in the description of knowledge.
This implementation part aims at illustrating the genericity, robustness, and portability of this approach, by showing its potential on four different application cases (presented in Section 2.1). These application cases have a common objective, to locate the elements (objects and geometries) contained in the data to structure them. The role of this part is not to be exhaustive by giving the modeling of all knowledge base components or by showing all steps followed by the process. Its role is to illustrate and explain the main steps of the proposed approach.

Chapter 12 presents the architecture implemented for the proposed approach.
Chapter 13 presents knowledge modeling for application cases.
Chapter 14 describes the object detection process for these application cases.
Chapter 15 presents the results obtained for the four considered application cases. It also compares the results obtained by the proposed approach with approaches from the literature.
Finally, Chapter 16 discusses the implementation choices and the results obtained by the proposed solution.

12 Processing architecture

Knowledge guides the object detection process by determining the suitable combination of algorithms to execute them. Object detection through algorithms execution from knowledge requires two main processes: knowledge management and algorithm management. The implementation processes depends on the requirements of knowledge processing and algorithm execution. Therefore this part presents the requirements of these two processes and compares processing supports to finally presents the architecture of the system.

12.1 Processing requirements

The knowledge processing is composed of two main functionalities that are the knowledge management for the detection process and the knowledge integration to enrich the system.

Software environment for knowledge management in the detection process requires the *manipulation of OWL* (c.f. Section 4.3.2). Moreover, it must allow reasoning on the model to apply the expert knowledge on the detection process. The system has been designed to be able to process a large variety of use cases, implying a large variety of knowledge, that is why this knowledge is stored in a triplestore. However, the use of a triplestore implies also the use of *SPARQL* to access and manage knowledge inside.

The extension of the system implies a functionality for the integration of new knowledge to enrich the knowledge base according to the enrichment of algorithm libraries. The system must be able to *read* and *load RDF/XML* files to enrich the semantic model.

Concerning the algorithm execution, processing vast point cloud requires fast algorithms executions, whose efficiency depends on the chosen programming language. This is why low-level programming languages like *C* are the most suitable.

12.2 Comparison of processing software environment

Among the different interfaces used to manage semantic technologies, two interfaces are emerging: the interface *JENA* for the language *JAVA* and *OWLCPP* for the language *C/C++* . Table 12.1 compares the two interfaces according to the system requirements previously presented.

Ability	OWLCPP	Jena
Manipulation of RDF	yes	yes
Manipulation of OWL	yes	yes
Use reasoner	yes	yes
Support SPARQL	no	yes
Serialize RDF/XML	no	yes
Load RDF/XML	yes	yes
Load/Write Turtle file	no	yes

Table 12.1: Comparative table between *OWLCPP* and *JENA*.

The interface *JENA* for the language *JAVA* is more suitable for knowledge management than *OWLCPP* for the language *C/C++*. Nevertheless, the language *JAVA* runs through a virtual machine (*JVM*), which makes it slower than languages like *C/C++*. On the other hand, *C/C++* has outstanding libraries such as [*Rusu and Cousins*, 2011] for point cloud processing. It is, therefore, more relevant to use the language *C/C++* to execute algorithms than to use the language *JAVA*. Choosing *JAVA* to implement the system would affect the efficiency of algorithm execution, whereas choosing *C/C++* would affect the efficiency of knowledge management. That is why the optimized implementation is to use *JAVA* for the knowledge management and *C/C++* for algorithm execution.

12.3 Software architecture

The comparison of processing software environment shows the optimal implementation of the system is based on a combination of *JAVA* and *C/C++*. However such architecture requires a *JAVA* interface to perform functions in *C/C++*. Thanks to the interface *JNI* of *JAVA*, it is possible to combine the two languages and take advantages of each other. Thus the architecture of our processing system is based on the interface *JENA* and *JNI* of the language *JAVA* to interface respectively with the knowledge base (described in *OWL* and using *SPARQL*) and the algorithms implemented in the language *C/C++*, as shown in Figure 12.1.

The use of the two interfaces (*JNI* and *JENA*) allows bidirectional management of

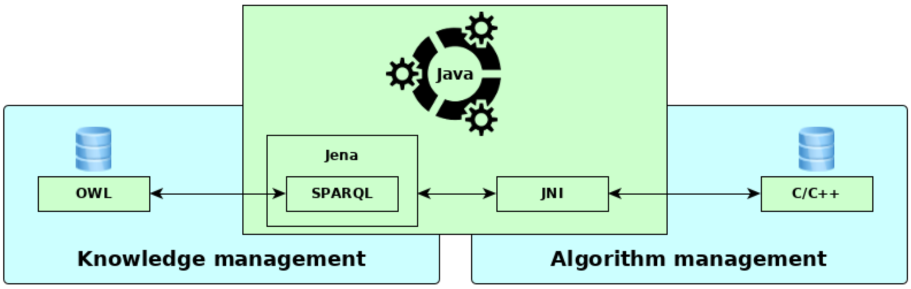


Figure 12.1: System processing architecture.

processing algorithms and knowledge by the processing system. Indeed, *JNI* allows the call and execution of processing algorithms in C++. It also allows for retrieving the results of the algorithms and transferring them to the processing systems. The processing system can then interpret these results as a bold semantic node in the *JENA* interface and then integrate them into the knowledge base.

This bi-directional architecture provides a high degree of flexibility in the detection process. Indeed, JAVA with the interface *JENA* enables for performing tasks like the choice of algorithms, their inputs, and their parameters. Each algorithm is then dynamically executed by the *JNI* interface. The results of these executions are then retrieved by the *JNI* interface and interpreted in JAVA to enrich the knowledge base using the *JENA* interface. The process can then be iterated to adapt the next task to be performed based on the new information acquired. This architecture provides a fully dynamic detection process, linking both the algorithmic paradigm and the semantic paradigm.

13 Knowledge modeling

The knowledge in the domain of Data, Scene, and Data Processing is required to handle each application case in an adapted way.

This knowledge is modeled in "OWL2" in an ontology. The knowledge about the Data Processing domain is common to all application cases, whereas the knowledge about the other domains (Data and Scene) contains specificities from the application cases. Nevertheless, application cases may share information. This is particularly the case for the knowledge of elementary objects (for example, walls, floors, rooms).

The application cases discussed in this thesis have been chosen to illustrate the adaptability of the approach presented and to show its robustness to changing characteristics. Therefore, the data for each use case were acquired using a different sensing process and contain contextual information that is different from the others.

13.1 Data processing knowledge

This section presents the knowledge modeling in the data processing domain.

The data processing is applied through algorithms. Thus, the knowledge of data processing concerns algorithms. These algorithms are modeled according to their inputs (by the property "*works on*"), parameters (by the property "*has parameter*"), prerequisites (by the property "*has prerequisite*"), and outputs (by the property "*produces*"). Algorithms used for the data processing and describing its knowledge domain are presented in Appendix A.

Let us here take the example of the normal region growing algorithm to describe the modeling of an algorithm that constitutes the knowledge of data processing. Figure 13.1 illustrates the knowledge modeling of this algorithm.

This normal region growing algorithm *works on* a *point cloud* to *produce segments*. It has two prerequisites. It requires that the point cloud has a *normal* for each point and has *less than one million points*. It also has two parameters that are the *normal*

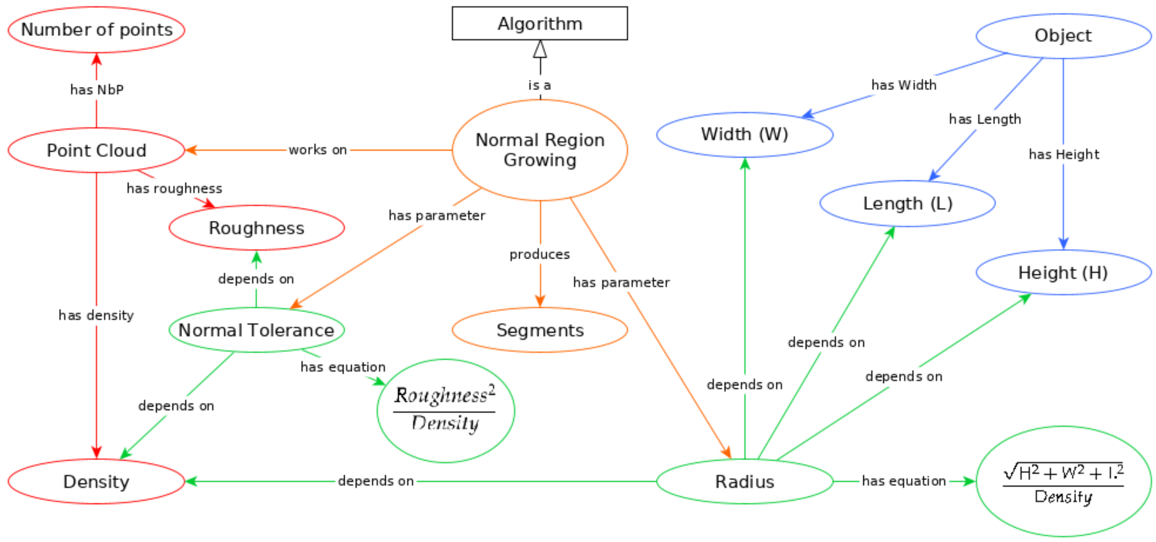


Figure 13.1: The knowledge data processing modeling for the region growing algorithm.

tolerance and the radius. The *normal tolerance* is computed from the *roughness* and the *density* of the data (see Equation 13.1).

$$NomalTolerance = \frac{Roughness^2}{Density} \tag{13.1}$$

The *radius* is computed from the *density* of the data and the *dimension* (height, width, and length) of the object to detect (see Equation 13.2).

$$Radius = \frac{\sqrt{H^2 + W^2 + L^2}}{Density} \tag{13.2}$$

The value of these parameters is computed during the configuration step of the detection process according to the sought object.

13.2 Data knowledge

This section presents the data knowledge. This knowledge comes from its modeling, but also from inference on the modeling that enriches the knowledge base. Therefore, a first part describes the data knowledge modeling, and a second part presents the deduced knowledge.

13.2.1 Data knowledge modeling

The knowledge about data consists of the knowledge about its acquisition, its characteristics, and objects that it contains. The knowledge of the acquisition process is information about the acquisition process as well as external factors that can influence the characteristics of the data.

Some data characteristics are stored as metadata (e.g. resolution, size, and dimension). The other characteristics of the data can be automatically deduced from the knowledge of the acquisition process (e.g. occlusion, density). In contrast, the other knowledge (acquisition process, external factors, objects, and geometries contained) must be specified by the user.

For the application cases studied here, only six data characteristics are sufficient and necessary: density, resolution, size, dimension (e.g. 2D, 3D), occlusion, and noise. Some of these characteristics are more decisive than others. In particular, the occlusion, noise, and density characteristics which have a significant influence on the choice and configuration of algorithms (c.f. Chapter 4). Besides, these characteristics often vary within the same dataset. For example, one portion of the data may be very dense and low-noise, while another portion may be low-density and high-noise. This information can be defined by the user but is mainly inferred from the knowledge of the acquisition process. In other cases of application, other characteristics must be taken into account.

Let us take the example of the application case 2.1.3. Figure 13.2 illustrates the knowledge modeling for the data related to this application case.

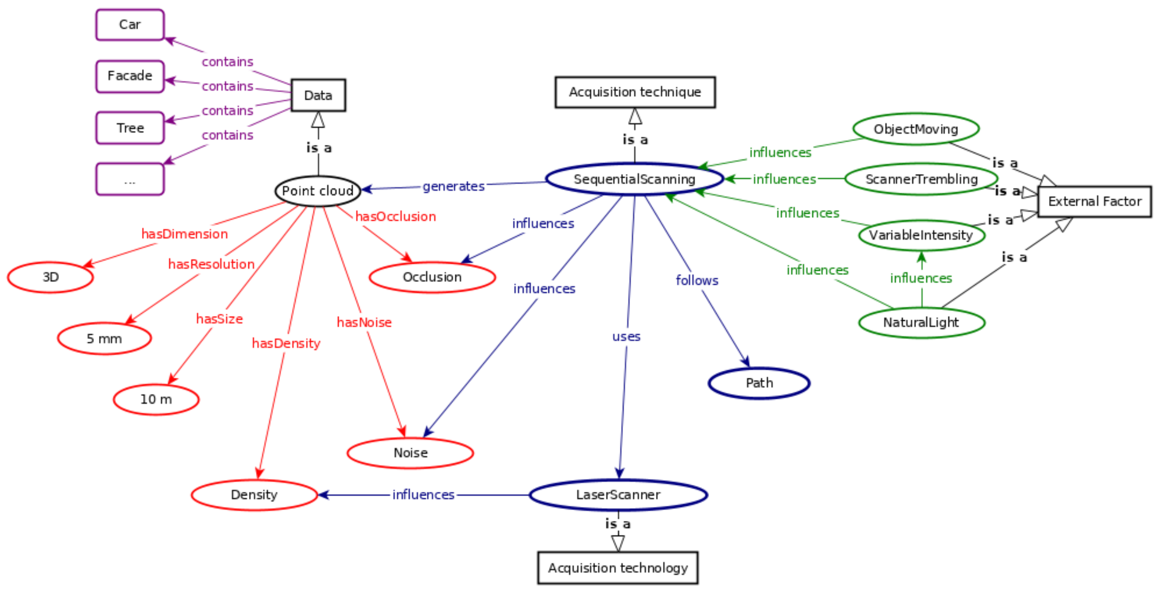


Figure 13.2: Data knowledge modeling for the use case 2.1.3.

In this application case, the data is acquired by a laser scanner that performs a sequential scanning of the scene following a straight trajectory. The natural light and the mobility of both the scanner and acquired objects (e.g. pedestrian, car) are external factors that influence the acquisition process. The objects that compose the data are objects that belong to an urban scene. They can, therefore, be trees, cars, pedestrians, facades, vegetation, traffic signs, or streetlights. This information allows for modeling several concepts. The central concept is the "data" that contains objects concepts (shown in purple in Figure 13.2). These concepts correspond to the objects sought. Then the concept of "*SequentialScanning*" (shown in blue in Figure 13.2) is defined as a kind of "*Acquisition technique*". It uses a "*laserScanner*", defined as a kind of "*Acquisition technology*" and follows a "*path*". The concept of "*laserScanner*" influences the density characteristic of the data, while the concept of "*SequentialScanning*" influences the noise and occlusion characteristics of the data. The other concepts "*NaturalLight*", "*VariableIntensity*", "*ScannerTrembling*", and "*ObjectMoving*" (shown in green in Figure 13.2) are defined as "*ExternalFactor*" which influences the concept "*SequentialScanning*". The influences between the concepts are described using *SPARQL* queries.

Finally, information on the resolution, size and dimension of the data is dynamically translated into a description logic (shown in red in Figure 13.2) by linking their value to the data by properties ("*hasResolution*","*hasSize*", and "*hasDimension*").

13.2.2 Data knowledge inference

SPARQL queries correlate contexts information and execute algorithms to infer new knowledge. It is through these queries that the occlusion, density variation, and noise characteristics can be identified. It is necessary to understand the causes that produce these characteristics in order to transcribe them into a rule of inference or *SPARQL* query.

In the continuity of the previous example (i.e. the application case 2.1.3), the concept "*SequentialScanning*" generates occlusions in the data.

Code 13.1 shows the *SPARQL* query that characterizes the influence of the "Occlusion" concept on the data.

```
CONSTRUCT {
?occlusion rdf:type OccludedArea.
?data hasOccludedArea ?occlusion.
 } WHERE {
```

```
?data rdf:type PointCloud.
?data ?hasAcquisitionMethodology ?sec.
?sec rdf:type SequentialScanning.
?sec follows ?p.
?object rdf:type Object.
?object hasHeight ?h.
?object hasWidth ?w.
?object hasLenght ?l.
?object hasLocalization ?local.
?occlusion getOcclusions(
  "pointcloud=?data" "path=?p" "height=?h" "width=?w" "lenght=?l" "localization=?local"
  )
}
```

Code 13.1: *SPARQL* query for occlusions identification

The query states that if the method used to acquire the data ("*?data*") is a sequential scan ("*?sec*") following a path ("*?p*"), then the presence of an object ("*?object*") causes occlusions ("*?occlusion*"). These occlusions are then calculated according to the acquisition path ("*?p*"), the position ("*?local*") and the size of the object ("*?h*",*?w*",*?l*") by calling an algorithm ("*getOcclusions*"). The occlusions calculated by the algorithm are then assigned to the data by a property ("*hasOccludedArea*"). This query is designed only once and will be automatically applied to all data that is acquired by sequential scanning along a path. Besides, it is based on the detection of elements. Thus, the more the object detection process locates objects (without necessary identify them), the more precisely the data can be characterized.

The location of the occlusion areas provides essential information for the classification step (as explained in Section 14.1.2).

Similarly, the influence of the instrument on data density is described in the form of queries using built-in algorithms. In the case of a laser scanner, the scanner remoteness from the acquired objects reduces the density of the points. Thus the corresponding query states that the further away the scanner is from the objects, the less dense the acquired data is. This query is used to characterize areas of low, medium, or high density. Figure 13.3 shows the characterization of the data of the application case 2.1.3 characterized by density areas.

The density areas provide relevant information for selecting and configuring algorithms (as shown in Section 13.1).

Finally, external factors such as the characteristic of "*Variable Intensity*" of light, "*ScannerTrembling*", and "*ObjectMoving*" influence both the noise and the regularity of the shapes that the data may contain. In this case, these factors provide global

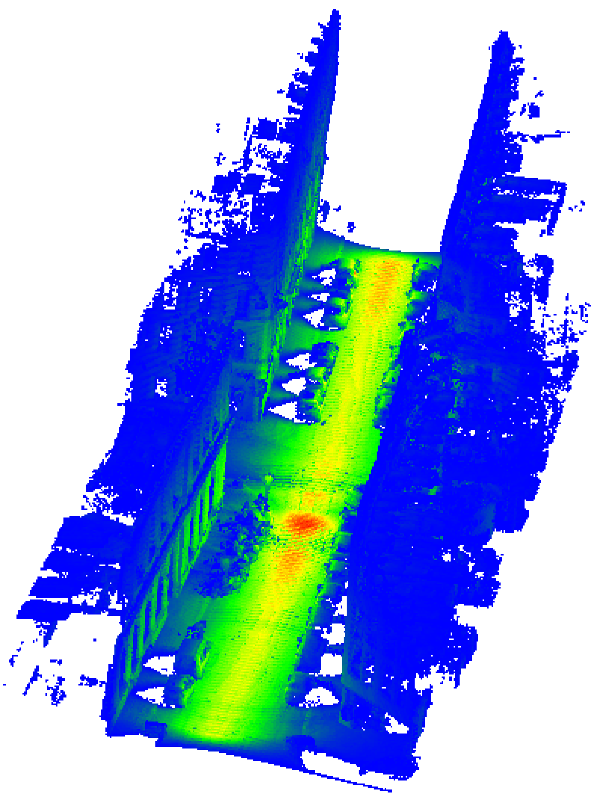


Figure 13.3: Density computation results based on the knowledge of the sensing process: high density in red and yellow, medium density in green, and low density in blue.

and additional information about the data. This information cannot be located unless the user mentions it. However, it does indicate that the data may be noisy and may have irregular shapes. The notion of "possibility" is modeled in "OWL2" by "owl:someValuesFrom" type constraints. This information allows for the addition of denoising algorithms in the detection process as well as providing more understanding of the content to classify and identify segments (portions of data).

The knowledge about data characteristics and the acquisition process provides an essential basis for understanding and structuring data. Data characteristics are used to guide the choice and configuration of algorithms. The knowledge about the acquisition process and the external factors influencing it can be combined with other knowledge. For example, the knowledge about light and the measuring instrument is combined with the knowledge about digitized objects to locate better areas of low or high density (as shown in Section 13.3.1).

13.3 Scene knowledge

The scene knowledge is represented through the modeling of objects and the scene. The object modeling is defined through its characteristics, its geometric composition or objects composition and its topological relationship with other objects. These topological relationships allow for describing the location of an object to the other in the scene. The scene is thus represented through topological relationships, and context that impacts the representation of the objects.

13.3.1 Object modeling

In the proposed approach, all the elements that compose the data are defined as "objects". The application cases considered in this thesis have some objects in common (e.g. floor and walls) and some specific objects according to the context (e.g. the watermill, bowls, tables, cars, traffic signs, trees). The knowledge on each of these objects must be modeled, as explained in Section 10.3.

On the one hand, an object is modeled through its characteristics, its geometry or object composition, and its topological relationships. This modeling allows for identifying segments (portions of data) that correspond to the definition of an object and classify them in this object category. On the other hand, object modeling describes object characteristics that can influence the acquisition of data (e.g. roughness, color, materials, height, length, width).

Description logics for identification

Let us take the example of the knowledge modeling about the floor and walls (i.e. portions of data having a specific particularity), which are common to our application cases. The geometry of the walls corresponds to a vertical planar surface. They have the specific characteristic of having a height greater than 2m. Finally, their topological characteristics are to be perpendicular to the floor and on the floor. Code 13.2 shows their semantic description in the Manchester syntax.

```
Wall:
Object
 and (hasGeometry some
    (Plane
     and (hasOrientation only Vertical)))
 and (isPerpendicular min 1 Floor)
 and (on min 1 Floor)
 and (hasHeight exactly 1 xsd:double[> "2.0"^^xsd:double])
```

Code 13.2: Wall modeling

Similarly, the geometry of the floors corresponds to a horizontal planar surface. Their specific characteristics are that they have a surface area greater than 1 m^2. Finally, their topological characteristics are that they can be perpendicular to walls. Code 13.3 shows their semantic description in the Manchester syntax.

```
Floor:
Object
 and (hasGeometry some
    (Plane
     and (hasOrientation only Horizontal)))
 and (isPerpendicular some Wall)
 and (hasSurface exactly 1 xsd:double[> "1.0"^^xsd:double])
```

Code 13.3: Floor modeling

Objects can also be defined as being composed of several other objects. For example, a room is defined as composed of a floor parallel to a ceiling, and both are connected to at least two common walls.

Code 13.4 shows the semantic description of a room in the Manchester syntax.

```
Room:
Object
 and (isComposedOf exactly 1
```

```
  Floor that isParallel exactly 1 Ceiling
   and isConnectedTo min 2 Wall)
```

Code 13.4: Room modeling

Similarly, the watermill, which is the main element required in the case of application 2.1.2, is defined as being composed of two connected rooms, each with its own dimensions.

Description logics for inference

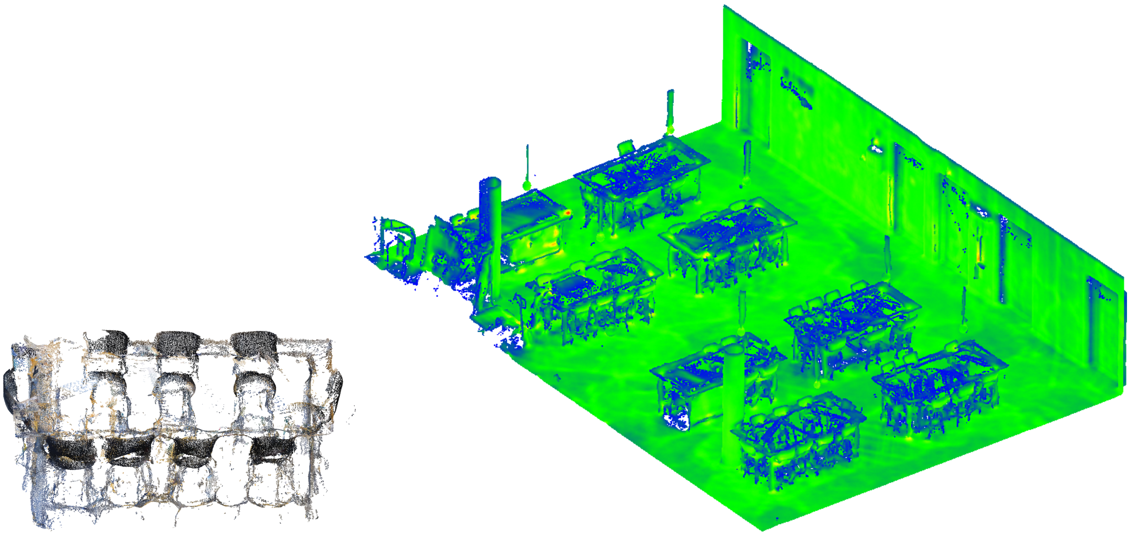
The knowledge about objects such as material and color can be combined with the knowledge of the acquisition process and external factors to refine the characteristics of the objects. For example, the rule of inference shown in Equation 13.3 allows to deduce that if an object ("*?o*") is made of glass ("*?m*") and that the instrument used ("*?i*") in the acquisition process ("*?a*") is a laser scanner, then the object will be represented with a low density ("*?l*").

$$\begin{gathered} Object(?o) \wedge hasMaterial(?o, ?m) \wedge Glass(?m) \wedge \\ Data(?d) \wedge Contains(?d, ?o) \wedge Acquisition(?a) \wedge \\ hasInstrument(?a, ?i) \wedge LaserScanner(?i) \wedge Generates(?a, ?d) \\ \Rightarrow LowDensity(?l) \wedge hasDensity(?o, ?l) \end{gathered} \tag{13.3}$$

Moreover, data with low density or containing occlusions would generally produce over-segmentation of an object. This knowledge means that an object occluded or with low density can be represented by several segments after the data segmentation instead of a single segment. The knowledgebase contains a class called *SegmentsSet*, defined as composed of several similar segments (having similar characteristics). Therefore, if an object is represented with a low density or occluded, then it is represented through a set of segments. The rule of inference in Equation 13.4 represents this knowledge.

$$\begin{gathered} Object(?o) \wedge (LowDensity(?l) \wedge hasDensity(?o, ?l)) \\ \vee (Occlusion(?oc) \wedge hasOcclusion(?o, ?oc)) \Rightarrow SegmentsSet(?o) \end{gathered} \tag{13.4}$$

Such rule facilitates the search for objects in the data and their classification (see Section 14.1.1. Figure 13.4 shows the example of glass tables in a point cloud acquired by laser scanning from the company NavVis and the density of the point cloud. This figure illustrates the relationship described by the rule.



(a) A glass table from NavVis point cloud.

(b) Density of a point cloud NavVis (medium density in green, low density in blue).

Figure 13.4: Example of glass tables in a point cloud acquired by laser scanning from the company NavVis.

Knowing a glass table has a low density in this point cloud allows for searching it in areas with a low density (in Figure 13.4).

13.3.2 Scene modeling

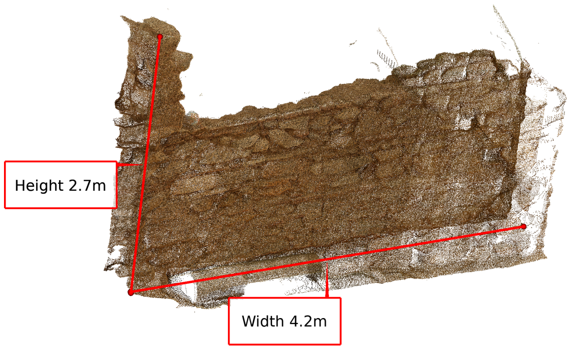
The scene modeling is composed of its context that impacts the object representation in the acquired data. The scene content is represented by objects modeling and their topological relationships between each other.

Scene context

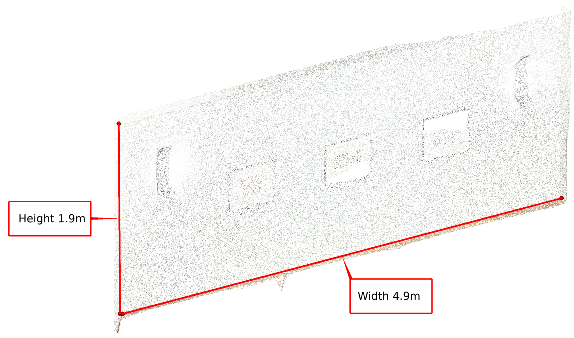
It exists diverse representations of an object. Searching to model all representations of an object would be a long time-consuming task not to provide a significant improvement. However, adapting the description logics of an object according to its context, would provide a more adapted description to search the object. Therefore, the scene modeling describes its context, but also its impact on objects in such context. Thus, the reasoning on the scene context modeling and standard description logics of an object, adds automatically a new concept describing objects in this context, in a better way.

Let us take the example of a wall representation in the data for the application cases studied. Figure 13.5 shows the difference in the representation of the walls between these different application cases.

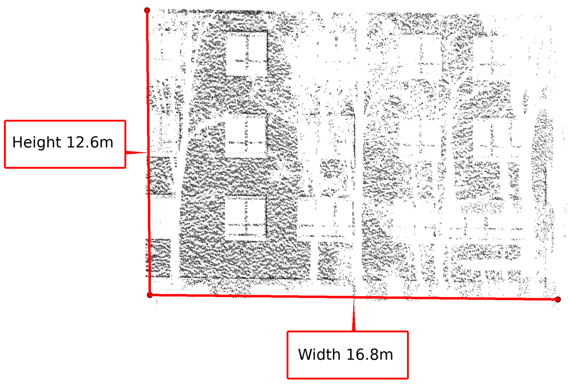
(a) Walls representation in the use case 2.1.4.



(b) Walls representation in the use case 2.1.2.



(c) Walls representation in the use case 2.1.1.



(d) Walls representation in the use case 2.1.3.

Figure 13.5: Different walls representation in the study cases considered.

In these different cases of applications, the walls do not have the same dimension (height, width, length) and do not have the same geometry (planar for cases 2.1.1, 2.1.4, 2.1.3 and non-planar for 2.1.2). They may also not have the same topological links (e.g. case walls 2.1.3 are not perpendicular to a floor).

Consequently, due to the variance of objects and the many factors influencing objects appearance, it is not possible to model the knowledge of objects in such a way as to describe all possible representations of objects. Thus, it is necessary to understand and model the reasons that cause these various representations of objects. That is why the knowledge on the scene context and its impact on objects representation is modeled through rules of inference allowing to adapt the description of the objects significantly (as explained in Section 10.3.3).

Let us pursue the example of the wall description logics, which is an object common to the different application cases studied. The context of the digitized scenes of the application cases 2.1.4 and 2.1.1 does not influence the description of the walls. In contrast, the context of the digitized scene in the application case 2.1.2 influences the description of the size, geometry, and roughness of the walls. This impact is modeled through the rule presented in Equation 13.5.

$$\begin{aligned} Object(?o) \wedge Scene(?s) \wedge belongsTo(?o, ?s) \wedge \\ DestroyedObject(?d) \wedge hasCharacteristics(?s, ?d) \\ \Rightarrow IrregularShape(?i) \wedge hasShape(?o, ?i) \wedge hasHeigh(?o, > 0m) \end{aligned} \tag{13.5}$$

The interpretation of this rule through a *SPARQL* update query on the standard wall description (c.f. Code 13.2) results in the definition shown in Code 13.5 of a wall for the use case 2.1.2 according to the Manchester syntax.

```
WallCH:
Object
 and (hasGeometry some
    (IrregularShape
     and (hasOrientation only Vertical)))
 and (isPerpendicular min 1 Floor)
 and (on min 1 Floor)
 and (hasHeight exactly 1 xsd:double[> "0.0"^^xsd:double])
```

Code 13.5: Wall modeling in cultural heritage use case (2.1.2)

Similarly, the context of the digitized scene in the application case 2.1.3 influences the description of the wall height and enables to remove the obligation of the topological link between the floor and the walls. This impact is modeled through the rule presented in Equation 13.6.

$$\begin{aligned} Wall(?w) \wedge UrbanScene(?s) \wedge OutdoorScene(?s) \wedge belongsTo(?o, ?s) \\ \Rightarrow hasArea(?w, > 17m^2) \wedge hasHeigh(?w, > 10m) \end{aligned} \tag{13.6}$$

The wall description for the use case 2.1.3 is impacted by the previous rule (Equation 13.6) on the scene context, but also, by rules defining the impact of the acquisition process on its definition. All of these impacts result in the following description logic for the Wall in use case 2.1.3. This description is presented in Code 13.6 according to the Manchester syntax.

```
UrbanWall:
(Object or SegmentsSet)
 and (hasGeometry some
```

```
    (Plane
     and (hasOrientation only Vertical)))
 and (isPerpendicular some Floor)
 and (on some Floor)
 and (hasHeight exactly 1 xsd:double[> "10.0"^^xsd:double])
 and (hasArea exactly 1 xsd:double[> "17.0"^^xsd:double])
```

Code 13.6: Wall modeling in urban outdoor use case (2.1.3)

Topological hierarchy

The topological relationships between objects, contained in their modeling, allows for describing the scene. The topological relationships can provide geometric relationships (as perpendicular, parallel, connected, composition) or relative location (as distance, under, on). These relationships allow for identifying the research area of an object or determining a logical order of detection. For example, a room, which is composed of a floor and walls, requires the detection of walls and floors before to be detected. Another example is the property *on* used in all use cases to guide the logical order of detection. Figure 13.6 illustrates the modeling of the topological relationship *on* for the different use cases.

