



HAL
open science

3D urban cartography incorporating recognition and temporal integration

Ahmad Kamal Aijazi

► **To cite this version:**

Ahmad Kamal Aijazi. 3D urban cartography incorporating recognition and temporal integration. Other [cond-mat.other]. Université Blaise Pascal - Clermont-Ferrand II, 2014. English. NNT: 2014CLF22528 . tel-01166334

HAL Id: tel-01166334

<https://theses.hal.science/tel-01166334>

Submitted on 22 Jun 2015

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

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

Numéro d'ordre D.U. : 2528
EDSPIC : 678

UNIVERSITÉ BLAISE PASCAL – CLERMONT II

Ecole Doctorale Sciences Pour l'Ingénieur de Clermont-Ferrand

Thèse présentée par

AHMAD KAMAL AIJAZI

pour obtenir le grade de

DOCTEUR D'UNIVERSITÉ

Spécialité : Vision pour la robotique

3D Urban Cartography Incorporating Recognition and Temporal Integration

Soutenue publiquement le 15 décembre 2014 devant le jury :

M. Thierry Chateau	PR - Université Blaise Pascal	Président
M. Robert Zlot	DR - CSIRO Australia	Rapporteur et examinateur
M. Uwe Stilla	PR - Technische Universitaet Muenchen	Rapporteur et examinateur
M. Nicolas Paparoditis	DR - IGN France	Rapporteur et examinateur
Mme Beatriz Marcotegui	MR - MINES ParisTech	Examineur
M. Paul Checchin	MCF, HDR - Université Blaise Pascal	Co-directeur de thèse
M. Laurent Trassoudaine	PR - Université Blaise Pascal	Directeur de thèse

Acknowledgments

This dissertation would not have been possible without the guidance and help of many whom in one way or another contributed their valuable time and efforts.

Firstly, I would like to thank my supervisors, Professor Laurent TRASSOUDAINÉ and Paul CHECCHIN, Associate Professor-HDR, for their unflinching support and confidence in me during the whole PhD tenure. Their encouragement and guidance from the beginning to the end not only enabled me to gain important insight and understanding of the subject but also helped in developing my own ideas which laid the foundation for this thesis. The diverse technical and non-technical discussions, work meetings in both Clermont-Ferrand and Montluçon and our various travels together constitute some of my fondest memories.

It is also a pleasure for me to thank all my colleagues at the Institut Pascal, specially Michel DHOME (DR CNRS), Roland CHAPUIS (PR) and Thierry CHATEAU (PR) for their timely and important help at different stages of my PhD. Similarly, I also feel obliged to extend my gratitude to Vanessa CHAUDRON for handling all the tedious paperwork for my different technical visits and conferences. Equally, I would also like to appreciate my colleagues at the "IUT de Montluçon" for hosting me all these years in an excellent working environment. I am also grateful to the thesis committee members specially Nicolas PAPORADITIS, Robert ZLOT and Uwe STILLA for taking out their valuable time to go through this manuscript. Their suggestions, insight and critique proved to be invaluable for this thesis.

Last but not least, I am indebted to the omnipotent and omnipresent God, and then to my family especially my parents, brother and sister for all their support and sacrifices. Finally, I would like to give a special thanks to my wife for providing a welcome distraction from work, for bringing me happiness and for her encouragement and love.

This research was funded by the "Conseil Général de l'Allier". Their support is also gratefully acknowledged.

Résumé

Au cours des dernières années, la cartographie urbaine 3D a suscité un intérêt croissant pour répondre à la demande d'applications d'analyse des scènes urbaines tournées vers un large public. Conjointement les techniques d'acquisition de données 3D progressaient. Les travaux concernant la modélisation et la visualisation 3D des villes se sont donc intensifiés.

Des applications fournissent au plus grand nombre des visualisations efficaces de modèles urbains à grande échelle sur la base de des imageries aérienne et satellitaire. Naturellement, la demande s'est portée vers des représentations avec un point de vue terrestre pour offrir une visualisation 3D plus détaillée et plus réaliste. Intégrées dans plusieurs navigateurs géographiques comme Google Street View, Microsoft Visual Earth ou Géoportail, ces modélisations sont désormais accessibles et offrent une représentation réaliste du terrain, créée à partir des numériseurs mobiles terrestres.

Dans des environnements urbains, la qualité des données obtenues à partir de ces véhicules terrestres hybrides est largement entravée par la présence d'objets temporairement statiques ou dynamiques (piétons, voitures, etc.) dans la scène. La mise à jour de la cartographie urbaine via la détection des modifications et le traitement des données bruitées dans les environnements urbains complexes, l'appariement des nuages de points au cours de passages successifs, voire la gestion des grandes variations d'aspect de la scène dues aux conditions environnementales constituent d'autres problèmes délicats associés à cette thématique. Plus récemment, les tâches de perception s'efforcent également de mener une analyse sémantique de l'environnement urbain pour renforcer les applications intégrant des cartes urbaines 3D.

Dans cette thèse, nous présentons un travail supportant le passage à l'échelle pour la cartographie 3D urbaine automatique incorporant la reconnaissance et l'intégration temporelle. Nous présentons en détail les pratiques actuelles du domaine ainsi que les différentes méthodes, les applications, les technologies récentes d'acquisition des données et de cartographie, ainsi que les différents problèmes et les défis qui leur sont associés. Le travail présenté se confronte à ces nombreux défis mais principalement à la classification des zones urbaines l'environnement, à la détection automatique des changements, à la mise à jour efficace de la carte et l'analyse sémantique de l'environnement urbain.

Dans la méthode proposée, nous effectuons d'abord la classification de l'environnement urbain en éléments permanents et temporaires. Les objets classés comme temporaire sont ensuite retirés du nuage de points 3D laissant une zone perforée dans le nuage de points 3D. Ces zones perforées ainsi que d'autres imperfections sont ensuite analysées et progressivement éliminées par une mise à jour incrémentale exploitant le concept de multiples passages. Nous montrons que la méthode d'intégration temporelle proposée permet également d'améliorer l'analyse sémantique de l'environnement urbain, notamment les façades des bâtiments. Les résultats, évalués sur des données réelles en utilisant différentes métriques, démontrent non seulement que la cartographie 3D résultante est précise et bien mise à jour, qu'elle ne contient que les caractéristiques permanentes exactes et sans imperfections, mais aussi que la méthode est également adaptée pour opérer sur des scènes urbaines de grande taille. La méthode est adaptée pour des applications liées à la modélisation et la cartographie du paysage urbain nécessitant une mise à jour fréquente de la base de données.

Mots-clés : cartographie urbaine, données LiDAR, nuages de points 3D, segmentation, classification, multi-passages, détection des changements, mise à jour incrémentale, analyse sémantique.

Abstract

Over the years, 3D urban cartography has gained widespread interest and importance in the scientific community due to an ever increasing demand for urban landscape analysis for different popular applications, coupled with advances in 3D data acquisition technology. As a result, in the last few years, work on the 3D modeling and visualization of cities has intensified.

Lately, applications have been very successful in delivering effective visualizations of large scale models based on aerial and satellite imagery to a broad audience. This has created a demand for ground based models as the next logical step to offer 3D visualizations of cities. Integrated in several geographical navigators, like Google Street View, Microsoft Visual Earth or Geoportail, several such models are accessible to large public who enthusiastically view the real-like representation of the terrain, created by mobile terrestrial image acquisition techniques.

However, in urban environments, the quality of data acquired by these hybrid terrestrial vehicles is widely hampered by the presence of temporary stationary and dynamic objects (pedestrians, cars, etc.) in the scene. Other associated problems include efficient update of the urban cartography, effective change detection in the urban environment and issues like processing noisy data in the cluttered urban environment, matching / registration of point clouds in successive passages, and wide variations in environmental conditions, etc. Another aspect that has attracted a lot of attention recently is the semantic analysis of the urban environment to enrich semantically 3D mapping of urban cities, necessary for various perception tasks and modern applications.

In this thesis, we present a scalable framework for automatic 3D urban cartography which incorporates recognition and temporal integration. We present in details the current practices in the domain along with the different methods, applications, recent data acquisition and mapping technologies as well as the different problems and challenges associated with them. The work presented addresses many of these challenges mainly pertaining to classification of urban environment, automatic change detection, efficient updating of 3D urban cartography and semantic analysis of the urban environment.

In the proposed method, we first classify the urban environment into permanent and temporary classes. The objects classified as temporary are then removed from the 3D point cloud leaving behind a perforated 3D point cloud of the urban environment. These perforations along with other imperfections are then analyzed and progressively removed by incremental updating exploiting the concept of multiple passages. We also show that the proposed method of temporal integration also helps in improved semantic analysis of the urban environment, specially building façades. The proposed methods ensure that the resulting 3D cartography contains only the exact, accurate and well updated permanent features of the urban environment. These methods are validated on real data obtained from different sources in different environments. The results not only demonstrate the efficiency, scalability and technical strength of the method but also that it is ideally suited for applications pertaining to urban landscape modeling and cartography requiring frequent database updating.

Keywords : urban cartography, LiDAR data, 3D point clouds, segmentation, classification, multi-passages, change detection, incremental updating and semantic analysis.

Résumé étendu : contexte et synthèse du mémoire

Contexte

Mon travail doctoral a été réalisé au sein de l'équipe de recherche PERSYST (Perception Systems : systèmes de perception) de l'INSTITUT PASCAL (UMR 6602 CNRS/UBP), une unité mixte de recherche du CNRS (Centre National de la Recherche Scientifique), de l'Université Blaise Pascal et de l'IFMA (Institut Français de Mécanique Avancée). PERSYST fait partie de l'axe ISPR (Image, Systèmes de Perception, Robotique) dont les recherches sont centrées sur la perception artificielle, la robotique et la vision par ordinateur. Plus particulièrement, PERSYST développe des approches globales de perception exploitant plusieurs capteurs en vue de la compréhension de scène, des approches de reconstructions 3D denses d'environnements complexes avec différentes modalités (caméra, lidar, radar), des solutions de localisation 2D/3D et de guidage de robots mobiles par approche mono et multisensorielle (caméra, lidar, proprioceptifs). Elles intègrent éventuellement la cartographie de l'environnement à construire (SLAM) ou des informations de type SIG (Système d'Information Géo-référencé). L'ensemble des activités est historiquement fondé sur une très forte culture de projet mettant en oeuvre des systèmes de perception temps réel sur des plates-formes réalistes et performantes.

Cette thèse a pu être menée grâce à l'aide d'une bourse octroyée par le « Conseil Général de l'Allier ». Les travaux effectués s'inscrivent, en partie, au sein du projet iSpace&Time de l'ANR (Agence Nationale de la Recherche) (ANR-10-CONT-23). L'objectif de ce projet est le développement et la mise en place d'un portail, démonstrateur d'un Système d'Information Géographique 4D (espace et temps) de la ville sur le web.

Les travaux de recherche sur la modélisation et la visualisation 3D de l'environnement urbain se sont intensifiés ces dernières années compte tenu du succès rencontré par des applications grand public. Ainsi, en mai 2007, Google a mis en ligne *Google Street View* qui permet de se déplacer de manière immersive dans un ensemble de panoramiques acquis par des *Google cars* qui sillonnent les grandes villes. Le projet ANR iTOWNS (déposé en février 2007 et coordonné par l'Institut National de l'Information Géographique et Forestière (IGN)) a permis de développer un navigateur équivalent à celui de *Street-View* (mais avec des images de meilleure qualité géométrique et radiométrique) tout en y rajoutant un modèleur participatif de saisie 3D d'objets destiné à la mise à jour des bases de données. La technologie proposée par la société EARTHMINE propose également des solutions de relevés 3D par modèleur dédiés aux collectivités et aux gestionnaires d'infrastructures pour saisir et/ou mettre à jour leurs bases de données 3D.

Un des difficiles défis à aborder en cartographie mobile urbaine concerne toujours la gestion des occultations qui se produisent lors de la digitalisation 3D d'un site. Imaginons un véhicule de numérisation terrestre hybride laser-image qui circule sur un site ouvert de type centre ville. Des parties des infrastructures statiques du paysage urbain (façades des bâtiments, chaussées, etc.) sont masquées par la présence d'objets mobiles sur les voies de circulation ou de certains objets temporairement présents et figés lors du passage du véhicule instrumenté. Il en résulte une dégradation de la qualité des relevés qui nuit à l'obtention d'un modèle 3D photo-réaliste.

Au cours des années, ce problème de la suppression des occultations a été intensément étudié, notamment dans la communauté de la vision par ordinateur et des graphistes. La plupart des approches reposent sur un détournement manuel.

Le travail qui va être décrit vise à apporter une solution par une approche multi-passages ou

multi-sessions. Plusieurs relevés du même site, obtenus à des instants différents, seront exploités. Chaque passage est traité pour en retirer les parties occultantes. La carte ainsi perforée est ensuite mise à jour par les prochaines images elles-mêmes perforées. Les zones vides d'information sont ainsi peu à peu comblées. La cartographie s'enrichit donc peu à peu de nouveaux détails, correspondant à des entités permanentes.

Introduction

Cartographie (du grec *χάρτης*, *khartēs*, « carte » ; et *γράφειν*, *graphein*, « écrire ») est un terme qui désigne l'étude et l'art de dresser des cartes de géographie [The14]. En complète synergie avec leur époque, les travaux de cartographie combinent science, technique et esthétique afin de démontrer que la réalité peut être modélisée pour restituer efficacement l'information spatiale. Une définition moderne de la cartographie 3D est qu'elle met en oeuvre « un ensemble de connaissances et d'activités scientifiques, techniques et artistiques, impliquées dans le cycle de vie complet des cartes 3D, et notamment : la collecte de données, le processus de production et de distribution, les indispensables technologies et algorithmes, ainsi que l'ensemble des utilisations possibles » [Gór09].

La cartographie 3D urbaine suscite donc un grand intérêt dans la communauté scientifique en raison de la demande croissante d'analyses et de modélisation de l'environnement urbain. De façon concomitante, les progrès de la technologie d'acquisition des données ne sont pas étrangers à cette évolution. Aussi, au cours des dernières années, les travaux sur la modélisation et la visualisation 3D des villes se sont intensifiés. Dernièrement, des applications fournissent à destination d'un large public des visualisations efficaces de modèles urbains à grande échelle sur la base de l'imagerie aérienne et satellitaire. C'est très logiquement que la demande s'est portée vers des représentations avec un point de vue terrestre et ainsi offrir une visualisation 3D, plus détaillée et toujours plus réaliste, des villes urbaines. Intégrées dans plusieurs navigateurs géographiques comme Google Streetmap Viewer, Microsoft Visual Earth ou Géoportail, ces modélisations sont accessibles au grand public qui découvre avec enthousiasme la représentation réaliste du terrain, créée à partir des techniques d'acquisition de données mobiles terrestres.

Outre les différentes applications possibles des données cartographiques 3D dans un contexte urbain, le chapitre 2 du présent manuscrit dresse un rapide panorama des différents véhicules de numérisation terrestre exploitant des procédés à base de technologie laser, image ou hybride. La figure 1 donne une illustration de deux de ces véhicules. Ceux-ci ont la particularité d'être à l'origine d'une partie des données que nous avons traitées au cours des travaux réalisés. Ce chapitre présente également les différentes techniques nécessaires pour collecter les mesures et les exprimer dans un référentiel commun, permettant ainsi une exploitation dans ce contexte particulier.

Si ce type de véhicule constitue un progrès important dans le domaine de la cartographie 3D, leur usage s'accompagne d'un certain nombre de limites ou de challenges qui sont également évoqués dans ce même chapitre. Ainsi, dans des environnements urbains, la qualité des données obtenues à partir de ces véhicules terrestres hybrides est largement entravée par la présence d'objets temporairement statiques et dynamiques (piétons, voitures, etc.) dans la scène. En conséquence, l'occultation de certaines zones reste toujours un problème (cf. figure 2). Les objets dynamiques et certains objets temporairement stationnés cachent certaines zones de la scène urbaine (parties de bâtiments, des routes, etc.).

Un autre problème auquel on est confronté avec ce type d'applications concerne la mise à jour effective de la cartographie urbaine. Pour mettre à jour ces cartes ou des modèles 3D de l'environnement, les mêmes zones doivent être cartographiées régulièrement. Par exemple, en ce qui concerne Google Earth, une mise à jour est effectuée toutes les deux semaines. Pour ce qui est de Google Street View, une réactualisation est opérée après seulement quelques mois, mais uniquement pour quelques régions géographiques particulières (source Google inc.). La plupart



FIGURE 1 : Les véhicules numérisateurs terrestres (a) STÉRÉOPOLIS II (source [Ste14]) et (b) Vis center's VizTruck.