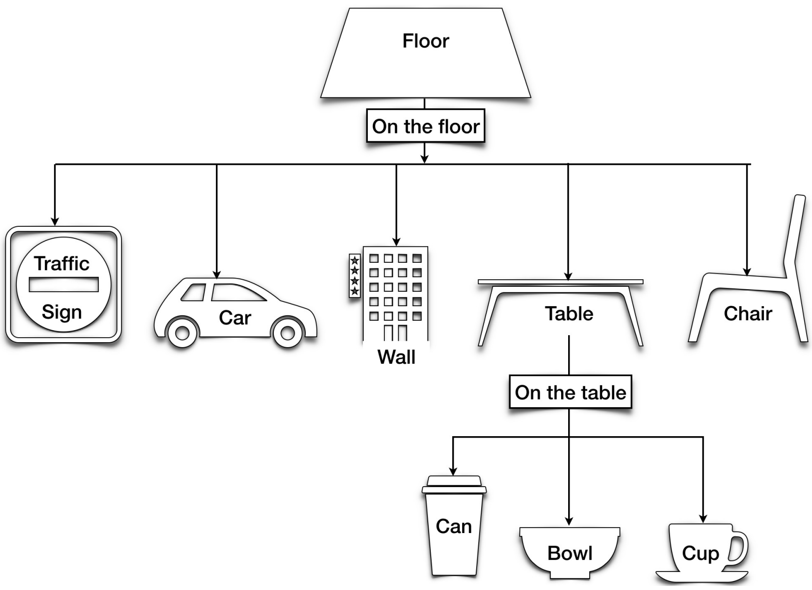


Figure 13.6: Illustration of topological hierarchy between objects.

13.4 Discussion

The user provides the knowledge modeling on data processing, data, and scene domain as description logics in an ontology. This knowledge base is at first enriching through the linking of information about the different domains. The knowledge about the factors that influence the data characteristics is described using rules of inference and queries. Such rules and queries allow the system to deduce the different possible representations of objects or to delimit the search field of this type of object. This enrichment of the knowledge base aims at providing a knowledge description more adapted to the processed application case. Then, the knowledge base is used to guide the process of detection. The more precise the description of objects is, the better their detection in the data is. That is why the first enrichment of the knowledge base is essential to provide an optimal knowledge model to give a maximum of information for guiding the detection process, presented in the next chapter 14.

14 Data processing

The knowledge explained in the previous chapter 13 fully drives the detections process. The selection of algorithms is made according to the characteristics of the objects and data (explained in Sections 11.1.1 and 11.1.2). Besides, the characteristics of objects and data are adapted to each application case by knowledge of the sensing process, external factors, acquisition context, and scene. Thus the selection algorithms for processing data and supporting object detection are adapted to each application case.

The complete processing to identify objects in the data and thus structure the data, start by selecting the relevant algorithms to process the data (as explained in Section 11.1.1). Then the selected algorithms are configured (as explained in Section 11.1.2) and executed (as explained in Section 11.1.3). The results of their execution enrich the knowledge on objects, geometries, and data and allow the classification of data portions (as explained in Section 11.2). Finally, the classification of objects and geometry in the data allows a self-learning process. The self-learning process uses how objects are represented in the data as well as how the data is structured. This self-learning process allows for "intelligent" reasoning capable of formulating and testing hypotheses designed to improve the accuracy of knowledge (as explained in Section 11.3).

This chapter shows these different steps through the results they produce when used to detect different objects in the application cases studied.

14.1 Detection process

The detection process is a cycle composed of a data processing step and a classification step. The data processing step consists in selecting, configuring, and executing algorithms. If an algorithm detects a characteristic of an object (e.g. size, shape, orientation), and all its prerequisites are met (e.g. normal estimated, data denoised), then it is selected for the application case considered. Moreover, algorithms that meet the prerequisites (e.g. size, normal estimation, denoising) of a selected algorithm are also selected. The selected algorithms are configured based on the knowledge available on their parameters. They are then executed through queries, as explained in Section 11.1.3. Algorithms are adapted according to the knowledge learned from their inputs, outputs, and prerequisites. Thus an algorithm satisfying the prerequisites of another is executed before it.

The hierarchical strategy of detection depends on the topological links between the objects and their size. The topological relationship provides information about the location of an object according to others. This information allows for reducing the area of object search, and the size determines the priority of object detection (i.e. the biggest objects obtains the highest priority). For example, a can is defined as "being on a table," and a table is defined as "being on the floor". A can is smaller than a table and a table smaller than the floor; the detection process begins by detecting the floor, then the table, and finally the can. In addition to this combination of size and topological relationship to prioritize the object detection, the relationship of composition describes a specific hierarchy relation. The meaning of the composition relationship between objects translates a requirement to detect the parts of an object before to be itself detected. Therefore, this composition relationship defines by itself a detection hierarchy. For example, the identification of a room requires firstly, to detect walls and floors.

According to the description of knowledge of the application cases, the detection process begins with the floor detection. Then it detects objects on the floor such as walls, tables, chairs, cars, motorcycles, and traffic signs. Among these objects on the floor, the walls are the biggest objects. Therefore they are detected in priority. Finally, it detects objects located on a table for the application case 2.1.4 such as cans, bowls, cereal boxes, and cups.

14.1.1 Detection of floors

The main steps of the floor detection are illustrated in this section. Firstly, the data processing step selects and configures algorithms according to the definition of a floor, and then execute them.

Data processing for the floor detection

Figure 14.1 shows the algorithms graph for the floor detection. The modeling of the algorithms used is provided in Appendix A.

The floor has a geometry defined by a horizontal orientation that is common to the

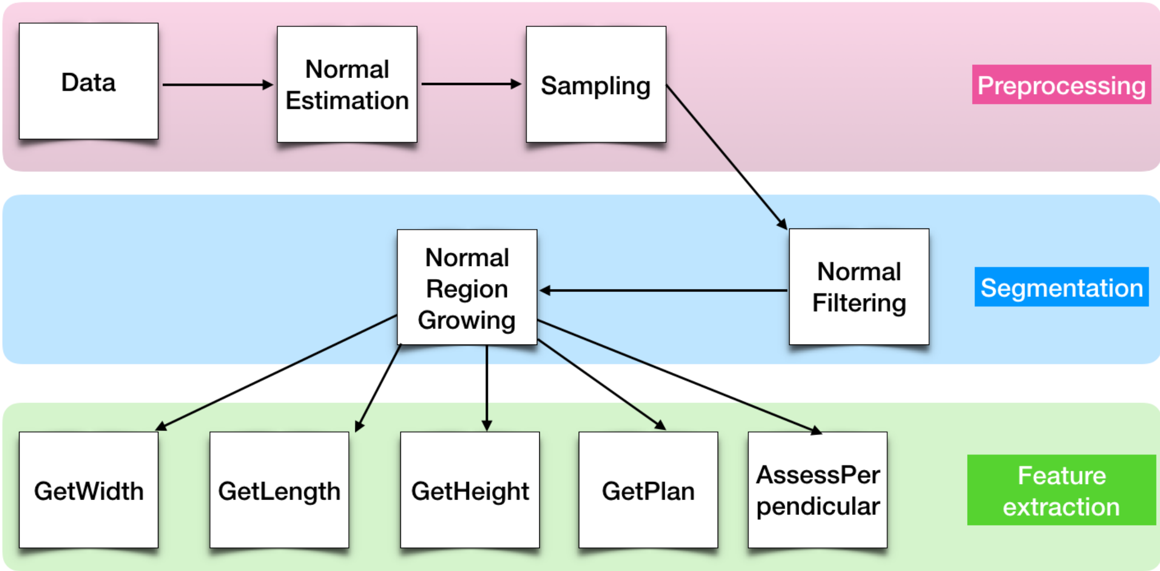


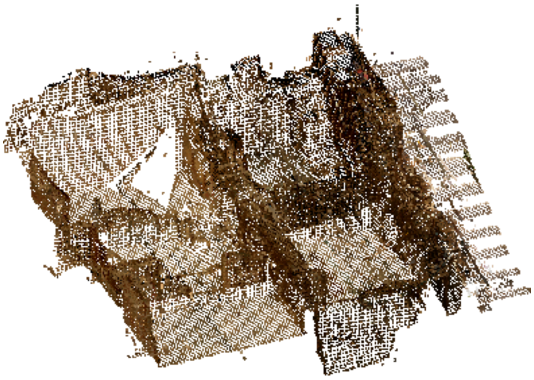
Figure 14.1: Algorithms graph for the floor detection.

application cases studied. This orientation implies that the normals at each point of the data of the application cases are estimated. That is why the first algorithm executed is a normal estimation algorithm ("*Normal Estimation*" in Figure 14.1).

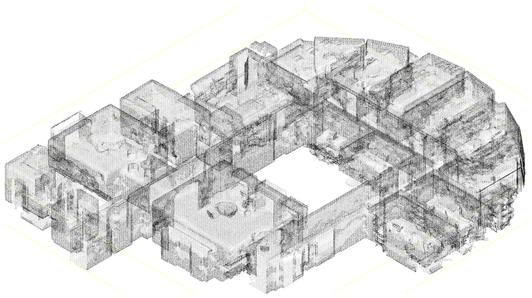
The second algorithm executed for the application cases 2.1.2, 2.1.3, and 2.1.1 is a sampling algorithm. This algorithm is selected to reduce the size of the data. It satisfies the prerequisite of the segmentation algorithm ("*Normal Region Growing*" in Figure 14.1) chosen to segment the data. Indeed, this segmentation algorithm requires that the data be as small as possible. Contrary to the data in the application case 2.1.4, which is small (less than one million points), the three application cases 2.1.2, 2.1.3, and 2.1.1 have large data sizes (several million points) that require to be reduced before applying the algorithm of "*Normal Region Growing*".

The sampling algorithm is configured according to the minimum dimensions of the objects to be detected. Therefore, for the application cases 2.1.3, 2.1.1, and 2.1.2, its execution produces a dataset sampled according to the minimum floor size. Figure 14.2 shows the results of the sampling algorithm for the three application cases concerned.

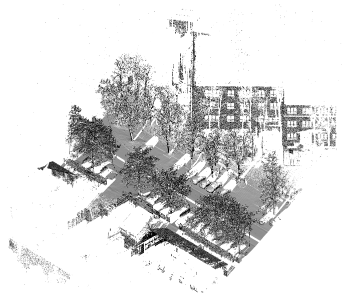
Floors are defined as having a single orientation (horizontal), in the case of the applications studied. That is why the third algorithm to execute is a filtering algorithm ("*Normal Filtering*" in Figure 14.1) that filters the data according to the orientation of the objects sought. This algorithm allows reducing the search field of objects. It is configured according to the orientation of the searched objects and produces new data composed only of the portions having an orientation corresponding to the searched objects. Figure 14.3 shows the results of the "*Normal*



(a) Results of the sampling algorithm on the use case 2.1.2.



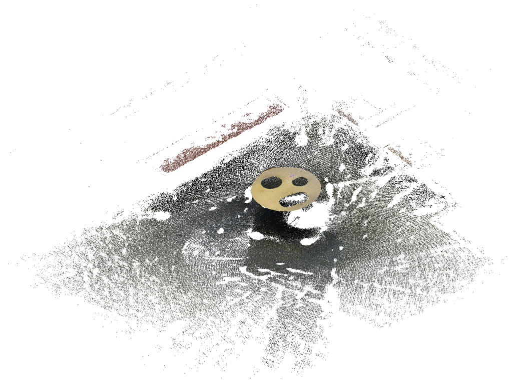
(b) Results of the sampling algorithm on the use case 2.1.1.



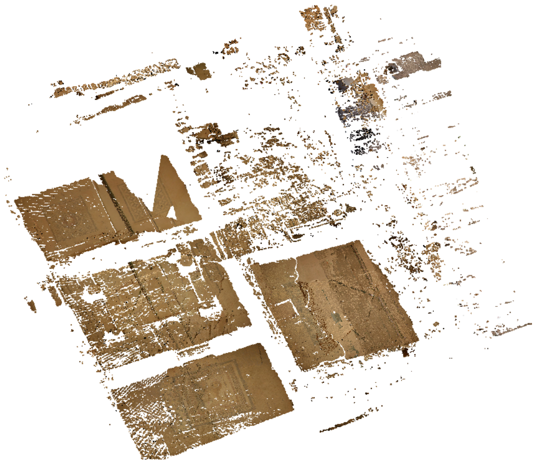
(c) Results of the sampling algorithm on the use case 2.1.3.

Figure 14.2: Results of the sampling algorithm for the three application cases concerned.

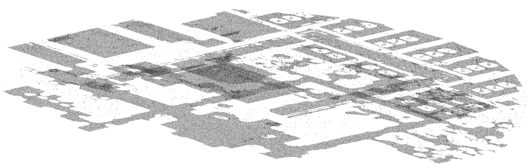
Filtering" algorithm on the four application cases studied.



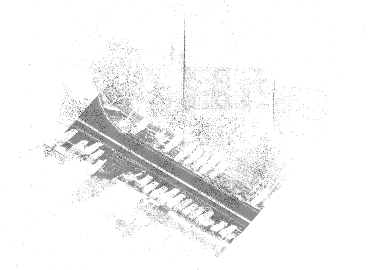
(a) Results of the horizontal filtering algorithm on the use case 2.1.4.



(b) Results of the horizontal filtering algorithm on the use case 2.1.2.



(c) Results of the horizontal filtering algorithm on the use case 2.1.1.



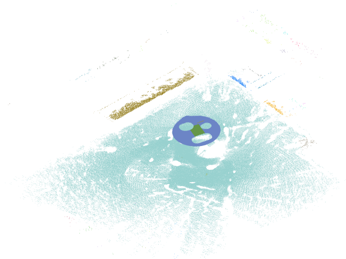
(d) Results of the horizontal filtering algorithm on the use case 2.1.3.

Figure 14.3: Results of the "*Normal Filtering*" algorithm for the floor detection in application cases studied.

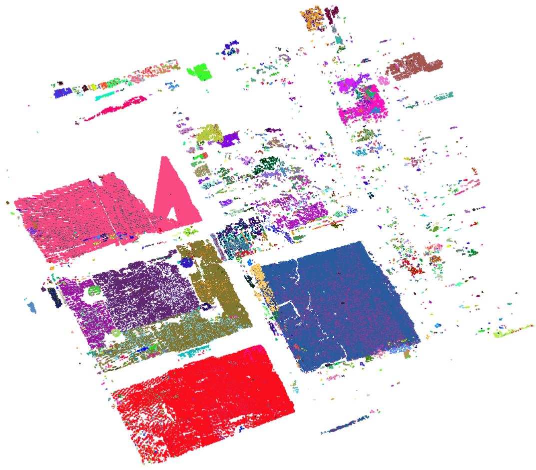
Floors are defined as segments (data portion), so it is necessary to use segmentation algorithms to divide the data into segments. Moreover, floors are defined mainly by their orientation. Thus a "Normal Region Growing algorithm" ("*Normal Region Growing*" in Figure 14.1) is more relevant for detecting segments in agreement with object characteristics, than segmentation algorithms based on other characteristics such as color. That is why this algorithm is executed on the application cases studied. Figure 14.4 shows the results of the "*Normal Region Growing*" algorithm on the application cases studied.

Floors are defined by a planar geometric shape for application cases 2.1.3, 2.1.4, and 2.1.1. A plane detection algorithm is therefore relevant to detect such objects for application cases 2.1.3, 2.1.4, and 2.1.1.

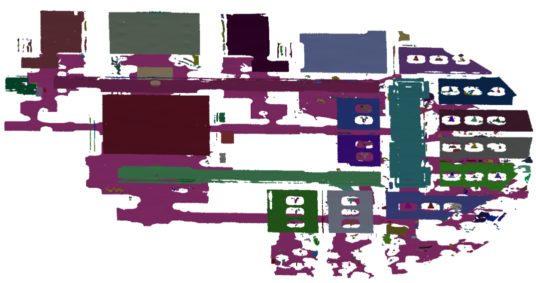
Knowledge of the context and the data acquisition process of the application case 2.1.2 enables to adapt the shape of floors defined as a planar surface in defining them as irregular.



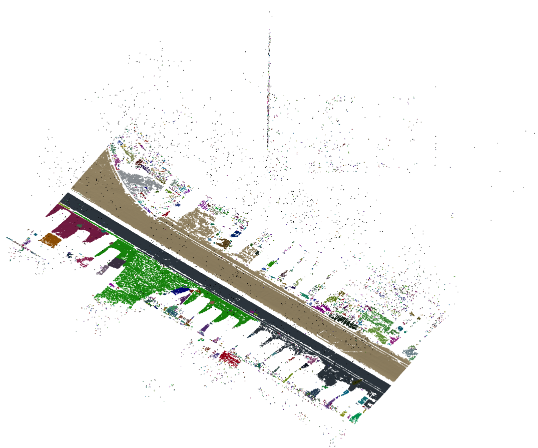
(a) Results of the segmentation applied to the horizontally filtered data of the use case 2.1.4.



(b) Results of the segmentation applied to the horizontally filtered data of the use case 2.1.2.



(c) Results of the segmentation applied to the horizontally filtered data of the use case 2.1.1.



(d) Results of the segmentation applied to the horizontally filtered data of the use case 2.1.3.

Figure 14.4: Results of the "*Normal Region Growing*" algorithm for the floor detection in the application cases studied. A single color is assigned to each segment.

Similarly, various algorithms for estimating characteristics such as dimensions (height, width, length), volume, area or topological relationships (e.g. parallel, perpendicular, in contact) are applied to the segments.

The results of the algorithms are integrated into the knowledge that describes the segments.

The knowledge about the segments is then analyzed in the classification process to identify objects in the data.

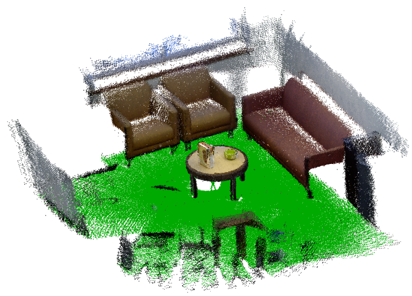
Classification

The *OWL2* formalism in which knowledge is modeled allows classifying a segment by the application of automatic logical reasoning. A segment is classified as an object if it satisfies the constraints defined for this object. A segment satisfies the constraints on an object geometry if it has all geometrical characteristics defined for this object.

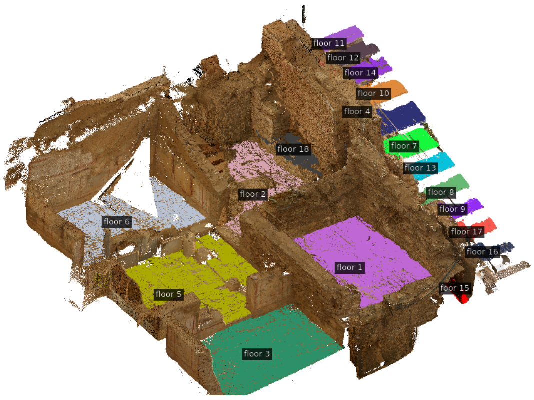
However, the logical reasoning mechanisms used until now by reasoners do not allow to work with reasonable execution times (several hours or even several days to perform reasoning). This problem is aggravated by the fact that the classification process must be repeated several times.

That is why we propose to translate the description logics of objects and geometry into a rule of inference which is then interpreted into SPARQL queries. The application of SPARQL queries is performed in a few seconds on the ontology used for the application case compared to the application of rules of inference such as SWRL or inference reasoning that both take several hours to apply. The classification of segments into objects is based on a semantic description of the objects, which is composed of geometric characteristics, object-specific characteristics, and topological links.

According to the definition of a floor presented in Section 13.3.1, a floor is a horizontal plane with a surface area greater than 1 square meter. After the execution of the normal region growing, the horizontal segments resulting from this algorithm are classified as horizontally oriented planes. After the execution of the feature extraction algorithms (e.g. *getHeight*, *getWidth*), each of these segments is characterized. The surface and volume characteristics are automatically calculated from the segment dimensions. Thus, among the segments identified as horizontal planes, those with an area greater than 1 square meter are identified as floors. Figure 14.5 shows the floor classification results for each application case.

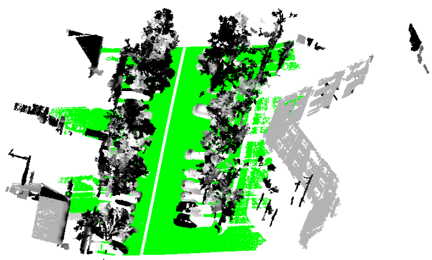


(a) Results of floors detection (in green) on the use case 2.1.4.



(b) Results of floors detection on the use case 2.1.2. A single color is assigned to each floor.

(c) Results of floors detection on the use case 2.1.1.



(d) Results of floors detection on the use case 2.1.3.

Figure 14.5: Results of floors detection on application cases studied.

14.1.2 Detection of objects on the floor

Among the different use cases, several objects require floor detection to be detected. In all cases, there are walls. In the urban outdoor environment, there are cars, traffic signs, trees, motorcycles, and street lights. In the indoor environment, the Kinect application case (2.1.4) aims at detecting table, chair, and computer on the floor. This section illustrates at first the detection of walls as common to all cases and particularly required for the room detection in use cases 2.1.2 and 2.1.1. Then, it presents the light street detection in the urban outdoor application case 2.1.3 due to its specificities of geometric and topological relationships. Finally, it shows the table detection in the Kinect use case (2.1.4) to follow the topological hierarchy of detection presented in Section 13.3.2, which is pursued in the next section through the detection of objects on the table.

Objects on the floor are searched among all points above the floor.

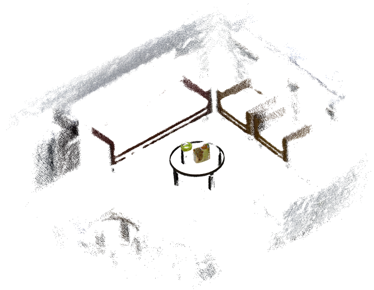
Walls detection

The algorithm selection of the wall is similar to the algorithm selection of floors (c.f. Figure 14.1). The steps of pre-processing being similar and yet executed, there are not executed again. The difference of processing is in the algorithm configuration for the segmentation and the feature extraction, due to the difference of description between a floor and a wall. For example, the parameter of normal filtering for a wall is horizontal normal, whereas the configuration of this algorithm for the floor is vertical normal. Figure 14.6 shows the results of the "*Normal Filtering*" algorithm on the four application cases studied.

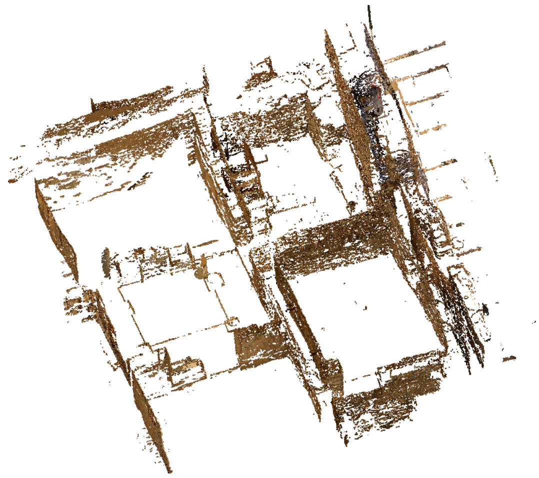
Then, another normal region growing is executed on the results from the normal filtering configurated for wall detection. Figure 14.7 shows the results of the "*Normal Region Growing*" algorithm on the application cases studied.

Finally, feature extraction is executed on the results of the normal region growing configurated for wall detection.

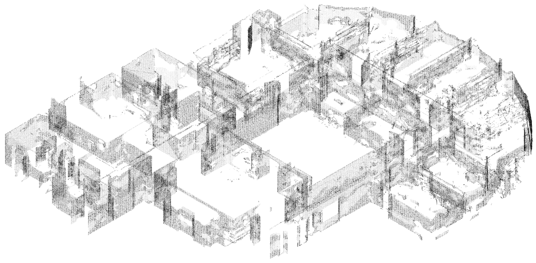
Walls classification The classification is executed after each enrichment of the knowledge base. Thus, when all segments are characterized, those respecting the description logic of a wall are classified as a wall. Wall detection in an outdoor environment is impacted by occlusions that divide a wall into several walls. As explained previously, the occlusion areas are detected by inference on the acquisition process knowledge and the knowledge of the scene. The knowledge of the scene includes the description of segments and objects such as their location and size.



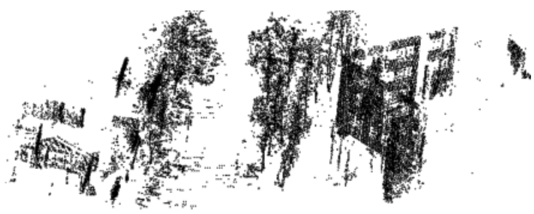
(a) Results of the vertical filtering algorithm on the use case 2.1.4.



(b) Results of the vertical filtering algorithm on the use case 2.1.2.

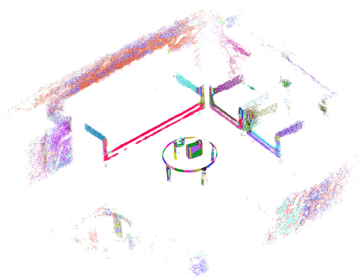


(c) Results of the vertical filtering algorithm on the use case 2.1.1.

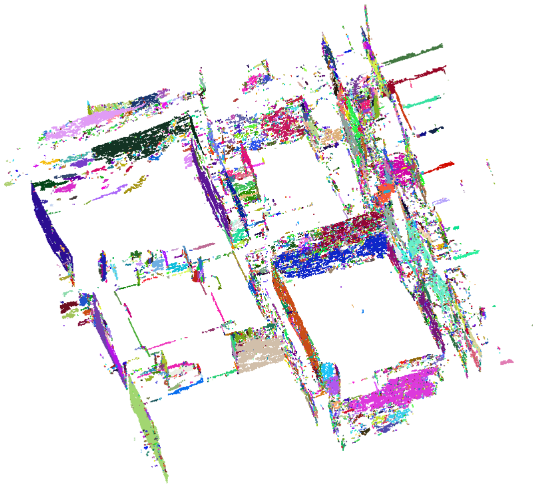


(d) Results of the vertical filtering algorithm on the use case 2.1.3.

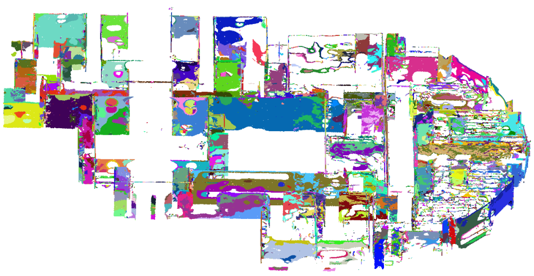
Figure 14.6: Results of the "*Normal Filtering*" algorithm for walls detection in application cases studied.



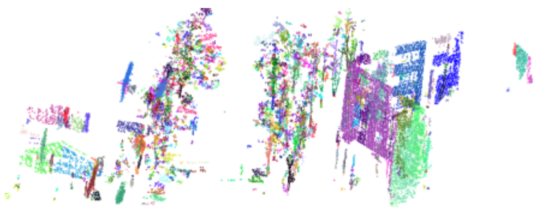
(a) Results of the segmentation applied to the vertically filtered data of the use case 2.1.4.



(b) Results of the segmentation applied to the vertically filtered data of the use case 2.1.2.



(c) Results of the segmentation applied to the vertically filtered data of the use case 2.1.1.



(d) Results of the segmentation applied to the vertically filtered data of the use case 2.1.3.

Figure 14.7: Results of the "*Normal Region Growing*" algorithm for walls detection in the application cases studied. A single color is assigned to each segment.

Detection of occlusion areas allows for establishing a topological connection link between the adjacent segments, as explained in Section 11.2.2. Moreover, the knowledge describing the segments stipulates that two segments are considered to belong to the same set of segments if they both have the same geometry and are connected by a topological link. Thus, the unification of the segments is carried out by logical reasoning on the knowledge of the segments.

This unification allows for the reconstruction of objects that have been segmented into several distinct segments. Figure 14.8 shows the rebuilding of a wall, which was initially segmented into several parts due to occlusion, in the case of an application 2.1.3.

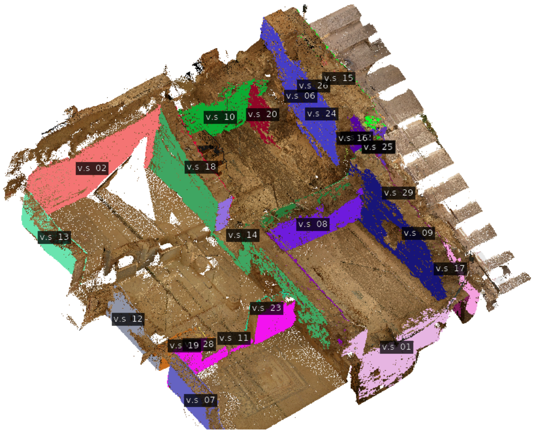
(a) Detection of a wall before the inference on the occluded areas.

(b) Detection of a wall after the inference that allows to unify segments.

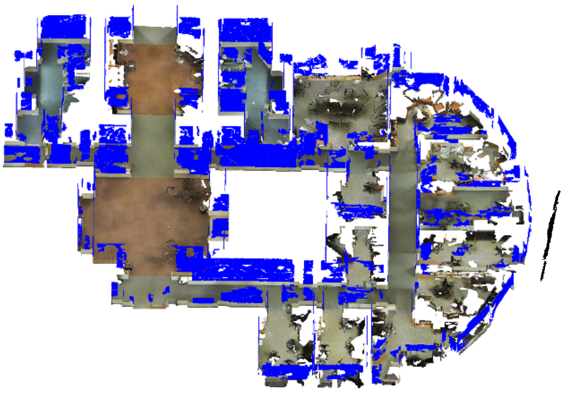
Figure 14.8: Detection of a wall from occluded areas for the applications case 2.1.3.

Similarly to the floor detection process, each algorithm execution generates an enrichment of knowledge from its results. This enrichment is followed by a classification that increases the data understanding at each step of the detection process. Thus, in the case of the wall detection process, the segments are first classified as vertical planes. After the characteristics and topological links enrichment of these vertical planes, those respecting the definition of walls as presented in Section 13.3 are identified as walls. Figure 14.9 shows the results of the wall detection in the four application cases considered.

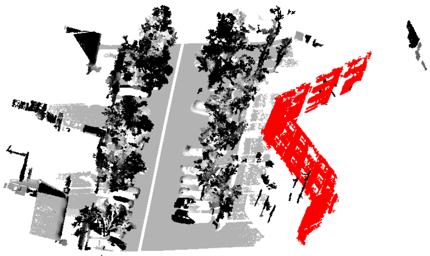
(a) Results of walls detection on the use case 2.1.4.



(b) Results of walls detection on the use case 2.1.2 A single color is assigned to each wall.



(c) Results of walls detection on the use case 2.1.1.

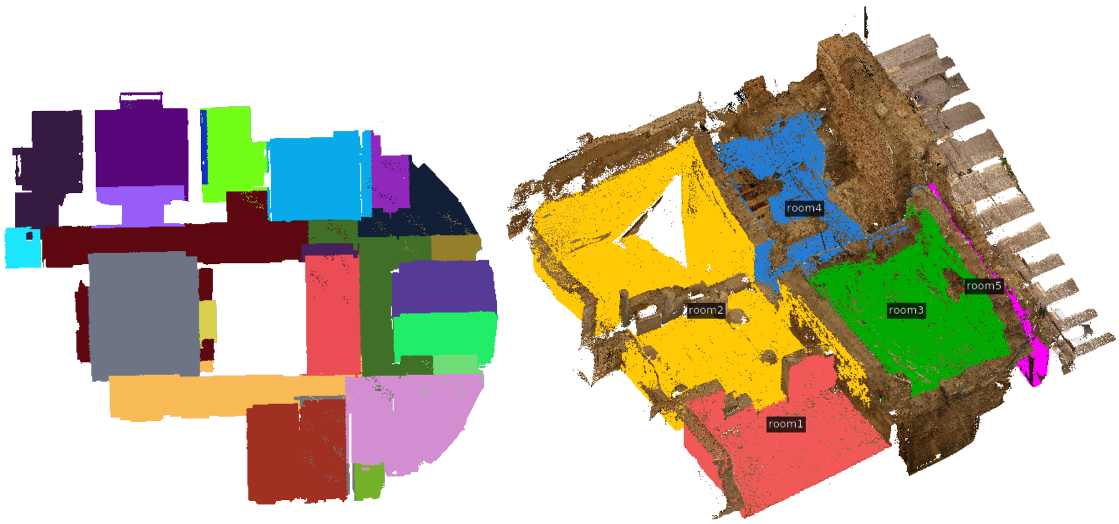


(d) Results of walls detection on the use case 2.1.3.

Figure 14.9: Results of walls detection on application cases studied.

Rooms classification

The application cases 2.1.2 and 2.1.1 aim at detecting rooms. Rooms are described as composed of walls and floors, as discussed in Section 10.3.2. More precisely, a room consists of a floor connected to at least three walls. Thus, after the classification of floors and walls, and the enrichment of their topological relationships, the rooms are automatically classified based on their detection of floors and walls. Figure 14.10 shows the results of the rooms classification on the application cases 2.1.1 and 2.1.2.



(a) Results of the rooms classification on the application case 2.1.1.

(b) Results of the rooms classification on the application case 2.1.2.

Figure 14.10: Results of the rooms classification.

Watermill classification in use case 2.1.2

Finally, in the application case 2.1.2, the room detection aims at detecting a water mill, which is composed of a large room and a small room whose dimensions are specified and that share the same wall. Figure 14.11 shows several representations of the watermill on the application case 2.1.2.

The semantic description of the watermill allows its identification in the data from the rooms detection and their relationships identification. Figure 14.12 shows the segmentation results of the watermill for the application case 2.1.2.

Street lights detection

Among objects on the floor, the application case 2.1.3 contains urban street lights. The street lights are described as objects that have a vertical line (as this corresponds to their main shape) and have a height between 4 m and 8 m. They are also

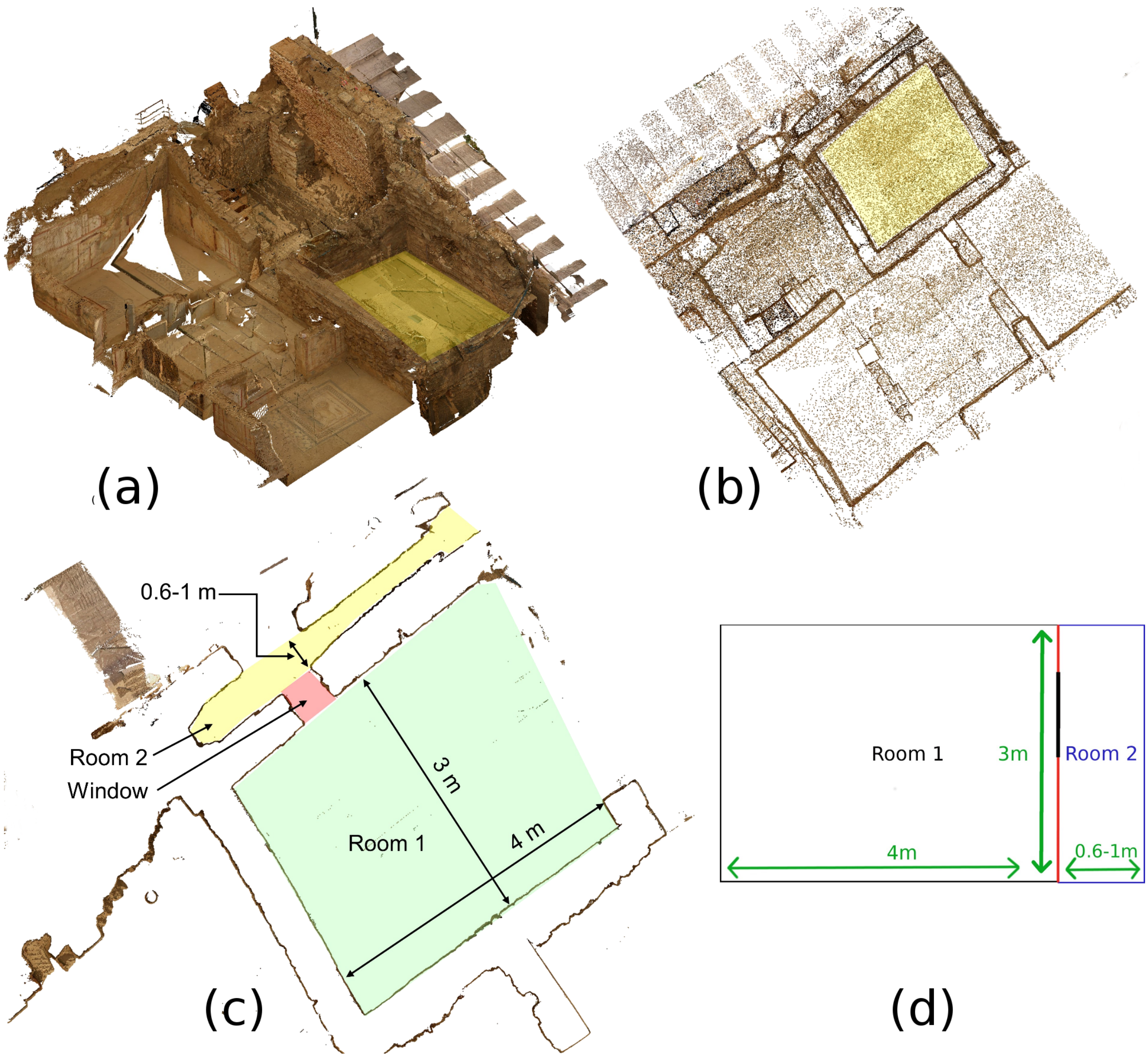


Figure 14.11: Descriptions of the watermill for the application case 2.1.2 (a) Point cloud with watermill (room illustrated in yellow); (b) Floor plot of the watermill; (c) Schematic geometric descriptions; (d) Schematic knowledge representation.

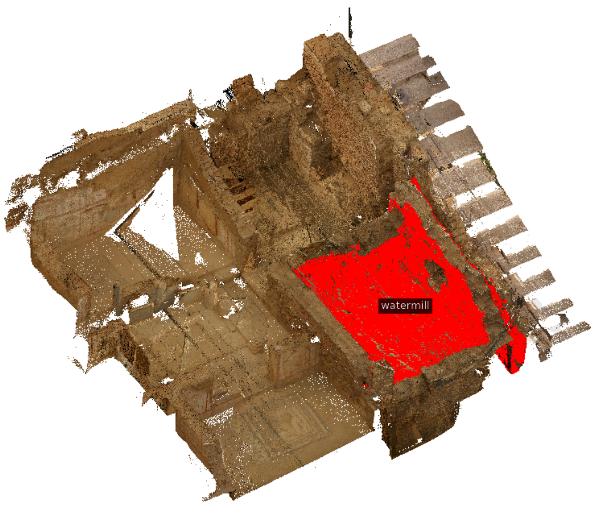
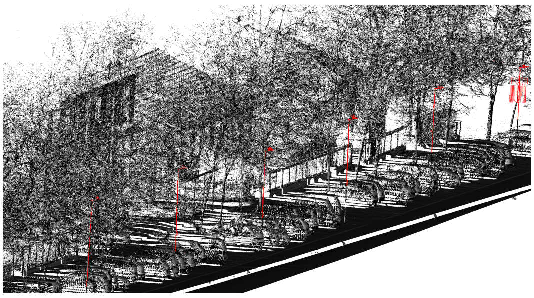


Figure 14.12: Results of watermill recognition (in red) on the application case 2.1.2.

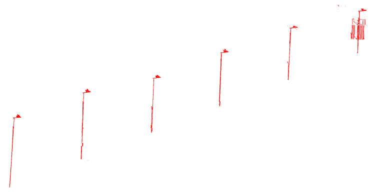
described as having a distance spacing of about 30 m (between 29m and 31m).

The segmentation step for the detection of the street lights is a Euclidean segmentation followed by a RANSAC line recognition algorithm, applied to results of the segmentation.

Then, feature extraction algorithms such as *getHeight* and *getDistance* are executed to characterize the vertical line segments. Several segments have a height corresponding to the height of an urban street light and have as geometry a vertical line. However, only some of these segments are spaced about 30 m from each other and are thus, identified as urban street lights. Figure 14.13 shows the results of the urban street lights detection for the application case 2.1.3.



(a) Global view of the results of the urban street lights detection (in red).



(b) Isolate view of the results of the urban street lights detection (in red).

Figure 14.13: Results of the urban street lights detection (in red) on the application case 2.1.3.

Tables detection

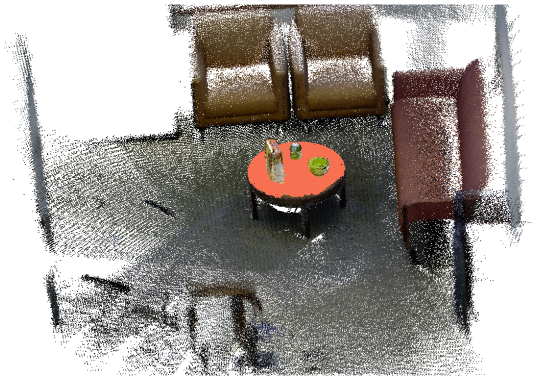
Among objects on the floor in the application case 2.1.4, tables are themself required for the detection of kitchen elements. A table is semantically described as composed of a tray, which is a horizontal plane at a distance between 50 cm and 70 cm from the floor and having an area superior to 0.20 m^2.

Therefore, the process selects firstly, normal filtering, and normal region growing as suitable for the detection of each table. These both algorithms requiring the same configuration than the configuration used for floor detection, these algorithms with such configurations are yet executed and do not need to be re-executed. Their results can be directly used for the continuation of the table detection. Then, the process selects Ransac plane recognition is suitable for table detection. This one is executed on segments above the floor resulting from the normal region growing configurated for horizontal plane detection.

Finally, features as area (through *getArea*) and the distance (through *getDistance*), whose the distance to the floor are estimated for each segments resulting from the Ransac plane recognition algorithm. Thanks to these data processing and knowledge that it provides, the process classifies the segments corresponding to the description of a table.

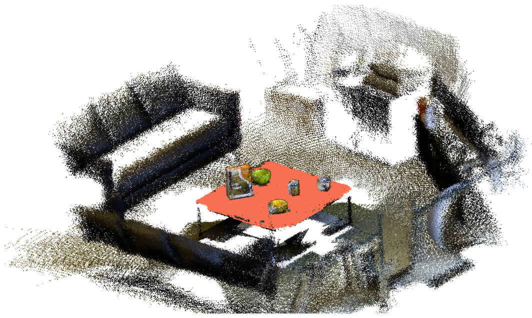
Figure 14.14 shows the results of table detection in point clouds acquired by Kinect (c.f. use case in Section 2.1.4).

(a) Point cloud of the first scene.



(b) Point cloud of the second scene.

(c) Point cloud of the third scene.



(d) Point cloud on the fourth scene.

Figure 14.14: Tables detection in point clouds of the use case 2.1.4.

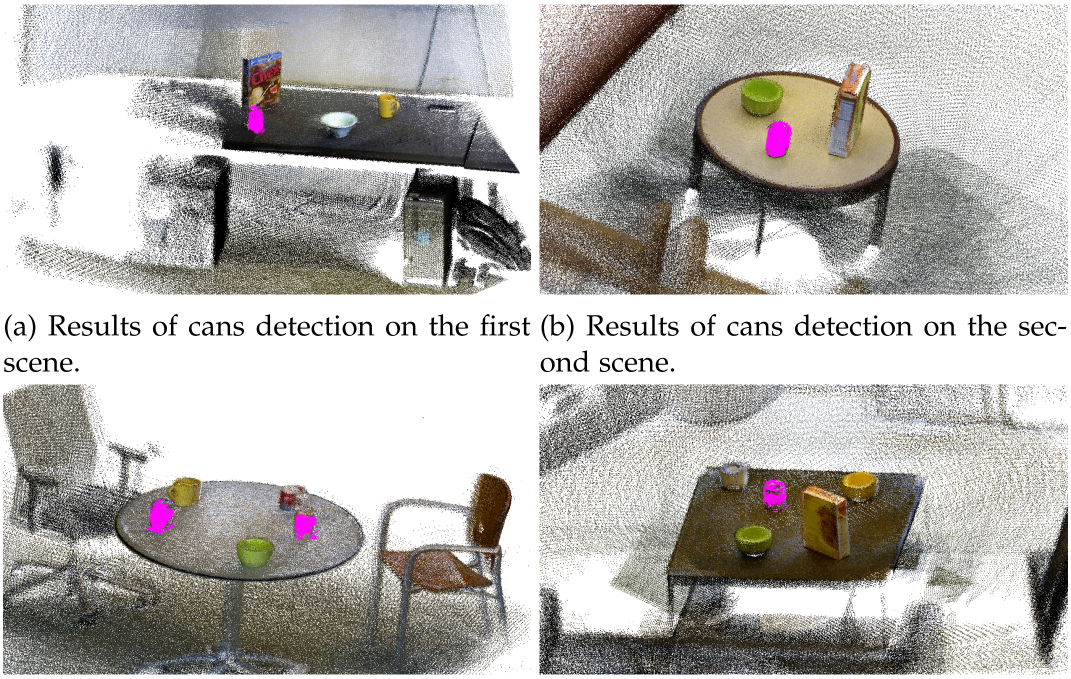
14.1.3 Detection of objects on tables

In the application case 2.1.4, there are kitchen elements as bowls, cans, cups, and cereal boxes that are on tables. Therefore, their detection is facilitated by table detection. Let us take the example of the can detection to illustrate the impact of table detection on the kitchen elements detection process. Firstly, a set of points is created by retrieving all points above the table.

Then, a Euclidean segmentation algorithm is selected. Its parameter of distance tolerance is configurated according to the point density and the size (height, length, and width) of the object. This segmentation is followed by the execution of Ransac cylinder recognition algorithm.

Finally, the width (through *getWidth*), the height (through *getHeight*), and the length (through *getLength*) are estimated for each cylinder. Thanks to the identification of cylinders and their characterization, cylinders respecting the can description are classified as can.

Figure 14.15 shows the results of the cans detection obtained from the four point clouds of the application case presented in Section 2.1.4.



(a) Results of cans detection on the first scene.

(b) Results of cans detection on the second scene.

(c) Results of cans detection on the third scene.

(d) Results of cans detection on the fourth scene.

Figure 14.15: Results of cans detection on the scenes of the application case 2.1.4.

14.1.4 Discussion

The detection process considers all objects to be detected in the processed data and uses their topological relationships to determine a logical order of detection between the objects. The knowledge of the objects and data processed allows the algorithms to be selected and configured according to their characteristics to execute them. The execution of an algorithm enriches the knowledge base based on these results. The reasoning based on this enrichment classifies the information coming from the data processing. Data processing, followed by knowledge base enrichment and classification provides a step-by-step understanding of the data. The data processing takes into account this understanding. It allows reducing the search areas of some objects. It uses the results of the previous executions to continue the detection process for each of the objects. Thus it allows minimizing the processing by not executing algorithms that would provide a result already obtained. Thus the detection process is tailor-made according to the knowledge of the data processed, the objects to be detected, and the processing performed. However, the knowledge used is limited to known factors and information. As a result, unknown factors and missing information can create unpredictable situations that make it difficult to understand the data. The proposed approach is composed of a self-learning process step based on the knowledge acquired during the first detection process to overcome this lack of knowledge. The self-learning process aims at enriching the knowledge base by a more precise knowledge of each application case to re-execute the detection process. Its objective is to improve the detection process by a more precise knowledge. Next section presents this self-learning process.

14.2 Knowledge-based self-learning process

Following the classification step, some objects have been identified. The analysis of these identified objects by the Knowledge-based self-learning process (explained in Section 11.3) allows improving the characteristics of the objects to identify them better.

This analysis consists first of all in enriching the information on the objects detected by using algorithms designed for the extraction of characteristic (e.g. size, shape, orientation, volume, area).

In the case of application 2.1.1 and 2.1.2, the enrichment of the characteristics allows to establish parallelism links between the walls and calculate the distance of each wall from the other.

The self-learning process then brings together recurring and common characteristics of a set of objects of the same type to formulate hypotheses.

Thus in the case of applications, 2.1.1 and 2.1.2, groups of walls have as recurring characteristics to have the same length or width, being parallel to each other and being connected to the same wall. The hypothesis shows in Equation 14.1, represented as a rule of inference, is therefore automatically formulated based on these characteristics.

$$
\begin{gathered}
Segment(?s) \wedge Wall(?w) \wedge areParallel(?s, ?w) \wedge (\\
(hasLenght(?s, ?l) \wedge hasLenght(?w, ?l2) \wedge equals(?l, ?l2)) \vee \\
(hasWidth(?s, ?wd) \wedge hasWidth(?w, ?wd2) \wedge equals(?wd, ?wd2))) \wedge \\
Wall(?w2) \wedge arePerpendicular(?s, ?w2) \wedge arePerpendicular(?w, ?w2) \Rightarrow Wall(?s)
\end{gathered} \tag{14.1}
$$

The new rule defines that if a segment ("*?s*") is parallel to a wall ("*?w*") and both have the same length ("*?l*" and *?l2*") or the same width ("*?wd*" and "*?wd2*"), and that a same wall ("*?w2*") is perpendicular with both elements (the segment "*?s*" and the wall "*?w*") then the segment is a wall.

This hypothesis is then validated automatically if its application does not produce any inconsistency in the knowledge (segment identified as corresponding to two different types of an object). If it is validated, then the knowledge it produces enriches the knowledge base. This enrichment of knowledge about objects impacts the entire detection process.

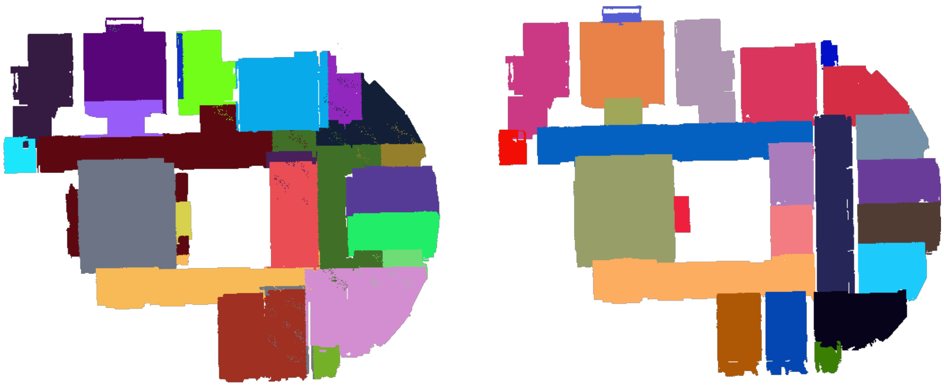
Hypothesis formulation allows objects to be detected even when their geometric representation in the data differs significantly from the geometry expected and defined in the knowledge.

In the case of application 2.1.1, several hypotheses in addition to the hypothesis illustrated in Equation 14.1, allow improving the classification of objects. First, in this application case, every ceiling is upper than 2.15 m. Meanwhile, all other objects defined in the knowledge base are lower than this value. Therefore, a new hypothesis is formulated and specify that every element upper than 2.15 m in the data of the case studied 2.1.1 is a ceiling. The same type of analysis is also applied for the ground description. The description of grounds, ceilings, and walls are thus improved.

Figure 14.16 shows the results of rooms detection before (a) and after (b) the iteration of new knowledge provided by such hypotheses.

The analysis process of the learning process enables to adapt the description of objects to the data dynamically and thus, significantly improves the object detection process. The rooms are better-segmented thanks to the improved knowledge of the walls, grounds, and ceilings. The relationships that unify them as rooms are also better controlled.

The higher the recurrence of the objects to be detected (number of objects of the same type) for an application case is, the more effective and relevant the learning



(a) Results of rooms detection before the learning. (b) Results of rooms detection after the learning.

Figure 14.16: Results of rooms detection on the application case 2.1.1.

is.

For example, in the application case 2.1.4, only one representation of each object type is defined. Thus there is no learning possible on the specific object types. However, general knowledge can be deduced by some hypothesis. For example, in this application case, all objects that have a small volume (less than $0.3m^3$) are on a table. Thus the assumption that "if an object has a small volume, it is placed on a table", can be applied to all application cases that are defined as similar to the application case 2.1.4. Thus this hypothesis allows faster detection of these objects in the other application cases.

Unlike the application case 2.1.4, the application case 2.1.3 object to detect present high recurrences, in particular for different types of objects such as cars, trees, street lights. Thus, the formulation of hypotheses is varied, and their validation is reinforced by the multiple situations tested.

Among the different hypotheses formulated, the hypothesis discussed earlier in Section 11.3 allows identifying cars (in Frauhhofer's dataset, c.f. Section 2.1.3) even if the segments that compose them are not representative of the geometry defined for cars. This assumption states that if a segment is close to a car and is aligned with a car, then this segment is a car.

Figure 14.17 shows the results of car detection before (a) and after (b) the iteration of new knowledge provided by such a hypothesis.

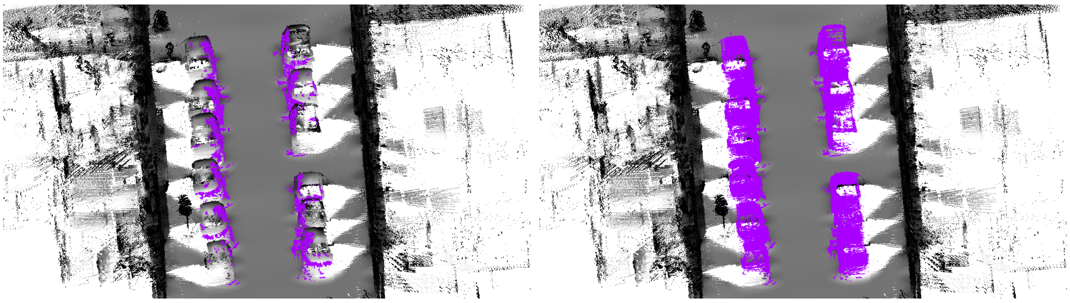
In the case of the application based on the Paris-Rue Madame dataset, the self-learning process has validated the following assumption: "If a segment is close to a car and does not belong to any other element then it is a car." The integration of this kind of assumption into the knowledge base has improved the accuracy of

(a) Results of cars detection before the self-learning process.

(b) Results of cars detection after the self-learning process.

Figure 14.17: Results of cars detection on the application case 2.1.3.

the cars detection process. Figure 14.18 illustrates the difference of accuracy for car detection on the Paris-Rue-Madame dataset before and after the self-learning process.



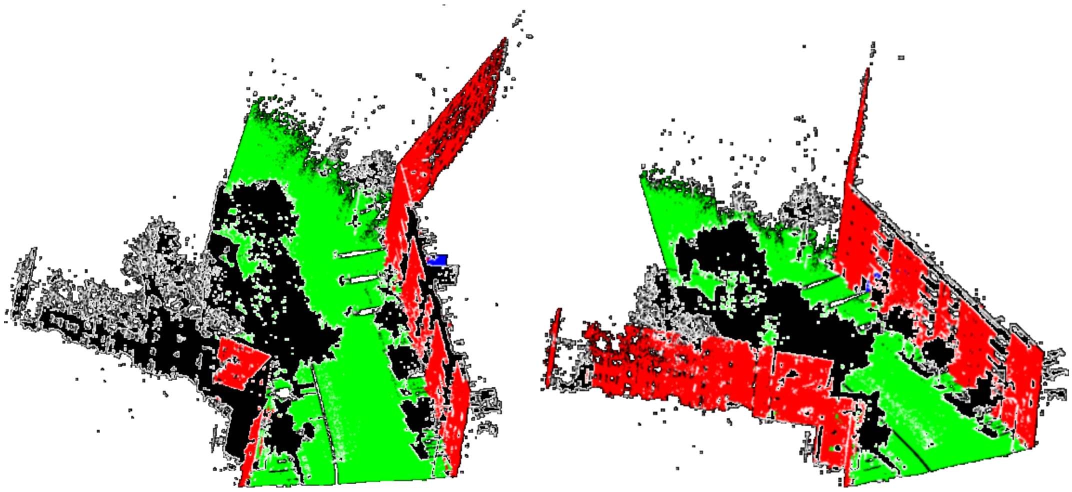
(a) Results of cars detection before the self-learning process.

(b) Results of cars detection after the self-learning process.

Figure 14.18: Comparison of cars detection results without and with the self-learning process on Paris-Rue Madame dataset.

Other hypotheses can only concern the geometry of an object. For example, facades could be defined as walls with a height greater than 13m in the case of application 2.1.3. However, facades were not identified because they are too far away and often isolated (no connection with objects). The characteristics common to the facades have been grouped during the analysis process (see Section 11.3.2) to detect these facades better. Grouping the common characteristics allows statistical analysis useful for hypothesis formulation, such as the calculation of confidence intervals for each value. In this example, the confidence interval was calculated for the value of the facade height. Its values are $I_c = [12.3; 15.6]$. This interval allows to formulate a hypothesis that a segment is a facade if its height is within this interval, i.e. between [12.3;15.6]. If a segment already classified as an object (such as a tree) has such a height, then the hypothesis would be invalidated and would be reformulated by integrating other characteristics (such as length, or width). Validation of this hypothesis allows for the adaptation of knowledge on façades and thus for better detection.

Figure 14.19 shows the results of facade detection before (a) and after (b) the iteration of new knowledge provided by such hypothesis.



(a) Results of facades detection before the self-learning process. (b) Results of facades detection after the self-learning process.

Figure 14.19: Results of facades detection on the application case 2.1.3.

As shown in Figure 14.19, all facades present in the case of application 2.1.3 are now detected.

Similarly to facades and cars, the self-learning process improves the semantic description of most of the objects described in a scene (such as vegetation, trees, pedestrians, walls, facades, street lights, traffic signs). The improvement of this knowledge leads to better detection of objects.

14.3 Discussion

The data processing presented in chapter 11 and illustrated in this chapter through the use cases consists of a detection process and knowledge-based self-Learning.
The detection process is built step by step according to the knowledge of the objects, data, and processing already performed. Its role is to adapt to each application case according to the explicitly defined knowledge of that application case. Its objective is to understand the data.
This understanding of the data, therefore, depends on explicitly defined knowledge. Inaccuracies or lack of information in the definition of this knowledge is often the limit to the performance of knowledge-based approaches. The role of

self-learning process is to overcome this limitation by enriching the knowledge base with knowledge specific to the addressed application case.

This learning requires an analytical basis on which to formulate hypotheses. The first execution of the detection process using the initial knowledge base provides a basis for understanding the data. This one is then enriched by an in-depth analysis of the objects detected through the extraction of characteristics and topological relationships between these objects. The initial knowledge base enriched by the detection process and an analysis of the detected objects provides the necessary support for the self-learning process. This self-learning process is based on the creation and verification of hypotheses. The validated assumptions are used to update the knowledge base according to the addressed application case.

The self-learning process thus provides a more specialized and accurate knowledge base for the addressed application case. This new, more specific knowledge base allows creating a detection process more adapted to the data processed and thus to improve the quality of its understanding. Next chapter presents an analysis of the results obtained for the considered application cases as well as an evaluation of the self-learning process effectiveness.

15 Results

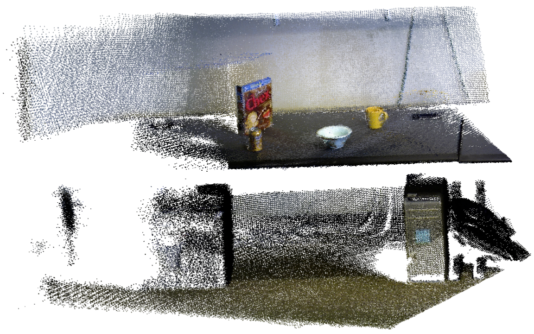
This chapter aims at presenting the results of the approach proposed in this thesis. A first section presents the results obtained for each application case presented in Section 2.1. A second section compares results obtained by this approach with other approaches. This comparison is applied through the indoor application case 2.1.1 and the outdoor application case 2.1.3.

15.1 Use cases results

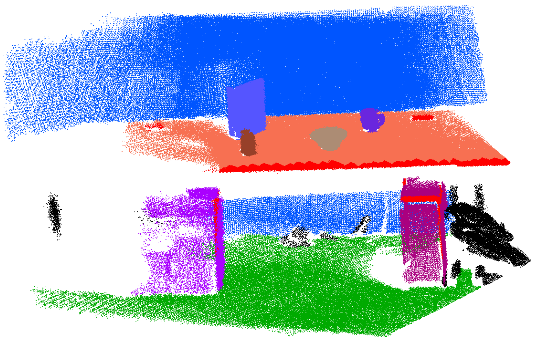
This section presents the results obtained for each of the four application cases studied. Their explanation begins with the type of application case and a description of objects searched in the data (means objects described semantically into the knowledge base). Then, results are presented firstly visually and then, through an assessment of the detection. Finally, the strength and weakness of the approach are discussed.

15.1.1 Results on time-of-flight use case

The use case 2.1.4 is a point cloud composed of four indoor scenes representing a workroom and three lounging rooms. Figure 15.1 presents for each of these scenes, its original part of the point cloud and the visual results of its understanding. In these scenes, three main categories of objects are described in the knowledge base. The first category of objects are objects defining a room as floors (in green in Figure 15.1) and walls (in blue in Figure 15.1). The second category of objects are desk furnitures as seats (in bright green in Figure 15.1), chairs (in dark green in Figure 15.1), tables (in orange in Figure 15.1), computer tower (in pink in Figure 15.1), and boxes (in bright purple in Figure 15.1). The last category of objects are kitchen elements, which are on tables, as cereal boxes (in blue-purple in Figure 15.1), cups (in purple in Figure 15.1), cans (in brown in Figure 15.1), and bowls (in grey in Figure 15.1). Not classified points, which do not belong to noise, appears in black in Figure 15.1), and also in red in Figure 15.1(b).

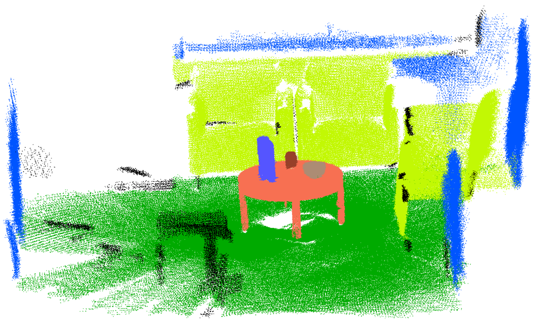


(a) Original point cloud of the first scene.



(b) Results on the first scene.

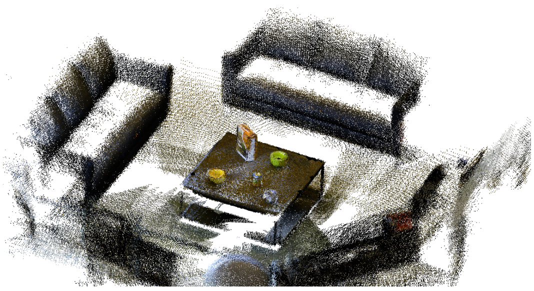
(c) Original point cloud of the second scene.



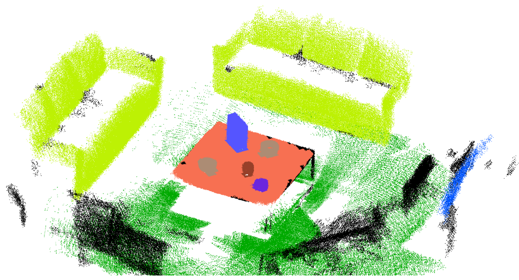
(d) Results on the second scene.

(e) Original point cloud of the third scene.

(f) Results on the third scene.



(g) Original point cloud of the fourth scene.



(h) Results on the fourth scene.

Figure 15.1: Results on the application case 2.1.4: each object classified as belonging to the same type are represented by the same color, elements colored in black and red are unclassified elements.