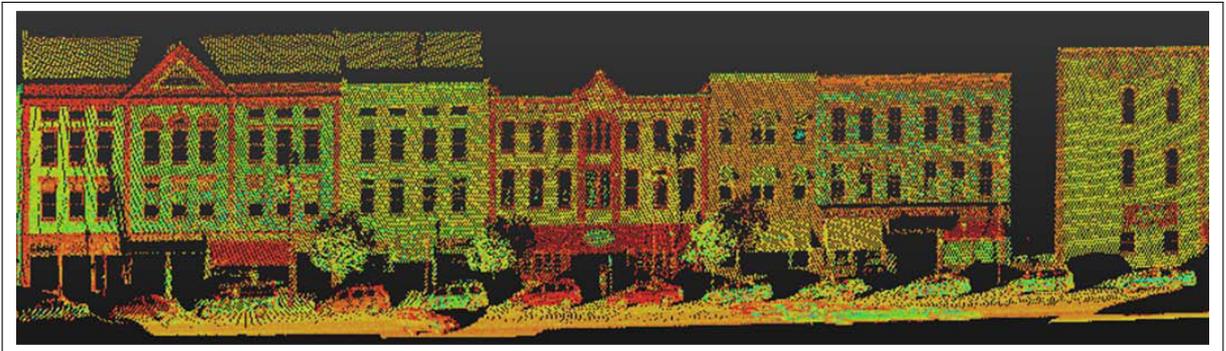


FIGURE 2 : Différentes zones occultées apparaissant dans le nuage de points 3D.

des services de cartographie actuels s'appuient également sur les informations en provenance des utilisateurs qui indiquent les zones modifiées, où une nouvelle acquisition est nécessaire. On peut éventuellement alors optimiser et prioriser les calendriers de relevés terrain. Toutefois, dans la pratique courante, comme aucune méthode de détection automatique de changement n'est utilisée afin d'efficacement identifier et localiser la/les partie(s) modifiée(s), beaucoup de ressources sont gaspillées. De plus, en raison de la procédure de mise à jour souvent inefficace, une grande quantité de données inutiles sont accumulées avec le temps. Par exemple, Google a déjà stocké plus de 20 péta-octets de données pour ses différents services de cartographie. L'utilisation de ces données redondantes pour les différentes applications spécialisées demeure un sujet difficile de la recherche.

Un autre aspect qui a attiré beaucoup l'attention récemment est l'analyse sémantique de l'environnement urbain afin d'enrichir la cartographie 3D des villes. Aujourd'hui, cette tâche apparaît de plus en plus nécessaire pour diverses applications de perception, telles que l'aide à la conduite, la navigation autonome en environnement urbain, le rendu temps réel, le tourisme virtuel et la planification des différentes missions urbaines.

Enfin, bien d'autres problèmes associés à cette thématique pourraient être listés comme, par exemple, le traitement des données bruitées dans les environnements urbains complexes, l'appariement des nuages de points au cours de passages successifs, voire la gestion des grandes variations d'aspect de la scène dues aux conditions environnementales.

Dans cette thèse, nous traitons de ces problèmes et proposons une nouvelle méthode de cartographie 3D urbaine incorporant la reconnaissance et l'intégration temporelle (en exploitant le concept de passages multiples). La cartographie 3D finale que l'on cherche à obtenir ne devra

contenir que les éléments permanents, exacts, précis et correctement mis à jour de l'environnement urbain. Un survol des points détaillés dans les chapitres du manuscrit sont à présent évoqués dans la section suivante.

Travaux menés

La qualité des données obtenues à partir de systèmes mobiles de cartographie dans les environnements urbains est donc largement entravée par la présence d'objets temporairement statiques et dynamiques (piétons, voitures, etc.) dans la scène. Une première étape pour la cartographie urbaine 3D peut être alors de déterminer les entités permanentes de l'environnement urbain. A cette fin, nous proposons d'effectuer une classification des éléments de l'environnement urbain vers des classes d'entités dites permanentes ou temporaires (chapitre 3).

Les objets classés temporaires sont ensuite retirés du nuage de points 3D pour obtenir un nuage de points 3D perforé de l'environnement urbain. Ces zones perforées ainsi que d'autres imperfections sont ensuite analysées et éliminées progressivement à l'aide d'un procédé de mise à jour incrémentale exploitant le concept de passages multiples (chapitre 4).

Nous montrons également que la méthode d'intégration temporelle proposée aide aussi à l'amélioration de l'analyse sémantique de l'environnement urbain, spécialement pour ce qui est des façades de bâtiments (chapitre 5).

Ce document comporte trois parties principales qui sont à présent résumées.

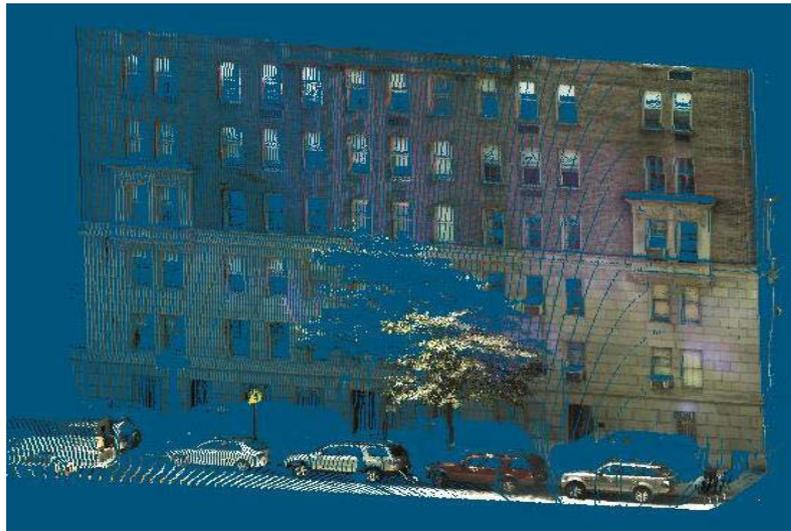
Segmentation et classification de nuage des points 3D : une approche par super-voxels

Dans le chapitre 3, une méthode de classification de données 3D éparses obtenues à partir de relevés laser terrestres en environnement urbain est présentée. Elle est fondée sur une technique de segmentation manipulant des super-voxels. Le nuage de points 3D est tout d'abord divisé en voxels caractérisés par plusieurs attributs. Ils deviennent ainsi des super-voxels qui sont liés entre eux pour former des objets par une méthode de chainage plutôt que par une croissance de régions classique. Ces objets sont ensuite classifiés à partir de modèles géométriques et de descripteurs locaux en 5 classes différentes : bâtiment, route, voiture, poteau et arbre.

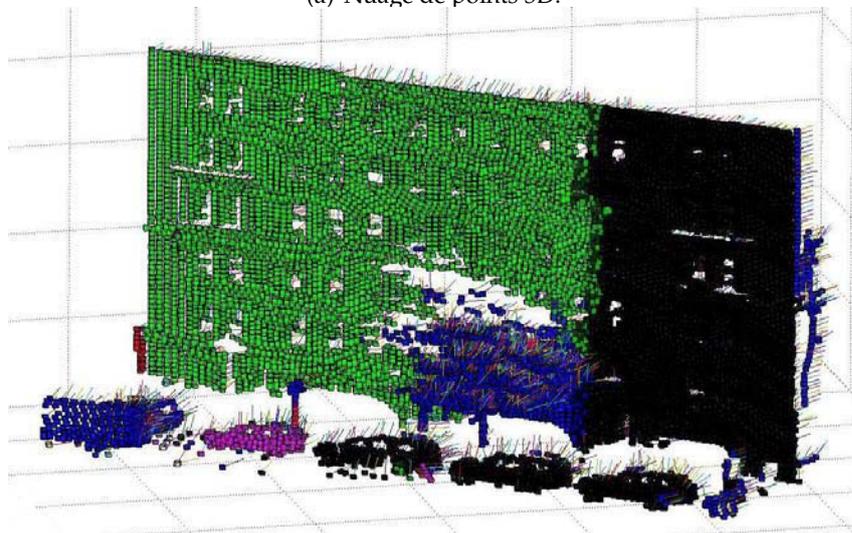
Outre la description de l'approche proposée, une étude bibliographique est menée. Elle est scindée en deux parties relatives respectivement à la segmentation et à la classification. Les approches au sein de celles-ci sont regroupées par tendances (descripteurs spécialisés et discontinuités surfaciques, Graph clustering, primitives géométriques ou modèles pour la classification, etc.). Dans le travail proposé, et dont le positionnement vis à vis de l'existant est discuté, nous utilisons des modèles géométriques construits à partir des caractéristiques locales et des descripteurs pour classer les différents objets segmentés d'une scène urbaine représentés par des groupes de voxels. Le terrain est supposé plan et utilisé comme un séparateur d'objets.

Un échantillon des résultats est présenté figure 3. Ils sont évalués et comparés avec une autre méthode de l'état de l'art en utilisant les différentes métriques d'estimation des performances existantes ainsi qu'une nouvelle métrique qui caractérise à la fois les résultats de la segmentation et de la classification.

L'influence sur les résultats de classification de la taille du voxel et de l'incorporation des informations couleur RVB et intensité de réflectance dans le super-voxel est aussi analysée. Cette étude montre que la précision de classification est accrue si la taille des voxels est diminuée mais ceci au détriment d'un coût calculatoire supérieur. Une solution pour en déterminer une valeur optimale est présentée. Cette analyse montre également l'importance d'utiliser des valeurs d'intensité et de couleurs RVB dans la segmentation et la classification.



(a) Nuage de points 3D.



(b) Voxelisation et segmentation.

■ Building
 ■ Road
 ■ Pole
 ■ Car
 ■ Tree
 ■ un-classified



(c) Classification.

FIGURE 3 : Illustrations de résultats de segmentation et de classification.

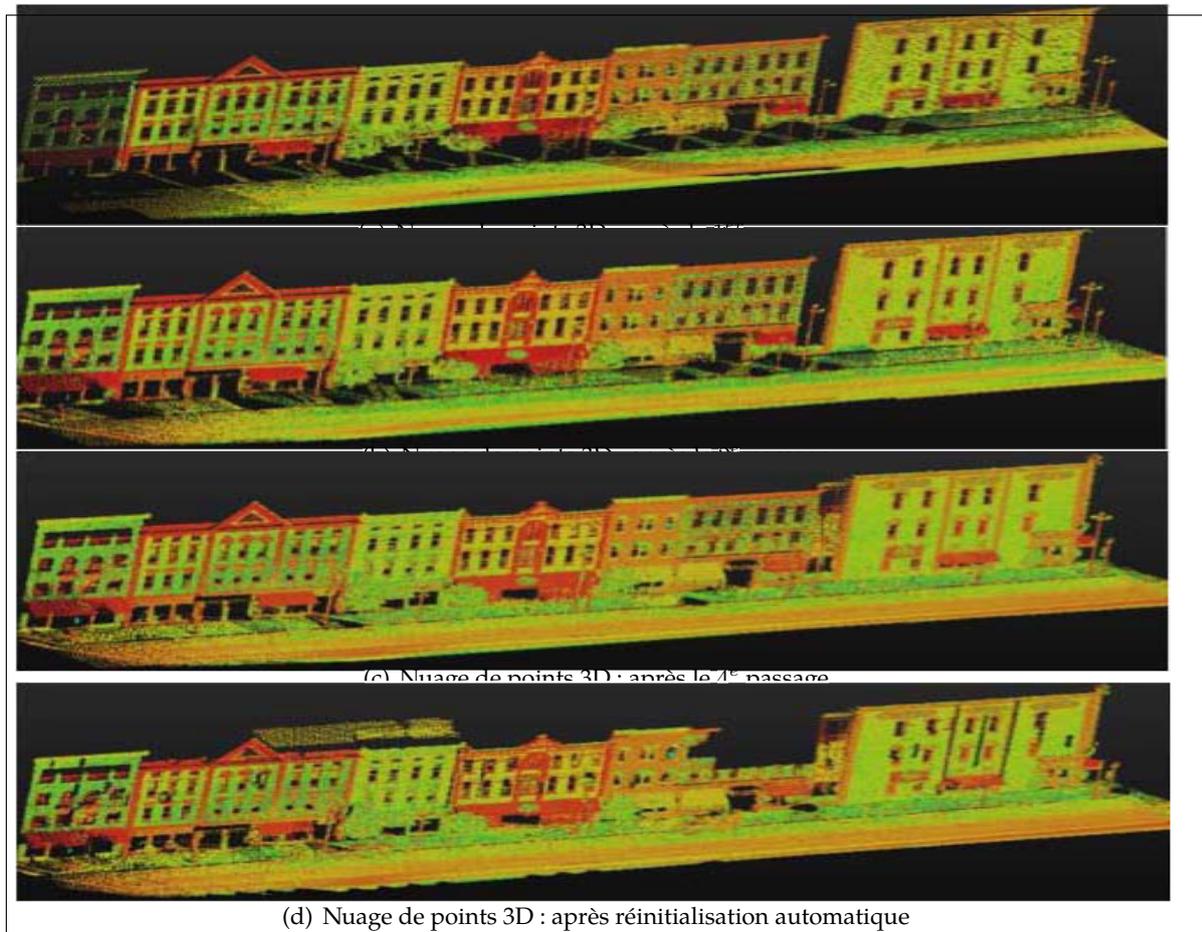


FIGURE 4 : (a) Nuage de points initial sans objets temporaires. (b) et (c) Mise à jour après détection de changements. (d) Réinitialisation automatique après détection de changements.

Cartographie urbaine 3D incorporant une classification et une intégration temporelle

Cette nouvelle méthode de cartographie urbaine 3D effectuée de façon automatique incorpore à la fois classification et intégration temporelle, en exploitant le concept de passages multiples. Dans la méthode proposée, les nuages de points 3D sont d'abord classés en 3 grandes catégories d'objets présents dans la scène : permanent, temporairement statique et mobile.

Les objets classés comme temporairement statiques et mobiles sont ensuite retirés des nuages de points 3D pour obtenir un nuage de points 3D perforé de la scène urbaine. Ces nuages de points 3D ainsi perforés, acquis lors de passages successifs en un même lieu à des jours et horaires différents sont ensuite combinés pour enrichir la cartographie. Les différentes modifications qui se produisent dans le paysage urbain au cours de cette période sont détectées et analysées à l'aide de fonctions cognitives de similarité. La cartographie 3D obtenue est progressivement modifiée en conséquence comme l'illustre la figure 4. Différentes fonctions spécialisées sont également introduites pour aider à éliminer les différentes imperfections dues à des occultations, des erreurs de classification et les différents changements qui se produisent dans l'environnement au cours du temps. Elles augmentent la robustesse de la méthode.

Les résultats, évalués sur des données réelles en utilisant différentes métriques, démontrent non seulement que la cartographie 3D résultante est précise et bien mise à jour, qu'elle ne contient que les caractéristiques permanentes exactes et sans imperfections, mais aussi que la méthode est adaptée pour opérer sur des scènes urbaines de grande taille.

Détection automatique et estimation des paramètres de fenêtres pour affiner l'analyse des façades des bâtiments

Cette partie du travail se concentre sur l'analyse sémantique des bâtiments de l'environnement urbain et plus précisément sur la détection et l'estimation des paramètres des fenêtres des façades. Comme expliqué dans la section précédente, en raison de la demande toujours plus forte de modèles urbains 3D plus réalistes, possibles notamment grâce aux progrès récents des technologies LiDAR terrestres, l'extraction d'informations sémantiques des façades de bâtiments, tels que les fenêtres, présente un grand intérêt. Cependant, en raison par endroits de la densité insuffisante des points dans les nuages des données LiDAR, il est très difficile de détecter et d'estimer de manière fiable la géométrie des fenêtres sans interaction humaine. Aussi, dans cette dernière partie des travaux effectués, décrits dans le chapitre 5, nous présentons une nouvelle méthode qui détecte automatiquement des fenêtres de formes différentes, dans les nuages de points 3D.

La méthode proposée effectue tout d'abord une segmentation du nuage de points 3D afin d'isoler ceux appartenant aux façades des bâtiments. Les points ainsi isolés sont projetés sur un plan 2D parallèle à la façade du bâtiment afin, ensuite, d'extraire l'ensemble des contours fermés délimitant les fenêtres potentielles. Les caractéristiques géométriques des fenêtres sont ensuite estimées en exploitant à la fois les fenêtres symétriquement correspondantes de la façade et des fenêtres temporellement correspondantes détectées lors de passages successifs. Pour cela, une technique d'analyse de variance (ANOVA) est mise en oeuvre. Cette solution originale de fusion d'information permet non seulement de faire face à la disposition asymétrique des fenêtres sur la façade mais contribue également à compléter les éléments occultés manquants. Les positions estimées des fenêtres sont ensuite exploitées pour affiner les régions correspondantes des nuages de points 3D de la façade comme l'illustre la figure 5 ci-dessous. Les résultats, évalués sur des données réelles en utilisant de nouvelles métriques, démontrent l'efficacité de la méthode proposée.

Principales contributions

Les principaux objectifs de cette thèse étaient de répondre aux différents problèmes rencontrés actuellement en cartographie 3D des villes urbaines en développant de nouvelles méthodes utilisant des données LiDAR acquises à partir de véhicules de numérisation terrestres. Les travaux menés dans ce cadre ont enrichi l'état de l'art en apportant plusieurs contributions qui sont détaillées dans cette thèse.

- ▷ Une méthode de classification de l'environnement urbain fondée sur la segmentation des nuages de points 3D en utilisant une approche par super-voxels, suivie par la caractérisation des nuages de points 3D urbains en utilisant une nouvelle technique d'appariement de points.
- ▷ Un procédé de gestion des occultations en exploitant le concept de passages multiples.
- ▷ Une nouvelle méthode pour la détection automatique de changements et la mise à jour incrémentale de nuages de points 3D urbains en éliminant les différentes imperfections.
- ▷ La détection et l'estimation automatiques des paramètres des fenêtres pour affiner l'analyse des façades des bâtiments en exploitant des correspondances symétriques et temporelles.

Les différents travaux menés au cours de cette thèse ont fait l'objet de publications dans 3 revues internationales indexées, 2 chapitres d'ouvrage et 8 conférences internationales notamment. Les détails des références de ces publications sont présentés dans l' [Appendix A](#).

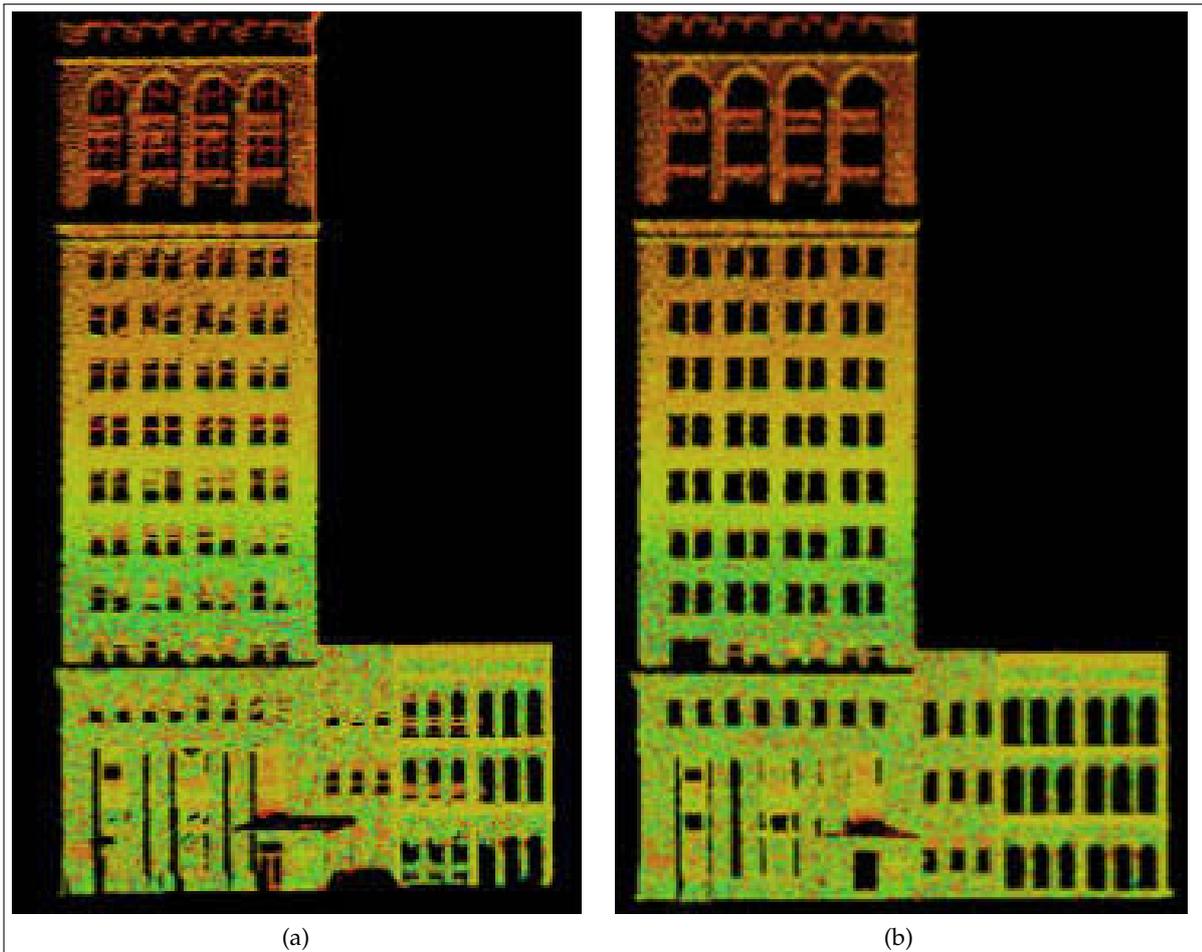


FIGURE 5 : Illustration de la détection des fenêtres des façades des bâtiments. (a) nuage de points 3D originel. (b) nuage de points 3D après la détection des fenêtres et affinage des zones correspondantes.

Mots-clés : cartographie urbaine, données LiDAR, nuages de points 3D, segmentation, classification, multi-passages, détection des changements, mise à jour incrémentale, analyse sémantique

Contents

1	Introduction	1
1.1	Context	2
1.2	Main Contributions	3
1.3	Organization of the Document	3
2	3D Urban Cartography	7
2.1	Introduction	8
2.2	Applications of 3D Urban Cartography	8
2.2.1	Navigation	8
2.2.2	Virtual Tourism and Preservation of Cultural Heritage Sites	9
2.2.3	Infrastructure Analysis and Management	9
2.2.4	Road Asset Management and Driver Assistance	10
2.2.5	Urban City Analysis and Planning	11
2.3	Terrestrial Mobile Mapping Systems	12
2.3.1	Mobile Mapping Systems: Standard Technologies	12
2.3.2	Direct Geo-referencing	14
2.3.3	Modes of Operation	16
2.3.4	Existing Mobile Mapping Systems (MMS)	18
2.3.5	Mobile Mapping Services	22
2.4	Limitations and Challenges	26
2.5	Proposed Methodology	30
3	Segmentation and Classification of 3D Urban Point Clouds	34
3.1	Introduction	35
3.1.1	Segmentation of 3D Data	35
3.1.2	Classification of 3D Data	38
3.2	Voxel-based Segmentation	40
3.2.1	Voxelisation of Data	41
3.2.2	Transformation of Voxels into Super-Voxels	42
3.2.3	Clustering by Link-Chain Method	42
3.3	Classification of Segmented Objects	44
3.4	Evaluation Metrics	46
3.5	Results	47
3.5.1	3D Datasets of Blaise Pascal University	47
3.5.2	3D Urban Data Challenge Dataset	47
3.5.3	Comparison of Results with Existing Evaluation Methods	51
3.5.4	Performance Evaluation and Discussion	55
3.5.5	Effect of Voxel Size on Classification Accuracy and Choice of Optimal Values	56
3.5.6	Influence of RGB Color and Reflectance Intensity	58
3.5.7	Limitations	58
3.6	Comparison with Existing Techniques	59

3.6.1	Comparison with Standard SVM based Classification Method	59
3.6.2	Comparison with a State of the Art Method	60
3.7	Conclusion	67
4	3D Urban Cartography - Classification and Incremental Updating	71
4.1	Introduction	72
4.2	3D Scan Registration	77
4.3	Classification of 3D Urban Environment	78
4.4	Combining Multiple Views for Scene Completion	81
4.5	3D Cartography and Removal of Imperfections Exploiting Multiple Passages	84
4.5.1	3D Evidence Grid Formulation	84
4.5.2	Similarity Map Construction	85
4.5.3	Associated Uncertainty	87
4.5.4	Specialized Functions	89
4.5.5	Automatic Checks and Balances	93
4.6	Results, Evaluation and Discussion	93
4.6.1	Change Detection and Reset Function	93
4.6.2	Evolution of Similarity Map Size	98
4.6.3	Handling Misclassifications and Improvement in Classification Results	98
4.6.4	Accuracy Evaluation	99
4.7	Conclusion	100
5	Detection and Feature Estimation of Windows for Refining Building Façades	103
5.1	Introduction	104
5.2	Segmentation of Building Façades and Road Surface from 3D Urban Point Clouds	108
5.3	Window Segmentation	108
5.3.1	3D Point Projection	108
5.3.2	Watertight Boundary Estimation	109
5.3.3	Point Inversion and Window Segmentation	111
5.4	Window Features Estimation	113
5.4.1	Exploiting Façade Symmetry	113
5.4.2	Exploiting Multiple Passages	113
5.4.3	ANOVA-based Estimation	114
5.5	Refining and Updating the Building Façades in the 3D Point Clouds	117
5.6	Automatic Checks and Balances	118
5.7	Results, Evaluation and Discussion	119
5.7.1	Evaluation	119
5.7.2	Limitations	124
5.8	Conclusion	124
6	Conclusion and Future Work	127
6.1	Summary	128
6.2	Directions for Future Research	129
	Appendix A	135
	Appendix B	140

List of Figures

1	Exemples de véhicules numérisateurs terrestres	IX
2	Différentes zones occultées apparaissant dans le nuage de points 3D.	IX
3	Illustrations de résultats de segmentation et de classification	XI
4	Détection de changements et mise à jour incrémentale	XII
5	Illustration de la détection des fenêtres des façades des bâtiments	XIV
1.1	Panoramic views acquired by STÉRÉOPOLIS	2
1.2	Crowd and traffic flow simulation	3
2.1	3D data acquisition of Capitol Hill	8
2.2	Use of the data for navigation purposes	9
2.3	Use of the data for virtual tourism	9
2.4	3D point cloud data of Sydney Opera House	10
2.5	Using the data for Infrastructure analysis	10
2.6	Using the data for road asset management and Driver assistance	11
2.7	Using the data for Urban city analysis	11
2.8	The basic composition of terrestrial Mobile Mapping Systems	12
2.9	The different sensors mounted on terrestrial Mobile Mapping Systems	13
2.10	The different laser scanners mounted on terrestrial Mobile Mapping Systems	14
2.11	Direct geo-referencing	15
2.12	Principle of On-the-Fly mode	16
2.13	3D data acquired using On-the-Fly mode	17
2.14	Principle of Stop-and-Go mode	17
2.15	3D data acquired using Stop-and-Go mode	18
2.16	Common Mobile Mapping Systems	19
2.17	Hybrid Mobile Mapping Systems	20
2.18	True color coded point cloud	20
2.19	Comparison of some commercially available Mapping systems	21
2.20	Topcon's IPS2 mobile mapping system	22
2.21	Lynx Mobile Mapper	23
2.22	Tele Atlas fleet	24
2.23	NAVTEQ mobile mapping vehicle	24
2.24	Google mobile mapping fleet	25
2.25	Occluding objects in Mobile Mapping System data	26
2.26	Density variation in 3D MMS data.	27
2.27	Problem of not-so-regular updating	28
2.28	Problem of different updating schedules	29
2.29	Misalignment errors when merging two 3D point clouds	29
2.30	An overview of the workflow	31
3.1	Illustration of segmentation methods based on surface discontinuities	35
3.2	Illustration of segmentation methods based on Graph-cuts	36

3.3	Illustration of modified FH algorithm with surface normals and color differences	37
3.4	Segmentation results with ellipsoids as geometric primitives	38
3.5	Segmentation of 3D point clouds based on morphological modeling	39
3.6	Conditional Random Fields based classification of 3D outdoor terrestrial laser scanned data	40
3.7	Rule based classification of 3D outdoor mobile terrestrial laser data	40
3.8	Voxelisation of data	41
3.9	Clustering of <i>s</i> -voxels using a link-chain method	43
3.10	Segmented objects in a scene with prior ground removal	44
3.11	surface normals of building and road surface <i>s</i> -voxels	45
3.12	Bounding boxes for buildings, trees, cars, pedestrians and poles	45
3.13	Segmentation and classification results for dataset 1	48
3.14	Segmentation and classification results for dataset 2	49
3.15	Segmentation and classification results for dataset 3	50
3.16	Segmentation and classification results for scene-A	52
3.17	Segmentation and classification results for scene-B	53
3.18	Segmentation and classification results for scene-C	54
3.19	Effects of maximum voxel size variation	57
3.20	Limitations due to partial occlusions	59
3.21	Limitations due to different appearance	60
3.22	Google map photo of Rue Madame in Paris, France	61
3.23	Detection results for both voxel-based and morphological transformation based methods	63
3.24	Comparison of segmentation and classification results	64
3.25	Misclassification of some 3D points at boundary regions of road surface	65
3.26	Misclassification of some 3D points belonging to a sign post with two traffic signs	66
3.27	Overall segmentation quality	67
3.28	Segmentation error due to variation in color and intensity values	68
3.29	Segmentation error due to disjoint building	68
3.30	Comparison of motorcycle segmentation results	69
3.31	Comparison of car segmentation results	70
4.1	Occlusion removal using object-specific detectors	72
4.2	Combining multiple images to remove occlusions	73
4.3	Occlusion removal by foreground and background segmentation	74
4.4	Occlusion removal by point-to-point comparison	75
4.5	Occlusion removal method based on scanning origin and geometry	76
4.6	The Vis Center’s LiDAR truck	79
4.7	Voxel-based segmentation and classification	79
4.8	Data representation in <i>x</i> -plane configuration	81
4.9	Characterization of classified objects	82
4.10	Extraction of occluding objects	83
4.11	Combining multiple views to complete occluded regions	83
4.12	Formulation of 3D evidence grids and similarity map	85
4.13	Tversky’s model of similarity	86
4.14	Uncertainty distribution map of a busy neighborhood	88
4.15	Registration of 3D point clouds	90
4.16	Change detection results (qualitative)	91
4.17	The uncertainty variation of changing cells	92
4.18	3D urban cartography (Results)	94
4.19	3D urban cartography (Results)	95
4.20	3D urban cartography (Results)	96

4.21	3D urban cartography (Results)	97
4.22	Evaluation of change detection method	98
4.23	The progressive reduction of the size of similarity map	99
4.24	3D model of building	100
4.25	Evaluation of accuracy	100
5.1	Representation of a building using LoD0 - LoD4	104
5.2	3D point cloud containing building façades in the urban environment	105
5.3	Detection of windows as blobs in a color image	106
5.4	Local lattice fitting and voting scheme	106
5.5	Problem of window detection at façade interface	107
5.6	Segmentation of windows from 3D points belonging to building façades	110
5.7	3D point cloud of the building façade showing varying resolution	112
5.8	Window segmentation exploiting façade symmetry	114
5.9	Finding Temporal correspondences in multiple passages	115
5.10	Finding common profile points for temporal correspondences	117
5.11	Partial and completely shut/closed windows on the building façade	118
5.12	Window detection results	120
5.13	Qualitative results of window detection and refinement of different building façades	121
5.14	Qualitative results of window detection and refinement of repeated window structures	122
5.15	Successful detection and estimation of window parameters	123
5.16	Evolution of window detection performance	124
6.1	Mapping adjoining alleys and narrow streets	130
6.2	Semantic and thermal mapping	131
6.3	Sensor equipped public transport	132
6.4	Crowd and traffic flow simulation	132

List of Tables

3.1	Segmentation results of 3D datasets of Blaise Pascal University.	47
3.2	Classification results of Blaise Pascal University dataset 1 with the new evaluation metrics.	51
3.3	Classification results of Blaise Pascal University dataset 2 with the new evaluation metrics.	51
3.4	Classification results of Blaise Pascal University dataset 3 with the new evaluation metrics.	55
3.5	Classification results of 3D Urban Data Challenge dataset (scene-A) with the new evaluation metrics.	55
3.6	Classification results of 3D Urban Data Challenge dataset (scene-B) with the new evaluation metrics.	56
3.7	Classification results of 3D Urban Data Challenge dataset (scene-C) with the new evaluation metrics.	56
3.8	Classification results evaluated using different metrics	58
3.9	Overall segmentation and classification accuracies when using RGB-Color and reflectance intensity values.	58
3.10	Parameters/Features description.	59
3.11	Comparison results for classification methods.	60
3.12	Comparison of classification results.	62
4.1	Object classification chart.	81
4.2	Type of changes.	86
4.3	Change detection evaluation.	97
4.4	Classification results for Permanently Static objects (F_1 -Score).	99
5.1	Choice of weights for estimation.	116
5.2	Comparison of detection result.	119
5.3	Window detection evaluation using metrics defined in [Vih12].	123

INTRODUCTION

This chapter introduces the context of the work presented in this thesis. It first describes in detail the background of the project framework in which this work is realized and then explains the problem statement. Whereas, the overall aim of the project is to develop and make available to the wide public a web based 4D Geographical Information System

(GIS) for cities, the work presented in this thesis contributes in the development of automatic, efficient and effective 3D urban cartography in this regard. The main aims along with the major contributions presented in the thesis are then briefly explained. Finally, the organization of the manuscript along with a brief summary of the chapters contained in this document is presented.

1.1 Context

This PhD thesis is funded by the "Conseil Général de l'Allier" and is realized in the framework of ANR (French National Research Agency) project iSpace&Time (ANR-10-CONT-23). The aim of this project is to develop and make available to the wide public a web based 4D Geographical Information System (GIS) for cities.

As part of its mission and its activities on digitization and modeling of cities and territories, IGN (Institut National de l'Information Géographique et Forestière) provides among other services, management, development, administrative assistance and Support to "Géoportail". The "Géoportail" is a public access portal of Geographical Information System of the French territory with high resolution aerial and satellite images. In the last two years, an infrastructure similar to Géoportail THD (terrestrial) was developed under the project ANR iTowns [iTo14] (ANR MDCO 2007 certified by CapDigital pole) which is managed by IGN. This infrastructure enables general public to explore (view, navigate and search) cities from the ground level through a tool for immersive web browsing in a database of panoramic images, acquired by a mobile terrestrial mapping system called STÉRÉOPOLIS as shown in Fig. 1.1. The technology used in iTowns information system allows one to move in a street view like manner in a panoramic topological graph but jumping discretely from one image to another (hot-spot jumping) and also to query a search engine whose database is enriched via automatic annotation of images.



Figure 1.1: Panoramic views of Paris area from iTowns acquired by a mobile terrestrial mapping system called STÉRÉOPOLIS (source [iTo14]).

The limitation of such systems like that of iTowns is that it does not take into account the dimension of time, evolutionary infrastructure and variable phenomenon in the city. The idea of having a GIS based on the concept of a "living city" obliges the regular update of information like traffic flows (cars, pedestrians, boats) and weather conditions etc. in the database. iTowns does not allow the integration of knowledge from other external (e.g. Wikipedia) or existing databases, to not only enrich the research information but also make the interaction with users more fruitful.