15.1. USE CASES RESULTS

Visually, two sources of unclassified points can be observed. Figure 15.1(b) highlights the two different sources of not classified points. The first source resulting in black points corresponds to segments that the approach has not classified due to a lack of knowledge description of these objects. For example, on the right of Figure 15.1(b), a backpack appears as a black shape. The ununderstanding of this segment is due to the absence of a semantic description of a backpack. The second source of classification lack that is highlighted in red corresponds to ambiguous segments, which could be classified into two object types, and are therefore not classified neither in both types to avoid inconsistency of the knowledge base, neither in one type to avoid arbitrary choice that conducts to wrong classification. This red segments can also be not classified because they do not fit with the semantic description. For example, a table tray is described semantically as a horizontal plane; however, in the example illustrated by Figure 15.2, the table has round edges. Therefore, the process detects the table by considering only the plane surface of the tray and not the round edges. This red segments can be considered as errors because they should be classified into an object type. Thus, these results illustrate that the approach produces "edge effects" in the data understanding process.

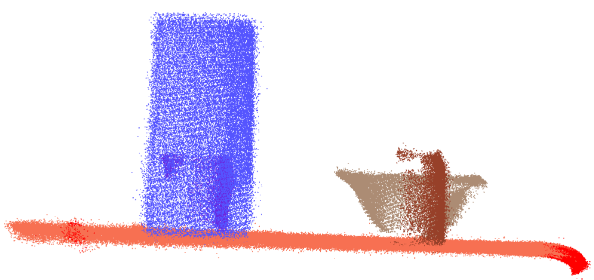


Figure 15.2: Illustration of round edge in red of the table.

Table 15.1 provides three information about the results. Firstly, it provides the number of objects classified according to objects present in the scene. Secondly, it gives the distribution of points between points representing noise, classified, and unclassified points. Finally, it shows an estimation of success and fails percentage according to the proportion of classified and unclassified points respectively, without considering points belonging to noise.

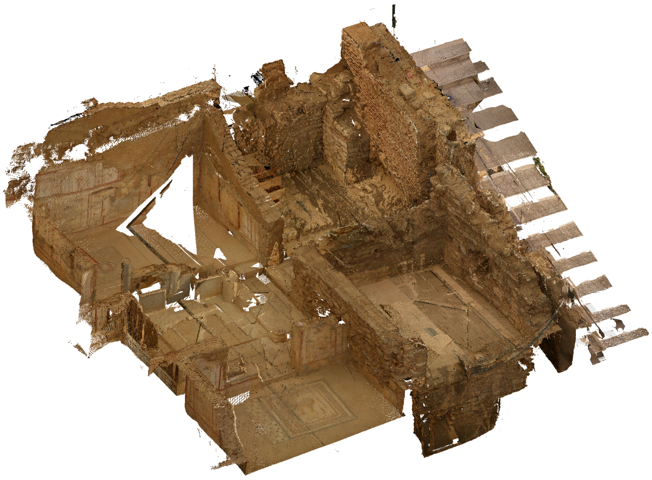
The number of classified objects shows that all considered objects (in a sense, semantically described in the knowledge base) have been detected in all scenes. The percentage of noisy points in each scene highlights variations of noise among the different scenes. Scene 4 is particularly noisy, with around 15% of noisy points. This noise can impact the result of the approach since the approach obtains its higher fail percentage in this scene. However, despite the noise, the approach succeeds to detect all objects of this scene, and this with an accuracy of 94% of points successfully classified. Moreover, the approach shows good robustness to the noise around 5%, by obtaining an accuracy of around 99% for the understanding of scene 2 and 3. Finally, the approach obtains an average accuracy of 97.66% for the understanding of this application case, and thus, a lack of 2.23% of data understanding.

Objects/Scene		Scene 1	Scene 2	Scene 3	Scene 4	All scene
Wall	present	1	4	1	1	7
	classified	1	4	1	1	7
Floor	present	1	1	1	1	4
	classified	1	1	1	1	4
Seat	present	0	3	0	2	5
	classified	0	3	0	2	5
Chair	present	0	0	2	0	2
	classified	0	0	2	0	2
Table	present	1	1	1	1	4
	classified	1	1	1	1	4
Computer	present	1	0	0	0	1
	classified	1	0	0	0	1
Box	present	1	0	0	0	1
	classified	1	0	0	0	1
Cereal box	present	1	1	0	1	3
	classified	1	1	0	1	3
Cup	present	1	0	1	1	3
	classified	1	0	1	1	3
Can	present	1	1	2	1	5
	classified	1	1	2	1	5
Bowl	present	1	1	2	2	6
	classified	1	1	2	2	6
Total	present	9	12	10	10	41
	classified	9	12	10	10	41
Percentage of noise		0.57%	5.43%	4.90%	14.98%	6.47%
Percentage of classified points		97.12%	93.50%	94.91%	80.16%	91.42%
Success (without considering noise)		97.67%	98.87%	99.8%	94.3%	**97.66%**
Percentage of not classified points		2.31%	1.07%	0.19%	4.86%	2.11%
Fail (without considering noise)		2.33%	1.13%	0.2%	5.7%	**2.34%**

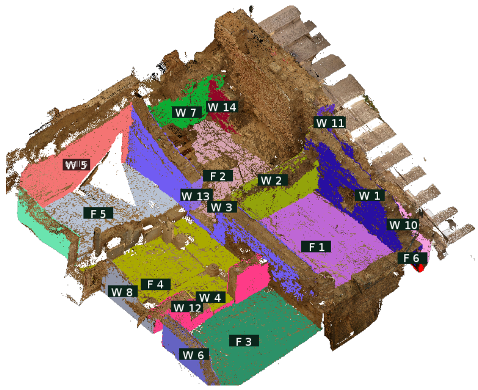
Table 15.1: Results of objects detection in scenes of the use case 2.1.4.

15.1.2 Results on cultural heritage use case

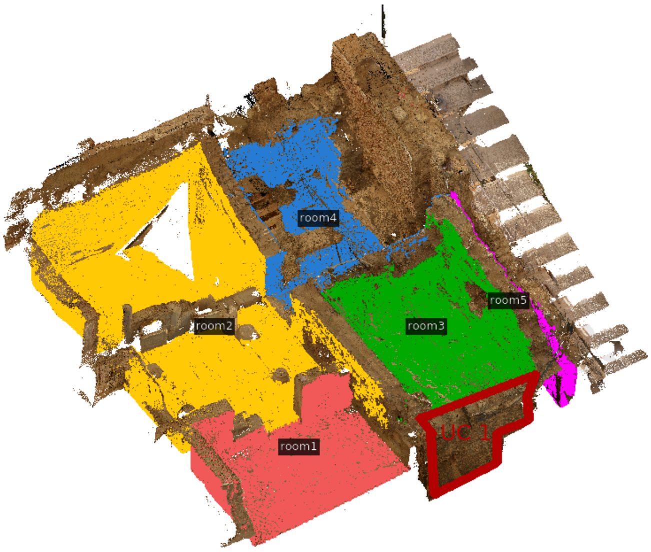
The application case 2.1.2 is a point cloud coming from a context of ruin excavation. The processing of this application case aims at detecting a watermill as described in Section 2.1.2. This watermill is semantically described as a composition of two rooms, which are themselves composed of a floor and walls. Figure 15.3 shows the detection results obtained for this application case.



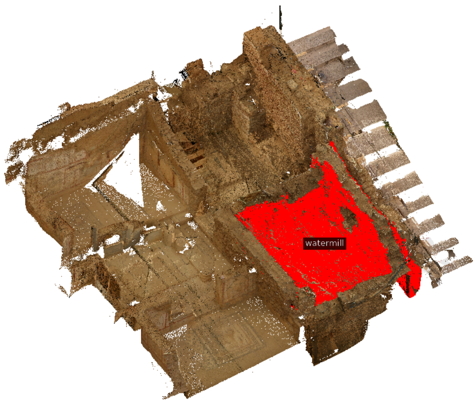
(a) Original point cloud.



(b) Results of walls and floors detection.



(c) Results of rooms detection.



(d) Results of the watermill detection.

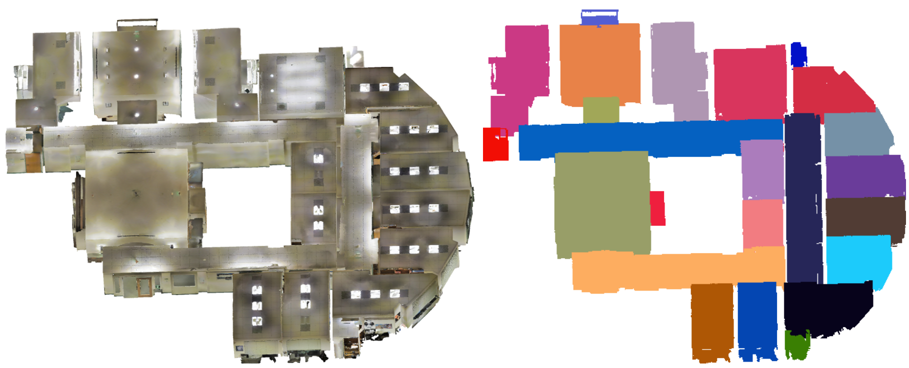
Figure 15.3: Results of objects detection on the application case 2.1.2.

One unclassified wall (UC 1 area on the bottom right part in Figure 15.3(c)) is due to the vertical position of the wall that is partially under the floor and thus not fulfilling the topological relationship with the floor and with other walls. However, the missed classification of this wall does not impact the correct classification of the room related to this wall thanks to the flexibility of the semantics that combined different kinds of information provided by the data description, and thus all rooms are correctly classified. Thanks to the proper detection of each room, the semantic classification identifies the watermill.

The main benefit of the proposed approach on this application case is the adaptation of general knowledge to the complex context of a ruin excavation. This case shows that the proposed approach is able to detect a wall that has been partially destroyed. Although only a majority of data points have been labeled with an object name, the results show a proper detection of objects. This fact highlights a limit in the detection accuracy. Despite the significant irregularity of shape, the omission of a part of the 3D data during processing and analysis still enables the recognition of the semantically defined objects. The proposed approach, therefore, provides the necessary flexibility needed, especially for cultural heritage objects. On the one hand, similar cultural heritage objects varying in shape and arrangement could also be automatically recognized across varying data sets as soon as a sound semantic description exists. On the other hand, existing semantic descriptions of cultural heritage objects could be adapted or reused for the recognition of other, more complex objects.

15.1.3 Results on indoor modern building use case

The proposed approach firstly detects and classifies walls, ceilings, and grounds. Secondly, the approach builds rooms according to the links between walls, ceilings, and grounds. Figure 15.4 shows the detection results obtained with the application case 2.1.1.



(a) Original point cloud. (b) Results of rooms detection.

Figure 15.4: Results on the application case 2.1.1.

In this application case, all expected rooms are detected. Nevertheless, some parts of rooms (small parts in *room 13* as illustrated in Figure 15.11) are not classified due to their specific characteristics. As illustrated in Figure 15.5 (in red), two exterior elements are associated to the *room 13* in Figure 15.11.

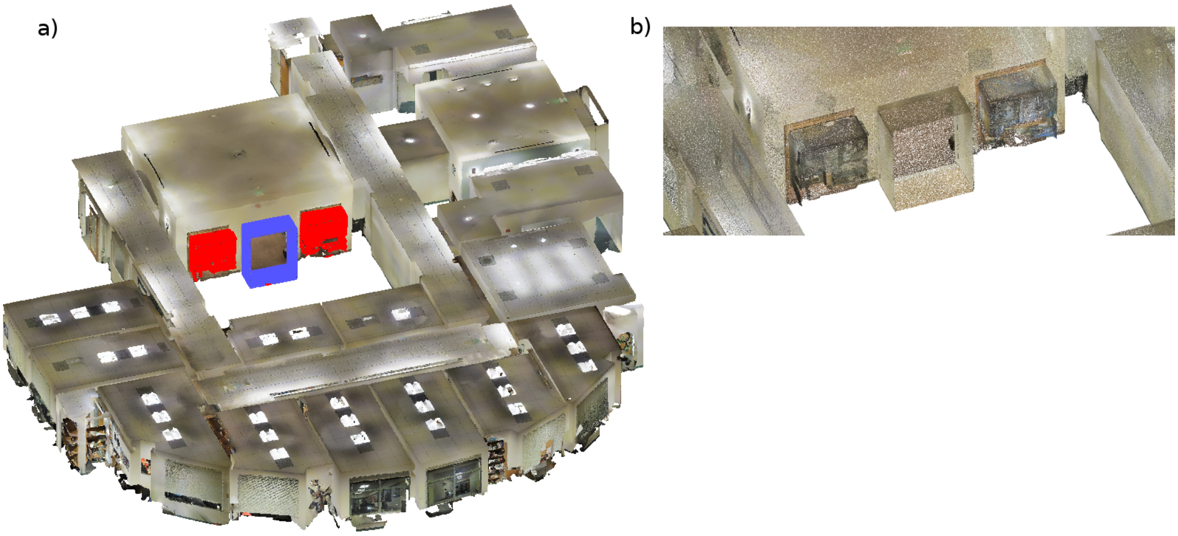


Figure 15.5: a) Illustration of rooms parts not classified in red and a wrong classified in purple, b) Zoom on the representation of these three room parts.

Due to their different features (e.g. ceiling not at the same height than the room), these elements are not correctly classified. An addition of knowledge about this type of elements would solve this classification problem. In this application case, the classification of these elements is considered as not relevant since they do not impact the overall classification of rooms and semantically, these elements are not themselves a room and are outside all detected rooms. It is thus logical to not detect as a part of the room. On the other hand, the center element (illustrated in purple in Figure 15.5) classified as another room has the same characteristics than *room 1* shown in Figure 15.11. Thus it is correct to detect it as another room (see Figure 15.11(d)).

However, the reasoning process considers that all points classified in more than one room are inconsistent. Thus, all points at the border of two rooms cannot be classified in any room (illustrated by the white gap between rooms in Figure 15.11 (d)) that produces "edge effects" in the detection results. Indeed, the decision to classify these points depends on the goal of the application case. Considering the flexibility and the automatic adaptation of the knowledge module, the addition of further knowledge about building elements could improve the detection process already implemented.

To go further in the analysis of the results, an estimation of the detection quality has been computed for this application case according to the metric of *Recall*, *Precision*, and *F1-score* (as explained in [*Zheng et al.*, 2019]):

- Recall: represents the proportion of points considered as negatively well classified. It is computed from the number of points similarly classified in the assessed set and the reference set divided by the number of all points from the assessed set.

- Precision: represents the proportion of points considered as positively well classified. It is computed from the number of points similarly classified in the assessed set and the reference set divided by the number of all points from the reference set.

- F1-score: represents the harmonic average between the precision and recall (with the best value at 1, and the worst at 0). It is computed from the precision and the recall scores.

Figure 15.6 illustrates the point cloud used as reference for the computation of metric scores.

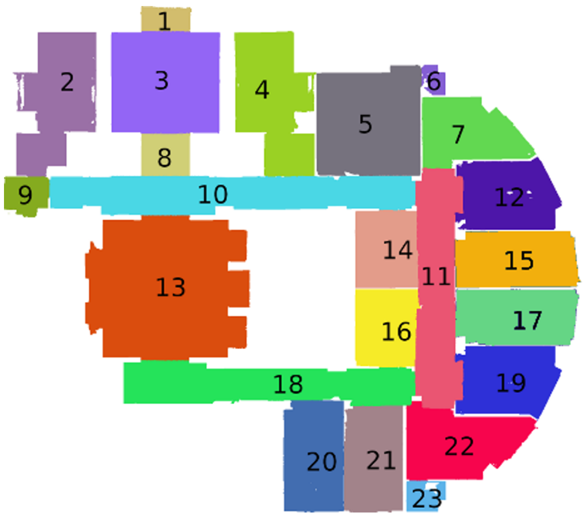


Figure 15.6: Ground truth of the point cloud used as reference.

Table 15.2 presents the three metric scores for each room represented in the point cloud.

This metrics assessment shows that the proposed approach has a good recall globally with an average of 0.889, but a medium precision with an average of 0.633. This proposed approach obtains a correct average F1-score of 0.722. Although a medium precision, the proposed approach has the advantage to detect all the rooms automatically.

Room	Recall	Precision	F1-score
Room 1	0.602	0.843	0.702
Room 2	0.964	0.820	0.886
Room 3	0.940	0.820	0.875
Room 4	0.925	0.722	0.811
Room 5	0.915	0.503	0.649
Room 6	0.974	0.707	0.819
Room 7	0.706	0.591	0.643
Room 8	0.887	0.779	0.829
Room 9	0.963	0.453	0.616
Room 10	0.898	0.768	0.828
Room 11	0.634	0.840	0.723
Room 12	0.976	0.542	0.697
Room 13	0.958	0.666	0.785
Room 14	0.639	0.545	0.588
Room 15	0.970	0.434	0.600
Room 16	0.956	0.464	0.624
Room 17	0.921	0.594	0.722
Room 18	0.768	0.819	0.793
Room 19	0.980	0.565	0.717
Room 20	0.968	0.520	0.677
Room 21	0.965	0.461	0.624
Room 22	0.962	0.576	0.720
Room 23	0.985	0.527	0.687
Average	0.889	0.633	0.722

Table 15.2: Quality of room detection in use case 2.1.1.

15.1.4 Results on outdoor urban use cases

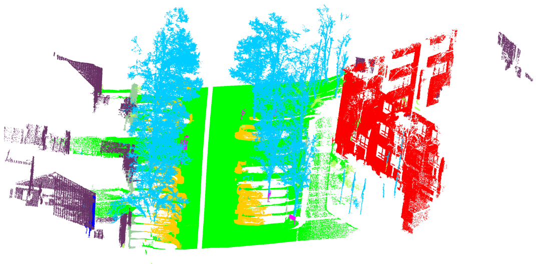
The application case of the urban outdoor environment is composed of two test point clouds acquired by laser-scanner: the point cloud from Fraunhofer GmbH and the point cloud from "Paris-rue-Madame database: MINES ParisTech 3D mobile laser scanner dataset from Madame street in Paris"[1] ([*Serna et al.*, 2014]). Objects to detect in this application case are mainly cars, traffic signs, walls (sometimes also called facade in this context), floor, and vegetation.

[1]MINES ParisTech© copyright. MINES ParisTech created this special set of 3D MLS data for the purpose of detection-segmentation-classification research activities, but does not endorse the way they are used in this project or the conclusions put forward.

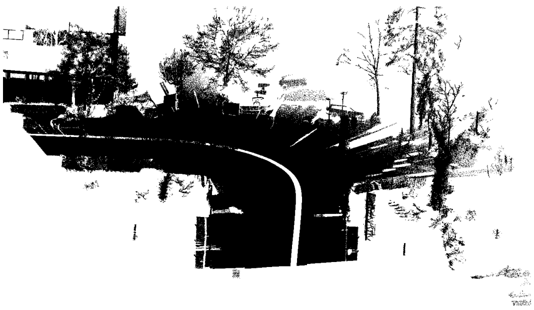
Frauhhofer's point cloud

Figure 15.7 shows the results of the detection applied to two parts of Frauhhofer's point cloud.

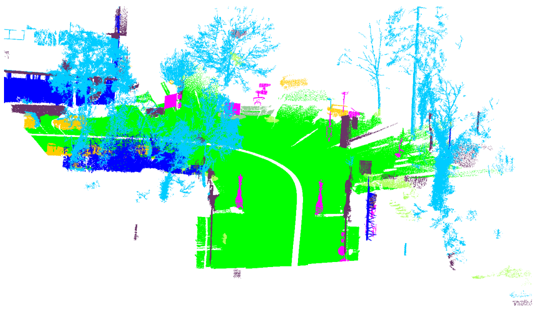
(a) First part of original point cloud from Fraunhofer GmbH.



(b) Results of detection on the first part.



(c) Second part of original point cloud from Fraunhofer GmbH.



(d) Results of detection on the second part.

Figure 15.7: Results on the application case 2.1.3: facades in red, walls in dark blue, building elements in purple, the floor in bright green, cars in yellow, traffic signs in magenta, and trees in bright blue.

The application of the proposed approach on this point cloud allows for detecting a wide variety of different objects regardless of their size (such as very large walls or smaller traffic signs), geometric complexity (such as cars) or different representations (such as trees). This capability to detect such a variety of objects shows the flexibility of the approach.

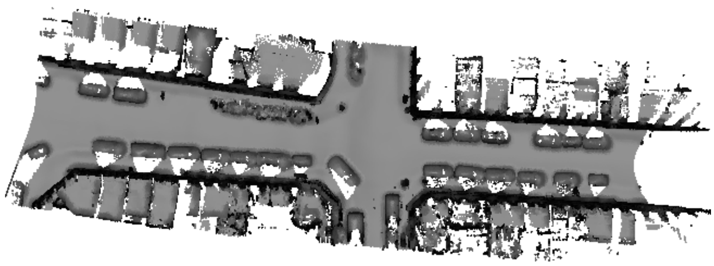
Moreover, this application case contains several challenging situations of detection as a divergence of the object representations inside the data (e.g. cars and trees), objects occluded by others or having a low point density due to the acquisition process (scanner too far from the object for example). Yet, even in these challenging situations, objects are correctly detected that shows the robustness and efficiency of the proposed approach.

Nevertheless, 3.62% of the data is still not understood. The reasons for this lack are similar than for the previous use case (e.g. ambiguity limits, object representation too far from their semantic description or absence of semantic description).

To assess more accurately, the precision and the efficiency of the proposed approach, the next results on Paris-rue-Madame dataset in a similar application case are assessed through metric values.

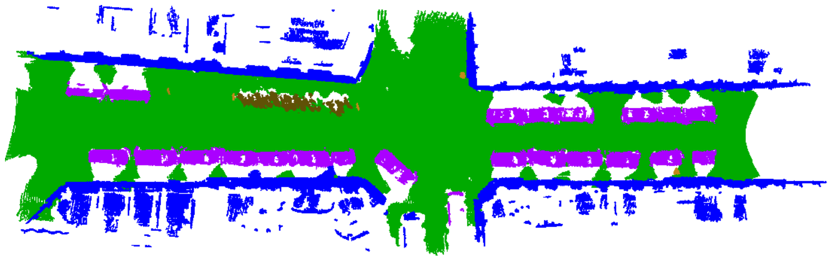
Paris-rue-Madame dataset

Figure 15.8 shows the results of the detection applied to "Paris-rue-Madame database: MINES ParisTech 3D mobile laser scanner dataset from Madame street in Paris"[2] ([*Serna et al.*, 2014]).



(a) Original point cloud from Paris-rue-Madame dataset, MINES ParisTech.[*Serna et al.*, 2014].

(b) Annotated dataset used as the reference for the computation of metrics [*Serna et al.*, 2014].



(c) Results of detection.

Figure 15.8: Results on the application case 2.1.3 from "Paris-rue-Madame database: MINES ParisTech 3D mobile laser scanner dataset from Madame street in Paris".

From visual observations, similar conclusions than that obtained from the previous application on Fraunhofer's point cloud can be deduced. That is why to go fur-

[2]MINES ParisTech© copyright. MINES ParisTech created this special set of 3D MLS data for the purpose of detection-segmentation-classification research activities, but does not endorse the way they are used in this project or the conclusions put forward.

ther in the analysis of the results, an estimation of the detection quality has been computed for this application case according to the metric of *Recall*, *Precision*, and *F1-score* (as explained in Section 15.1.3).

Table 15.3 shows the metric values obtained for these three metrics and for each of the following object types: Facade, Ground, Car, Motorcycle, and Traffic sign.

Metrics/Objects	Facade	Ground	Car	Motorcycle	Traffic sign
Recall	0.923	0.9978	0.935	0.737	0.8806
Precision	0.834	0.964	0.94	0.6405	0.7253
F1-score	0.8765	0.9806	0.939	0.685	0.7955

Table 15.3: Results of metric values obtained for each considered object type.

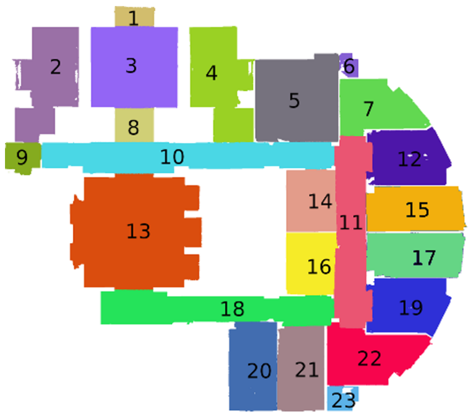
These metric values show that the approach is globally efficient and precise with an average *F1-score* of 0.8552, an average precision value of 0.82076 and an average recall value of 0.89468. The most efficiency and precision of the approach is obtained at first for the object ground, and then for the object car. The efficiency in car detection shows the robustness of the approach for managing the detection of objects having a diversity of representation in the data. The object facade has efficiency and precision near to the average efficiency and precision of the approach. However, a loss of efficiency and precision is observed for smaller objects as motorcycle and traffic sign. This loss of efficiency is due to the smallest objects provide a lower quantity of information that makes their detection more challenging than big objects as the ground. Moreover, the comparison of these results with other approaches in Section 15.2.2 will show these results are good results for such small objects since the proposed approach obtains the better results in their detection.

15.1.5 Discussion on the efficiency of the self-learning process

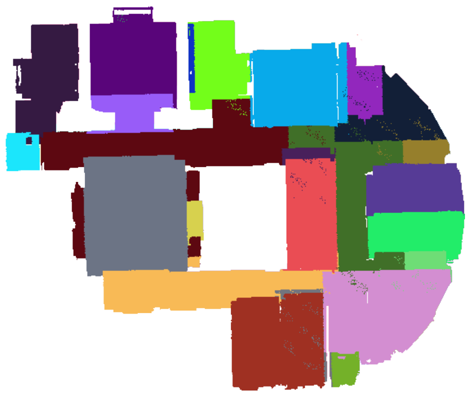
The results obtained by the proposed approach on the different use cases highlight its ability to provide good detection results for different objects and in different contexts. This approach uses explicit knowledge to guide the detection process, as well as a self-learning process. The learning objective is to improve the knowledge according to each application case so that it is more precise. Obtaining accurate knowledge is intended to improve the detection process. As it is guided by knowledge, it is directly impacted by its accuracy. A comparison of results obtained before and after the self-learning process was carried out. This comparison aims at estimating the benefit of this learning. This comparison was made on the two application cases for which the metrics Recall, Precision, and F1-score were evaluated. The application cases are the application cases on Stanford (c.f. Section 2.1.1) and Paris-Rue-Madame dataset (c.f. Section 2.1.3).

Efficiency of the self-learning process applied to Stanford dataset

Figure 15.9 presents the results obtained on Stanford dataset before and after the self-learning process.



(a) Ground truth point cloud.



(b) Results obtained before the self-learning process.

(c) Results obtained after the self-learning process: green circles highlight the data understanding improvement and orange circles highlight the loss of data understanding.

Figure 15.9: Results of detection on Stanford dataset.

This Figure highlights the improvements (c.f. green circles in Figure (c)) and losses (c.f. orange circles in Figure (c)) generated by the self-learning process. The improvements have resolved the problems of non-detection on the one hand, and misclassification on the other hand. The losses related to this process correspond to an accuracy loss for some rooms, whose some points have not been classified. The quality difference of results obtained by the detection processes before and after the self-learning process has been evaluated according to the recall, precision, and F1-score metrics. Table 15.4 presents the three metric scores before and after the self-learning process execution for each room of the dataset.

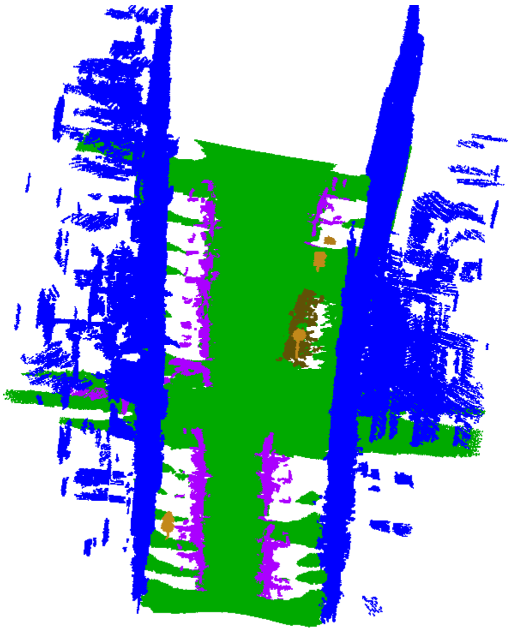
Room	Recall		Precision		F1-score	
	Before	After	Before	After	Before	After
Room 1	0.100	0.602	0.838	0.843	0.179	0.702
Room 2	0.989	0.964	0.815	0.820	0.894	0.886
Room 3	0.888	0.940	0.757	0.820	0.817	0.875
Room 4	0.985	0.925	0.517	0.722	0.678	0.811
Room 5	0.816	0.915	0.798	0.503	0.807	0.649
Room 6	0.187	0.974	0.559	0.707	0.280	0.819
Room 7	0.395	0.706	0.376	0.591	0.385	0.643
Room 8	0.489	0.887	0.845	0.779	0.619	0.829
Room 9	0.977	0.963	0.696	0.453	0.813	0.616
Room 10	0.664	0.898	0.588	0.768	0.624	0.828
Room 11	0.519	0.634	0.606	0.840	0.559	0.723
Room 12	0.984	0.976	0.226	0.542	0.367	0.697
Room 13	0.888	0.958	0.672	0.666	0.765	0.785
Room 14	0.505	0.639	0.122	0.545	0.197	0.588
Room 15	0.825	0.970	0.813	0.434	0.819	0.600
Room 16	0.498	0.956	0.763	0.464	0.603	0.624
Room 17	0.907	0.921	0.636	0.594	0.748	0.722
Room 18	0.932	0.768	0.640	0.819	0.760	0.793
Room 19	0.983	0.980	0.165	0.565	0.282	0.717
Room 20	0.535	0.968	0.798	0.520	0.641	0.677
Room 21	0.455	0.965	0.697	0.461	0.551	0.624
Room 22	0.564	0.962	0.837	0.576	0.674	0.720
Room 23	0.996	0.985	0.713	0.527	0.831	0.687
Average	0.699	0.889	0.629	0.633	0.605	0.722

Table 15.4: Comparison of metrics obtained before and after the self-learning process on Stanford dataset.

The average evaluation of the metrics showed little overall improvement in accuracy (with an increase of only 0.004), but slightly greater improvement in recall with an increase of 0.190. The self-learning process thus obtains an overall average increase of 0.117 on its F1-score. The room-by-room observation of the results shows a significant increase (about 0.5) in the F1-score for rooms 1, 6, 14, and 19. Rooms 4, 7, 8, 10, 11, 12, and 21 also benefit from an increase (about 0.2) in the F1-score. The self-learning process has little impact on rooms 2, 3, 13, 16, 17, 18, 20, and 22. Rooms 5, 9, 15, and 23 suffer a loss of about 0.2 of the F1-score. This decrease in quality is due to a loss of accuracy. However, this loss of accuracy on these four rooms remains low compared to the benefits provided on the eleven parts that obtained a quality increase between 0.2 and 0.6.

Efficiency of the self-learning process applied to Paris-Rue-Madame dataset

Figure 15.10 shows the results obtained on Paris-Rue-Madame dataset before and after the self-learning process.



(a) Results obtained before the self-learning process.

(b) Results obtained after the self-learning process.

Figure 15.10: Results on Paris-Rue-Madame dataset [*Serna et al.*, 2014].

Table 15.5 presents the three metric scores before and after the self-learning process execution for each considered objects of the dataset.

Metrics/Objects		Facade	Ground	Car	Motorcycle	Traffic sign
Recall	Before S-L	0.923	0.997	0.565	0.403	0.880
	After S-L	0.923	0.998	0.935	0.737	0.881
Precision	Before S-L	0.834	0.964	0.498	0.282	0.725
	After S-L	0.834	0.964	0.94	0.641	0.725
F1-score	Before S-L	0.876	0.980	0.530	0.332	0.795
	After S-L	0.877	0.981	0.939	0.685	0.796

Table 15.5: Comparison of metrics obtained before and after Self-Learning process (S-L) on Paris-Rue-Madame dataset.

The comparison of the detection scores before and after the self-learning process application, presented in the table 15.5, shows that self-learning process has no impact on the detection of facades, ground and traffic signs. However, the self-learning process allows for a significant improvement in the detection of cars and motorcycles. The recall, precision, and F1-scores averaged double for the detection of these two objects after the self-learning process. This detection improvement for cars and motorcycles is also visible in Figure 15.10.