So, in this project it is proposed to use all this data to browse and query the GIS in present time (the real flow of traffic, the real weather conditions etc.) and retrospectively the evolution of these conditions over time. Also, it will allow to simulate different moving objects (like pedestrians, cars, etc.) in the scenes (as shown in Fig. 1.2) and also predict the impact of changing conditions like traffic flows etc., thus, paving way for a real like representation and management of a realistic 4D GIS for cities. The objective of the project iSpace&Time, a logical continuation of project iTowns, is to fill these gaps and thus develop such a portal integrating different technologies like: web2.0, virtual and augmented reality, animation and simulation.

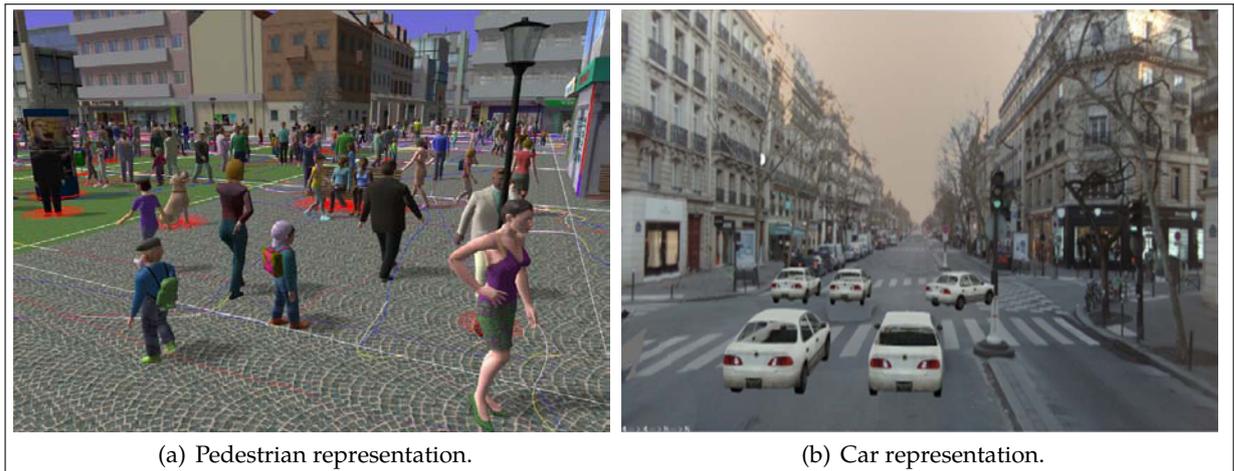


Figure 1.2: (a) & (b) present crowd [YMP+09] and traffic flow information [Bré13] respectively in the GIS using simulated representations of pedestrians and cars.

1.2 Main Contributions

The main aims of this PhD thesis is to address the different problems currently faced in 3D urban cartography by developing new methods of 3D cartography using 3D LiDAR data acquired from terrestrial mobile mapping systems.

The work conducted in this framework not only advanced the state of the art but also made several important contributions which are discussed in this thesis:

- ▷ A segmentation based method for classification of 3D urban point clouds using a super-voxel based approach followed by the characterization of 3D urban point clouds using a new point matching technique.
- ▷ A method of occlusion management exploiting the concept of multiple passages.
- ▷ A novel method for automatic change detection, imperfection removal and incremental updating of 3D urban cartography.
- ▷ Automatic detection and parameter estimation of windows for refining building façades exploiting symmetrical and temporal correspondences.

This work has, so far, produced 15 publications including 3 international journal papers and 2 book chapters. Details of these publications are presented in [Appendix A](#).

1.3 Organization of the Document

This document contains 6 chapters which are summarized as follows:

Chapter 2, entitled **3D Urban Cartography**, introduces the domain of 3D urban cartography which has gained widespread interest and importance in the scientific community due to an increasing demand for urban landscape analysis in different popular applications, coupled with advances in 3D data acquisition technology. As a result, it is observed that in the last few years, work on the 3D modeling and visualization of cities has intensified. Integrated in several geographical navigators like Google Street View, Microsoft Visual Earth or Géoportail, these models are accessible to a wide public who enthusiastically view the real-like representation of the terrain, created by mobile terrestrial mapping systems. In this chapter we also discuss the different methods, applications, recent data acquisition and mapping technologies as well as the different problems and challenges associated with this technology. A general workflow of our proposed methodology is then presented at the end of this chapter.

In *Chapter 3*, entitled **Segmentation and Classification of 3D Urban Point Clouds**, after shedding light on the existing state of the art techniques for segmentation and classification of urban environment, a method to classify urban scenes based on a super-voxel segmentation of sparse 3D data obtained from LiDAR sensors is presented. The 3D point cloud is first segmented into voxels which are then characterized by several attributes transforming them into super-voxels. These are joined together by using a link-chain method rather than the usual region growing algorithm to create objects. These objects are then classified using geometrical models and local descriptors. In order to evaluate the results, a new metric is presented which combines both segmentation and classification results simultaneously. The effects of voxel size and incorporation of RGB color and laser reflectance intensity on the classification results are also discussed. The method is evaluated on standard datasets using different evaluation metrics to demonstrate its efficacy.

Chapter 4 on **3D Urban Cartography - Classification and Incremental Updating** first gives a detailed overview of the different state of the art techniques for 3D urban cartography and then presents a new method of automatic 3D urban cartography incorporating classification and incremental updating, exploiting the concept of multiple passages. In the proposed method, the 3D point clouds are first classified into 3 main object classes: Permanently Static, Temporarily Static and Mobile using a point matching technique. The Temporarily Static and Mobile objects are then removed from the 3D point clouds leaving behind a perforated 3D point cloud of the urban scene. These perforated 3D point clouds obtained from successive passages (in the same place) on different days and at different times are then matched together to complete the 3D urban landscape. The changes occurring in the urban landscape over this period of time are detected and analyzed using cognitive functions of similarity and the resulting 3D cartography is progressively modified accordingly. Different specialized functions are also introduced, to help remove the different imperfections due to occlusions, misclassifications and different changes occurring in the environment over time, thus increasing the robustness of the method. The results, evaluated on real data, demonstrate that not only is the resulting 3D cartography accurate, containing only the exact permanent features free from imperfections, but the method is also suitable for handling large urban scenes.

Chapter 5, entitled **Detection and Feature Estimation of Windows for Refining Building Façades**, discusses our work on automatic detection and parameter estimation of windows for refining building façades. The chapter discusses the different state of the art practices in this domain and explains that due to the ever-increasing demand for more realistic 3D urban models coupled with recent advancements in ground based LiDAR technologies, recovering details of building façade structures, such as windows, has gained considerable attention. However, insufficient raw laser information available makes it very difficult to detect and recover reliable geometry of windows without human interaction. So, in this chapter we present a new method that automatically detects windows of different shapes in 3D LiDAR point clouds obtained from mobile terrestrial data acquisition systems in the urban environment. The proposed method first segments out 3D points belonging to the building façade from the 3D urban point cloud and then projects them onto a 2D plane parallel to the building façade. After point inversion within a watertight boundary, windows are segmented out based on geometrical information. The window features/parameters are then estimated exploiting both symmetrically corresponding windows in the façade as well as temporally corresponding windows in successive passages, based on ANOVA measurements. This unique fusion of information not only accommodates for lack of symmetry but also helps complete missing features due to occlusions. The estimated windows are then used to refine the 3D point cloud of the building façade. The results, evaluated on real data using different standard evaluation metrics, not only demonstrate the efficacy but also the technical edge of the proposed method.

Finally the conclusions and perspectives are presented in *Chapter 6*.

3D URBAN CARTOGRAPHY

3D urban cartography has gained widespread interest and importance in the scientific community due to an increasing demand for urban landscape analysis in different popular applications, coupled with advances in 3D data acquisition technology. As a result, in the last few years, work on the 3D modeling and visualization of cities has intensified. Integrated in several geographical navigators like Google Street View, Mi-

crosoft Visual Earth or Géoportail, these models are accessible to a wide public who enthusiastically view the real-like representation of the terrain, created by mobile terrestrial image acquisition techniques. In this chapter, we give a brief introduction of 3D urban cartography; discussing the different methods, applications, recent data acquisition and mapping technologies as well as the different problems and challenges associated.

2.1 Introduction

Cartography (from Greek *χάρτης*, *khartēs*, "map"; and *γράφειν* *graphein*, "write") is the study and art of making maps [The14]. Evolving over time, it combines science, aesthetics, and technique, to build on the premise that reality can be modeled in ways that communicate spatial information effectively. A modern definition of 3D cartography is that: "it is the totality of scientific, technical and artistic knowledge and activities, concerning the complete lifecycle of 3D maps, including: the collection of data, the process of production and distribution, the underlying technologies and algorithms, as well as the forms of use and applications" [Gór09].

Over the years, 3D urban cartography have gained widespread interest and importance in the scientific community due to an ever increasing demand for urban landscape analysis for different popular applications, coupled with advances in 3D data acquisition technology. As a result, in the last few years, work on the 3D modeling and visualization of cities has intensified.

The type of cartographic data depends upon the application for which it is acquired. Figure 2.1 is a simple example of cartographic data of Capitol Hill in Washington, USA acquired to create 3D city models.

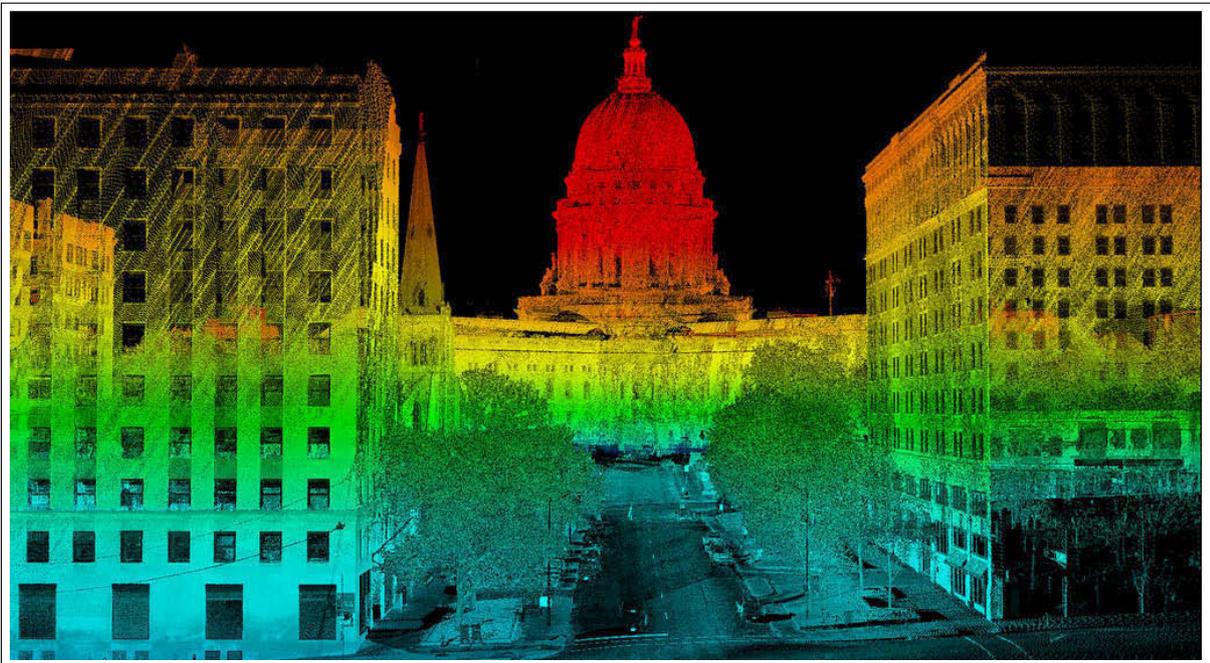


Figure 2.1: 3D data acquisition of the Capitol Hill area in Washington USA using a terrestrial scanner (source [3D 14]).

In this chapter, we will first present different applications of the cartographic data in the urban environment and then discuss the most recent data acquisition technologies, mainly terrestrial Mobile Mapping Systems (MMS).

2.2 Applications of 3D Urban Cartography

The nature of the cartographic data depends upon the different applications envisaged. In this section, we present some of the more recent and popular applications of 3D urban cartography.

2.2.1 Navigation

One of the most common and widely used applications of 3D urban cartography is geographical navigation. The different 3D models of the environment, registered with existing road maps

allow precise localization with the help of GNSS (Global Navigation Satellite System) and hence allow easy navigation of both vehicles and pedestrian in the urban environment. Figure 2.2 shows some examples of this type of applications.



Figure 2.2: (a) & (b) show the use of the data for navigation purposes (source [Ter14]).

2.2.2 Virtual Tourism and Preservation of Cultural Heritage Sites

The recent advancements in 3D cartography of different urban areas and specific locations have improved the quality, usability and accessibility of virtual tours considerably (Google Street View [GBR+08]). With a single click, such realistic 3D models allow the viewer to view these sites with great detail at the comfort of his own home. Figure 2.3 shows the use of cartographic data of Arc de Triomphe and Rue Soufflot in Paris for virtual tours.



Figure 2.3: (a) & (b) using the data for virtual tourism (source [Urb14]).

Similarly, more and more cultural heritage sites are also acquired not only to provide virtual tours but also to conserve a digital model of the site in actual condition for analysis, studies, research and for future references. In Fig. 2.4, you can see the rendered 3D point cloud data of the famous Sydney Opera House (a UNESCO World Heritage site) in Australia, acquired using a Faro scanning system for CyArk in 2013 (source [CyA14]).

2.2.3 Infrastructure Analysis and Management

3D urban cartography is also used effectively for the analysis and management of different city infrastructures such as carriageways sidewalks, retaining walls, bridges, tunnels, road drainage

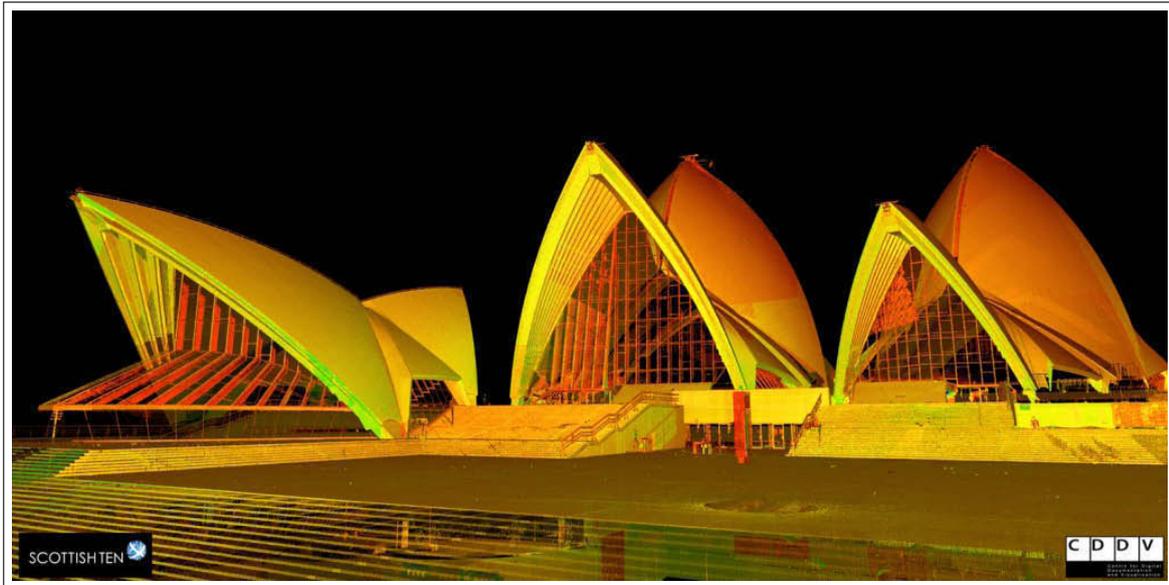


Figure 2.4: 3D point cloud data of Sydney Opera house (source [CyA14]).

structures, etc. [RH06]. Another application is in case of transportation of oversize cargo, where accurate and rapid mapping of route clearance (bridge height, road width) can be performed for the entire route [HCK06]. Figure 2.5(a) shows the data being used for analyzing power lines in the area, which can be used for low wire identification, vegetation encroachment and sag measurement while Figure 2.5(b) shows the data being used to study bridges.

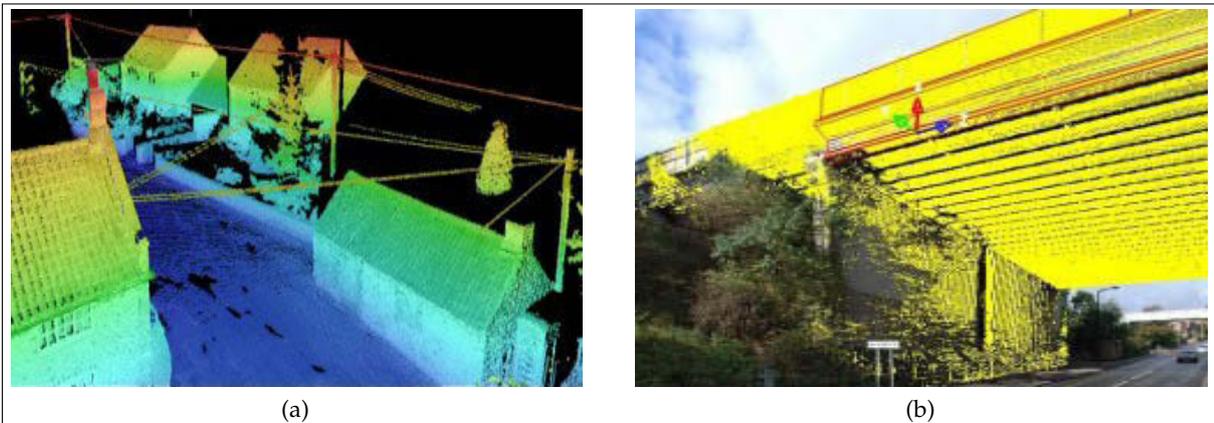


Figure 2.5: Using the data for Infrastructure analysis (source [3DI14]).

2.2.4 Road Asset Management and Driver Assistance

The 3D urban cartography can be effectively used to collect accurate road and highway data. Several government transportation agencies are increasingly expected to document and monitor every asset on their roads to meet federal mandates, increase road efficiency and improve safety. This could also be used to conduct surveys without disruption to traffic which can help improve maintenance and asset management of road surfaces [LC05] and street/railway furniture e.g. signs, markings, etc. [RH06]. The benefits range from protecting the environment to saving lives. Driver assistance technologies are also being developed by researchers to also improve road safety by providing computer assistance with braking, steering, parking, collision avoidance and other things. With access to very accurate road and sign data, among other things, the car's

computer could also sense if the car was leaving the road or alert the driver if exceeding the speed limit. An example is shown in Fig. 2.6.

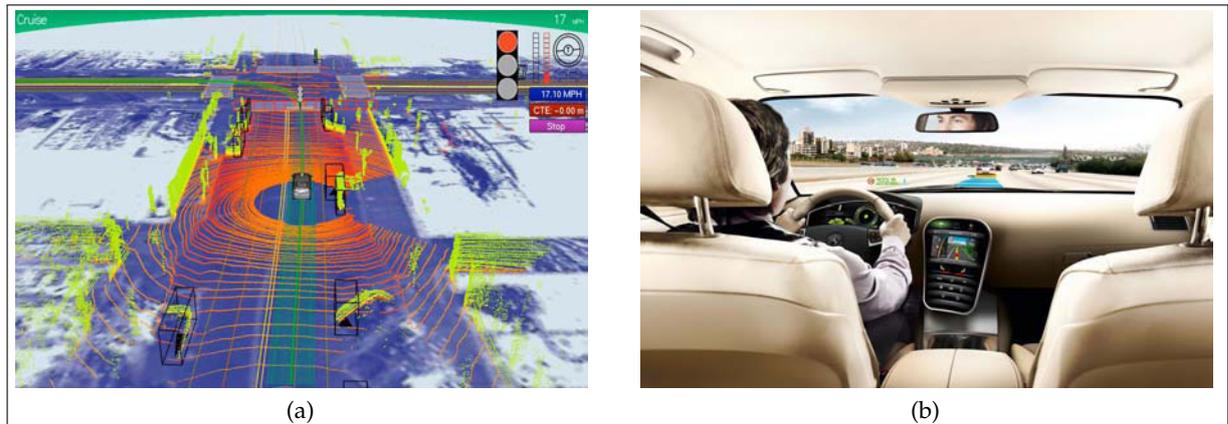


Figure 2.6: Using the data for road asset management and driver assistance (source web *Google X*).

2.2.5 Urban City Analysis and Planning

In recent times, 3D urban cartography data is also used for urbanism. The data is used for analyzing, planning and simulating different aspects related to urban growth, population distribution and management of natural resources, etc. Figure 2.7 shows one such example in which the acquired data is used for city modeling.

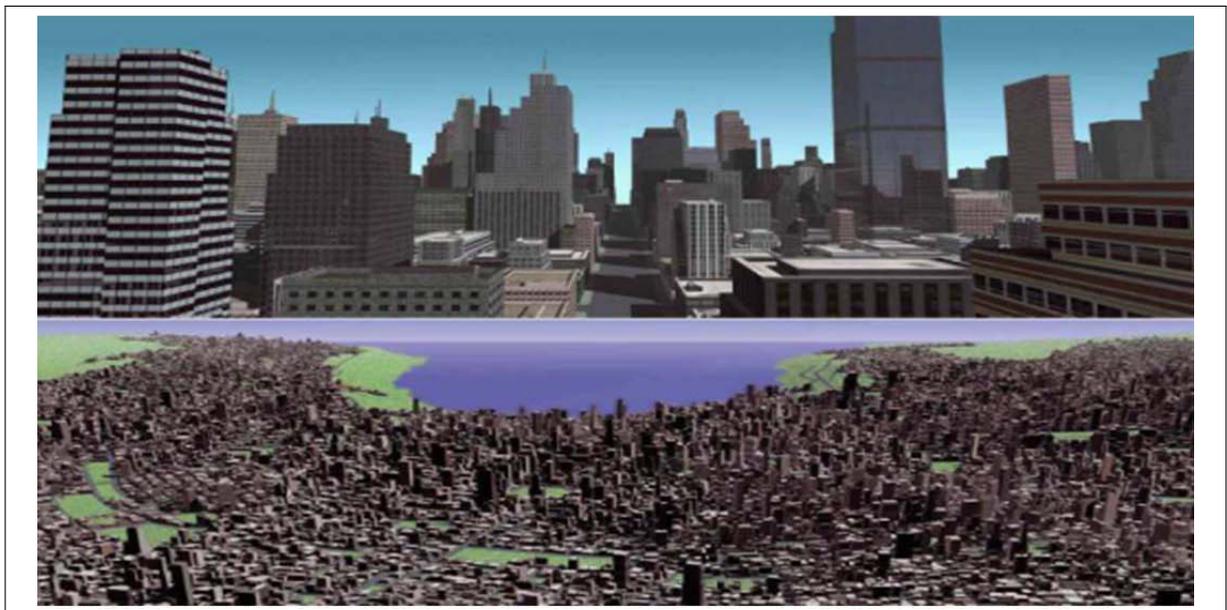


Figure 2.7: Using the data for Urban city analysis (source [[Ter14](#)]).

Other applications include 3D modeling for simulations of urban disaster or terrorism scenarios, increasing immersive experience in video games and other multimedia applications such as films and animations, etc.

2.3 Terrestrial Mobile Mapping Systems

The increasing demand of acquiring detailed 3D data of large urban environment for different cartography applications in both limited time and budget has led to the rapid advancements in the 3D data acquisition technologies.

The aerial data acquisition systems allows rapid acquisition of large areas. However, the acquired data does not contain sufficient details. On the other hand, the static terrestrial data acquisition systems provide high level of details but are impractical to acquire large mapping areas like entire streets, neighborhoods or cities, etc. Hence, the terrestrial mobile mapping systems are a best compromise of these two systems.

A definition of the Mobile Mapping Systems (MMS) according to Ellum and El-Sheimy [EES02] is: "MMS integrate navigation sensors and algorithms together with sensors that can be used to determine the positions of points remotely. All the sensors are rigidly mounted together on a platform; the former sensors determine the position and orientation of the platform, and the later sensors determine the position of the points external to the platform".

As also indicated in the above mentioned definition, Mobile Mapping Systems collect geospatial data using a mobile platform typically equipped with a range of photographic, radar, LiDAR or other remote sensors. Such sensors are also coupled with time synchronized navigation sensors and the primary output from such systems includes GIS data, digital maps, and geo-referenced images, videos and 3D point clouds. As shown in Fig. 2.8, the system is mainly composed of three parts; 1) Perception system, 2) Localization system and 3) Mobile Platform. The perception system consists of different data acquisition sensors like cameras and LiDAR scanners while the localization system consists of different sensors like GPS, Inertial Measurement Unit (IMU), odometer, etc. The Mobile Platform consists of not only the vehicle itself, but also the sensor mounts, the main control unit and other accessories.

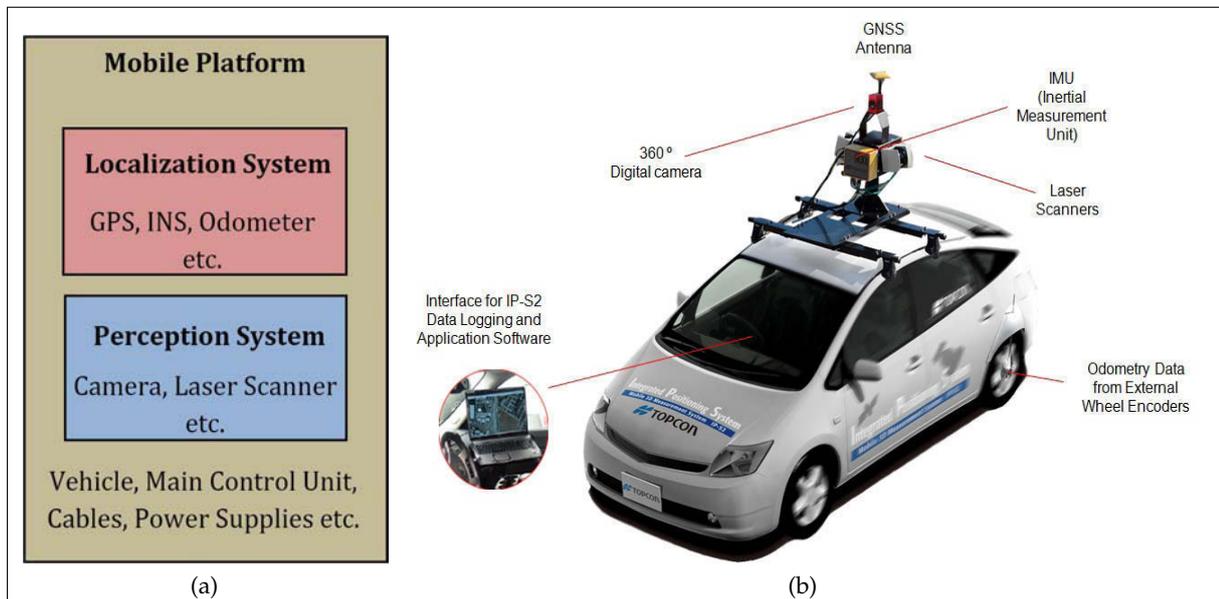


Figure 2.8: (a) The basic composition of terrestrial Mobile Mapping Systems. (b) TOPCON IP-S2 system equipped with different components.

2.3.1 Mobile Mapping Systems: Standard Technologies

In general, the most common configuration of a mobile mapping system integrates digital frame cameras, laser scanner, GPS receiver and antenna, Inertial Navigation System (INS) for

acceleration and orientation measurements of the moving platform and, in most cases, a wheel-mounted Distance Measuring Indicator (DMI), that provides accurate vehicle velocity updates.

The positioning sensors, namely GPS, INS and DMI, are vehicle-oriented. They are used to determine the absolute locations of the mobile mapping platform with respect to a global coordinate system, e.g., WGS-84. The mapping sensors (laser scanners, video or digital cameras, etc.) are feature-oriented. They provide the positional information of objects (features) relative to the vehicle in a local coordinate system. In addition, attributes of features can be obtained from the mapping sensors. The use of multiple camera arrays to provide 360° panoramic images in the horizontal plane is very common (Fig. 2.9(a)), however frequently fully integrated multiple cameras are used (e.g. Ladybug unit used in the IP-S2 system, Fig. 2.9(b)).

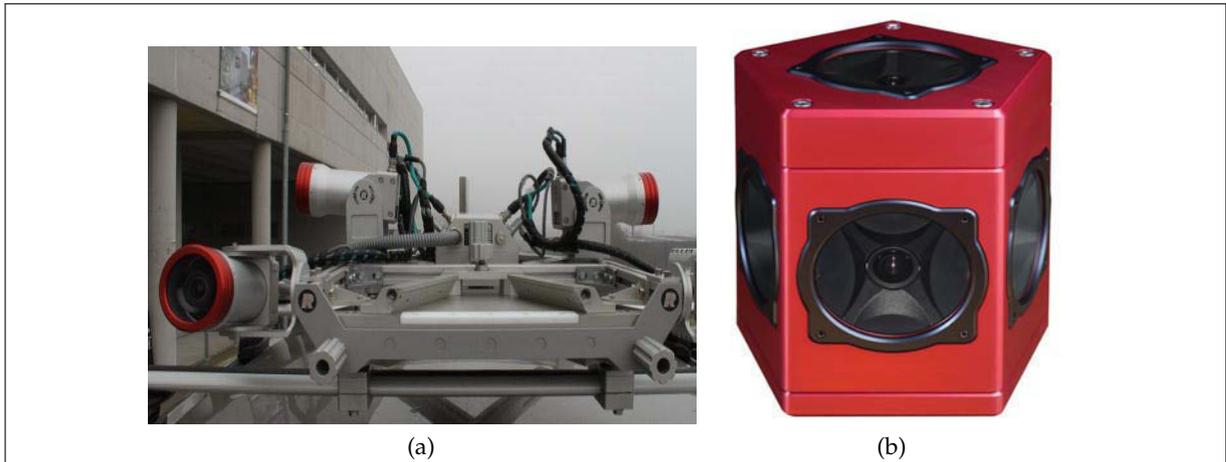


Figure 2.9: (a) TVMX-250-CS6 Camera System supporting up to six digital color cameras. (b) Ladybug with a set of six Sony CCD cameras arranged in a similar circular five-camera configuration plus a single vertical camera (source [Poi14]).

A basic laser scanner measures its surroundings using LiDAR for measurements of range and angle. Currently, range measurements for mobile laser scanning systems are divided mainly into two techniques: time-of-flight and phase shift. A Time-Of-Flight (TOF) scanner sends a short laser pulse to the target; the time that elapses between emitted pulse and received pulse correlates to the range. Examples of laser scanners using this principle include: Riegl (2011) (Fig. 2.10(a)), Leica (2011), Trimble (2011), Optech (2011), 3rdTech (2011) and Sick (2011). Phase based laser scanners, Zoller+Fröhlich (2011) and Faro (2011) (Fig. 2.10(b)), infer the range by measuring the phase difference between the emitted and received backscattered signal of an Amplitude Modulated Continuous Wave (AM CW).

Phase shift laser scanners have better accuracy, although the measurement range is lower. Highly accurate xyz-coordinates of the ground points for each laser pulse can be calculated by combining the laser range, scan angle, laser position from GPS and orientation of the laser platform from INS.

The laser pulse repetition rate (PRR) in combination with scanning mirror deflecting pattern determines LiDAR data collection rate. In the most advanced commercially available LiDAR systems, the data measurement rate is typically 50 kHz to 500 kHz, which allows the user to collect highly accurate data of required ground point density within a very short period of time.

Inertial navigation provides real-time indication of position and velocity of a moving vehicle using IMU sensors that react on the basis of Newton's laws of motion. Two primary types of IMUs are accelerometers, which sense linear acceleration in the inertial frame, and gyroscopes, which sense the inertial rotational motion (angular rates, angular increments or total angular displacements from an initial known orientation relative to inertial space). Considering an accelerometer, the first integration of acceleration gives the velocity and the second integration the position or rather, change of position along the accelerometer's axis.



Figure 2.10: (a) 2 Laser scanners VQ-250 mounted on the VMX-250 system supplied by Riegl. (b) Faro PHOTON 120 mounted on ROAD-SCANNER system.

The three types of IMUs that are most commonly used in mobile mapping systems are: (i) MEMS (Micro Electro-Mechanical Systems) Gyros, that utilize tiny quartz tuning forks as sensors integrated on to silicon chips. They are widespread used in the imaging systems that are employed on less demanding applications. Not much used in laser scanners. (ii) Fibre Optic Gyros (FOG) are much more expensive than MEMS gyros but gives a very acceptable performance that satisfies many laser scanning applications. Much used nowadays. (iii) Ring Laser Gyros (RLG) are the most accurate type but there are the most expensive to produce, which limits their use to high-end imaging systems and laser scanners and to only the most demanding applications in terms of accuracy.

In principle, the only external information the INS requires is initial calibration (initialization and alignment), which includes the externally provided 3D initial position, velocity and attitude. The sensor errors grow with time; therefore INS must be re-calibrated periodically to maintain reliable navigation quality. In stand-alone mode, the INS navigation results are primarily affected by initial sensor misorientation, accelerometer biases and gyro drifts that cause time-dependant positioning errors.

Precise calibration is required to geometrically align the positioning sensors and mapping sensors together. A sophisticated software unit controls these components and accurately synchronizes the measurements recorded through a time stamp.

Ground based navigation solution, integrated with the laser scanner, is a critical component for any mobile LiDAR system as it is used to obtain geo-referenced coordinates of the collected LiDAR data (discussed in the next section).

2.3.2 Direct Geo-referencing

Direct geo-referencing is defined as the process of determining the time-variable spatial position of a point (or points) scanned by a mobile mapping sensor with reference to a global coordinate system. For accurate geo-referencing of the scanned point, it is essential to determine the 3D instantaneous coordinates of the center of mass of the LiDAR sensor with reference to the mapping frame (i.e. the local or a global coordinate system). The 3D coordinates of the sensor's center of mass is a time varying function that depends on the observed measurements of the sensor's position and orientation. At present, the key component responsible for taking the position and orientation measurements in both terrestrial and airborne LiDAR mapping sensors is the integrated GPS/IMU system. Presuming that the GPS/IMU systems are able to determine the orientation of the integration platform at any moment, then the reference frame from the

GPS/IMU to the laser can be transferred, if firstly the laser data is synchronized with the GPS/IMU observations and secondly the spatial transformation between the GPS/IMU frame and the laser frame is known, i.e. determined in a calibration process.