Discussion

The evaluation of the self-learning process on the Stanford and Paris-Rue-Madame datasets allows for identifying its limitations and benefits. The main limitation observed on the Stanford dataset is a slight loss of accuracy related to the previously discussed "edge effects" (see Sections 15.1.1 and 15.1.3).
The benefit of the self-learning process lies in its ability to solve non-detection problems as well as to correct misclassifications. This benefit was observed both on the cars and motorcycles detection in the Paris-Rue-Madame dataset and on half of the rooms in the Stanford dataset. Although the detection of some objects (e.g. wall and floor) is not significantly affected by the self-learning process, the detection of some other objects is significantly improved.

15.2 Comparison with other approaches

15.2.1 Approaches comparison on indoor environment

The comparison of different approaches results in the indoor environment is applied to the use case 2.1.1. [*Armeni et al.*, 2017] makes available some data enabling comparisons. These data correspond to the third rooms shown in Figure 15.11 (a) of a bigger point cloud composed of more than 22 millions of points. For this use case, the approach results of this thesis are compared with the approaches of [*Armeni et al.*, 2016] and [*Bobkov et al.*, 2017] to demonstrate the potential and the efficiency of the proposed approach. These two approaches come from Machine learning approaches and have been described earlier in the related work.

Figure 15.11 visually compares the results of these two approaches with the approach presented in this thesis.

The approach [*Armeni et al.*, 2016] badly detects the *room 19* and *room 22*, whose a part of the room *room 19* is detected as the *room 22*. Moreover, this approach

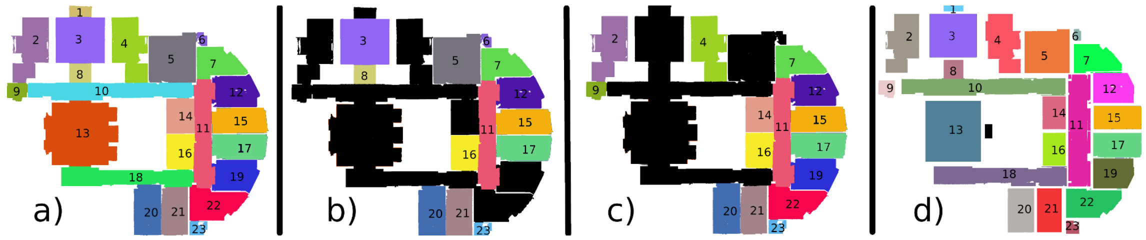


Figure 15.11: Illustration of the results comparison, the black color represents rooms wrongly detected; other colors represent rooms well detected: a) the ground truth, b) the reconstruction results of [*Armeni et al.*, 2016], c) the reconstruction result of [*Bobkov et al.*, 2017], d) the result of the presented approach.

does not detect the room *room 1*. Further problem of over-segmentation have been observed with this approach for *room 2*, *room 4*, and *room 14*. Finally, the rooms *room 9*, *room 10*, *room 13*, and *room 18* are detected as the same room. Figure 15.11 (b) shows in black every room, which is not well detected by this approach.

This last problem of rooms segmentation is also observed with the approach [*Bobkov et al.*, 2017] that detects six rooms (*room 1*, *room 3*, *room 8*, *room 10*, *room 13*, and *room 18*) as a same room. Furthermore, two other (*room 5*, and *room 6*) are also detected as a same room. Figure 15.11 (c) shows in black every room, which is not well detected by this approach.

Table 15.6 compares numerically the results of the two approaches with the approach presented in this thesis. The approach proposed detects 22 rooms and wrongly classify only one room (*room 13* in Figure 15.11 (d)) while the approaches [*Armeni et al.*, 2016] and [*Bobkov et al.*, 2017] respectively wrongly detect 10 and 8 rooms.

Approach	Correctly classified room (max 22)	Sucess(%)
Presented approach	21	95%
[*Armeni et al.*, 2016]	12	55%
[*Bobkov et al.*, 2017]	14	63%

Table 15.6: Results comparison between the presented approach and two other approaches on the point cloud corresponding to the use case 2.1.1.

According to the results obtained, we can claim that our approach is more robust (95% of rooms are detected) than the two other approaches (55% and 63% of rooms are detected). This higher performance comes from full management by semantic technologies and dynamical knowledge adaptation. The knowledge used by semantics technologies allows for adapting the process to all variations encountered during the sensing process included external factor, acquisition context, and scene characteristics. Moreover, knowledge used to drive the detection process fully is dynamically enriched by the knowledge-based self-learning process. Thus, this approach ensures proper object detection through a safe and smart detection process that uses reasoning and consistency checking to avoid any wrong detection.

15.2.2 Approaches comparison on outdoor environment

Two assessments of the approach efficiency on the outdoor environment have been done through the comparison with other approaches. The first comparison is based on the results obtained for the point cloud from Fraunhofer. These results have been presented in 3D Tag conference [*Ponciano et al.*, 2019*a*]. The second comparison is based on results obtained for "Paris-rue-Madame database: MINES ParisTech 3D mobile laser scanner dataset from Madame street in Paris"[3] ([*Serna et al.*, 2014]).

Comparison of walls and ground detection on Fraunhofer's point cloud

The presented approach is compared to two other approaches specialized in the detection of walls and ground. The three approaches have been applied to the same test point cloud to obtain comparable results. In the paper [*Anagnostopoulos et al.*, 2016], the authors proposed a method, which exclusively detects walls and floors in point cloud through the combination of linear algorithms. This method well detects the ground but fails to detect some walls due to many missing parts in it. In the paper [*Xing et al.*, 2018], the authors proposed a method of feature recognition based on the application of SWRL-rules to classify walls and ground in an urban context. This method proposes to apply a planar segmentation and uses then the semantic to classify planar segments. Unfortunately, this approach depends on the planar segmentation results and thus, fails to detect walls which are primarily composed of vegetation. The results of this comparison is illustrated in Table 15.7.

Approach	Correctly classified (%)	Wrongly classified (%)
Presented approach	96.38%	3.62%
[*Anagnostopoulos et al.*, 2016]	90.18%	9.82%
[*Xing et al.*, 2018]	86.06%	13.94%

Table 15.7: Results comparison between the presented approach and two other approaches on the same point cloud part.

[3]MINES ParisTech© copyright. MINES ParisTech created this special set of 3D MLS data for the purpose of detection-segmentation-classification research activities, but does not endorse the way they are used in this project or the conclusions put forward.

Generally, generic object detection approaches are less accurate than detection approaches dedicated to the detection of one or two objects. However, in this comparison study, our approach is more precise (more than 96%) than the two other approaches (90% and 86.06%) that are specialized in ground and wall detection. The proposed approach thus shows that it is both generic and more robust than the two other approaches.

Comparison of detection approaches on Paris-rue-Madame dataset

The application case of urban outdoor point clouds is one that has the biggest number of 3D annotated datasets. Therefore, these annotated datasets allow the training of machine learning approaches. Moreover, among the different approaches to structure data in an urban context, approaches using Neighborhood approximation, and a feature extraction followed by a classification using machine learning classifier have stood out for their effectiveness. That is why the efficiency of the proposed approach is compared with three approaches of this type, which are the approaches [*Hackel et al.*, 2016*b*], [*Weinmann et al.*, 2015*b*], and [*Weinmann et al.*, 2014]. The approach [*Hackel et al.*, 2016*b*] uses descriptors such as Shape Context 3D (SC3D) ([*Frome et al.*, 2004]) and Signature of Histogram of Orientations (SHOT) ([*Tombari et al.*, 2010]) to extract the characteristics. The approach [*Weinmann et al.*, 2014] uses a supervised classification, whereas the approaches [*Weinmann et al.*, 2014] and [*Hackel et al.*, 2016*b*] use a machine learning classifier, which is Random Forest classifier to identify data elements. These approaches provide excellent results on the application case "Paris-rue-Madame." This application case has the advantage to have an open annotated dataset of reference (illustrated in Figure 15.8(b)), provided by "Paris-rue-Madame database: MINES ParisTech 3D mobile laser scanner dataset from Madame street in Paris"[4] ([*Serna et al.*, 2014]), which allows the computation of the *Recall*, *Precision*, and *F1-score* metrics ([*Zheng et al.*, 2019], explained in Section 15.1.4). That is why this application case is used to assess the proposed approach through these three metrics and to compare its efficiency with the approaches [*Hackel et al.*, 2016*b*], [*Weinmann et al.*, 2015*b*], and [*Weinmann et al.*, 2014]. Table 15.8 shows the values obtained by each approach for the three metrics and for each following object type: Facade (F), Ground (G), Car (C), Motorcycle (M), and Traffic sign (T).

The comparison of results on the three metrics between the different approaches

[4]MINES ParisTech© copyright. MINES ParisTech created this special set of 3D MLS data for the purpose of detection-segmentation-classification research activities, but does not endorse the way they are used in this project or the conclusions put forward.

Objects/Approaches		F	G	C	M	T
[*Hackel et al.*, 2016*b*]	Recall	0.9799	0.9692	0.9786	0.9796	0.9939
	Precision	0.9902	0.9934	0.9086	0.4792	0.3403
	F1-score	0.9851	0.9811	0.9423	0.6435	0.5070
[*Weinmann et al.*, 2015*b*]	Recall	0.9527	0.8650	0.6476	0.7198	0.9485
	Precision	0.9620	0.9782	0.7948	0.0980	0.0491
	F1-score	0.9573	0.9182	0.7137	0.0934	0.5070
[*Weinmann et al.*, 2014]	Recall	0.958	0.911	0.603	0.657	0.978
	Precision	0.964	0.960	0.768	0.136	0.058
	F1-score	0.961	0.935	0.676	0.225	0.109
Proposed approach	Recall	0.923	0.9978	0.935	0.737	0.8806
	Precision	0.834	0.964	0.94	0.6405	0.7253
	F1-score	0.8765	0.9806	0.939	0.685	0.7955

Table 15.8: Results of metric values obtained by each approach for each considered object class: the best score by metric and object is highlighted by a green background, whereas the worst score is highlighted in red.

shows the strength and weakness of the proposed approach. The weakness of the proposed approach for this application case appears in facade detection, where the proposed approach obtains the worst values of recall, precision, and F1-score. This low quality of the proposed approach is due to many points behind the facade that are not considered as a facade part by this approach, contrary to the other approaches, whose reference dataset.

The strength of the proposed approach appears in the detection of small objects as motorcycles, traffic signs, and cars. Although a lower recall value, the proposed approach obtains the best value of precision and F1-score for the detection of motorcycle and traffic sign. This strength comes from the use of topological relationship that provides further information, which improves their detection. The worst recall value obtained for the detection of traffic signs is certainly due to the integration of segments having similar characteristics than the traffic sign and that are thus, integrated as one of them.

Finally, the proposed approach obtains the best recall value and F1-score (slightly highlighted in green in Table 15.8), quasi equivalent to the best F1-score for the object ground. It also obtains the best precision value for the detection of cars. It loses some quality in the car detection due to some overlap between floor and cars, whose the boundaries are sometimes not accurate enough. However, although the proposed approach does not obtain the best results for the detection of the ground and cars, their results are excellent and very near to the best results values.

Therefore, the proposed approach is one of the most efficient approaches to process this application case.

16 Discussion

This implementation part has presented the choices of implementation and results obtained by the implemented approach.

16.1 Implementation choices

The implementation choices concern firstly, the implementation of the processing architecture and then, the implementation of the studied application cases.

The choice of implementation techniques for the processing depends on their capacity to satisfy the requirements of the proposed solution and on their performances. That is why the data processing is realized in *C++*, whereas the knowledge management is realized in *Java*, through the use of *Jena* that allows among others, the manipulation of *OWL* knowledge model and the use of *SPARQL* query. Moreover, the conceptual bridge for the exchange between knowledge and data processing paradigms is realized technically in *Java* through the combination of *Jena* and *JNI* libraries. The communication between the two paradigms is done through SPARQL that uses the library *JNI* to execute the data processing in *C++*, and retrieve its result.

The choices on application cases implementation cannot exhaustively be shown due to the number of objects modeled for the four application case. That is why the knowledge modeling and processing chapters provide an overview of these choices and their impact on the processing through examples from the studied application case.
Concerning the data modeling, the modeling example of the application case 2.1.3 shows the influence of diverse external factors on the acquisition technique. It also illustrates the influence of the acquisition techniques and technologies on the data characteristics (e.g. occlusion and density). These influences are expressed through SPARQL queries construct to compute the data characteristics (as occluded areas)

of the processed data automatically.
Examples of object modeling based on the description of geometry, characteristic, and topological relationships are given through the semantic description of a wall and a floor. These examples are interesting firstly because they are shared between the different application cases, and secondly, because their representation varies according to the context of the application. Thus, they allow the illustration of the adaptation of generic characteristics of an object type according to the application case.

The steps of data processing (algorithm selection, configuration, and execution followed by the classification) are applied for detecting all objects according to their characteristics. That is why some objects must be detected before the others. These elements are called elementary objects and are identified according to their topological relationship with others, their relationship of composition, and their size. The others are called secondary objects. The detection of secondary objects is facilitated by the detection of elementary objects that reduces their area of research on some point clouds portions. The identification of secondary objects can also require the identification of elementary objects in case of composition relation (e.g. a room is a secondary object because it requires the identification of walls and floors that are elementary objects to be identified). Thanks to the illustration of data processing through diverse examples of elementary and secondary objects, the chapter 14 also illustrates the adaptation of algorithm selection according to objects description, as well as, a dynamic and continuous classification according to results of each data processing steps. Thus, the chapter 13 and chapter 14 show the approach flexibility through the adaptation capability of both the model and the processing according to object types and context diversity. Finally, the role and performance of the self-learning process illustrated on the enrichment of wall description and the improvement of the application case 2.1.1, shows the generalizability of the approach through its capability to improve both the knowledge model and the data processing according to each application case.

16.2 Results discussion

The study of final results on each application case allows for identifying the strengths and weaknesses of the approach. The main strengths of the approach identified through the analysis of the results are its flexibility (e.g. the capability to be applied to four different application cases) and its robustness to detect objects in complex situation (e.g. detection of watermill in a point cloud of ruin excavation, or detection of all searched objects in a point cloud composed of 15% of noise). The main weakness of the approach is the ambiguity that creates "edge effects" in the data understanding (c.f. results on application cases 15.1.1 and 15.1.3). To conclude in more detail on the strengths and weaknesses of the proposed approach, let us take the criteria used to analyze the related work to analyze in its turn the proposed approach.

Quality

The quality of data understanding has two aspects. The first aspect is the quantitative quality represented, for example, by the percentage of detected objects or the percentage of data understand (equivalent to the percentage of well-classified points). The approach has good quantitative quality since:

1. Data processing in Section 15.1.1 results in recognition of all searched objects, and an average of 97.66% of the data is understood.

2. Data processing in Section 15.1.2 results in recognition of all rooms and the watermill.

3. Data processing in Section 15.1.3 results in recognition of all rooms. The comparison with other approaches to this application case shows this approach brings better results of data understanding.

The second aspect of quality is the accuracy of object detection. It can be assessed through the number of elements wrongly classified or not classified, but also by the detected proportion of an object. The advantage of the proposed approach in accuracy is not to produce the wrong classification. Moreover, the application case section 15.1.4 shows the approach has globally a good efficiency and precision, and even one of the best efficiency among existing approaches (c.f. comparison in Section 15.2.2). However, some weakness lies in the unclassified element. Indeed, some elements as the wall in results 15.1.2 or the round edges of the table in Figure 15.2 have representation too far from the semantic description or as the backpack in Figure 15.1, have no description inside the knowledge base to be identified. Nevertheless, the major weakness that results in unclassified elements is ambiguity.

Ambiguity

Indeed, the proposed approach does not manage ambiguities. An element that could be classified into two disjoint object types would create an inconsistency of the knowledge. That is why such an object would not be classified in the two object types. It would also not be classified arbitrarily in one of the object types to avoid the wrong classification, and would thus stay as unclassified. The choice of preferred an unclassified element to wrongly classified elements is due to the impact of the wrong classification on the self-learning process. An unclassified element could be later be classified through the self-learning process, whereas a wrong classified element would provide wrong information on an object type as a base for the learning. Therefore, this "no management" of the ambiguities conducts the approach at producing "edge effects" as observed in results 15.1.3 between rooms and shown by red elements of Figure 15.1(b). These "edge effects" illustrate the lack of accuracy quality of the approach. This accuracy weakness is a compromise that guarantees the proper work of the self-learning process and benefits to the robustness and the generalizability of the approach.

Robustness

The robustness of this approach is shown through the good quantitative quality obtained for each application case; even for challenging situations. For example, results in 15.1.2 show the approach is able to detect rooms even if their walls are partially destroyed. Moreover, in the urban application case, many factors conduct to variations of data characteristics as differences of density or occluded area. However, the approach succeeds to gather parts of the same wall split by an occluded area. Urban application case also provides a broad diversity of representation for an object type as cars; the approach also succeeds to detect this variety of representations.

Flexibility

As proved by the application of this approach on four very different cases, this one is flexible. This approach can adapt face to:

1. noisy point cloud as scene 4 in 15.1.1,

2. point cloud with strong roughness as the application case of cultural heritage,

3. variations of object representation, density variations, occluded areas, and other variations of data characteristics as in urban application case.

Generalizability

Finally, the approach can be improved both by itself and by extensions from the user. The improvement capability, by itself, is provided by the self-learning process that improves both the knowledge and the data understanding process. The improvement of the approach by users is possible through the adding of new algorithms into the algorithm library and their semantic description into the knowledge base. Moreover, the user can also enrich the knowledge base by knowledge about the different domain represented by the ontology.

Part V

Conclusion

17 Conclusion and future work

Although we know of many solutions for the processing of different kind of data, they are generally based on an implicit understanding of the data and are inflexible and not robust to face and manage variations in data characteristics, which requires an adaptation of the way of processing. This lack of flexibility and robustness limits current performance to analyze unstructured data sets in changing contexts automatically.

17.1 Main contribution

This thesis addresses this limit by a new knowledge-based approach. The knowledge described through an ontology aims at representing the source of data characteristics variations and their impact on the data understanding process to manage these variations of data characteristics. The role of the knowledge is to guide the process of data understanding step by step through a continuous exchange of information between the knowledge base and the data processing. This continuous exchange is provided by a conceptual and technical bridge that allows the knowledge to guide the data processing and allows the data processing to enrich the knowledge. This bridge provides the expected flexibility of the approach by allowing a data understanding process that adapts according to the knowledge (dynamically enriched by the data processing). However, the content of the knowledge base cannot predict all cases of object representation variations to understand the data wholly. Therefore, the approach completes the data understanding through a self-learning process based on knowledge and using hypothesis formulation. This self-learning process aims at guaranty the robustness of the approach by enriching the knowledge base according to each use case specificity. This learning process is based on an analysis of the first results of the data understanding process to improve its result through a second data understanding process based on more specific knowledge.
Therefore, this new knowledge-based approach provides three key contributions, which are an ontology describing the domains of data, scene, and data processing,

a technical and conceptual bridge between knowledge and data processing, and a self-learning process.

Ontology describing the domains of data, scene, and data processing The ontology provides a knowledge model that allows users to describe scenes and data, which they have to process. The knowledge model also allows users to enrich the data processing by describing algorithms that can be added to the libraries of algorithms.
The knowledge provided by users about scenes and data, whose acquisition process, produces a base of reasoning that enables to deduce and anticipate data characteristics, as well as to object representation inside the data. Since the knowledge aims at guiding the data understanding process, this deduced knowledge allows for adapting the data understanding process more precisely to each processed use case.

Technical and conceptual bridge between knowledge management and data processing Contrary to other knowledge-based approaches, the approach presented in this thesis uses a continuous bi-directional exchange of information between the knowledge and the data processing. This continuous bi-directional exchange of information allows the knowledge to guide the data processing step by step, by taking into account the results of data processing at each step. At each step of the data understanding process, the knowledge guides the selection, configuration, and execution of algorithms according to the processed use case and the current step of data understanding. At each of these steps, the knowledge base interprets the results of executed algorithms automatically and combines the different knowledge available to understand and structure the data. Thus, the selection, the configuration, and the execution of algorithms are managed in line with the evolution of data processing and understanding.

Knowledge-based self-learning process using hypothesis formulation The described process based on a bridge between knowledge and processing allows for a data understanding based on a knowledge base content. The purpose of this content is not to describe the scene and data in detail, but to provide a base of knowledge sufficient to recognize objects. Such knowledge base has so, a lack of specific knowledge about the processed data that does not allow for recognizing objects, whose representation varies from their description in the knowledge base. The knowledge-based self-learning process proposed in this approach aims at enriching the knowledge base through a knowledge specific to each use case. It uses the data understanding process based on the sufficient knowledge provided by users as a learning base. This base of data understanding is analyzed to generate hypotheses automatically to improve the knowledge description. The formulation of a hypothesis depends firstly on an enrichment about scene knowledge (e.g. identification of object size, identification of topological relationships between objects), and secondly, on gathering common characteristics of objects belonging to the same type. The identified characteristics common to an object type are combined to formulate hypotheses to define a better description of this object type in the knowledge base. These hypotheses are tested to verify that their integration to the original knowledge base does not create any inconsistency. A hypothesis that does not create inconsistency is considered as coherent and consequently, integrated into the knowledge base. The adding of validated hypothesis into the knowledge base changes the behavior of the data processing, which is more adapted to the processed use case. This adaptation to the processed use case through a more specific knowledge provides a better data understanding.

The robustness and flexibility of the approach have been validated from four application cases with different characteristics, contexts, and objectives. This study shows that the proposed approach provides relevant results in terms of data understanding even in challenging situations such as data portions with low density. Its comparison with other existing approaches that have dealt with two of the four use cases results obtained shows that the approach proposed in this thesis significantly improves the understanding of unstructured data.

17.2 Advantages and limits

The particularity of this approach lies in the automatic enrichment of the knowledge according to each application case. First, knowledge about the representations of objects in the data is enriched by the reasoning process applied to the knowledge of the acquisition process, the digitalized scene, and the context. Then, self-learning process enriches the knowledge base by identifying new object characteristics and more specialized object descriptions for each application case.

These enrichments allow for the assumption of data characteristics and the adaptation of the knowledge on object representations to different application cases. For the knowledge-based approach, this leads to an adaptation of the data understanding process according to each application case. This flexibility of adaptation allows the approach to be applied to different application cases with different contexts (such as archaeological excavations, urban outdoor environments or indoor building environments), but also different acquisition processes (such as laser scanning, or time-of-flight scanning of a Microsoft Kinect).

Finally, the proposed approach is easily extensible by adding algorithms and knowledge to expand its intrinsic capabilities and to various fields of application.

The main limitation of this approach is the accuracy of elements detection. This limitation is mainly due to the management of identification ambiguities. Indeed, the approach considers as inconsistent any element that may belong to two disjoint classes. Thus, portions of data located at the boundary between two objects and which can, therefore, be classified either in one or the other, are not classified. This behavior causes "edge effects" on objects detection. However, this limit can also be an advantage in some application cases, since it avoids misclassifications. This lack of precision is also due to the sensitivity of the approach to variations in the algorithm parameterization. The parameterization of algorithms is performed automatically based on the knowledge of the characteristics of data, objects, and algorithms. Knowledge of algorithms requires a good understanding of its behavior and the impact of changing parameter values on its behavior. It is not always easy to formulate with high precision the knowledge on the configuration of algorithms. These difficulties may lead in some cases to a description of the knowledge of the algorithm that is insufficiently precise. This insufficient precision leads to a correct configuration of the algorithm but not sufficiently precise according to the sensitivity of the algorithm. Thus, the accuracy of detection depends above all on the description of the knowledge about object types and algorithm configuration.

17.3 Future work

This section proposes future work based on the results of this thesis. This future work is presented according to short, medium, and long-term planning. Short-term planning proposes possible but not necessary improvements to prove the effectiveness of the thesis work. The medium-term planning proposes possible extensions to enrich the current version of the work. Finally, long-term planning proposes a research problem based on the work of this thesis, allowing us to go further in the context of this work.

17.3.1 Short term

In the short term, it would be interesting to extend the analysis of the behaviour of various objects in different data sets in order to improve the understanding of their interactions, and thus to extend and improve the knowledge base. Such improvement would aim at obtaining a better accuracy for application cases that would require greater detection accuracy. We did not make these improvements because the applications on which the proposed approach has been tested do not require greater precision of results as the main objective of this study was to establish the proof of concept of the proposed approach.
Furthermore, it would be interesting to extend the approach to integrating image data. This extension would require the addition of image processing algorithms to the algorithm library and the addition of their representation to the knowledge base. Image data provides additional content that extends the base provided for detection. Such an improvement would simplify the detection process, make detection more robust, and increase the quality of the detection process.

17.3.2 Middle term

In the medium term, it would be interesting to integrate "free form" detection algorithms and point cloud descriptors to extend the efficiency of the currently implemented approach and to identify more quickly objects that would be very difficult to describe in the form of the current knowledge-based on classical geometries. The addition of these algorithms would require investigations to understand how to configure and use them, but also research to formalize this knowledge in the ontology.
Another interesting extension would be to process video data. This type of data would require extending knowledge modeling to the temporal domain. It would also allow for the exploration of learning opportunities based on the evolution of images within the video.

17.3.3 Long term

Finally, it would be interesting in the long term to automatically integrate knowledge from the Semantic Web in order to enrich the ontology with descriptions of objects and scenes that can be identified in data. The Semantic Web is a source that can provide various knowledge such as various descriptions of objects and geometries. However, the object modeling in the Semantic Web depends on the designers who contribute to its development. Therefore, there are many different models, both for modeling objects, and for modeling the same object. Thus, the integration of knowledge from the semantic web to enrich the knowledge base used by the approach proposed in this thesis should satisfy several needs. First, it would require retrieving the relevant knowledge to be integrated, and then adapting its modeling to the model used in this approach for data understanding. Besides, the great diversity of possible descriptions of an object could serve as a basis for the knowledge-based self-learning process to infer new knowledge. Indeed, the self-learning collects information from knowledge about individual objects to describe knowledge about the type of an object. Thus, it could collect information common to all descriptions for an object type, in order to generate new knowledge on the possible representations of these objects in the data. Moreover, these representations could be automatically adapted to each application case through reasoning on the knowledge domains described in the ontology.

Conclusion (en Français)

Conclusion et travaux futurs

Bien que nous connaissions de nombreuses solutions pour le traitement de différents types de données, elles reposent généralement sur une compréhension implicite des données. Elles sont également inflexibles et peu robustes pour appréhender et gérer les variations des caractéristiques des données, qui nécessitent une adaptation des méthodes de traitement. Ce manque de flexibilité et de robustesse limite les performances actuelles pour analyser automatiquement des ensembles de données non structurées dans des contextes changeants.

Contribution principale

Cette thèse aborde cette limite par une nouvelle approche basée sur la connaissance. Les connaissances décrites à travers une ontologie visent à représenter la source des variations des caractéristiques des données et leur impact sur le processus de compréhension des données pour gérer ces variations. Le rôle de la connaissance est de guider pas à pas le processus de compréhension des données à travers un échange continu d'informations entre la base de connaissances et le traitement des données. Cet échange continu est assuré par un pont conceptuel et technique qui permet aux connaissances de guider le traitement des données et permet au traitement des données d'enrichir les connaissances. Ce pont apporte la flexibilité attendue de l'approche en permettant un processus de compréhension des données qui s'adapte en fonction des connaissances (dynamiquement enrichi par le traitement des données). Cependant, le contenu de la base de connaissances ne peut pas prédire tous les cas de variations dans la représentation des objets pour comprendre les données dans leur intégralité. Par conséquent, l'approche complète la compréhension des données par un processus d'auto-apprentissage basé sur la connaissance et la formulation d'hypothèses. Ce processus d'auto-apprentissage vise à garantir la robustesse de l'approche en enrichissant la base de connaissances en fonction de chaque cas d'utilisation spécifique. Ce processus d'apprentissage est basé sur une analyse des premiers résultats du processus de compréhension des données afin d'améliorer son résultat par un second processus de compréhension des données basé sur des connaissances plus spécifiques. Par conséquent, cette nouvelle approche fondée sur le savoir apporte trois contributions clés : une ontologie décrivant les domaines des données, de la scène et du traitement des données, un pont technique et conceptuel entre le savoir et le traitement des données, et un processus d'auto-apprentissage.

Ontologie décrivant les domaines des données, de la scène et du traitement des données L'ontologie fournit un modèle de connaissance qui permet aux utilisateurs de décrire des scènes et des données qu'ils doivent traiter. Le modèle de connaissance permet également aux utilisateurs d'enrichir le traitement des données en décrivant les algorithmes qui peuvent être ajoutés aux bibliothèques d'algorithmes. Les connaissances fournies par les utilisateurs sur les scènes et les données, dont le processus d'acquisition produit une base de raisonnement qui permet de déduire et d'anticiper les caractéristiques des données, ainsi que de représenter les objets dans les données. Puisque la connaissance vise à guider le processus de compréhension des données, cette connaissance déduite permet d'adapter plus précisément le processus de compréhension des données à chaque cas d'utilisation traité.

Pont technique et conceptuelle entre la gestion des connaissances et le traitement des données Contrairement à d'autres approches basées sur la connaissance, l'approche présentée dans cette thèse utilise un échange continu bidirectionnel d'informations entre la connaissance et le traitement des données. Cet échange d'informations bidirectionnel continu permet de guider le traitement des données étape par étape, en prenant en compte les résultats du traitement des données à chaque étape. À chaque étape du processus de compréhension des données, les connaissances guident la sélection, la configuration et l'exécution des algorithmes en fonction du cas d'utilisation traité et de l'étape actuelle de compréhension des données. À chacune de ces étapes, la base de connaissances interprète automatiquement les résultats des algorithmes exécutés et combine les différentes connaissances disponibles pour comprendre et structurer les données. Ainsi, la sélection, la configuration et l'exécution des algorithmes sont gérées en fonction de l'évolution du traitement et de la compréhension des données.

Processus d'auto-apprentissage basé sur les connaissances et la formulation d'hypothèses Le processus décrit, basé sur un pont entre la connaissance et le traitement, permet une compréhension des données basée sur le contenu d'une base de connaissances. Le but de ce contenu n'est pas de décrire la scène et les données en détail, mais de fournir une base de connaissances suffisante pour reconnaître les objets. Une telle base de connaissances a donc, un manque de connaissances spécifiques sur les données traitées qui ne permet pas de reconnaître les objets, dont leur représentation dans les données diffère de leur description dans la base de connaissances. Le processus d'auto-apprentissage basé sur la connaissance proposé dans cette approche vise à enrichir la base de connaissances par une connaissance spécifique à chaque cas d'utilisation. Il utilise comme base d'apprentissage le processus de compréhension des données basé sur les connaissances suffisantes fournies par les utilisateurs. Cette base de données est analysée pour générer automatiquement des hypothèses afin d'améliorer la description des connaissances. La formulation d'une hypothèse dépend d'une part d'un enrichissement de la connaissance de la scène (ex : identification de la taille de l'objet, identification des relations topologiques entre objets), et d'autre part de la collecte des caractéristiques communes d'objets appartenant au même type. Les caractéristiques identifiées qui sont communes à un type d'objet sont combinées pour formuler des hypothèses afin de définir une meilleure description de ce type d'objet dans la base de connaissances. Ces hypothèses sont testées pour vérifier que leur intégration à la base de connaissances d'origine ne crée pas d'incohérence. Une hypothèse qui ne crée pas d'incohérence est considérée comme cohérente et, par conséquent, elle est intégrée dans la base de connaissances. L'ajout d'hypothèses validées dans la base de connaissances modifie le comportement du traitement des données, qui est plus adapté au cas d'utilisation traité. Cette adaptation au cas d'utilisation traité par une connaissance plus spécifique permet une meilleure compréhension des données.

La robustesse et la flexibilité de l'approche ont été validées à partir de quatre cas d'application aux caractéristiques, contextes et objectifs différents. Cette étude montre que l'approche proposée fournit des résultats pertinents en termes de compréhension des données, même dans des situations difficiles telles que des portions de données à faible densité. Sa comparaison avec d'autres approches existantes qui ont traité deux des quatre cas d'utilisation obtenus montre que l'approche proposée dans cette thèse améliore considérablement la compréhension des données non structurées.