The laser labels the beginning of each line with a precise internal clock. The synchronization is performed by relating the laser time system to the GPS time system. The laser internal clock can be reset by an external TTL signal. By employing this possibility, a modified pulse per second signal can be pseudo-randomly sent to the laser.

Once the GPS time at the start of each line is known, the corresponding GPS time of every subsequent laser point can be computed by simply adding the laser repetition period to the time of the previous laser point.

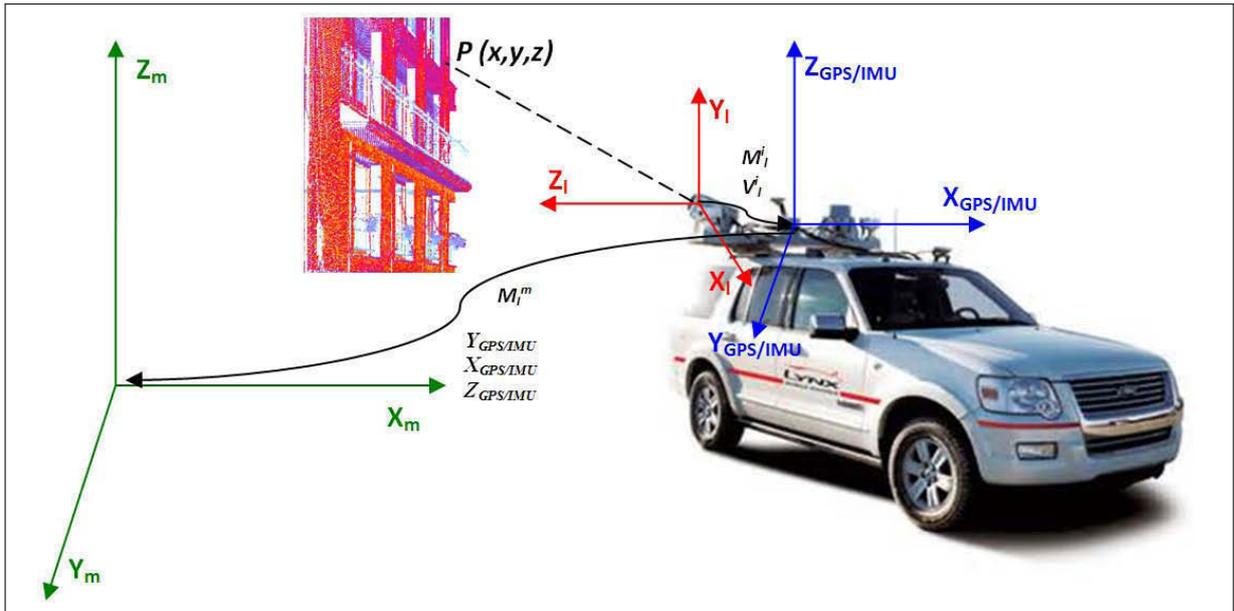


Figure 2.11: The direct geo-referencing of a 3D point P acquired by the laser scanner into the mapping frame of reference.

Hence, after every laser point is time labeled in GPS time, the transformation between the laser reference frame (X_l, Y_l, Z_l in Equation (2.1)) and the mapping reference frame (WGS84) can be done according to Equation (2.1) in two steps: a) transformation from the laser to the IMU reference frame and b) transformation from the IMU to the mapping reference frame (X_m, Y_m, Z_m in Equation (2.1)). The first step consists in determining the offset between the laser and the IMU reference frames (V_l^i in Equation (2.1)) and the misalignment matrix between the laser and the IMU reference frames (M_l^i in Equation (2.1)). The offset and the misalignment matrix (V_l^i and M_l^i) remain constant as long as both systems (IMU and laser) are rigidly mounted on the integration platform and the platform does not suffer from any distortion due to stress. These constant values are determined in a calibration process. In the second step, the rotation matrix (M_i^m in Equation (2.1)) and the translation vector ($X_{GPS/IMU}, Y_{GPS/IMU}, Z_{GPS/IMU}$ in Equation (2.1)) are determined using the integration of the GPS/IMU observations. It should be noted that the rotation and translation matrices applied in the second step are not constant and keep varying during the scanning process. Therefore, for each laser point a different translation and rotation matrix will be derived from the computed trajectory. The frame transformations are presented in Fig. 2.11.

$$\begin{pmatrix} X_m \\ Y_m \\ Z_m \end{pmatrix} = \begin{pmatrix} X_{GPS/IMU} \\ Y_{GPS/IMU} \\ Z_{GPS/IMU} \end{pmatrix} + \mathbf{M}_i^m \mathbf{M}_l^i \left(\begin{pmatrix} X_l \\ Y_l \\ Z_l \end{pmatrix} + \mathbf{V}_l^i \right) \quad (2.1)$$

The advantage of using integrated GPS/IMU data to directly orient laser data is that the transformation between the local laser reference frame and the mapping reference frame is known at any given moment (as long as the laser is synchronized), independently if the laser is collecting data in a static mode or in kinematic mode. Thus, the laser can be used as a push-broom sensor sweeping the scene with profiles while fixing the scan angles as the vehicle moves. The different modes of operation are discussed in the next section.

2.3.3 Modes of Operation

There are typically two operation modes for a mobile mapping system, depending upon the combination of perception and localization information [VM10]: "On-the-Fly" and "Stop-and-Go" mode. These modes are explained below.

2.3.3.1 On-the-Fly Mode

In On-the-Fly mode, the Mobile Mapping System (MMS) acquires data while in motion i.e. the sensors mounted on the mobile platform continue to function as the platform displaces. Figure 2.12 shows the principle of this mode. The large arrow represents the direction of motion of the vehicle while the small arrows represent the direction of data acquisition of the sensor, for example a 2D laser scanner. As the vehicle moves the acquisition of the sensor is in helical form. The sensor is used as a push-broom sensor, sweeping the scene with profiles while fixing the scan angles as the vehicle moves.

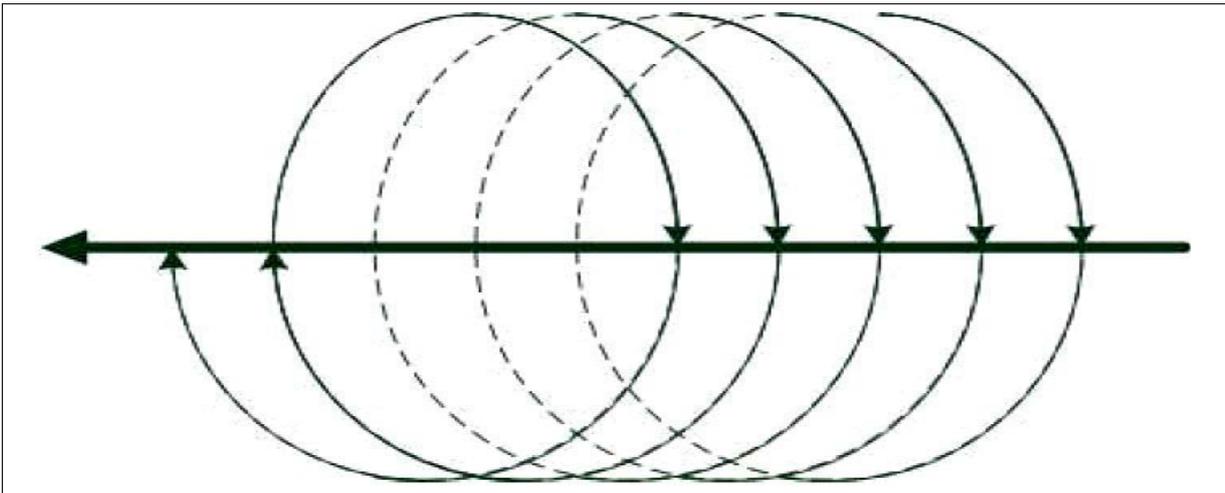


Figure 2.12: Principle of On-the-Fly mode.

The main advantage of this mode lies in the fast acquisition rate (as the acquisition time is equal to the displacement time of the vehicle). On the other hand, the data should be correctly synchronized in order to obtain coherent information. Most of the recent Mobile Mapping Systems [PGJMS+13; GCM+12] operate in this mode. In Fig. 2.13, the 3D data acquired by a Riegl MMS VMX-450 using On-the-Fly mode is shown.

2.3.3.2 Stop-and-Go Mode

The second type of operation mode is the Stop-and-Go mode. In this mode, as the name suggests, the mobile platform or vehicle stops intermittently during its displacement to scan the environment. Hence, the position and orientation of the platform remain static during acquisition. The MMS hence moves, stops and acquires data and then moves again. This process is repeated until all the area is scanned.



Figure 2.13: 3D data acquired by Riegl MMS VMX-450 using On-the-Fly mode (source [Rie14]).

Figure 2.14 shows the principle of the Stop-and-Go mode. The red circles/points represent the scanning (stopping) positions of the vehicle while the big arrow represents the direction of displacement of the mobile platform/vehicle. During the acquisition the vehicle remains stationary and the scanners scan in 2D. Once the scanning is complete, the vehicle moves to the next position.

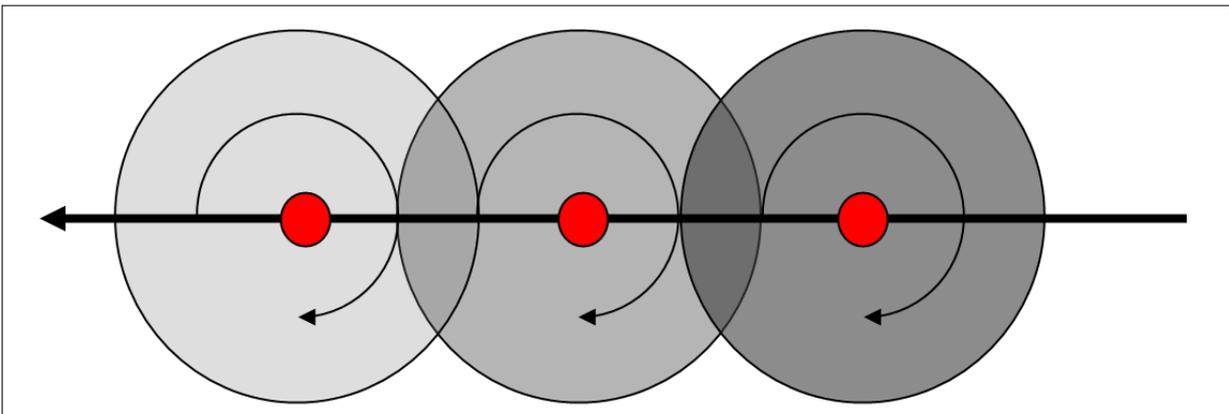


Figure 2.14: Principle of Stop-and-Go mode (Top View).

This operation mode is used by few Mobile Mapping Systems like CityGRID [Har07] or the system of NAIST Institute [AKY05]. Figure 2.15 shows the data (both image and LiDAR) acquired by the MMS of NAIST Institute. The advantage of this mode of operation is that synchronization between the perception and localization data is no longer necessary. Hence, the acquired data is much more precise than in case of On-the-Fly mode. However, as we need to stop every time for acquisition, the acquisition time is quite long which makes it less practical for scanning large areas.

In our work, as explained in this thesis, we mainly use the data acquired from different Mobile Mapping Systems employing On-the-Fly mode.

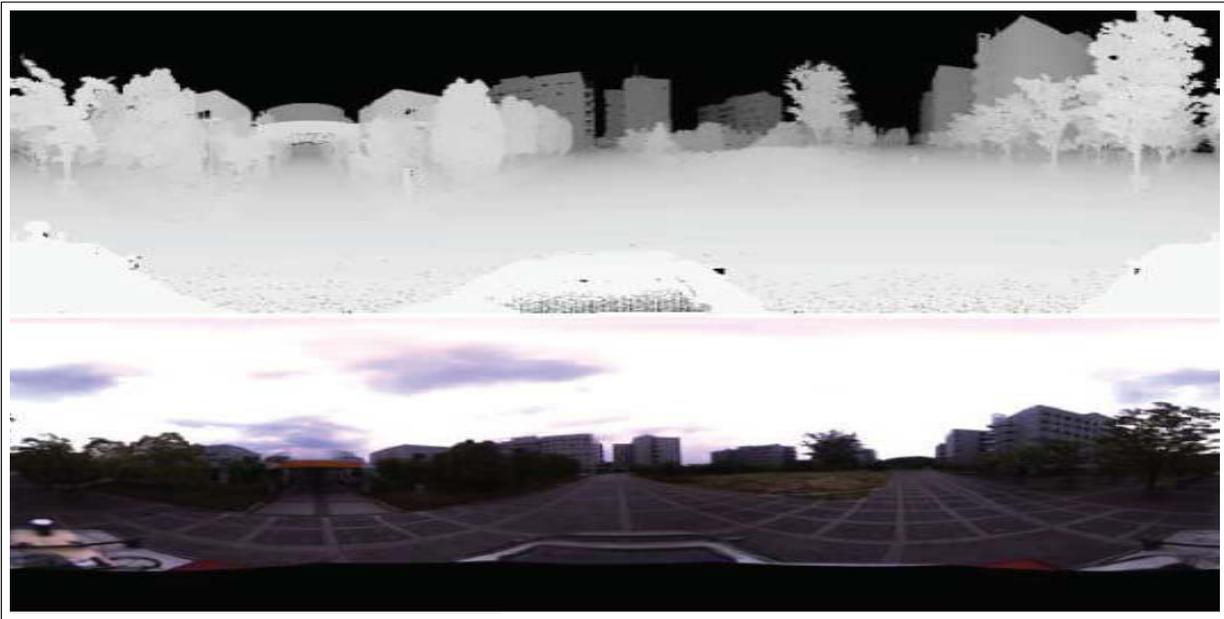


Figure 2.15: 3D data acquired by NAIST System using Stop-and-Go mode (source [AKY05])

2.3.4 Existing Mobile Mapping Systems (MMS)

The development of Mobile Mapping Systems (MMS) began in the late 1980s and early 1990s, when the first usable land-based MMS was developed by the Centre for Mapping at Ohio State University. Their system, called GPSVan™, integrated a code-only Global Navigation Satellite System (GNSS) receiver, two digital CCD cameras, two color video cameras and several dead-reckoning sensors (two gyroscopes and a distance measurement unit on each of the front wheels), all of which were mounted on a van [Goa91] and [Nov91]. Mobile Mapping Systems have evolved ever since. There are two main types of MMS; one that relies only on photogrammetry to acquire data of the urban environment for example "Google Car" of Google Inc. as shown in Fig. 2.16(a) and secondly those which use lasergrammetry to acquire data for instance StreetMapper-1 (version-2009) as seen in Fig. 2.16(b). Google car has a collection of nine cameras that acquire image data used in applications like "Street View" of Google Maps (source *Web Google Maps*). Whereas, the StreetMapper-1 is equipped with 4 Riegl LMS-Q120i scanners (source [3D114]) that acquires data in the form of point clouds.

Compared to these systems, most of the recent MMS are equipped with both cameras and LiDARs and acquire both images and 3D point clouds of the environment, for example the STÉRÉOPOLIS II system [PPC+12] (Fig. 2.17(a)) and VizTruck [Vis11] (Fig. 2.17(b)). Some of these systems like RIEGL VMX-450 [Rie14] (Fig. 2.17(d)) and Lynx SG1 [Opt14] (Fig. 2.17(c)) combines both camera and LiDAR information to obtain colored 3D point clouds [SBV+11], [CZ13] and [PME11] as shown in Fig. 2.18.

Commercial suppliers of MMS not only include several large companies such as TRIMBLE and TOPCON as well as some medium size companies like RIEGL, OPTECH, MDL or SITECO which are already well-established as suppliers of surveying instrumentation and laser scanning systems for mapping industry. Details of some of these systems are provided below whereas a comparison of few of these commercially available systems is presented in Fig. 2.19.

2.3.4.1 StreetMapper System

The StreetMapper system is specifically designed by the 3D Laser Mapping company for mobile mapping use when mounted on a suitable vehicle (Fig. 2.16(b)). The company has developed this system in close collaboration with the German systems supplier, IGI. For use in the StreetMapper,



Figure 2.16: (a) Google car with panoramic camera. (b) StreetMapper system of 3DLM with 4 LiDARs (source [3DI14]).

IGI supplies its TERRAcontrol DGPS/IMU system with different laser scanning and digital imaging systems. The dual-frequency GPS receiver can come from any one of several suppliers.

3D Laser Mapping supplies the hardware and software solutions that are used for the mission planning, the control of the laser scanners and the data storage within a StreetMapper system. The control unit and its computer are housed in a cabinet that is mounted inside the mapping vehicle. Until now, the multiple laser scanners that have been used on StreetMapper systems have been supplied by Riegl Inc. On most existing StreetMapper systems, between two and four of the older LMS-Q120 scanner units (with their 150 m range) have been fitted on a roof rack, together with the IMU and the GPS antenna. However the latest StreetMapper 360 systems utilize the newer Riegl VQ-180 or VQ-250 units, the former having a 100 degree FOV and a range of up to 150 m; the latter giving a full circle (360°) scan and ranges up to 300 m. A choice of video or digital still frame cameras from different manufacturers can be supplied in order to generate the higher quality images that will be needed to supplement the laser scanned data. On the software side, IGI also contributes its TERRAoffice software for the processing of the IMU data, while the differential GPS data is processed using the Graf-Nav package that is supplied by the Waypoint division of NovAtel, Canada. The TerraScan/TerraModeler/TerraMatch suite



Figure 2.17: (a) STÉRÉOPOLIS II system of (source [Ste14]). (b) Vis center's VizTruck. (c) Lynx SG1 mapping system (source [Opt14]). (d) Riegl VMX-450 mapping system (source [Rie14]).