17.3. FUTURE WORK

Avantages et limites

La particularité de cette approche réside dans l'enrichissement automatique des connaissances en fonction de chaque cas d'application. Premièrement, la connaissance des représentations des objets dans les données est enrichie par le processus de raisonnement appliqué à la connaissance du processus d'acquisition, de la scène numérisée et du contexte. Ensuite, le processus d'auto-apprentissage enrichit la base de connaissances en identifiant de nouvelles caractéristiques d'objets et des descriptions d'objets plus spécialisées pour chaque cas d'application.

Ces enrichissements permettent de prédire les caractéristiques des données et d'adapter les connaissances sur les représentations des objets aux différents cas d'application. Dans le cas de l'approche fondée sur la connaissance, cela conduit à une adaptation du processus de compréhension des données en fonction de chaque cas d'application. Cette souplesse d'adaptation permet d'appliquer l'approche à différents cas d'application dans différents contextes (fouilles archéologiques, environnements urbains extérieurs ou environnements intérieurs de bâtiments), mais aussi à différents processus d'acquisition (comme le balayage laser ou le balayage à temps de vol d'un instrument tel que Microsoft Kinect).

Enfin, l'approche proposée est facilement extensible par l'ajout d'algorithmes et de connaissances afin d'étendre ses capacités intrinsèques et à divers domaines d'application.

La principale limite de cette approche est la précision de la détection des éléments. Cette limitation est principalement due à la gestion des ambiguïtés de reconnaissance d'objets. En effet, l'approche considère comme incohérent tout élément pouvant appartenir à deux classes disjointes. Ainsi, les parties de données situées à la frontière entre deux objets et qui peuvent donc être classées dans l'un ou l'autre ne sont pas classées. Ce comportement provoque des "effets de bord" sur la détection d'objets. Toutefois, cette limite peut également être un avantage dans certains cas d'application, car elle permet d'éviter les erreurs de classification. Ce manque de précision est également dû à la sensibilité de l'approche aux variations du paramétrage des algorithmes. Le paramétrage des algorithmes s'effectue automatiquement en fonction de la connaissance des caractéristiques des données, des objets et des algorithmes. La connaissance des algorithmes nécessite une bonne compréhension de leur comportement et de l'impact de l'évolution des valeurs des paramètres sur leur comportement. Il n'est pas toujours facile de formuler avec une grande précision les connaissances sur la configuration des algorithmes. Ces difficultés peuvent conduire dans certains cas à une description insuffisamment précise de la connaissance de l'algorithme. Cette précision insuffisante conduit à une configuration correcte de l'algorithme mais pas suffisamment précise selon la sensibilité de l'algorithme. Ainsi, la précision de la détection dépend avant tout de la description des connaissances sur les types d'objets et la configuration des algorithmes.

Travaux futurs

Cette section propose des travaux futurs basés sur les résultats de cette thèse. Ce travail futur est présenté selon une planification à court, moyen et long terme. La planification à court terme propose des améliorations possibles mais non nécessaires pour prouver l'efficacité du travail de thèse. La planification à moyen terme propose des prolongements possibles pour enrichir la version actuelle de l'ouvrage. Enfin, la planification à long terme propose une problématique de recherche basée sur les travaux de cette thèse, permettant d'aller plus loin dans le cadre de ces travaux.

Court terme

À court terme, il serait intéressant d'étendre l'analyse du comportement des différents objets dans différents ensembles de données afin d'améliorer la compréhension de leurs interactions et donc de développer et améliorer la base de connaissances. Une telle amélioration viserait à obtenir une meilleure précision pour les cas d'application qui nécessitent une plus grande précision de détection. Nous n'avons pas apporté ces améliorations puisque les applications sur lesquelles l'approche proposée a été testée ne nécessitent pas une plus grande précision des résultats étant donné que l'objectif principal de cette étude était d'établir la preuve du concept de l'approche proposée. De plus, il serait intéressant d'étendre l'approche à l'intégration d'image. Cette extension nécessiterait l'ajout d'algorithmes de traitement d'images à la bibliothèque d'algorithmes et l'ajout de leur représentation à la base de connaissances. Les images fournissent un contenu supplémentaire qui élargit la base fournie pour la détection. Une telle amélioration simplifierait le processus de détection, rendrait la détection plus robuste et améliorerait la qualité du processus de détection.

Moyen terme

À moyen terme, il serait intéressant d'intégrer des algorithmes de détection de "forme libre" et des descripteurs de nuages de points pour étendre l'efficacité de l'approche actuellement mise en œuvre. Cela permettrait d'identifier plus rapidement des objets qui seraient très difficiles à décrire sous la forme des connaissances actuelles basées sur des géométries classiques. L'ajout de ces algorithmes nécessiterait des investigations pour comprendre comment les configurer et les utiliser, mais aussi des recherches pour formaliser ces connaissances dans l'ontologie. Une autre extension intéressante serait de traiter les données vidéo. Ce type de données nécessiterait d'étendre la modélisation des connaissances au domaine temporel. Cela permettrait également d'explorer les possibilités d'apprentissage en fonction de l'évolution des images de la vidéo.

Long terme

Enfin, il serait intéressant à long terme d'intégrer automatiquement les connaissances issues du Web sémantique afin d'enrichir l'ontologie des descriptions d'objets et de scènes identifiables dans les données. Le Web sémantique est une source de connaissances diverses qui peut fournir des descriptions variées d'objets et de géométries. Cependant, la modélisation d'objets dans le Web sémantique dépend des concepteurs qui contribuent à son développement. Il existe donc de nombreux modèles différents, tant pour la modélisation de différents objets que pour la modélisation d'un même objet. Ainsi, l'intégration des connaissances issues du web sémantique pour enrichir la base de connaissances utilisée par l'approche proposée dans cette thèse devrait satisfaire plusieurs besoins. Tout d'abord, il faudrait récupérer les connaissances pertinentes à intégrer, puis adapter leurs modélisations au modèle utilisé dans cette approche pour comprendre les données. En outre, la grande diversité des descriptions possibles d'un objet pourrait servir de base au processus d'auto-apprentissage basé sur la connaissance pour déduire de nouvelles connaissances. En effet, l'auto-apprentissage recueille des informations à partir de connaissances sur des objets individuels pour décrire la connaissance du type d'objet. Ainsi, il pourrait collecter des informations communes à toutes les descriptions d'un type d'objet, afin de générer de nouvelles connaissances sur les représentations possibles de ces objets dans les données. De plus, ces représentations pourraient être automatiquement adaptées à chaque cas d'application par un raisonnement sur les domaines de connaissance décrits dans l'ontologie.

Bibliography

[*Aboali et al.*, 2017] Aboali, Maged, Manap, Nurulfajar Abd, Darsono, Abd Majid, and Yusof, Zulkalnain Mohd. *Review on three-dimensional (3-d) acquisition and range imaging techniques*. Int. J. Appl. Eng. Res, 12:2409–2421 (2017).

[*Abuzaina et al.*, 2013] Abuzaina, Anas, Nixon, Mark S., and Carter, John N. *Sphere detection in kinect point clouds via the 3d hough transform*. In Richard Wilson, Edwin Hancock, Adrian Bors, and William Smith, editors, *Computer Analysis of Images and Patterns*, pages 290–297. Springer Berlin Heidelberg, Berlin, Heidelberg (2013). ISBN 978-3-642-40246-3.

[*Ackermann and Troisi*, 2010] Ackermann, Sebastiano and Troisi, Salvatore. *Una procedura di modellazione automatica degli edifici con dati lidar*. BOLLETTINO DELLA SOCIETÀ ITALIANA DI FOTOGRAMMETRIA E TOPOGRAFIA, 2:9–25 (2010).

[*Adan and Huber*, 2011] Adan, Antonio and Huber, Daniel. *3d reconstruction of interior wall surfaces under occlusion and clutter*. In *2011 International Conference on 3D Imaging, Modeling, Processing, Visualization and Transmission*, pages 275–281. IEEE (2011).

[*Ahsan and Shah*, 2006] Ahsan, Syed and Shah, Abad. *Data, information, knowledge, wisdom: A doubly linked chain*. In *the proceedings of the 2006 international conference on information knowledge engineering*, pages 270–278. Citeseer (2006).

[*Aijazi et al.*, 2013] Aijazi, Ahmad, Checchin, Paul, and Trassoudaine, Laurent. *Segmentation based classification of 3d urban point clouds: A super-voxel based approach with evaluation*. Remote Sensing, 5(4):1624–1650 (2013).

[*Alexandre*, 2012] Alexandre, Luıs A. *3d descriptors for object and category recognition: a comparative evaluation*. In *Workshop on Color-Depth Camera Fusion in Robotics at the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Vilamoura, Portugal*, volume 1, page 7 (2012).

[*Anagnostopoulos et al.*, 2016] Anagnostopoulos, Ioannis, Pătrăucean, Viorica, Brilakis, Ioannis, and Vela, Patricio. *Detection of walls, floors, and ceilings in point cloud data*. In *Construction Research Congress 2016*, pages 2302–2311 (2016). URL http://dx.doi.org/10.1061/9780784479827.229. Visited on 2019-08-1, https://ascelibrary.org/doi/pdf/10.1061/9780784479827.229.

[*Anguelov et al.*, 2005] Anguelov, Dragomir, Taskarf, B, Chatalbashev, Vassil, Koller, Daphne, Gupta, Dinkar, Heitz, Geremy, and Ng, Andrew. *Discriminative learning of markov random fields for segmentation of 3d scan data*. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 2, pages 169–176. IEEE (2005).

[*Armeni et al.*, 2017] Armeni, I., Sax, A., Zamir, A. R., and Savarese, S. *Joint 2D-3D-Semantic Data for Indoor Scene Understanding*. ArXiv e-prints (2017). URL https://arxiv.org/abs/1702.01105. Visited on 2019-08-1, 1702.01105.

[*Armeni et al.*, 2016] Armeni, Iro, Sener, Ozan, Zamir, Amir R., Jiang, Helen, Brilakis, Ioannis, Fischer, Martin, and Savarese, Silvio. *3d semantic parsing of large-scale indoor spaces*. In *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition* (2016).

[*Babahajiani et al.*, 2017] Babahajiani, Pouria, Fan, Lixin, Kämäräinen, Joni-Kristian, and Gabbouj, Moncef. *Urban 3d segmentation and modelling from street view images and lidar point clouds*. Machine Vision and Applications, 28(7):679–694 (2017).

[*Ballard*, 1981] Ballard, Dana H. *Generalizing the hough transform to detect arbitrary shapes*. Pattern recognition, 13(2):111–122 (1981).

[*Bassiliades*, 2018] Bassiliades, Nick. *Swrl2spin: Converting swrl to spin*. In *RuleML+RR* (2018).

[*Bernardini and Rushmeier*, 2002] Bernardini, Fausto and Rushmeier, Holly. *The 3d model acquisition pipeline*. In *Computer graphics forum*, volume 21, pages 149–172. Wiley Online Library (2002).

[*Berners-Lee et al.*, 2001] Berners-Lee, Tim, Hendler, James, Lassila, Ora et al. *The semantic web*. Scientific american, 284(5):28–37 (2001).

[*Bewley et al.*, 2011] Bewley, A., Shekhar, R., Leonard, S., Upcroft, B., and Lever, P. *Real-time volume estimation of a dragline payload*. In *2011 IEEE International*

Conference on Robotics and Automation, pages 1571–1576 (2011). ISSN 1050-4729. URL http://dx.doi.org/10.1109/ICRA.2011.5979898.

[*Bobkov et al.*, 2017] Bobkov, Dmytro, Kiechle, Martin, Hilsenbeck, Sebastian, and Steinbach, Eckehard. *Room segmentation in 3d point clouds using anisotropic potential fields*. In *2017 IEEE International Conference on Multimedia and Expo (ICME)*, pages 727–732. IEEE (2017).

[*Bock et al.*, 2008] Bock, Jürgen, Haase, Peter, Ji, Qiu, and Volz, Raphael. *Benchmarking owl reasoners*. In *ARea2008-Workshop on Advancing Reasoning on the Web: Scalability and Commonsense*. Tenerife (2008).

[*Boley et al.*, 2010] Boley, Harold, Paschke, Adrian, and Shafiq, Omair. *Ruleml 1.0: the overarching specification of web rules*. In *International Workshop on Rules and Rule Markup Languages for the Semantic Web*, pages 162–178. Springer (2010).

[*Borrmann et al.*, 2011] Borrmann, Dorit, Elseberg, Jan, Lingemann, Kai, and Nüchter, Andreas. *The 3d hough transform for plane detection in point clouds: A review and a new accumulator design*. 3D Research, 2(2):3 (2011).

[*Boykov et al.*, 1999] Boykov, Yuri, Veksler, Olga, and Zabih, Ramin. *Fast approximate energy minimization via graph cuts*. In *Proceedings of the Seventh IEEE International Conference on Computer Vision*, volume 1, pages 377–384. IEEE (1999).

[*Breiman*, 2001] Breiman, Leo. *Random forests*. Machine Learning, 45(1):5–32 (2001). ISSN 1573-0565. URL http://dx.doi.org/10.1023/A:1010933404324. Visited on 2019-08-1.

[*Camurri et al.*, 2014] Camurri, Marco, Vezzani, Roberto, and Cucchiara, Rita. *3d hough transform for sphere recognition on point clouds*. Machine Vision and Applications, 25(7):1877–1891 (2014). ISSN 1432-1769. URL http://dx.doi.org/10.1007/s00138-014-0640-3. Visited on 2019-08-1.

[*Cao et al.*, 2010] Cao, J., Tagliasacchi, A., Olson, M., Zhang, H., and Su, Z. *Point cloud skeletons via laplacian based contraction*. In *2010 Shape Modeling International Conference*, pages 187–197 (2010). URL http://dx.doi.org/10.1109/SMI.2010.25.

[*Carroll et al.*, 2004] Carroll, Jeremy J, Dickinson, Ian, Dollin, Chris, Reynolds, Dave, Seaborne, Andy, and Wilkinson, Kevin. *Jena: implementing the semantic web recommendations*. In *Proceedings of the 13th international World Wide Web conference on Alternate track papers & posters*, pages 74–83. ACM (2004).

[*Choi et al.*, 2008] Choi, Jung Hwa, Park, Young Tack, Suh, Il Hong, Lim, Gi Hyun, and Lee, Sanghoon. *Ontology-Based Semantic Context Modeling for Object Recognition of Intelligent Mobile Robots*, pages 399–408. Springer Berlin Heidelberg, Berlin, Heidelberg (2008). ISBN 978-3-540-76729-9. URL http://dx.doi.org/10.1007/978-3-540-76729-9_31. Visited on 2019-08-1.

[*Corman et al.*, 2018] Corman, Julien, Reutter, Juan L., and Savković, Ognjen. *Semantics and validation of recursive shacl*. In Denny Vrandečić, Kalina Bontcheva, Mari Carmen Suárez-Figueroa, Valentina Presutti, Irene Celino, Marta Sabou, Lucie-Aimée Kaffee, and Elena Simperl, editors, *The Semantic Web – ISWC 2018*, pages 318–336. Springer International Publishing, Cham (2018). ISBN 978-3-030-00671-6.

[*Cortes and Vapnik*, 1995] Cortes, Corinna and Vapnik, Vladimir. *Support-vector networks*. Machine learning, 20(3):273–297 (1995).

[*Couprie et al.*, 2011] Couprie, Camille, Bresson, Xavier, Najman, Laurent, Talbot, Hugues, and Grady, Leo. *Surface reconstruction using power watershed*. In Pierre Soille, Martino Pesaresi, and Georgios K. Ouzounis, editors, *Mathematical Morphology and Its Applications to Image and Signal Processing*, pages 381–392. Springer Berlin Heidelberg, Berlin, Heidelberg (2011). ISBN 978-3-642-21569-8.

[*Cregan et al.*, 2005] Cregan, Anne, Mochol, Malgorzata, Vrandecic, Denny, and Bechhofer, Sean. *Pushing the limits of owl, rules and protege - a simple example*. In *OWL: Experiences and Directions*. Galway, Ireland (2005).

[*Dai et al.*, 2016] Dai, Jifeng, Li, Yi, He, Kaiming, and Sun, Jian. *R-fcn: Object detection via region-based fully convolutional networks*. In *Advances in neural information processing systems*, pages 379–387 (2016).

[*Davenport et al.*, 1998] Davenport, Thomas H, Prusak, Laurence et al. *Working knowledge: How organizations manage what they know*. Harvard Business Press (1998).

[*De Deuge et al.*, 2013] De Deuge, Mark, Quadros, Alastair, Hung, Calvin, and Douillard, Bertrand. *Unsupervised feature learning for classification of outdoor 3d scans*. In *Australasian Conference on Robitics and Automation*, volume 2, page 1 (2013).

[*De Maesschalck et al.*, 2000] De Maesschalck, Roy, Jouan-Rimbaud, Delphine, and Massart, Désiré L. *The mahalanobis distance*. Chemometrics and intelligent laboratory systems, 50(1):1–18 (2000).

[*Deb and Agrawal*, 1998] Deb, Kalyanmoy and Agrawal, Samir. *Understanding interactions among genetic algorithm parameters.* In *FOGA*, pages 265–286 (1998).

[*Deschaud and Goulette*, 2010] Deschaud, Jean-Emmanuel and Goulette, François. *A Fast and Accurate Plane Detection Algorithm for Large Noisy Point Clouds Using Filtered Normals and Voxel Growing*. In *3DPVT*. Paris, France (2010). URL https://hal-mines-paristech.archives-ouvertes.fr/hal-01097361. Visited on 2019-08-1.

[*Diaz-Vilarino et al.*, 2015] Diaz-Vilarino, Lucia, Conde, Borja, Lagüela, Susana, and Lorenzo, Henrique. *Automatic detection and segmentation of columns in as-built buildings from point clouds*. Remote Sensing, 7(11):15651–15667 (2015). ISSN 2072-4292. URL http://dx.doi.org/10.3390/rs71115651.

[*Dietenbeck et al.*, 2017] Dietenbeck, Thomas, Torkhani, Fakhri, Othmani, Ahlem, Attene, Marco, and Favreau, Jean-Marie. *Multi-layer ontologies for integrated 3d shape segmentation and annotation*. In *Advances in Knowledge Discovery and Management*, pages 181–206. Springer (2017).

[*Dijkstra*, 1959] Dijkstra, Edsger W. *A note on two problems in connexion with graphs*. Numerische mathematik, 1(1):269–271 (1959).

[*Dobruschin*, 1968] Dobruschin, PL. *The description of a random field by means of conditional probabilities and conditions of its regularity*. Theory of Probability & Its Applications, 13(2):197–224 (1968).

[*Dong et al.*, 2014] Dong, Chao, Loy, Chen Change, He, Kaiming, and Tang, Xiaoou. *Learning a deep convolutional network for image super-resolution*. In *European conference on computer vision*, pages 184–199. Springer (2014).

[*Drost et al.*, 2010] Drost, Bertram, Ulrich, Markus, Navab, Nassir, and Ilic, Slobodan. *Model globally, match locally: Efficient and robust 3d object recognition*. In *2010 IEEE computer society conference on computer vision and pattern recognition*, pages 998–1005. Ieee (2010).

[*Du et al.*, 2018] Du, Xinxin, Ang, Marcelo H, Karaman, Sertac, and Rus, Daniela. *A general pipeline for 3d detection of vehicles*. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3194–3200. IEEE (2018).

[*Farjas et al.*, 2003] Farjas, Mercedes, García-Lázaro, Francisco J, Zancajo, Julio, and Mostaza, Teresa. *Automatic point-cloud surveys in prehistoric site documentation and modelling*. Topografía y Cartografía, 116:70–71 (2003).

[*Frome et al.*, 2004] Frome, Andrea, Huber, Daniel, Kolluri, Ravi, Bülow, Thomas, and Malik, Jitendra. *Recognizing objects in range data using regional point descriptors*. In *European conference on computer vision*, pages 224–237. Springer (2004).

[*Fürber and Hepp*, 2010] Fürber, Christian and Hepp, Martin. *Using sparql and spin for data quality management on the semantic web*. In Witold Abramowicz and Robert Tolksdorf, editors, *Business Information Systems*, pages 35–46. Springer Berlin Heidelberg, Berlin, Heidelberg (2010). ISBN 978-3-642-12814-1.

[*Fürsattel et al.*, 2015] Fürsattel, Peter, Placht, Simon, Balda, Michael, Schaller, Christian, Hofmann, Hannes, Maier, Andreas, and Riess, Christian. *A comparative error analysis of current time-of-flight sensors*. IEEE Transactions on Computational Imaging, 2(1):27–41 (2015).

[*Giel and Issa*, 2011] Giel, Brittany and Issa, Raja. *Using laser scanning to access the accuracy of as-built bim*. In *Computing in Civil Engineering (2011)*, pages 665–672 (2011). ISBN 978-0-7844-1182-7. URL http://dx.doi.org/10.1061/41182(416)82.

[*Girshick*, 2015] Girshick, Ross. *Fast r-cnn*. In *Proceedings of the IEEE international conference on computer vision*, pages 1440–1448 (2015).

[*Gómez-Romero et al.*, 2016] Gómez-Romero, Juan, García, Jesús, Patricio, Miguel A, Serrano, Miguel A, and Molina, José M. *Context-based situation recognition in computer vision systems*. In *Context-Enhanced Information Fusion*, pages 627–651. Springer (2016).

[*Griffiths and Boehm*, 2019] Griffiths, David and Boehm, Jan. *A review on deep learning techniques for 3d sensed data classification*. Remote Sensing, 11(12):1499 (2019).

[*Grilli et al.*, 2017] Grilli, E, Menna, F, and Remondino, F. *A review of point clouds segmentation and classification algorithms*. The International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences, 42:339 (2017).

[*Günther et al.*, 2011] Günther, Martin, Wiemann, Thomas, Albrecht, Sven, and Hertzberg, Joachim. *Model-based object recognition from 3d laser data*. In Joscha Bach and Stefan Edelkamp, editors, *KI 2011: Advances in Artificial Intelligence*, pages 99–110. Springer Berlin Heidelberg, Berlin, Heidelberg (2011). ISBN 978-3-642-24455-1.

[*Habib et al.*, 2008] Habib, AF, Al-Durgham, M, Kersting, AP, and Quackenbush, P. *Error budget of lidar systems and quality control of the derived point cloud*. In *Proceedings of the XXI ISPRS Congress, Commission I: Vol. XXXVII. Part B1*, pages 203–209 (2008).

[*Hackel et al.*, 2016*a*] Hackel, Timo, Wegner, Jan D, and Schindler, Konrad. *Contour detection in unstructured 3d point clouds*. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1610–1618 (2016*a*).

[*Hackel et al.*, 2016*b*] Hackel, Timo, Wegner, Jan D, and Schindler, Konrad. *Fast semantic segmentation of 3d point clouds with strongly varying density*. ISPRS annals of the photogrammetry, remote sensing and spatial information sciences, 3(3):177–184 (2016*b*).

[*Han et al.*, 2017] Han, Xian-Feng, Jin, Jesse S., Wang, Ming-Jie, Jiang, Wei, Gao, Lei, and Xiao, Liping. *A review of algorithms for filtering the 3d point cloud*. Signal Processing: Image Communication, 57:103 – 112 (2017). ISSN 0923-5965. URL http://dx.doi.org/https://doi.org/10.1016/j.image.2017.05.009. Visited on 2019-08-1.

[*Hana et al.*, 2018] Hana, Xian-Feng, Jin, Jesse S, Xie, Juan, Wang, Ming-Jie, and Jiang, Wei. *A comprehensive review of 3d point cloud descriptors*. arXiv preprint arXiv:1802.02297 (2018).

[*Hartley and Mundy*, 1993] Hartley, Richard I and Mundy, Joseph L. *Relationship between photogrammetry and computer vision*. In *Integrating photogrammetric techniques with scene analysis and machine vision*, volume 1944, pages 92–105. International Society for Optics and Photonics (1993).

[*He et al.*, 2015] He, Kaiming, Zhang, Xiangyu, Ren, Shaoqing, and Sun, Jian. *Spatial pyramid pooling in deep convolutional networks for visual recognition*. IEEE transactions on pattern analysis and machine intelligence, 37(9):1904–1916 (2015).

[*He et al.*, 2016] He, Kaiming, Zhang, Xiangyu, Ren, Shaoqing, and Sun, Jian. *Deep residual learning for image recognition*. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778 (2016).

[*Henn et al.*, 2013] Henn, André, Gröger, Gerhard, Stroh, Viktor, and Plümer, Lutz. *Model driven reconstruction of roofs from sparse lidar point clouds*. ISPRS Journal of photogrammetry and remote sensing, 76:17–29 (2013).

[*Hernandez and Marcotegui*, 2009] Hernandez, J. and Marcotegui, B. *Point cloud segmentation towards urban ground modeling*. In *2009 Joint Urban Remote Sensing Event*, pages 1–5 (2009). ISSN 2334-0932. URL http://dx.doi.org/10.1109/URS.2009.5137562.

[*Himmelsbach et al.*, 2010] Himmelsbach, Michael, Hundelshausen, Felix V, and Wuensche, H-J. *Fast segmentation of 3d point clouds for ground vehicles*. In *2010 IEEE Intelligent Vehicles Symposium*, pages 560–565. IEEE (2010).

[*Hitzler et al.*, 2009] Hitzler, Pascal, Krötzsch, Markus, Parsia, Bijan, Patel-Schneider, Peter F, and Rudolph, Sebastian. *Owl 2 web ontology language primer*. W3C recommendation, 27(1):123 (2009).

[*Hmida et al.*, 2013] Hmida, Helmi Ben, Cruz, Christophe, Boochs, Frank, and Nicolle, Christophe. *From 3d point clouds to semantic objects an ontology-based detection approach*. CoRR, abs/1301.4783 (2013). URL http://arxiv.org/abs/1301.4783. Visited on 2019-08-1, 1301.4783.

[*Horridge and Patel-Schneider*, 2009] Horridge, Matthew and Patel-Schneider, Peter F. *Owl 2 web ontology language manchester syntax*. W3C Working Group Note (2009).

[*Horrocks et al.*, 2004] Horrocks, Ian, Patel-Schneider, Peter F, Boley, Harold, Tabet, Said, Grosof, Benjamin, Dean, Mike et al. *Swrl: A semantic web rule language combining owl and ruleml*. W3C Member submission, 21(79):1–31 (2004).

[*Hu et al.*, 2018] Hu, Pingbo, Dong, Zhen, Yuan, Pengfei, Liang, Fuxun, and Yang, Bisheng. *Reconstruction of 3d models from point clouds with hybrid representation*. International Archives of the Photogrammetry, Remote Sensing & Spatial Information Sciences, 42(2) (2018).

[*Hulik et al.*, 2014] Hulik, Rostislav, Spanel, Michal, Smrz, Pavel, and Materna, Zdenek. *Continuous plane detection in point-cloud data based on 3d hough transform*. Journal of Visual Communication and Image Representation, 25(1):86 – 97 (2014). ISSN 1047-3203. URL http://dx.doi.org/https://doi.org/10.1016/j.jvcir.2013.04.001. Visited on 2019-08-1.

[*Husain et al.*, 2014] Husain, Farzad, Dellen, Labette, and Torras, Carme. *Recognizing point clouds using conditional random fields*. In *2014 22nd international conference on pattern recognition*, pages 4257–4262. IEEE (2014).

[*Hwang et al.*, 2006] Hwang, Wonil, Park, Jinyoung, Suh, Hyowon, Kim, Hyungwook, and Suh, Il Hong. *Ontology-based framework of robot context modeling* *and reasoning for object recognition*. In *International Conference on Fuzzy Systems and Knowledge Discovery*, pages 596–606. Springer (2006).

[*Jecić and Drvar*, 2003] Jecić, Stjepan and Drvar, Nenad. *The assessment of structured light and laser scanning methods in 3d shape measurements*. In *4th International Congress of Croatian Society of Mechanics* (2003).

[*Ji et al.*, 2012] Ji, Shuiwang, Xu, Wei, Yang, Ming, and Yu, Kai. *3d convolutional neural networks for human action recognition*. IEEE transactions on pattern analysis and machine intelligence, 35(1):221–231 (2012).

[*Johnston et al.*, 2008] Johnston, Benjamin, Yang, Fangkai, Mendoza, Rogan, Chen, Xiaoping, and Williams, Mary-Anne. *Ontology based object categorization for robots*. In Takahira Yamaguchi, editor, *Practical Aspects of Knowledge Management*, pages 219–231. Springer Berlin Heidelberg, Berlin, Heidelberg (2008). ISBN 978-3-540-89447-6.

[*Järvenpää et al.*, 2018] Järvenpää, Eeva, Hylli, Otto, Siltala, Niko, and Lanz, Minna. *Utilizing spin rules to infer the parameters for combined capabilities of aggregated manufacturing resources*. IFAC-PapersOnLine, 51(11):84 – 89 (2018). ISSN 2405-8963. URL http://dx.doi.org/https://doi.org/10.1016/j.ifacol.2018.08.239. 16th IFAC Symposium on Information Control Problems in Manufacturing INCOM 2018; visited on 2019-08-1.

[*Kalinowski*, 2010] Kalinowski, Pav. *Understanding confidence intervals (cis) and effect size estimation*. APS Observer, 23(4) (2010).

[*Kanimozhi and Christy*, 2013] Kanimozhi, T. and Christy, A. *Incorporating ontology and sparql for semantic image annotation*. In *2013 IEEE Conference on Information Communication Technologies*, pages 26–31 (2013). URL http://dx.doi.org/10.1109/CICT.2013.6558056.

[*Karmacharya et al.*, 2015] Karmacharya, A, Boochs, F, and Tietz, B. *Knowledge guided object detection and identification in 3d point clouds*. In *Videometrics, Range Imaging, and Applications XIII*, volume 9528, page 952804. International Society for Optics and Photonics (2015).

[*Khaloo and Lattanzi*, 2017] Khaloo, Ali and Lattanzi, David. *Robust normal estimation and region growing segmentation of infrastructure 3d point cloud models*. Advanced Engineering Informatics, 34:1–16 (2017).

[*Klasing et al.*, 2009] Klasing, K., Althoff, D., Wollherr, D., and Buss, M. *Comparison of surface normal estimation methods for range sensing applications*. In *2009 IEEE*

International Conference on Robotics and Automation, pages 3206–3211 (2009). ISSN 1050-4729. URL http://dx.doi.org/10.1109/ROBOT.2009.5152493.

[*Knublauch et al.*, 2011] Knublauch, Holger, Hendler, James A, and Idehen, Kingsley. *Spin-overview and motivation*. W3C Member Submission, 22:W3C (2011).

[*Knublauch and Kontokostas*, 2017] Knublauch, Holger and Kontokostas, Dimitris. *Shapes constraint language (shacl)*. W3C Recommendation, 20 (2017).

[*Krizhevsky et al.*, 2012*a*] Krizhevsky, Alex, Sutskever, Ilya, and Hinton, Geoffrey E. *Imagenet classification with deep convolutional neural networks*. In *Advances in neural information processing systems*, pages 1097–1105 (2012*a*).

[*Krizhevsky et al.*, 2012*b*] Krizhevsky, Alex, Sutskever, Ilya, and Hinton, Geoffrey E. *Imagenet classification with deep convolutional neural networks*. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 1097–1105. Curran Associates, Inc. (2012*b*). URL http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf.

[*Ku et al.*, 2018] Ku, Jason, Mozifian, Melissa, Lee, Jungwook, Harakeh, Ali, and Waslander, Steven L. *Joint 3d proposal generation and object detection from view aggregation*. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1–8. IEEE (2018).

[*Kumar et al.*, 2008] Kumar, V. S., Narayanan, S., Kurc, T., Kong, J., Gurcan, M. N., and Saltz, J. H. *Analysis and semantic querying in large biomedical image datasets*. Computer, 41(4):52–59 (2008). ISSN 0018-9162. URL http://dx.doi.org/10.1109/MC.2008.108.

[*Lague et al.*, 2013] Lague, Dimitri, Brodu, Nicolas, and Leroux, Jérôme. *Accurate 3d comparison of complex topography with terrestrial laser scanner: Application to the rangitikei canyon (nz)*. ISPRS journal of photogrammetry and remote sensing, 82:10–26 (2013).

[*Lai et al.*, 2011] Lai, Kevin, Bo, Liefeng, Ren, Xiaofeng, and Fox, Dieter. *A large-scale hierarchical multi-view rgb-d object dataset*. In *2011 IEEE international conference on robotics and automation*, pages 1817–1824. IEEE (2011).