Figure 2.18: True color coded point cloud acquired by Riegl VMX-450 mapping system (source [Rie14]).

Scanner	FARO Photon 120	SICK LMS 291	VQ-250	MDL scanner	LYNX laser scanner
MLS System	ROAD SCANNER	IP-S2	VMX-250	DYNASCAN	LYNX MOBILE MAPPER
Measuring principle	Phase difference	TOF	TOF	TOF	TOF
Maximum range	120 m (@ ρ 90%)	30 m (Max. 80 m @ ρ 10%)	200 m (for 300 kHz, @ ρ 80%)	Up to 500 m	200 m (@ ρ 20%)
Range precision	1 mm @ 25 m, ρ 90% 2.7 mm @ 25 m, ρ 10%	10 mm at range 1 to 20 m	5 mm @ 150 m, (1σ)		8 mm, 1σ
Range accuracy	\pm 2 mm @ 25 m	\pm 35 mm	10 mm @ 150 m, (1σ)	\pm 5 cm	\pm 10 mm, (1σ)
Laser measurement rate	122–976 kHz	40 kHz	50–300 kHz	36 kHz	75–500 kHz
Measurement per laser pulse			Practically unlimited		Up to 4 simultaneous
Scan frequency	48 Hz	75 Hz	Up to 100 Hz	Up to 30 Hz	80–200 Hz
Laser wavelength	785 nm (near infrared)	905 nm (near infrared)	Near infrared	–	1550 nm (near infrared)
Scanner field of view	H360°/V320°	180° ^a /90° ^b	360°	360°	360°
Operating temperature	5°–40 °C	0°–50 °C	10°–40 °C	–20°–60 °C	10°–40 °C
Angular resolution	H0.00076°/V0.009°	1° ^a /0.5° ^b	0.001°	0.01°	0.001°
Weight	14.5 kg	4.5 kg	Approx. 11 kg	11 kg	–
Main dimensions (L × W × H)	410 × 160 × 280 mm	155 × 156 × 120 mm	376 × 192 × 218 mm	595 × 240 × 255 mm	620 × – × 490 ^c mm

^a SICK LMS 291-S05 used in IP-S2 Compact system.
^b SICK LMS 291-S14 used in IP-S2 Compact system.
^c Wide length is not provided in the specification sheet.

Figure 2.19: Comparison of some commercially available Mapping systems (source [PGJMS+13]).

of programs from Terrasolid in Finland is then utilized for the processing of the laser scan data and its transformation into the final 3D elevation model data.

The StreetMapper system has been supplied to a number of international customers, including Geomaat (Netherlands), Transport and Road Research Institute (Lithuania), Geokosmos (Russia); Tecdawn (China) and Terrametrix and GeoDigital (USA). A StreetMapper system has also been used extensively by Halcrow, a large engineering consultancy company, to carry out corridor surveys along roads for highway asset management and to capture street level data in city centres in the United Kingdom.

2.3.4.2 Integrated Positioning System (IP-S2)

The IP-S2 has been introduced by Topcon Positioning Systems (as shown in Fig. 2.20) in 2009. There are more than 400 units of this system currently in use world-wide. One of its main users is Google Inc. The IP-S2 system includes a Topcon dual-frequency 40-channel GNSS receiver operating at 20 Hz, which is coupled to a Honeywell HG1700 tactical-grade IMU based on a ring laser gyro (RLG) that is operating at 100 Hz. The resulting DGPS/IMU positional data is supplemented by that generated by a wheel-mounted odometer with an angular encoder operating at 30 Hz to complete the overall positioning capability for the IP-S2 system. Besides these positioning devices, the imaging and laser scanning capabilities of the IP-S2 are based on well-known units that are available off-the-shelf. They include the Ladybug multi-camera unit from Point Grey Research that carries out the 360 degree panoramic imaging with framing rates of up to 15 frames per second. The laser scanning that is carried out using the standard configuration of the IP-S2 is provided by three Sick LMS 291 scanners operating at 75 Hz. One of these laser scanners is pointing directly forwards (or backwards) towards the road in front of (or behind) the vehicle, while the other two scanners point to each side to provide a continuous series of range or elevation profiles within the vertical plane. All of these imaging and scanning devices send their data to a central control box which then passes it via a high speed FireWire-B (IEEE1394-B) link to the PC that is mounted in the vehicle for the recording and processing of the data. An LCD display screen allows the vehicle's crew to monitor the connectivity and operation of all the various positioning, laser scanning and frame imaging devices. In the version of the system that is used in the Google Street View vehicles, the Ladybug multiple camera has been replaced by Google's own multiple camera system (Fig. 2.16(a)), which is based on the use

of Elphel digital frame cameras. The IP-S2 system is also used by the Geodis surveying and mapping company based in Brno in the Czech Republic.



Figure 2.20: Topcon's IPS2 mobile mapping system (source [Top14]).

2.3.4.3 Lynx Mobile Mapper

The Lynx Mobile Mapper (Fig. 2.21) was introduced by Optech Inc., which is based near Toronto in Canada towards the end of 2007. The system includes a purpose built spinning laser profiling system that is designed specifically for attachment to standard vehicle roof racks with mounts for two of these laser scanners and two calibrated digital frame cameras in its standard configuration. The Lynx system also includes an Applanix POS LV sub-system, complete with its IMU; a dual frequency GPS receiver and antenna; and a Distance Measuring Instrument (DMI), for coordinate positioning purposes. The laser scanners that are used in the Lynx system are built in-house by Optech and utilize a Class I laser as the basis for their laser rangefinders. They have a maximum range of 200 m; a full circle 360 degree angular coverage; a pulse measuring rate of upto 200 kHz ; and a scan rate of about 200 Hz . The system control unit with its embedded positioning and navigation solution that is based on the Applanix POS LV 420 DGPS/IMU sub-system can control up to four laser scanners simultaneously using the laptop computer that is attached to the unit. The Applanix POS PAC MMS (Mobile Mapping Suite) software is used to process the POS LV DGPS/IMU data, while Optech supplies its own LynxSurvey and DASHMap software for the final post-processing of the measured data. At the time of its introduction, Optech announced that Lynx systems had already been supplied to two European companies i.e. the Infoterramapping company based in the United Kingdom and the Sineco company in Italy. Since then, further systems have been sold to TopScan in Germany and Teccon in Belgium. Still more systems have been supplied to various North American users, e.g. to Aerial Data Service, Michael Baker, WH Pacific, Sanborn, Surveying and Mapping (SAM) Inc. and McKim and Creed in the USA Highway and railway infrastructure surveys and urban modeling surveys appear to be the main applications that have been undertaken by these mapping companies using their Lynx systems.

2.3.5 Mobile Mapping Services

There are several companies in the world that operate individual mobile mapping vehicles, offering their services especially to those agencies that are concerned with providing navigation solutions, online maps, highway management and maintenance, etc. by carrying out street-level imaging and mapping operations on a regional, national and international scale [Per10]. Some of the main service providers are presented below.

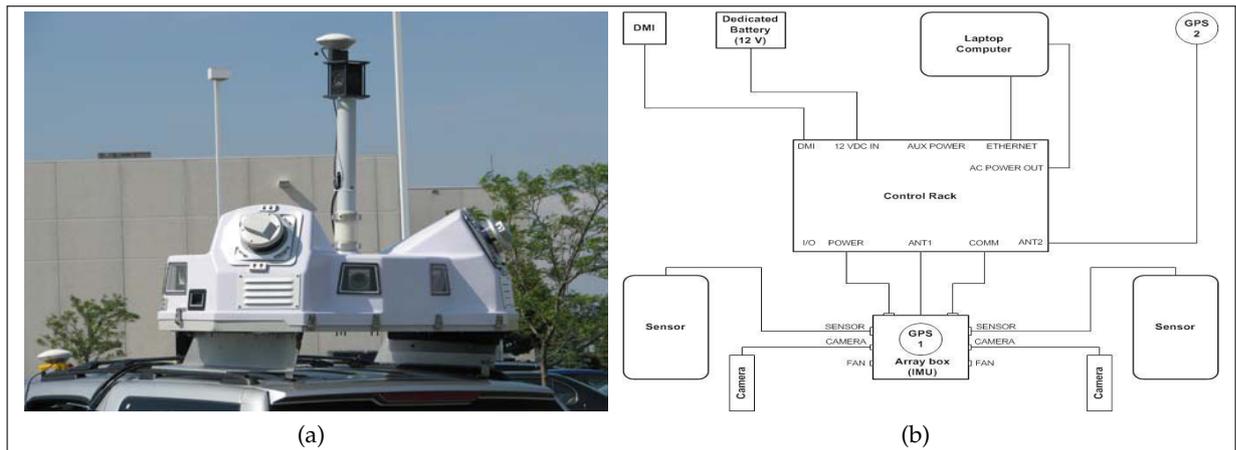


Figure 2.21: Lynx Mobile Mapper along with system configuration (source [Opt14]).

2.3.5.1 Tele Atlas

Tele Atlas is a Dutch-owned mapping company that is based in the town of 's-Hertogenbosch in the Netherlands. Originally an independent company, in July 2008, it was bought by and became a subsidiary of the TomTom company, which is a major Dutch supplier of car navigation systems. In practice, Tele Atlas still supplies digital map data to a wide spectrum of users besides TomTom. These have included Google, which, in October 2009, decided to stop using Tele Atlas map data for the USA i.e. which it now generates from its own mobile mapping activities. However Google still continues to use Tele Atlas map data in other countries. The processing and analysis of the data that has been acquired by its fleets of mobile mapping vehicles is carried out partly in Poland and partly in Noida, a suburb of the Indian capital, New Delhi.

Each of the Tele Atlas vans is equipped with either 4 or 6 digital video cameras in various configurations depending on the area being surveyed. The frame images that are generated by these cameras have a format size of $1,200 \times 960$ pixels and are acquired at the rate of three frames per second when travelling at normal speeds on the roads. The forward-pointing cameras can generate overlapping 3D stereo-images, which allows them to be used as photogrammetric source material. For precise positioning, each van is equipped with a GPS unit operating at 5 Hz , which makes use of Fugro's OmniSTAR wide-area differential GPS service employing satellite broadcast techniques. For use in tunnels and urban canyons, where the GPS signals are either lost or are much restricted, the vans are equipped with a single-axis gyroscope recording at 100 Hz that provides directional (heading) data and an odometer attached to one of the rear wheels of the vehicle. The image and positional data are continuously recorded on the PCs that are mounted in the back of the van. This data is also displayed continuously on monitor display screens for the crew to check their operation. A similar fleet of smaller Toyota vehicles was then developed and brought into service in North America. Besides a set of digital video frame cameras, similar to those being used in Europe, most of these vehicles are equipped with twin 2D laser scanners from SICK that generate a continuous series of range profiles across the surrounding landscape at right angles to the vehicle's direction of travel. Some of these survey vans are also equipped with full-blown inertial measuring units (IMUs) rather than the single-axis gyroscopes mentioned above. Besides the two large fleets of vans that are in operation in Europe and North America, an additional but much smaller number of vans have been deployed in south-eastern Asia; in Taiwan, Singapore and Thailand. More than 50 mobile mapping vans are currently being operated by Tele Atlas world wide as shown in Fig. 2.22.



Figure 2.22: (a) Tele Atlas van. (b) Smaller Toyota vehicles deployed in America (source *Web Tele Atlas*).

2.3.5.2 NAVTEQ

NAVTEQ is a large American mapping company with its headquarters in Chicago. In December 2007, the company was purchased by the Finnish Nokia organisation, which is a major supplier of telecom networks and cell phones on a world-wide scale. Nokia also provides its Ovi Maps product (previously called Nokia Maps), which can be downloaded free by those customers who have purchased the company's smart phones that are equipped with a suitable processor, display screen and operating system.



Figure 2.23: (a) NAVTEQ mobile mapping vehicle is equipped with roof-mounted cameras and a GPS antenna. (b) A close-up picture of the array of multiple cameras housed in a transparent perspex cover on the roof of a NAVTEQ mobile mapping vehicle (source [Nav14]).

However, besides supplying digital map data to Nokia for incorporation in these products, NAVTEQ appears to operate in a fairly independent manner. Like Tele Atlas, NAVTEQ still provides digital map databases for the navigation systems that are being installed in the cars that are being built by several different manufacturers. Besides which, the company also supplies digital map data for use in portable GPS sets and in the Internet-based map applications that are provided by Microsoft (Bing Maps) and Yahoo (Yahoo Live Maps). NAVTEQ has a large map data production centre located in Fargo, North Dakota, that is supplemented and supported by a network of smaller national and regional offices world-wide. Each of the vehicles (comprising cars and SUVs) in the NAVTEQ fleet is equipped with an array of six or eight digital video cameras (Fig. 2.23). These are placed on a specially designed tray that is mounted on the roof

of the vehicle and they are enclosed in a transparent Perspex cover. Each of the six (or eight) cameras acquires its images every 5 metres; in total, they provide a 270 (or 360) degree coverage of the road and its surroundings as seen from the mapping vehicle at each successive position where the images are being acquired. The new vehicles are also equipped with an Applanix POS LV IMU and a TrimbleGPS receiver, which utilizes a differential GPS service. The resulting measured image and positional data are recorded on a powerful PC.

2.3.5.3 Google

Google's Street View is a special feature of the well-known Google Maps and Google Earth services that can be accessed via the Internet. The Street View software gives access to the panoramic images that have been acquired at intervals of 10 to 20 m along the streets of many cities of the world. The numbers and types of mobile mapping cars that have been used to acquire Street View imagery in different countries has varied considerably from country to country. In the USA, Australasia and Japan, the cars were at first equipped with Ladybug multiple cameras. However, since then, these have been replaced by the now standard nine-camera system mounted on a sturdy mast that is itself attached to a roof rack that is fitted to the roof of the car (Fig. 2.24). The mast and camera system can be folded down on to the roof rack when not in use.

As discussed previously, the nine digital cameras from Elphel that make up the Google multiple camera system comprise eight that, in total, provide a 360 degree panorama in the horizontal plane, while the ninth camera points vertically upwards to record the undersides of bridges and overpasses and the top surfaces of tunnels. Each car is equipped with a combined DGPS/IMU system that has been supplied by Topcon, together with a wheel-mounted odometer that, in conjunction with the IMU, can help establish position wherever GPS coverage is poor or has been lost in tunnels or within high-rise urban areas.



Figure 2.24: Google mobile mapping fleet consisting of mapping vehicles, trikes, snow mobiles and portable backpacks vehicle (source [Goo14]).

The Google Street View cars also feature a pair of SICK LMS 291 laser scanners that continuously measure a series of range or elevation profiles on either side of the mapping vehicle. A third SICK scanner measures the road surface in front of the vehicle. Besides the car-based mobile mapping systems, Google has also introduced a number of pedal-powered tricycles (trikes), snow mobile and portable back packs that are equipped with a similar set of cameras,

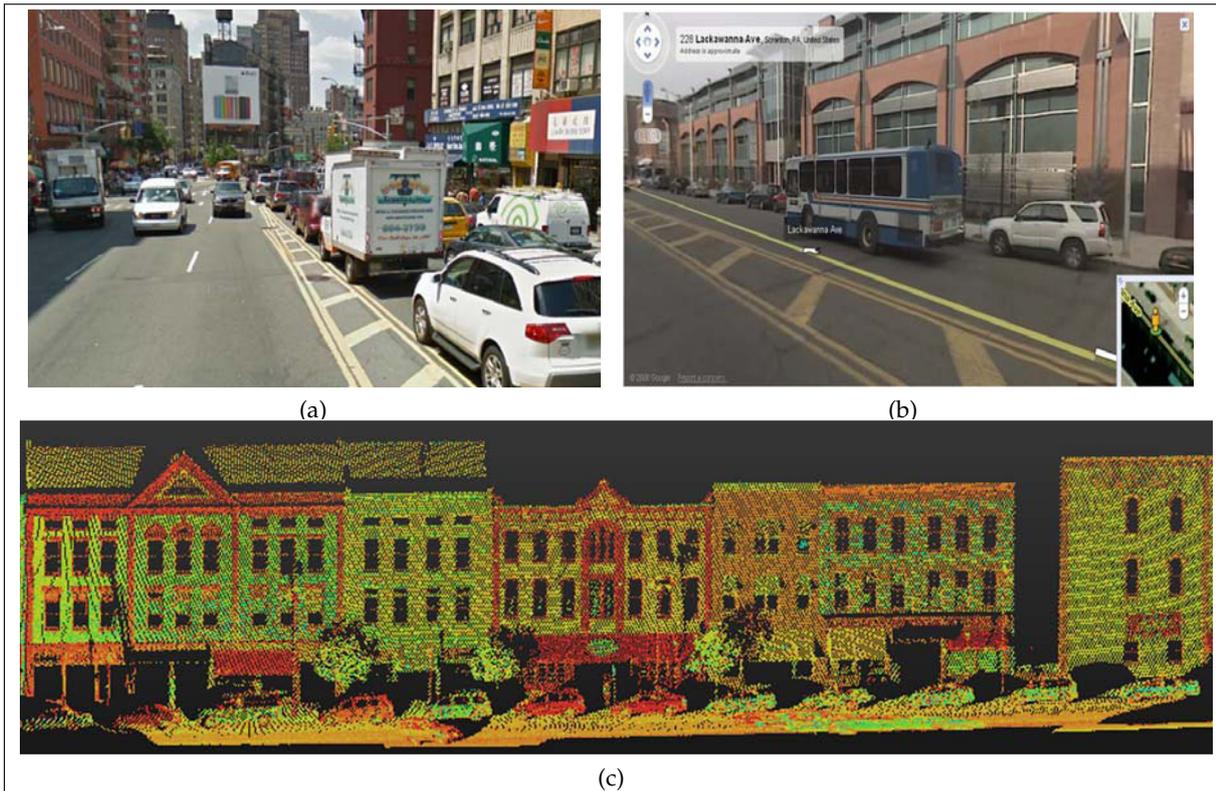


Figure 2.25: (a) & (b) show different occluding objects in the urban scene in image data (source *Web Google Maps*). (c) shows different occluding objects in the 3D LiDAR point clouds of the urban environment acquired from Vis Centre’s Viztruck.

laser scanners and positioning equipment (Fig. 2.24). These are being used for data collection in areas such as pedestrian precincts and public parks and along cycle tracks where cars cannot be operated or mountain ranges, etc.