[*Lehtomäki et al.*, 2016] Lehtomäki, Matti, Jaakkola, Anttoni, Hyyppä, Juha, Lampinen, Jouko, Kaartinen, Harri, Kukko, Antero, Puttonen, Eetu, and

Hyyppä, Hannu. *Object classification and recognition from mobile laser scanning point clouds in a road environment*. IEEE Transactions on Geoscience and Remote Sensing, 54(2):1226–1239 (2016).

[*Lemmens*, 2011] Lemmens, Mathias. *Terrestrial laser scanning*. In *Geo-information*, pages 101–121. Springer (2011).

[*Li*, 2017] Li, Bo. *3d fully convolutional network for vehicle detection in point cloud*. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1513–1518. IEEE (2017).

[*Li et al.*, 2016] Li, Zhuqiang, Zhang, Liqiang, Tong, Xiaohua, Du, Bo, Wang, Yuebin, Zhang, Liang, Zhang, Zhenxin, Liu, Hao, Mei, Jie, Xing, Xiaoyue et al. *A three-step approach for tls point cloud classification*. IEEE Transactions on Geoscience and Remote Sensing, 54(9):5412–5424 (2016).

[*Limberger and Oliveira*, 2015] Limberger, Frederico A. and Oliveira, Manuel M. *Real-time detection of planar regions in unorganized point clouds*. Pattern Recognition, 48(6):2043 – 2053 (2015). ISSN 0031-3203. URL http://dx.doi.org/https://doi.org/10.1016/j.patcog.2014.12.020. Visited on 2019-08-1.

[*Lin et al.*, 2018] Lin, Yangbin, Wang, Cheng, Zhai, Dawei, Li, Wei, and Li, Jonathan. *Toward better boundary preserved supervoxel segmentation for 3d point clouds*. ISPRS journal of photogrammetry and remote sensing, 143:39–47 (2018).

[*Liu*, 2015] Liu, Ming. *Robotic online path planning on point cloud*. IEEE transactions on cybernetics, 46(5):1217–1228 (2015).

[*Long et al.*, 2015] Long, Jonathan, Shelhamer, Evan, and Darrell, Trevor. *Fully convolutional networks for semantic segmentation*. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440 (2015).

[*Lu et al.*, 2009] Lu, Wei-Lwun, Murphy, Kevin P, Little, James J, Sheffer, Alla, and Fu, Hongbo. *A hybrid conditional random field for estimating the underlying ground surface from airborne lidar data*. IEEE Transactions on Geoscience and Remote Sensing, 47(8):2913–2922 (2009).

[*Lu et al.*, 2016] Lu, Xiaohu, Yao, Jian, Tu, Jinge, Li, Kai, Li, Li, and Liu, Yahui. *Pairwise linkage for point cloud segmentation*. ISPRS Annals of Photogrammetry, Remote Sensing & Spatial Information Sciences, 3(3) (2016).

[*Luo et al.*, 2018] Luo, Huan, Wang, Cheng, Wen, Chenglu, Chen, Ziyi, Zai, Dawei, Yu, Yongtao, and Li, Jonathan. *Semantic labeling of mobile lidar point clouds* *via active learning and higher order mrf*. IEEE Transactions on Geoscience and Remote Sensing, 56(7):3631–3644 (2018).

[*Maas*, 1999] Maas, Hans-Gerd. *The potential of height texture measures for the segmentation of airborne laserscanner data*. In *Fourth international airborne remote sensing conference and exhibition/21st Canadian symposium on remote sensing*, volume 1, pages 154–161. Citeseer (1999).

[*Macher et al.*, 2015] Macher, Hélène, Landes, Tania, and Grussenmeyer, Pierre. *Point clouds segmentation as base for as-built bim creation*. ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences, 2(5):191 (2015).

[*Marroquin et al.*, 2018] Marroquin, Roberto, Dubois, Julien, and Nicolle, Christophe. *Know beyond seeing: combining computer vision with semantic reasoning*. In *2018 IEEE 12th International Conference on Semantic Computing (ICSC)*, pages 310–311. IEEE (2018).

[*Maturana and Scherer*, 2015] Maturana, Daniel and Scherer, Sebastian. *Voxnet: A 3d convolutional neural network for real-time object recognition*. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 922–928. IEEE (2015).

[*McGuinness et al.*, 2004] McGuinness, Deborah L, Van Harmelen, Frank et al. *Owl web ontology language overview*. W3C recommendation, 10(10):2004 (2004).

[*Meditskos et al.*, 2014] Meditskos, Georgios, Kontopoulos, Efstratios, and Kompatsiaris, Ioannis. *Knowledge-driven activity recognition and segmentation using context connections*. In Peter Mika, Tania Tudorache, Abraham Bernstein, Chris Welty, Craig Knoblock, Denny Vrandečić, Paul Groth, Natasha Noy, Krzysztof Janowicz, and Carole Goble, editors, *The Semantic Web – ISWC 2014*, pages 260–275. Springer International Publishing, Cham (2014). ISBN 978-3-319-11915-1.

[*Meyer et al.*, 2019*a*] Meyer, Gregory P, Charland, Jake, Hegde, Darshan, Laddha, Ankit, and Vallespi-Gonzalez, Carlos. *Sensor fusion for joint 3d object detection and semantic segmentation*. arXiv preprint arXiv:1904.11466 (2019*a*).

[*Meyer et al.*, 2019*b*] Meyer, Gregory P, Laddha, Ankit, Kee, Eric, Vallespi-Gonzalez, Carlos, and Wellington, Carl K. *Lasernet: An efficient probabilistic 3d object detector for autonomous driving*. arXiv preprint arXiv:1903.08701 (2019*b*).

[*Miller*, 1998] Miller, Eric. *An introduction to the resource description framework*. Bulletin of the American Society for Information Science and Technology, 25(1):15–19 (1998).

[*Moosmann et al.*, 2009] Moosmann, Frank, Pink, Oliver, and Stiller, Christoph. *Segmentation of 3d lidar data in non-flat urban environments using a local convexity criterion*. In *2009 IEEE Intelligent Vehicles Symposium*, pages 215–220. IEEE (2009).

[*Motik and Studer*, 2005] Motik, B and Studer, Rudi. *Kaon2-a scalable reasoning tool for the semantic web*. In *The Future Web* (2005).

[*Musen et al.*, 2015] Musen, Mark A et al. *The protégé project: a look back and a look forward*. AI matters, 1(4):4 (2015). URL http://dx.doi.org/10.1145/2757001.2757003.

[*Nguyen and Le*, 2013] Nguyen, Anh and Le, Bac. *3d point cloud segmentation: A survey*. In *2013 6th IEEE conference on robotics, automation and mechatronics (RAM)*, pages 225–230. IEEE (2013).

[*Niemeyer et al.*, 2011] Niemeyer, Joachim, Wegner, Jan Dirk, Mallet, Clément, Rottensteiner, Franz, and Soergel, Uwe. *Conditional random fields for urban scene classification with full waveform lidar data*. In *ISPRS Conference on Photogrammetric Image Analysis*, pages 233–244. Springer (2011).

[*Ochmann et al.*, 2016] Ochmann, Sebastian, Vock, Richard, Wessel, Raoul, and Klein, Reinhard. *Automatic reconstruction of parametric building models from indoor point clouds*. Computers & Graphics, 54:94–103 (2016).

[*Oehler et al.*, 2011] Oehler, Bastian, Stueckler, Joerg, Welle, Jochen, Schulz, Dirk, and Behnke, Sven. *Efficient multi-resolution plane segmentation of 3d point clouds*. In Sabina Jeschke, Honghai Liu, and Daniel Schilberg, editors, *Intelligent Robotics and Applications*, pages 145–156. Springer Berlin Heidelberg, Berlin, Heidelberg (2011). ISBN 978-3-642-25489-5.

[*O'Riain et al.*, 2015] O'Riain, Seán, McCrae, John, Cimiano, Philipp, and Spohr, Dennis. *Using spin to formalise xbrl accounting regulations on the semantic web*. In Elena Simperl, Barry Norton, Dunja Mladenic, Emanuele Della Valle, Irini Fundulaki, Alexandre Passant, and Raphaël Troncy, editors, *The Semantic Web: ESWC 2012 Satellite Events*, pages 58–72. Springer Berlin Heidelberg, Berlin, Heidelberg (2015). ISBN 978-3-662-46641-4.

[*Othmani et al.*, 2010] Othmani, Ahlem, Meziat, Carole, and Loménie, Nicolas. *Ontology-driven image analysis for histopathological images*. In *International Symposium on Visual Computing*, pages 1–12. Springer (2010).

[*OuYang and Feng*, 2005] OuYang, Daoshan and Feng, Hsi-Yung. *On the normal vector estimation for point cloud data from smooth surfaces*. Computer-Aided Design, 37(10):1071 – 1079 (2005). ISSN 0010-4485. URL http://dx.doi.org/https://doi.org/10.1016/j.cad.2004.11.005. Visited on 2019-08-1.

[*Ouyang et al.*, 2015] Ouyang, Wanli, Wang, Xiaogang, Zeng, Xingyu, Qiu, Shi, Luo, Ping, Tian, Yonglong, Li, Hongsheng, Yang, Shuo, Wang, Zhe, Loy, Chen-Change et al. *Deepid-net: Deformable deep convolutional neural networks for object detection*. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2403–2412 (2015).

[*Overby et al.*, 2004] Overby, Jens, Bodum, Lars, Kjems, Erik, and M Ilsøe, Peer. *Automatic 3d building reconstruction from airborne laser scanning and cadastral data using hough transform*. International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences, XXXIV (2004).

[*Patel-Schneider*, 2017] Patel-Schneider, Peter F. *Ashacl: Alternative shapes constraint language*. arXiv preprint arXiv:1702.01795 (2017).

[*Pauling et al.*, 2009] Pauling, Frederick, Bosse, Michael, and Zlot, Robert. *Automatic segmentation of 3d laser point clouds by ellipsoidal region growing*. In *Australasian Conference on Robotics and Automation*, volume 10 (2009).

[*Pauly et al.*, 2002] Pauly, Mark, Gross, Markus, and Kobbelt, Leif P. *Efficient simplification of point-sampled surfaces*. In *Proceedings of the Conference on Visualization '02*, VIS '02, pages 163–170. IEEE Computer Society, Washington, DC, USA (2002). ISBN 0-7803-7498-3. URL http://dl.acm.org/citation.cfm?id=602099.602123.

[*Pauly et al.*, 2005] Pauly, Mark, Mitra, Niloy J, Giesen, Joachim, Gross, Markus H, and Guibas, Leonidas J. *Example-based 3d scan completion*. In *Symposium on Geometry Processing*, CONF, pages 23–32 (2005).

[*Ponciano et al.*, 2017] Ponciano, Jean-Jacques, Boochs, Frank, and Trémeau, Alain. *Knowledge-based object recognition in point clouds and image data sets*. gis.Science - Die Zeitschrift für Geoinformatik (2017). URL https://hal.archives-ouvertes.fr/hal-02047375. Visited on 2019-08-1.

[*Ponciano et al.*, 2019*a*] Ponciano, Jean-Jacques, Boochs, Frank, and Tremeau, Alain. *Identification and classification of objects in 3d point clouds based on a semantic concept*. In *3D-Tage*. Oldenburger, Germany (2019*a*). URL https://hal.archives-ouvertes.fr/hal-02014831. Visited on 2019-08-1.

[*Ponciano et al.*, 2019*b*] Ponciano, Jean-Jacques., Karmacharya, A., Wefers, S., Atorf, P., and Boochs, F. *Connected semantic concepts as a base for optimal recording and computer-based modelling of cultural heritage objects*. In Rafael Aguilar, Daniel Torrealva, Susana Moreira, Miguel A. Pando, and Luis F. Ramos, editors, *Structural Analysis of Historical Constructions*, pages 297–304. Springer International Publishing, Cham (2019*b*). ISBN 978-3-319-99441-3. URL http://dx.doi.org/10.1007/978-3-319-99441-3_31. Visited on 2019-08-1.

[*Ponciano et al.*, 2015] Ponciano, Jean-Jacques, Prudhomme, Claire, Tietz, Burkhard, and Boochs, Frank. *Detection and isolation of switches in point clouds of the german railway network*. In *2015 11th International Conference on Signal-Image Technology & Internet-Based Systems (SITIS)*, pages 96–102. IEEE (2015). URL http://dx.doi.org/10.1109/SITIS.2015.86.

[*Ponciano et al.*, 2019*c*] Ponciano, Jean-Jacques, Trémeau, Alain, and Boochs, Frank. *Automatic detection of objects in 3d point clouds based on exclusively semantic guided processes*. ISPRS International Journal of Geo-Information, 8(10) (2019*c*). ISSN 2220-9964. URL http://dx.doi.org/10.3390/ijgi8100442.

[*Poux et al.*, 2018] Poux, Florent, Neuville, Romain, Nys, Gilles-Antoine, and Billen, Roland. *3d point cloud semantic modelling: Integrated framework for indoor spaces and furniture*. Remote Sensing, 10(9) (2018). ISSN 2072-4292. URL http://dx.doi.org/10.3390/rs10091412.

[*Poux et al.*, 2017] Poux, Florent, Neuville, Romain, Van Wersch, Line, Nys, Gilles-Antoine, and Billen, Roland. *3d point clouds in archaeology: Advances in acquisition, processing and knowledge integration applied to quasi-planar objects*. Geosciences, 7(4) (2017). ISSN 2076-3263. URL http://dx.doi.org/10.3390/geosciences7040096.

[*Prudhomme et al.*, 2019] Prudhomme, Claire, Homburg, Timo, Ponciano, Jean-Jacques, Boochs, Frank, Cruz, Christophe, and Roxin, Ana-Maria. *Interpretation and automatic integration of geospatial data into the semantic web*. Computing, pages 1–27 (2019). URL http://dx.doi.org/10.1007/s00607-019-00701-y. Visited on 2019-08-1.

[*Prudhomme et al.*, 2017] Prudhomme, Claire, Homburg, Timo, Ponciano, Jean-Jacques, Boochs, Frank, Roxin, Ana, and Cruz, Christophe. *Automatic integration of spatial data into the semantic web*. In *WebIST 2017* . Porto, Portugal (2017). URL http://dx.doi.org/10.5220/0006306601070115. Visited on 2019-08-1.

[*Prud'hommeaux and Seaborne*, 2008] Prud'hommeaux, Eric and Seaborne, Andy. *SPARQL Query Language for RDF*. W3C Recommendation (2008). URL http://www.w3.org/TR/rdf-sparql-query/. http://www.w3.org/TR/rdf-sparql-query/.

[*Pu et al.*, 2011] Pu, Shi, Rutzinger, Martin, Vosselman, George, and Elberink, Sander Oude. *Recognizing basic structures from mobile laser scanning data for road inventory studies*. ISPRS Journal of Photogrammetry and Remote Sensing, 66(6):S28–S39 (2011).

[*Puttonen et al.*, 2013] Puttonen, Eetu, Lehtomäki, Matti, Kaartinen, Harri, Zhu, Lingli, Kukko, Antero, and Jaakkola, Anttoni. *Improved sampling for terrestrial and mobile laser scanner point cloud data*. Remote Sensing, 5(4):1754–1773 (2013). ISSN 2072-4292. URL http://dx.doi.org/10.3390/rs5041754. Visited on 2019-08-1.

[*Qi et al.*, 2017*a*] Qi, Charles R, Su, Hao, Mo, Kaichun, and Guibas, Leonidas J. *Pointnet: Deep learning on point sets for 3d classification and segmentation*. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 652–660 (2017*a*).

[*Qi et al.*, 2017*b*] Qi, Charles Ruizhongtai, Yi, Li, Su, Hao, and Guibas, Leonidas J. *Pointnet++: Deep hierarchical feature learning on point sets in a metric space*. In *Advances in Neural Information Processing Systems*, pages 5099–5108 (2017*b*).

[*Qian et al.*, 1992] Qian, Kai, Lu, Xun, and Bhattacharya, Prabir. *A 3d object recognition system using voxel representation*. Pattern recognition letters, 13(10):725–733 (1992).

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

[*Remondino et al.*, 2008] Remondino, Fabio, El-Hakim, Sabry F, Gruen, Armin, and Zhang, Li. *Turning images into 3-d models*. IEEE Signal Processing Magazine, 25(4):55–65 (2008).

[*Ren et al.*, 2015] Ren, Shaoqing, He, Kaiming, Girshick, Ross, and Sun, Jian. *Faster r-cnn: Towards real-time object detection with region proposal networks* (2015). URL http://arxiv.org/abs/1506.01497. Cite arxiv:1506.01497Comment: Extended tech report, visited on 2019-08-1.

[*Reutebuch et al.*, 2005] Reutebuch, Stephen E, Andersen, Hans-Erik, and McGaughey, Robert J. *Light detection and ranging (lidar): an emerging tool for multiple resource inventory*. Journal of Forestry, 103(6):286–292 (2005).

[*Rousseeuw and Driessen*, 1999] Rousseeuw, Peter J and Driessen, Katrien Van. *A fast algorithm for the minimum covariance determinant estimator*. Technometrics, 41(3):212–223 (1999).

[*Rue and Held*, 2005] Rue, Havard and Held, Leonhard. *Gaussian Markov random fields: theory and applications*. Chapman and Hall/CRC (2005).

[*Rusu and Cousins*, 2011] Rusu, Radu B and Cousins, S. *Point cloud library (pcl)*. In *2011 IEEE International Conference on Robotics and Automation*, pages 1–4 (2011).

[*Sareen et al.*, 2010] Sareen, K. K., Knopf, G. K., and Canas, R. *Rapid clustering of colorized 3d point cloud data for reconstructing building interiors*. In *2010 International Symposium on Optomechatronic Technologies*, pages 1–6 (2010). URL http://dx.doi.org/10.1109/ISOT.2010.5687331.

[*Schnabel et al.*, 2007] Schnabel, Ruwen, Wahl, Roland, and Klein, Reinhard. *Efficient ransac for point-cloud shape detection*. In *Computer graphics forum*, volume 26, pages 214–226. Wiley Online Library (2007).

[*Sedlacek and Zara*, 2009] Sedlacek, David and Zara, Jiri. *Graph cut based point-cloud segmentation for polygonal reconstruction*. In *International Symposium on Visual Computing*, pages 218–227. Springer (2009).

[*Serna and Marcotegui*, 2014] Serna, Andrés and Marcotegui, Beatriz. *Detection, segmentation and classification of 3d urban objects using mathematical morphology and supervised learning*. ISPRS Journal of Photogrammetry and Remote Sensing, 93:243–255 (2014).

[*Serna et al.*, 2014] Serna, Andrés, Marcotegui, Beatriz, Goulette, François, and Deschaud, Jean-Emmanuel. *Paris-rue-madame database - a 3d mobile laser scanner dataset for benchmarking urban detection, segmentation and classification methods*. In *ICPRAM* (2014).

[*Shearer et al.*, 2008] Shearer, Rob, Motik, Boris, and Horrocks, Ian. *Hermit: A highly-efficient owl reasoner.* In *OWLED*, volume 432, page 91 (2008).

[*Simonyan and Zisserman*, 2014] Simonyan, Karen and Zisserman, Andrew. *Very deep convolutional networks for large-scale image recognition*. arXiv preprint arXiv:1409.1556 (2014).

[*Sirin et al.*, 2007] Sirin, Evren, Parsia, Bijan, Grau, Bernardo Cuenca, Kalyanpur, Aditya, and Katz, Yarden. *Pellet: A practical owl-dl reasoner*. Web Semantics: science, services and agents on the World Wide Web, 5(2):51–53 (2007).

[*Son et al.*, 2015] Son, Hyojoo, Bosché, Frédéric, and Kim, Changwan. *As-built data acquisition and its use in production monitoring and automated layout of civil infrastructure: A survey*. Advanced Engineering Informatics, 29(2):172–183 (2015).

[*Strom et al.*, 2010] Strom, J., Richardson, A., and Olson, E. *Graph-based segmentation for colored 3d laser point clouds*. In *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2131–2136 (2010). ISSN 2153-0866. URL http://dx.doi.org/10.1109/IROS.2010.5650459.

[*Studer et al.*, 1998] Studer, Rudi, Benjamins, Richard, and Fensel, Dieter. *Knowledge engineering: Principles and methods*. Data & Knowledge Engineering, 25(1-2):161–198 (1998).

[*Sun et al.*, 2018] Sun, Z, Xu, Y, Hoegner, L, and Stilla, U. *Classification of mls point clouds in urban scenes using detrended geometric features from supervoxel-based local contexts.* ISPRS Annals of Photogrammetry, Remote Sensing & Spatial Information Sciences, 4(2) (2018).

[*Szegedy et al.*, 2015] Szegedy, Christian, Liu, Wei, Jia, Yangqing, Sermanet, Pierre, Reed, Scott, Anguelov, Dragomir, Erhan, Dumitru, Vanhoucke, Vincent, and Rabinovich, Andrew. *Going deeper with convolutions*. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9 (2015).

[*Tarsha-Kurdi et al.*, 2008] Tarsha-Kurdi, Fayez, Landes, Tania, and Grussenmeyer, Pierre. *Extended ransac algorithm for automatic detection of building roof planes from lidar data*. The Photogrammetric Journal of Finland, 21:97–109 (2008).

[*Tchapmi et al.*, 2017] Tchapmi, Lyne, Choy, Christopher, Armeni, Iro, Gwak, Jun-Young, and Savarese, Silvio. *Segcloud: Semantic segmentation of 3d point clouds*. In *2017 International Conference on 3D Vision (3DV)*, pages 537–547. IEEE (2017).

[*Tomaszuk*, 2016] Tomaszuk, Dominik. *Inference rules for rdf(s) and owl in n3logic*. ArXiv, abs/1601.02650 (2016).

[*Tombari et al.*, 2010] Tombari, Federico, Salti, Samuele, and Di Stefano, Luigi. *Unique signatures of histograms for local surface description*. In *European conference on computer vision*, pages 356–369. Springer (2010).

[*Tombari et al.*, 2011] Tombari, Federico, Salti, Samuele, and Di Stefano, Luigi. *A combined texture-shape descriptor for enhanced 3d feature matching*. In *2011 18th IEEE international conference on image processing*, pages 809–812. IEEE (2011).

[*Tongphu et al.*, 2012] Tongphu, S., Suntisrivaraporn, B., Uyyanonvara, B., and Dailey, M. N. *Ontology-based object recognition of car sides*. In *2012 9th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology*, pages 1–4 (2012). URL http://dx.doi.org/10.1109/ECTICon.2012.6254268.

[*Torr and Zisserman*, 2000] Torr, Philip HS and Zisserman, Andrew. *Mlesac: A new robust estimator with application to estimating image geometry*. Computer vision and image understanding, 78(1):138–156 (2000).

[*Torralba et al.*, 2010] Torralba, Antonio, Murphy, Kevin P, and Freeman, William T. *Using the forest to see the trees: exploiting context for visual object detection and localization*. Communications of the ACM, 53(3):107–114 (2010).

[*Triboan et al.*, 2017] Triboan, Darpan, Chen, Liming, Chen, Feng, and Wang, Zumin. *Semantic segmentation of real-time sensor data stream for complex activity recognition*. Personal and Ubiquitous Computing, 21(3):411–425 (2017). ISSN 1617-4917. URL http://dx.doi.org/10.1007/s00779-017-1005-5. Visited on 2019-08-1.

[*Triebel et al.*, 2007] Triebel, Rudolph, Schmidt, Richard, Mozos, Óscar Martínez, and Burgard, Wolfram. *Instance-based amn classification for improved object recognition in 2d and 3d laser range data*. In *Proceedings of the 20th international joint conference on Artifical intelligence*, pages 2225–2230. Morgan Kaufmann Publishers Inc. (2007).

[*Tsarkov and Horrocks*, 2006] Tsarkov, Dmitry and Horrocks, Ian. *Fact++ description logic reasoner: System description*. In *International joint conference on automated reasoning*, pages 292–297. Springer (2006).

[*Van Aelst and Rousseeuw*, 2009] Van Aelst, Stefan and Rousseeuw, Peter. *Minimum volume ellipsoid*. Wiley Interdisciplinary Reviews: Computational Statistics, 1(1):71–82 (2009).

[*Velizhev et al.*, 2012] Velizhev, Alexander, Shapovalov, Roman, and Schindler, Konrad. *Implicit shape models for object detection in 3d point clouds*. In *International Society of Photogrammetry and Remote Sensing Congress*, volume 2 (2012).

[*Vo et al.*, 2015] Vo, Anh-Vu, Truong-Hong, Linh, Laefer, Debra F., and Bertolotto, Michela. *Octree-based region growing for point cloud segmentation*. ISPRS Journal of Photogrammetry and Remote Sensing, 104:88 – 100 (2015). ISSN 0924-2716. URL http://dx.doi.org/https://doi.org/10.1016/j.isprsjprs.2015.01.011. Visited on 2019-08-1.

[*Vosselman*, 2013] Vosselman, George. *Point cloud segmentation for urban scene classification*. Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci, 1:257–262 (2013).

[*Vosselman et al.*, 2001] Vosselman, George, Dijkman, Sander et al. *3d building model reconstruction from point clouds and ground plans*. International archives of photogrammetry remote sensing and spatial information sciences, 34(3/W4):37–44 (2001).

[*Vrubel et al.*, 2009] Vrubel, Alexandre, Bellon, Olga RP, and Silva, Luciano. *A 3d reconstruction pipeline for digital preservation*. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 2687–2694. IEEE (2009).

[*Walsh et al.*, 2013] Walsh, Sara B, Borello, Daniel J, Guldur, Burcu, and Hajjar, Jerome F. *Data processing of point clouds for object detection for structural engineering applications*. Computer-Aided Civil and Infrastructure Engineering, 28(7):495–508 (2013).

[*Wang et al.*, 2007] Wang, Hai H., Li, Yuan Fang, Sun, Jing, Zhang, Hongyu, and Pan, Jeff. *Verifying feature models using owl*. Journal of Web Semantics, 5(2):117 – 129 (2007). ISSN 1570-8268. URL http://dx.doi.org/https://doi.org/10.1016/j.websem.2006.11.006. Visited on 2019-08-1.

[*Wefers*, 2015] Wefers, S. *La cascade de direction efesos*. Historique des études techniques sur la fourniture d'une antiquité tardive à la ville byzantine. Monographies RGZM, 118 (2015).

[*Weinmann et al.*, 2015*a*] Weinmann, Martin, Jutzi, Boris, Hinz, Stefan, and Mallet, Clément. *Semantic point cloud interpretation based on optimal neighborhoods, rel-*

evant features and efficient classifiers. ISPRS Journal of Photogrammetry and Remote Sensing, 105:286–304 (2015*a*).

[*Weinmann et al.*, 2014] Weinmann, Martin, Jutzi, Boris, and Mallet, Clément. *Semantic 3d scene interpretation: A framework combining optimal neighborhood size selection with relevant features*. ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences, 2(3):181 (2014).

[*Weinmann et al.*, 2015*b*] Weinmann, Martin, Urban, Steffen, Hinz, Stefan, Jutzi, Boris, and Mallet, Clément. *Distinctive 2d and 3d features for automated large-scale scene analysis in urban areas*. Computers & Graphics, 49:47–57 (2015*b*).

[*Wirges et al.*, 2018] Wirges, Sascha, Fischer, Tom, Stiller, Christoph, and Frias, Jesus Balado. *Object detection and classification in occupancy grid maps using deep convolutional networks*. In *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, pages 3530–3535. IEEE (2018).

[*Woo et al.*, 2002] Woo, H., Kang, E., Wang, Semyung, and Lee, Kwan H. *A new segmentation method for point cloud data*. International Journal of Machine Tools and Manufacture, 42(2):167 – 178 (2002). ISSN 0890-6955. URL http://dx.doi.org/https://doi.org/10.1016/S0890-6955(01)00120-1. Visited on 2019-08-1.

[*Wu et al.*, 2018] Wu, T., Liu, J., Li, M., Chen, R., and Hyyppä, J. *Automated large scale indoor reconstruction using vehicle survey data*. In *2018 Ubiquitous Positioning, Indoor Navigation and Location-Based Services (UPINLBS)*, pages 1–5 (2018). URL http://dx.doi.org/10.1109/UPINLBS.2018.8559828.

[*Xiang et al.*, 2018] Xiang, Binbin, Yao, Jian, Lu, Xiaohu, Li, Li, Xie, Renping, and Li, Jie. *Segmentation-based classification for 3d point clouds in the road environment*. International journal of remote sensing, 39(19):6182–6212 (2018).

[*Xing et al.*, 2018] Xing, Xu-Feng, Mostafavi, Mir-Abolfazl, and Chavoshi, Seyed. *A knowledge base for automatic feature recognition from point clouds in an urban scene*. ISPRS International Journal of Geo-Information, 7(1):28 (2018).

[*Xu et al.*, 2016] Xu, Bo, Jiang, Wanshou, Shan, Jie, Zhang, Jing, and Li, Lelin. *Investigation on the weighted ransac approaches for building roof plane segmentation from lidar point clouds*. Remote Sensing, 8(1) (2016). ISSN 2072-4292. URL http://dx.doi.org/10.3390/rs8010005.

[*Yang et al.*, 2017] Yang, Bisheng, Dong, Zhen, Liu, Yuan, Liang, Fuxun, and Wang, Yongjun. *Computing multiple aggregation levels and contextual features for road* *facilities recognition using mobile laser scanning data*. ISPRS Journal of Photogrammetry and Remote Sensing, 126:180–194 (2017).

[*Yang et al.*, 2015] Yang, Bisheng, Dong, Zhen, Zhao, Gang, and Dai, Wenxia. *Hierarchical extraction of urban objects from mobile laser scanning data*. ISPRS Journal of Photogrammetry and Remote Sensing, 99:45–57 (2015).

[*Yang et al.*, 2018] Yang, Zetong, Sun, Yanan, Liu, Shu, Shen, Xiaoyong, and Jia, Jiaya. *Ipod: Intensive point-based object detector for point cloud*. arXiv preprint arXiv:1812.05276 (2018).

[*Yohai et al.*, 1987] Yohai, Victor J et al. *High breakdown-point and high efficiency robust estimates for regression*. The Annals of Statistics, 15(2):642–656 (1987).

[*Zeng et al.*, 2018] Zeng, Jin, Cheung, Gene, Ng, Michael, Pang, Jiahao, and Yang, Cheng. *3d point cloud denoising using graph laplacian regularization of a low dimensional manifold model*. arXiv preprint arXiv:1803.07252 (2018).

[*Zhan et al.*, 2009] Zhan, Q, Liang, Yubin, and Xiao, Y. *Color-based segmentation of point clouds*. Laser scanning, 38(3):155–161 (2009).

[*Zhang et al.*, 2013] Zhang, Jie, Cao, Junjie, Liu, Xiuping, Wang, Jun, Liu, Jian, and Shi, Xiquan. *Point cloud normal estimation via low-rank subspace clustering*. Computers & Graphics, 37(6):697–706 (2013).

[*Zheng et al.*, 2019] Zheng, Mingxue, Wu, Huayi, and Li, Yong. *An adaptive end-to-end classification approach for mobile laser scanning point clouds based on knowledge in urban scenes*. Remote Sensing, 11(2):186 (2019).

[*Zhou and Tuzel*, 2018] Zhou, Yin and Tuzel, Oncel. *Voxelnet: End-to-end learning for point cloud based 3d object detection*. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4490–4499 (2018).

List of Figures

List of Tables

Part VI

Appendix

A Algorithms processing

The structuring of algorithm knowledge explained in Section (10.4) allows the easy integration of specialized computer vision libraries (such as PCL or openCV). These libraries provide a wide variety of algorithms for processing data.

There are three essential aspects considered in the modeling of algorithms: The first aspect is the role that each parameter fulfills for the algorithm. The second aspect is the influence on the result of each parameter. Finally, the last aspect is the influence of one parameter on another, the latter being the most complicated aspect to understand. It is modelled using equations to relate the various influences of a parameter.

The main algorithms described to handle most application cases, and in particular, those presented in Section 2.1, are divided into three categories: preprocessing, segmentation, and feature extraction. Preprocessing algorithms prepare the data for the processing. Segmentation algorithms isolate and detect elements. Finally, the algorithms for extracting features allow characterizing portions of data (segments) for the identification of objects by classification (explained in Section 11.2).

In the following sections, we will describe with the same formalism the different algorithms that we have implemented in our work in order to process the study cases presented in Section 2.1.

The objective of this Section is not to be exhaustive but to illustrate the generic nature of the modeling implemented.

A.1 Data preprocessing

Data loader

Files in different formats allow for storing 3D point clouds. Depending on the format, the file data is encoded either in binary or Ascii. Among the different formats, the most commonly used are as follows: PCD, PLY, XYZ, XYZRGB, and XYZRGBNxNyNyNz.

For each of these formats, specialized algorithms, called "data loader", are required to extract the information from the files and load the point cloud into memory. These algorithms sequentially read the information contained in a file to create a representation of this information as a 3D point cloud that can contain color or orientation. This representation is accessible in memory through a pointer.

These algorithms work on a file to produce a point cloud. To do this, they use the path of the data file to load (the data *source*) as input. Furthermore, each algorithm supports a format. This format is a prerequisite for the algorithm. The format and source, which are data features, allow linking the processed data to the suitable data loader algorithm.

Structuring

The treatment of large point clouds requires subdividing it into subsets with smaller size. Indeed, the larger is point cloud used by algorithms, the more time-consuming is the execution of the algorithms. Therefore, dividing a cloud into small subsets reduces the complexity of algorithms execution. The most common method for subdividing a point cloud is to use an octree. Indeed, an octree allows dividing several times a point cloud into subsets until reaching the desired size of the subsets. Thanks to its usage, the size of the subsets used by the algorithms can be effectively controlled and optimized for each algorithm. Figure A.1 shows the structure of an octree.

The process to build an octree consists in dividing each node into eight sub-nodes. Initially, a volume defines the maximum size of a tree leaf. This volume is the only parameter of an octree building algorithm. As long as the tree has at least one leaf bigger than this volume, it continues to divide these leaves into eight.

The generation of an octree has a logarithmic complexity, which allows it to perform even on large point clouds (several million points).

Sampling

When the point clouds are massive, the subdivision of this cloud into subsets is not always sufficient. Handling massive point clouds may require reducing the complexity of the point cloud by performing a sampling or compression of the point cloud.

A commonly used sampling algorithm consists of using an octree to sampling the point cloud according to a specific volume. Thus, the algorithm, called "*Metric*

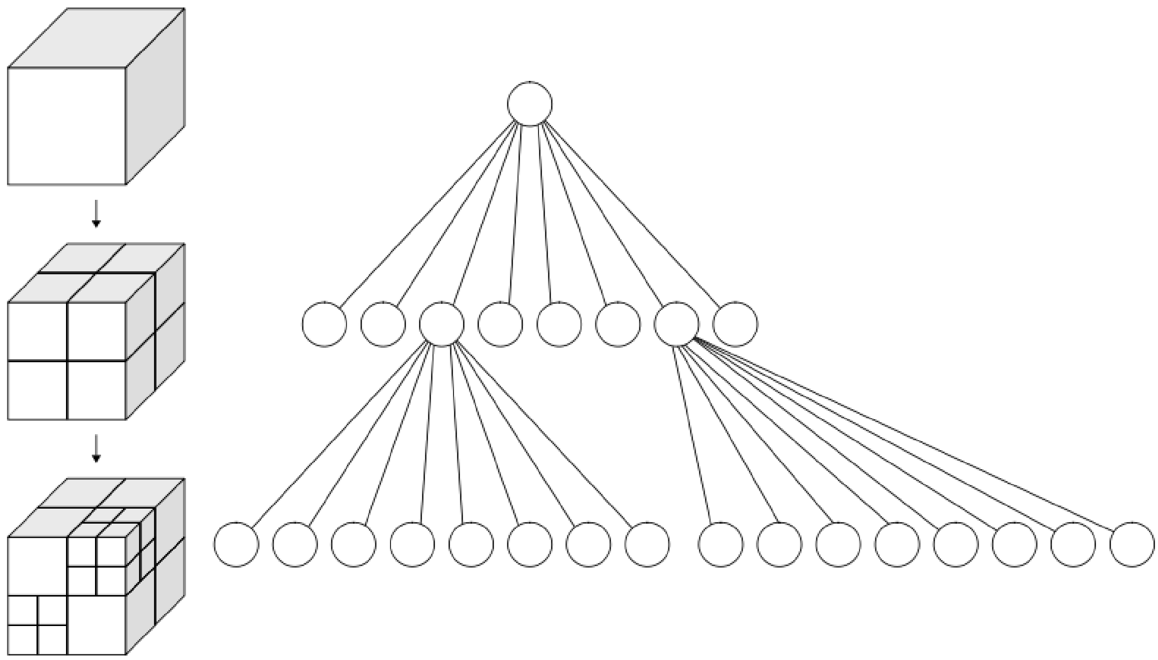


Figure A.1: Diagram of an Octree.

Sampling" subdivides the point cloud into subsets and then calculate for each subset a single point representing it. This compression of a subset into a single point allows an efficient reduction of the point cloud. However, the subdivision of the point cloud into a subset can separate an object into several distinct subsets. A too intensive subdivision can lead to an over-segmentation of the object.

Therefore, the maximum volume defined as a parameter of the algorithm must be determined according to the object to be detected. Figure A.2 shows the choice of the size of the volume (in red) for the detection of a sidewalk.

The volume computation depends on the size of the object (H for height, W for width, L for length). Its computation follows Equation A.1.

$$DataVolume = \frac{H \times L \times W}{100} \tag{A.1}$$

Equation A.1 has been empirically determined by tests performed on the different application cases presented in Section 2.1.

Figure A.3 shows the sampling of a point cloud by the "*Metric Sampling*" algorithm.

This algorithm works on an octree, which is a type of data and produces a sampled

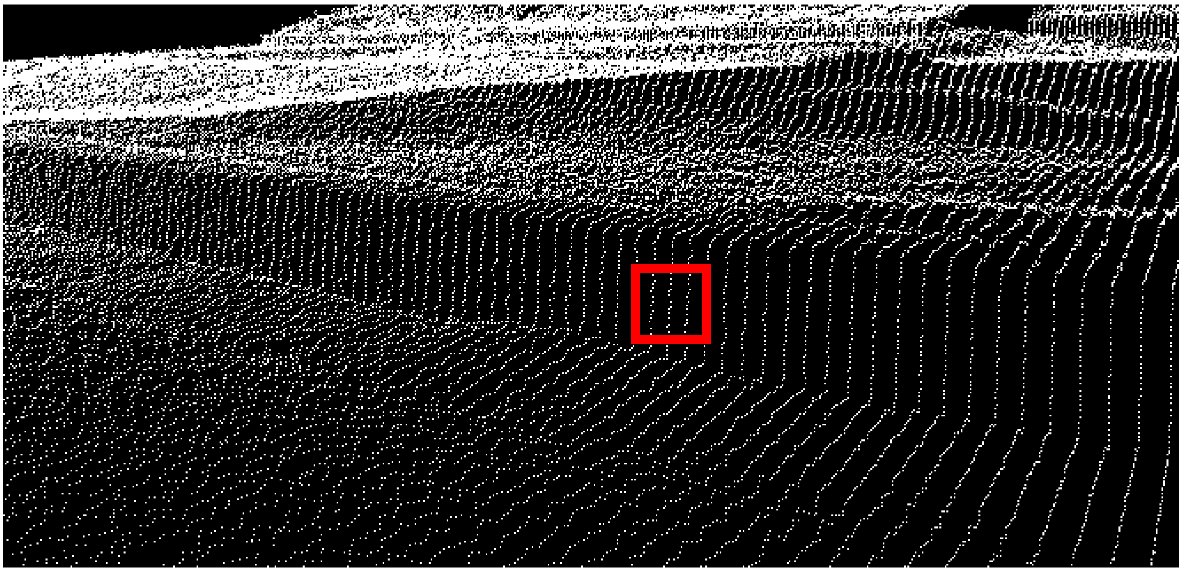
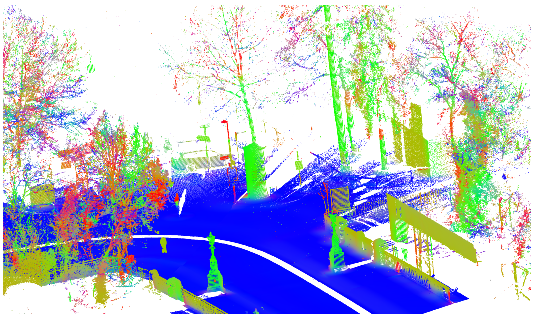
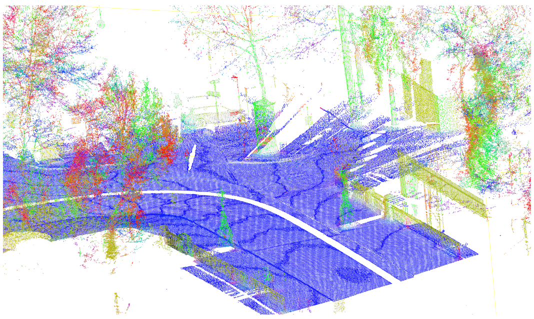


Figure A.2: Illustration of the volume choice in red for a sidewalk.



(a) Origin point cloud.



(b) Point cloud after sampling.

Figure A.3: Illustration of the sampling process (point clouds are colored for better viewing) point cloud, corresponding to a point cloud with a reduced number of point.

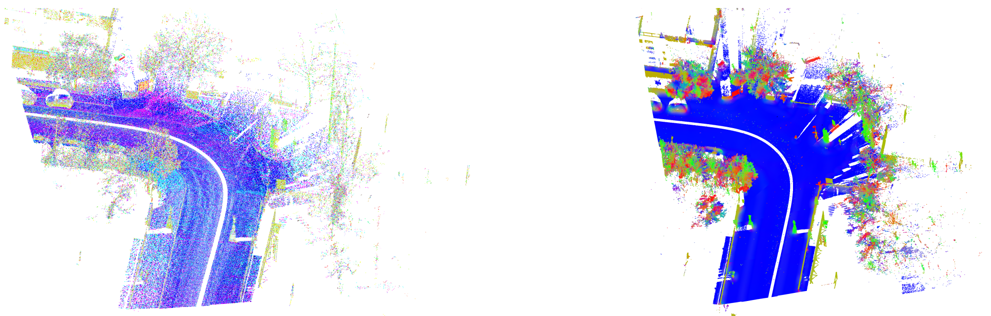
A.2 Segmentation

The purpose of segmentation algorithms is to characterize the data and divide them into segments describing a particular geometry of an object.

Normal estimation

The surface of an object is one of the most important features for identifying the geometry of an object. In 3D point clouds, the surface of an object is described by the normal of each point composing the surface. Thus it is necessary to estimate the normals at each point if the original cloud has not one. Commonly the calculation of the normal of a point is done according to these two nearest neighbors. However, calculating the normal of a point based only on its two nearest neighbors does not provide a segmentation with high quality.

This problem of quality can be solved by performing plane detection algorithms around the point to describe. The plane detection, as explained in Section A.3, aims at determining its tangent at the point whose, the normal must be calculated. These results are compared in Figure A.4.



(a) Normal estimation by taking the two closest neighbors of each point.

(b) Normal estimation using RANSAC to detect planes.

Figure A.4: Comparison of the results of the normal estimation. Normals are colored in red, green, and blue, according to their X,Y,Z values respectively.

The implementation of this type of algorithm called "*Normal Estimation*" requires having a radius to select the neighbors of each point to form subsets. These subsets are then used to detect the tangent plane at the investigated point. The radius is calculated according to both the dimensions of the object searched and the resolution of the data acquisition system. This radius corresponds to the calculation of the diagonal of a rectangular parallelepiped formed by the maximum dimensions (H for height, W for width, L for length) of the object divided by the density of the data. Equation A.2 is the computation of the radius.

$$Radius = \frac{\sqrt{H^2 + W^2 + L^2}}{Density} \tag{A.2}$$

Thus, the smaller the object sought is, and the higher the density of the data is, the smaller the radius is. The respect of this proportion allows a more accurate estimation of the normals for each point.

Moreover, the determination of the tangent plane requires a precision value to define whether or not a point belongs to a plane (as explained in Section A.3). This precision value is called "Tolerance" and is computed by Equation A.3.

$$Tolerance = \frac{\sum_{i=1}^{i=n} min(\sum_{j=1}^{j=n} \sqrt{(p_i.x - p_j.x)^2 + (p_i.y - p_j.y)^2 + (p_i.z - p_j.z)^2})}{n} \tag{A.3}$$

Thus, the "*Normal Estimation*" algorithm works on a point cloud to produce a point cloud with normals, and it requires a *radius* and a *tolerance* as a parameter.

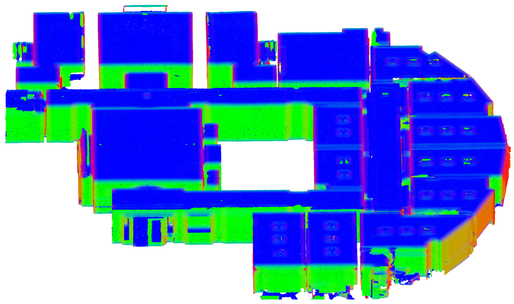
Filtering

Although some objects can have complex shapes, most objects, especially large objects such as floors, walls, and ceilings, have a regular surface. Indeed, the surfaces of these objects are composed only of a limited set of points with a different normal. Thus, filtering the point cloud to keep only the points with the same normal characterizing the surface simplifies the detection process. Therefore, filtering the point cloud according to a normal allows for detecting such type of object. A significant number of these object types are present in the application cases presented in Section 2.1.

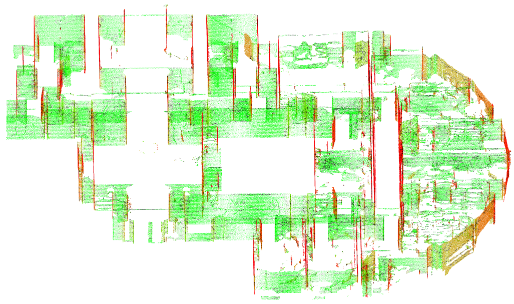
Thus, the algorithm called "*Normal Filtering*" aims at filtering the data by keeping only points which have a possible orientation of the object. It can thus produce a filtered point cloud, from a point cloud whose the normals are estimated.

The estimation of normals at each point of the point cloud is a prerequisite for

"*Normal Filtering*". Figure A.5 shows the result of filtering a point cloud that only keeps points with a horizontal normal.



(a) Origin point cloud.



(b) Point cloud after filtering retaining only vertical points.

Figure A.5: Illustration of the filtering of a point cloud.

The "*Normal Filtering*" algorithm requires a "*normal set*" as a parameter to be able to filter the points to be collected. The prerequisite of the algorithm is, thus, the estimation of normals of the point cloud. It also requires a "*normal tolerance*" parameter to determine if the normal of a point belongs or not to the set of normals provided. Its calculation depends on the roughness of the object and the density of the point cloud by Equation A.4.

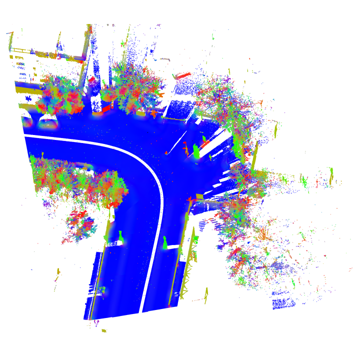
$$NormalTolerance = \frac{Roughness^2}{Density} \tag{A.4}$$

Region growing

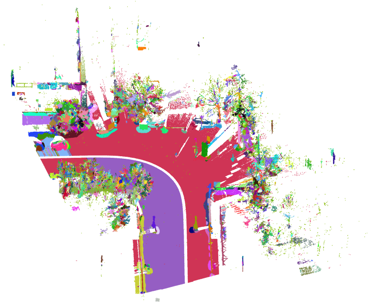
The segmentation of data to detect objects composed of various surfaces (e.g. traffic signs, car, tree) can be carried out in different ways, as presented in the literature review in Section 6.1. Among the different segmentation algorithms, algorithms based on growing regions have the advantage of being quite flexible. They can be used in different application cases and can be adapted to detect different objects. These algorithms use a gregarious criterion to form regions and segment the data. Among the different possible criteria, color and the normal at a point are proper representative criteria of objects in most cases.

These algorithms work on small point clouds to produce segments. Figure A.6 shows the application of a region growing to segment the different surfaces of a data.

A.2. SEGMENTATION



(a) Origin point cloud.



(b) Point cloud after segmentation (a unique color is assigned to each segment.

Figure A.6: Results of a segmentation by a region growing algorithm.

They require a point cloud with a small size (e.g. not exceed one million points) and without noise. They also need a "*radius*" as a parameter. This parameter corresponds to the maximum distance between a point and the other points constituting a region for the point considered as being in the region. It is calculated according to Equation A.2.

They also need a "*tolerance*" parameter, which is the maximum deviation between the criteria (used for the segmentation) of two points to be considered in the same region. Tolerance depends on the type of criteria used.

Two versions of this type of algorithm are implemented to consider two essential characteristics of objects: their surface and their color. Thus an algorithm called "*Normal Region Growing*" aims to segment the data according to the normal of each point. It is a prerequisite that the normals at each point of the point cloud are estimated. The *tolerance* of this algorithm is calculated by Equation A.4.

The second algorithm, called "*Color Region Growing*," aims at segmenting the data according to the color of each point. It is a prerequisite to apply it on the colored point clouds. The *"color tolerance"* is directly related to the *"color variation"* characteristic of the object. The light of the scene does not impact the color variation but only the colors.

A.3 Features extraction

Features are a crucial concept that guides the process of detection through selecting and parameterizing algorithms. The algorithm features are linked to objects and data features. An expert can firstly provide some feature of the data. However, the more precise the data have characteristics, the more efficiently the detection system can select, parameterize, and execute algorithms. That is why the system requires extraction algorithms for characterizing data. Secondly, the segments resulting from the segmentation must be characterized to identify the object represented by this segment. Therefore, the system requires also extraction algorithms for characterizing these segments according to object features. Thus the feature extraction algorithms must allow detecting the characteristics of the data described in Section 10.2 and the objects described in Section 10.3. Therefore, two main types of algorithms are necessary, "getter algorithm" and "Shape recognition algorithm".

Getter algorithm: Getter algorithms extract a specific feature from the data. They do not take any parameters and work on data or specific data, which is a segment (created by data segmentation processes).

The list of the main getter algorithms are as follows:

GetHeight : characterizes the height of a segment.

GetWidth : characterizes the width of a segment.

GetLength : characterizes the length of a segment.

GetVolume : characterizes the volume of a segment.

GetArea : characterizes the area of a segment.

GetLocation : characterizes the location of a segment.

GetPointCount : characterizes the point number of a point cloud.

GetMeanColor : characterizes the mean color of a segment.

GetMeanNormal : characterizes the mean normal of a segment.

GetResolution : characterizes the resolution of a point cloud.

GetDistance : characterizes the euclidean distance between two segments.

Shape recognition algorithm

There is a multitude of algorithms to detect geometric shapes in 3D point clouds. The most commonly used algorithms are the algorithms from model-driven approaches that extract characteristics on the shape of the object, as explained in chapter 5. Among these different algorithms, the RANSAC algorithm discussed in Section 5.2 is an effective method of detecting geometric shapes such as lines, planes, and spheres. Figure A.7 shows the organizational chart of the RANSAC algorithm. Thus three different implementations of RANSAC is created to detect lines, planes,

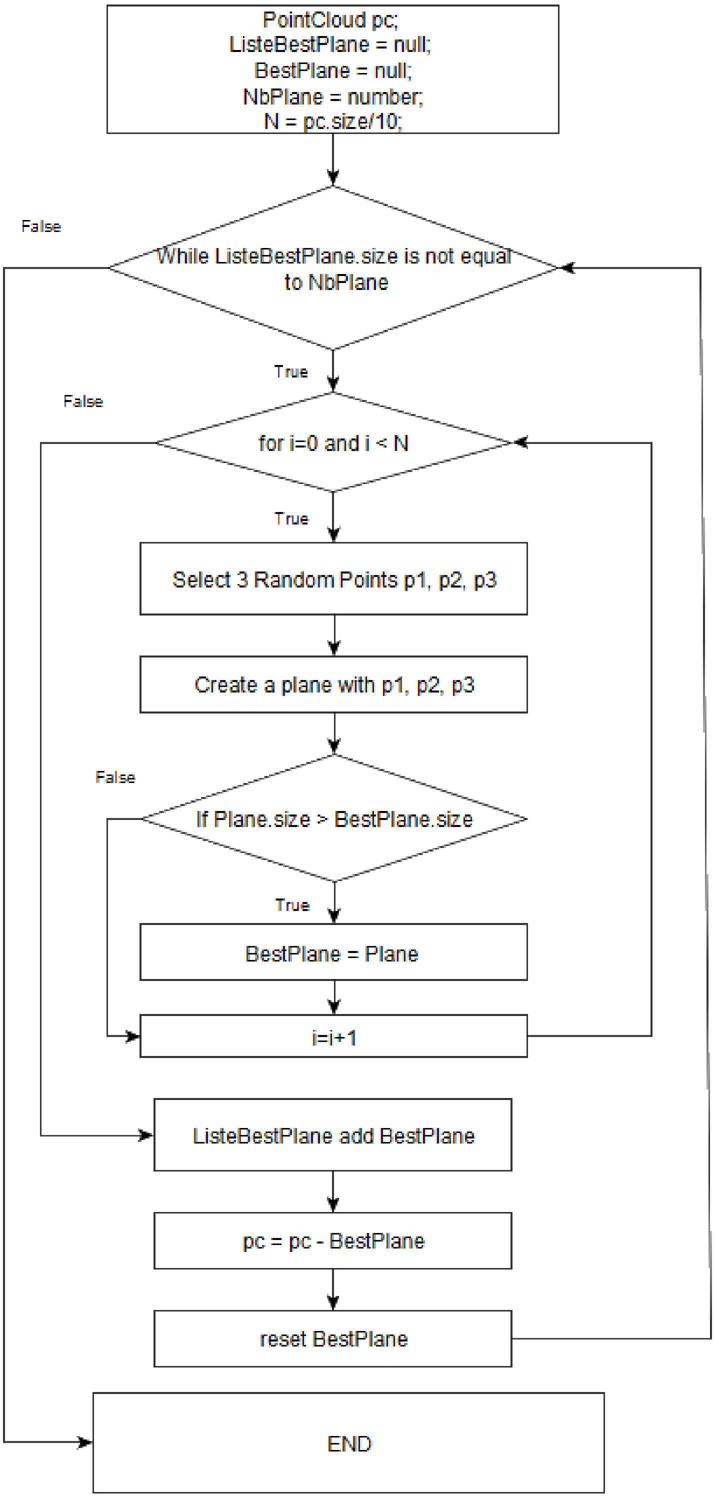


Figure A.7: Organizational chart of the RANSAC algorithm.

and spheres in the data. These three algorithms work on segments. The first algorithm, called "*RANSAC line detection*" computes the mathematical model of a line to produce lines that characterize the segment. The second algorithm, called "*RANSAC plane detection*" computes the mathematical model of a plane to produce planes that characterize the segment. The third algorithm, called "*RANSAC sphere detection*" computes the mathematical model of a sphere to produce spheres that characterize the segment. The mathematical models are constructed by a random selection of points in the segment. The number of points selected depends on the model: two points for a line, three for a plane and four for a sphere.

However, the points constituting the segment do not perfectly match the calculated mathematical model. Therefore, these algorithms need a distance as a parameter. This parameter aims to threshold the points of the segment and the computed mathematical model. This distance is called "tolerance". It is calculated according to the density of the segment by Equation A.3 previously explained. This equation uses n the number of point of the segment and p_i the i^{th} point of the segment.

This tolerance allows effective shape detection in point clouds, even if they are noisy. Figure A.8 shows the result of a sphere detection by RANSAC algorithm in a noisy point cloud.

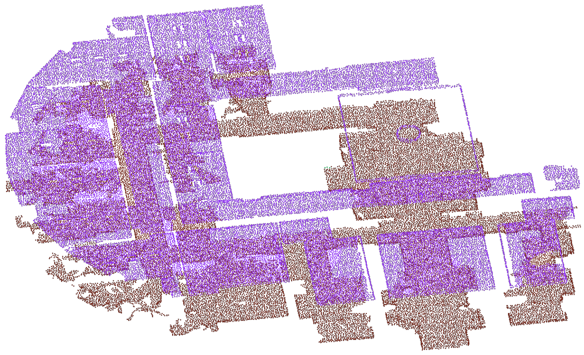
(a) Origin point cloud. (b) Point cloud after sphere detection.

Figure A.8: Results of detection of a unique sphere by the RANSAC algorithm.

Another essential parameter for these three algorithms is the number of iterations needed to find the best mathematical models in the segment. This iteration count is computed according to the number of points in the segment. It corresponds to 10% of the number of points in the segment. Such parameterization provides flexibility and efficiency in detecting geometric shapes, as illustrated in Figure A.9 that shows the detection of two planes in a point cloud composed of more than 22 Million points.

Each iteration produces a desired geometric model (plane, line, or sphere) containing points of the segments. The number of models selected is equivalent to the

(a) Origin point cloud.

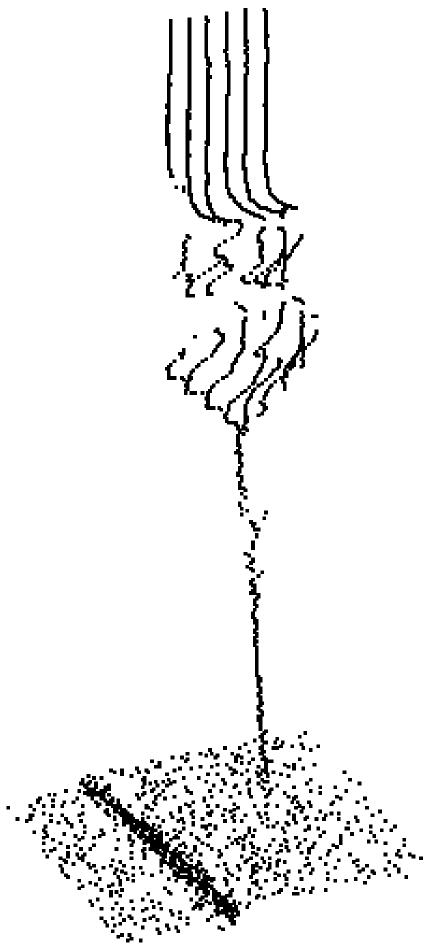


(b) Point cloud after plane detection (a unique color is assigned at each plane).

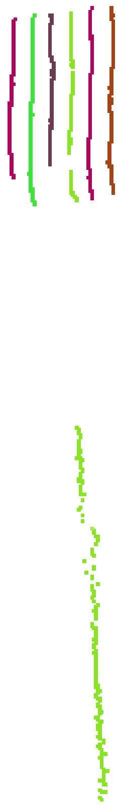
Figure A.9: Results of detection of two planes by the RANSAC algorithm.

number of mathematical models required to detect an object. This number is a parameter common to all three algorithms and is called "number of elements". For example, let us take the case of a traffic sign defined as consisting of six main lines. The RANSAC algorithm corresponding to line detection will, therefore, search for six lines in the segments to allow their identification. Figure A.10 shows the result of the detection of six lines by the RANSAC algorithm.

The three algorithms for line, plane and sphere detection have all the same input ("*segment*") and parameters ("*tolerance*", "*iteration count*", "*number of element*"). Moreover, they do not have any prerequisites, only what they produce (line, plane, or sphere) differs.



(a) Origin point cloud.



(b) Point cloud after lines detection (a unique color is assigned at each line).

Figure A.10: Results of lines detection obtained by the RANSAC algorithm.

B Object detection overview

The detection of objects and geometries in the data is entirely knowledge-driven. Thus, it provides knowledge that is then analyzed in a learning phase to generate new and more accurate knowledge about the application case and thus improve the effectiveness of detection. Figure B.1 shows an overview of facades detection process in an urban point cloud.

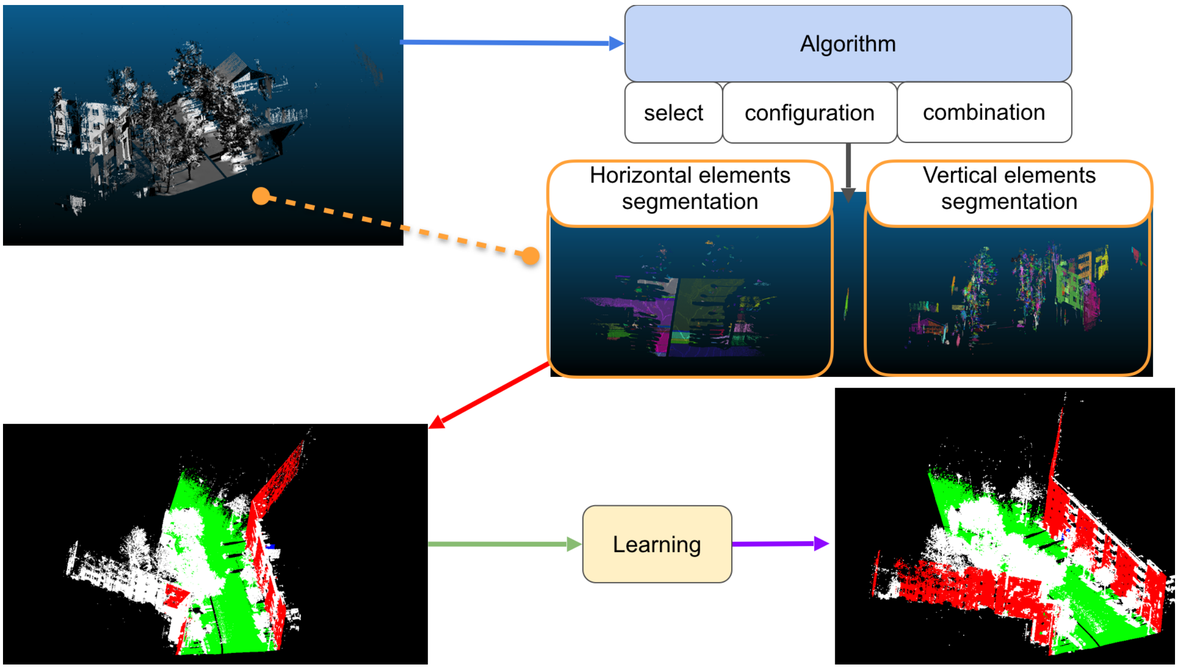


Figure B.1: Overview for an example of facades detection process.

Data processing is performed using algorithms (illustrated by the blue arrow in Figure B.1). These algorithms must be selected, configured and combined (shown in the blue box in Figure B.1) according to the application case (e.g. acquisition context, objects sought) and the prerequisites of each algorithm (e.g. low noise, high density, estimated normal).

Let us take the example of a facade defined as a planar surface perpendicular to the ground with a height of at least 12 meters. The detection of this facade requires algorithms for plane detection, size estimation, and topological link assessment (e.g. parallel, perpendicular). These algorithms may have prerequisites such as normal estimation, or whether the data is small in size or not noisy. Thus other algorithms such as normal estimation, denoising, sampling algorithms must be in some cases combined in a sequence to satisfy the needs of some selected algorithms.

The results provided by each executed algorithm are analyzed and correlated (shown by orange frames in Figure B.1) to identify objects and geometries in the data (shown by the red arrow in Figure B.1). In this case, the analysis and correlation of algorithms results allow identifying ground (in green) and some facades (in red).

The management of algorithms and the interpretation of results are entirely driven by reasoning. This reasoning is based on the knowledge presented in chapter 10. However, it is not possible to formulate the knowledge in such a way as to describe all possible cases.

In the case illustrated in Figure B.1, parts of the facades are not detected because the information on these parts (height less than 12 meters, not connected to the ground) differs from the knowledge on the facades (height greater than 12 meters and perpendicular to the ground). The discrepancy between the information collected and the knowledge at the preliminary stage may be due to multiple factors (e. g. sensitivity of the acquisition instrument, acquisition condition, insufficient light, objects too far away). Therefore the knowledge needs to be adjusted according to the application case and the information computed.

Knowledge adaptation requires understanding how objects are represented in the data to anticipate possible variations and to compensate for the discrepancy between the information obtained and the knowledge. Understanding the representation of objects and geometries ("learning" frame in Figure B.1), requires analyzing the objects and geometry detected to formulate hypotheses on the characteristics allowing them to be better identified.

In the case presented by Figure B.1, the analysis of the detected facades allows inferring that facades can have a height between 10 and 13 meters and may not be connected to the ground that has discontinuities (areas without ground). This new knowledge is integrated with prior knowledge and allows changing the behavior of the detection process. Thus the detection strategy becomes specialized for the application case, which leads to better results (shown by the purple arrow in Figure B.1).