2.4 Limitations and Challenges

Although the use of terrestrial MMS for urban cartography has several advantages such as fast acquisition speed, highly accurate and dense data and the ability of map large areas easily (even multiple times), but there are also some limitations. In urban environments, the quality of the data obtained from these hybrid terrestrial vehicles is widely hampered due to the presence of temporarily static and dynamic objects (pedestrians, cars, etc.) in the scene. As a result, the occlusion of some features remains a problem. Moving objects or certain temporarily stationed objects (parked cars, traffic, pedestrian, etc.) present in the area hide certain features of the urban landscape (parts of buildings, road sides, etc.) as shown in Fig. 2.25.

Similarly, the fast acquisition of such a large amount of data (modern acquisition platforms facilitate data collection rates of over two billion points per hour) not only requires large storage space but also makes it practically impossible to process this data in real time for any higher level applications.

Also, due to the fact that the vehicle carrying the mapping system moves mainly on streets and roads, there are sometimes accessibility issues and only limited viewing angles are available compared to static scanners which can be moved around a certain building or area to acquire data from multiple views.

In static scans, points are collected radially outward with the point density dropping off sharply with distance. In MMS, the points only drop in density perpendicular to the line of

travel whereas along the line of travel the density varies with the speed of the vehicle as shown in Fig. 2.26(b). Moreover, the density also depends upon the spatial orientation of surfaces and their distance for each acquired frame (heterogeneous density). For example, in the ground plane, the points acquired close to the vehicle have a high density whereas the far-off points have a more low density as shown in Fig. 2.26(a).

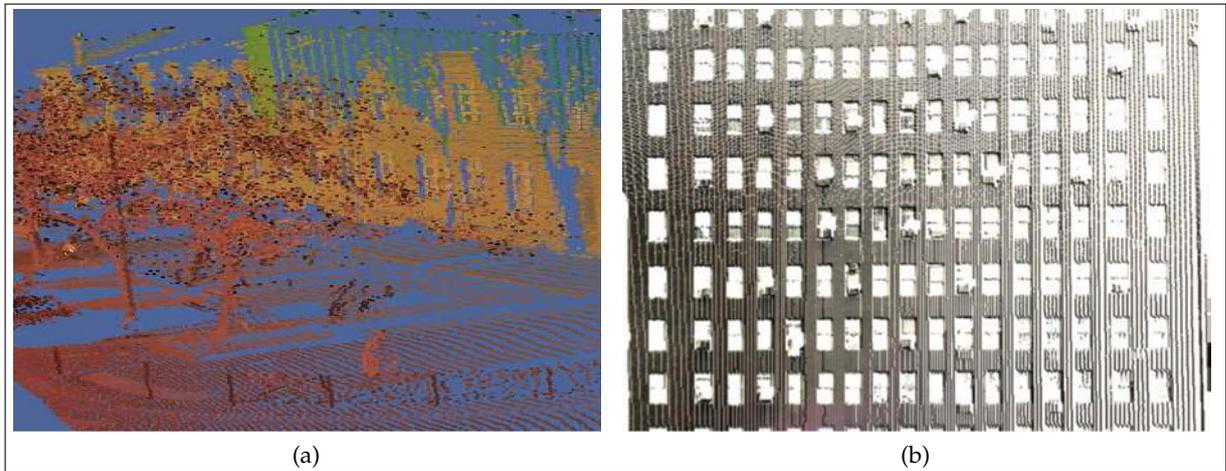


Figure 2.26: Density variation in 3D MMS data. (a) & (b) show the point density dropping perpendicular to the line of travel of the data acquiring vehicle. In (b) we can clearly see that the point density decreases from left to right due to the varying speed of the vehicle.

In case of image data not only the effects of illumination and weather conditions play an important role but also different post processing techniques such as automatic face detection and blurring, etc. is also required for privacy.

Another common problem currently faced is the effective update of the urban cartography as shown in Fig. 2.27 and Fig. 2.28. In order to update these maps or resulting 3D models, the same areas are mapped repeatedly and regularly for example for Google Earth after every two weeks and Google Street View up to few months in several areas (source Google inc.).

Apart from scheduled mapping of different areas, most mapping services also rely on user feedback to manually indicate changed areas where new mapping is required. This also helps optimize and prioritize mapping schedules. However, as in common practice, no automatic change detection methods are used that could efficiently identify and pinpoint only the changed portion, a lot of unnecessary computational power and time are wasted. Similarly, inefficient updating methodology also results in just piling of large amount of data over time, for example Google had already stocked over 20 petabytes of data for its different mapping services. Further use of this redundant data for different higher level applications still remains a challenging topic of research.

Another difficulty that arises when attempting to detect effective changes occurring in the urban environment is to differentiate the changes caused by the presence of transient objects i.e. frequently pedestrians and vehicles in the scene with the actual structural changes occurring in the cartography due to addition, removal or modification of certain parts of the environment.

Sometimes, the scanning system itself offers several issues to consider when dealing with the detection of changes. When employing LiDAR scanners, for instance, problems such as no-reflectance regions, range limits, and noisy ranges need to be resolved. Regions of no-reflectance refer to areas towards which pulses have been transmitted, but due to absorption or specularly, not enough energy (if any) is returned to trigger a range measurement (some common examples include objects like, windows, and low reflectance surfaces). At times when data from the other scan exists in these "hole" regions they will be considered as changes while not being so. Similarly with ranging limits, the different scanning positions will leave areas seen in one scan uncovered by the other. This relatively trivial fact suggest that the lack of information cannot be attributed



Figure 2.27: Problem of not-so-regular updating can result in incorrect urban features in the visualization and cartography of the environment. (a) shows the original Street View screenshot of a house in Dauphine Street, New Orleans taken at 15-07-2012 while the actual photo taken on the same date is presented in (c). (b) shows the original Street View screenshot of K-doe's mother in law lounge in New Orleans taken on 21-06-2012 while the actual photo of the place taken on the same date is presented in (d).

only to actual changes (or geometric constraints that the scene imposes) in the region. Therefore, the consideration of the range limits should be handled appropriately. Also, important is the noise which usually accompanies laser data, especially around object edges and corners. Unless treated properly, noise can also be interpreted as changes in the scene.

Another important issue related to multiple scanning and mapping of the same areas for updating is the misalignment of the acquired 3D points in case of LiDAR data. Although, usually the acquired 3D points are directly geo-referenced as explained in Section 2.3.2, there are still some misalignment errors when merging these 3D point clouds due to different inherent errors of the acquisition and positioning sensors as shown in Fig. 2.29. Different applications and precise analysis necessitate a more accurate/fine registration of these point clouds.



Figure 2.28: Problem of different updating schedules. The figure clearly shows that both the Google Map and Street view were not updated at the same time. As the Google map suggests to the user to drive through Wood house lane, Manchester, while the Street View image shows it is blocked (year 2012).

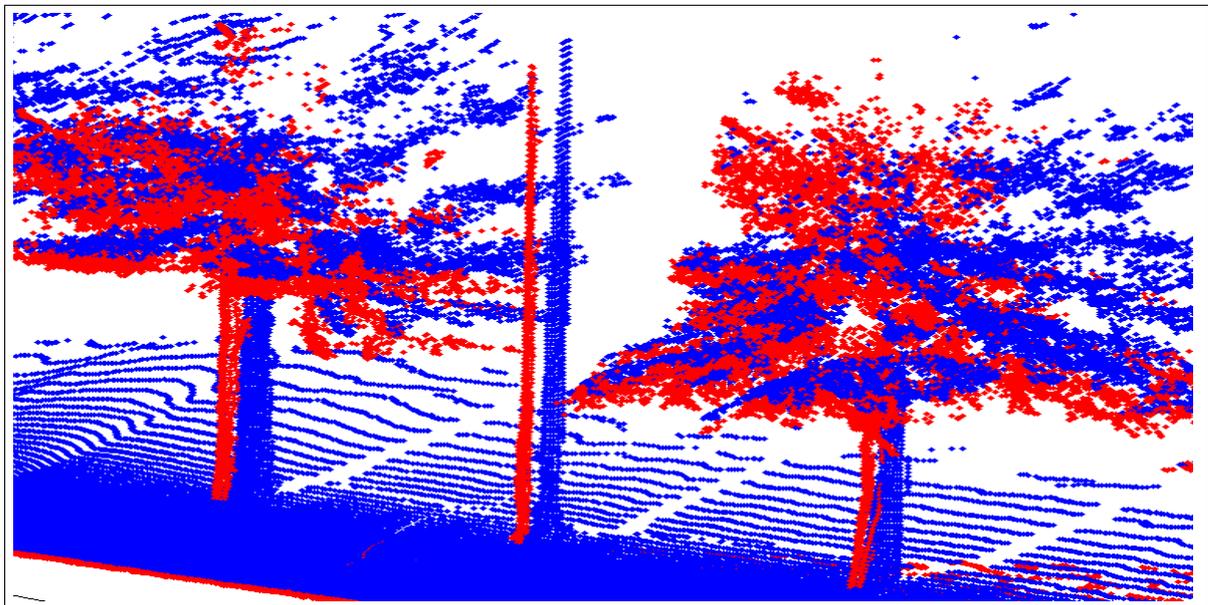


Figure 2.29: Misalignment errors when merging two 3D point clouds (in red and blue respectively) of the same urban scene (source [GS10]).

In this thesis, we effectively address these problems and propose a new method of 3D urban cartography incorporating recognition and temporal integration. The resulting 3D cartography contains only the exact, accurate and well-updated, permanent features of the urban environment.

2.5 Proposed Methodology

As we discussed in the last section that in urban environments the quality of the data obtained from Mobile Mapping Systems (MMS) are widely hampered due to the presence of temporarily static and dynamic objects (pedestrians, cars, etc.) in the scene as shown in Fig. 2.25. So the first step for 3D urban cartography is to obtain the permanent part of the urban environment. So, in order to obtain the permanent part of the urban environment, we first classify the urban environment into permanent and temporary classes (Chapter 3). The objects classified as temporary are then removed from the 3D point cloud leaving behind a perforated 3D point cloud of the urban environment. These perforations along with other imperfections are then analyzed and progressively removed by incremental updating exploiting the concept of multiple passages (Chapter 4). We also show that the proposed method of temporal integration also helps in improved semantic analysis of the urban environment, specially building façades (Chapter 5).

These methods ensure that the resulting 3D cartography contains only the exact, accurate and well-updated, permanent features of the urban environment. An overview of the workflow is provided in Fig. 2.30.

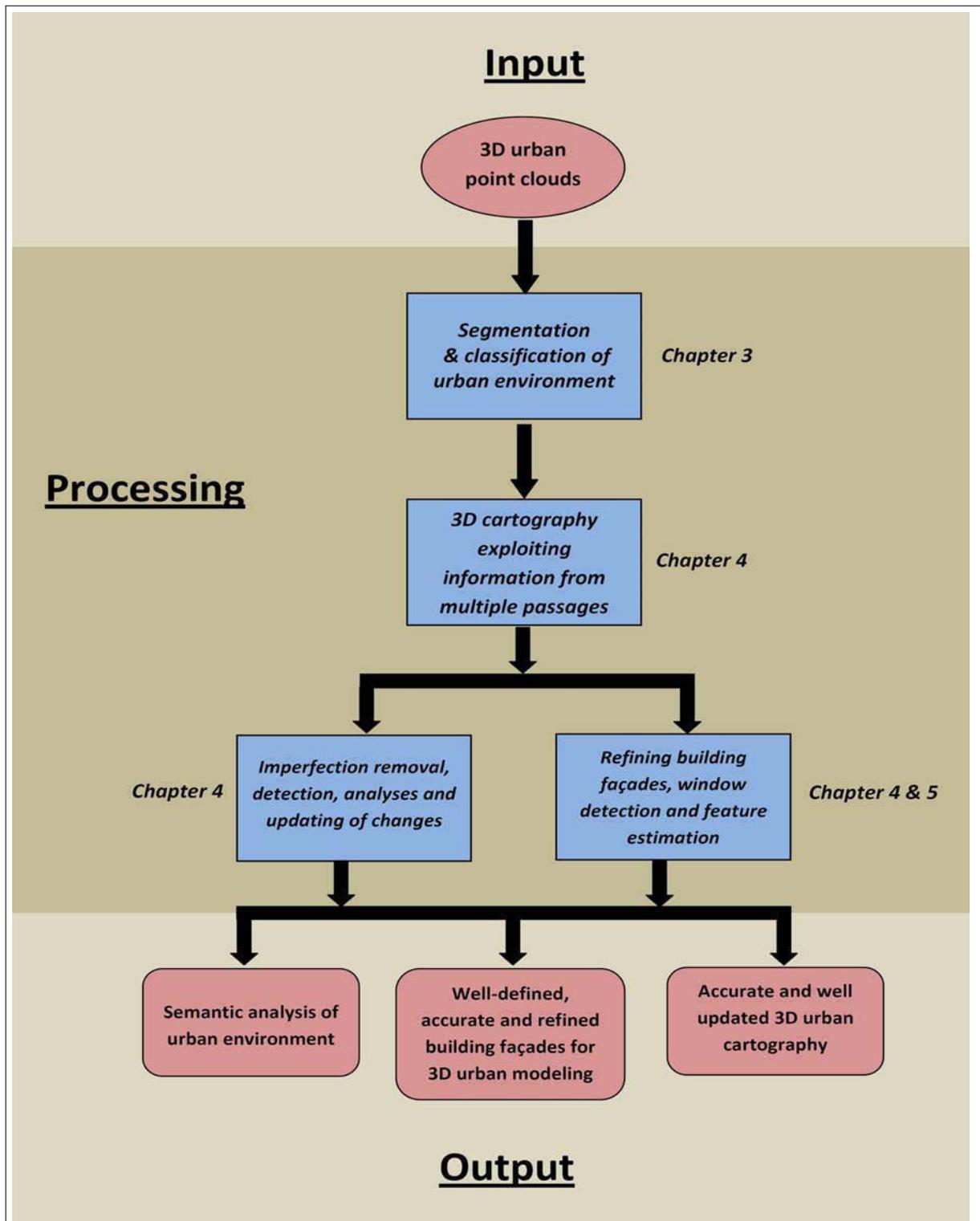


Figure 2.30: An overview of the workflow.

SEGMENTATION AND CLASSIFICATION OF 3D URBAN POINT CLOUDS

Segmentation and classification of urban range data into different object classes offer several challenges due to certain properties of the data such as density variation, inconsistencies due to missing data and the large data size which require heavy computation and large memory. In this chapter, after shedding light on the existing state of the art techniques for segmentation and classification of urban environment, a method to classify urban scenes based on a super-voxel segmentation of sparse 3D data obtained from LiDAR sensors is presented. The 3D point cloud is first segmented into voxels which are then charac-

terized by several attributes transforming them into super-voxels. These are joined together by using a link-chain method rather than the usual region growing algorithm to create objects. These objects are then classified using geometrical models and local descriptors. In order to evaluate the results, a new metric is presented which combines both segmentation and classification results simultaneously. The effects of voxel size and incorporation of RGB color and laser reflectance intensity on the classification results are also discussed. The method is evaluated on standard datasets using different metrics to demonstrate its efficacy.

3.1 Introduction

The automatic segmentation and classification of 3D urban data have gained widespread interest and importance in the scientific community due to the increasing demand of urban landscape analysis and cartography for different popular applications, coupled with the advances in 3D data acquisition technology. The automatic extraction (or partially supervised) of important urban scene structures such as roads, vegetation, lamp posts, and buildings from 3D data has been found to be an attractive approach to urban scene analysis because it can tremendously reduce the resources required for analyzing the data for subsequent use in 3D city modeling and other algorithms.

3.1.1 Segmentation of 3D Data

In order to fully exploit 3D point clouds, for scene understanding and object classification, effective segmentation has proved to be a necessary and critical pre-processing step in a number of autonomous perception tasks. For instance, Malisiewicz and Efros [ME07] show that prior segmentation improves classification in computer vision applications.

3.1.1.1 Specialized Features and Surface Discontinuities

Earlier works, including [RDHV06] and [SV03], employed the use of small sets of specialized features, such as local point density or height from the ground, to discriminate only few object categories in outdoor scenes, or to separate foreground from background. In literature survey, we also find some segmentation methods based on surface discontinuities as described by Moosmann, Pink, and Stiller [MPS09] who used surface convexity in a terrain mesh as a separator between objects as shown in Fig. 3.1.

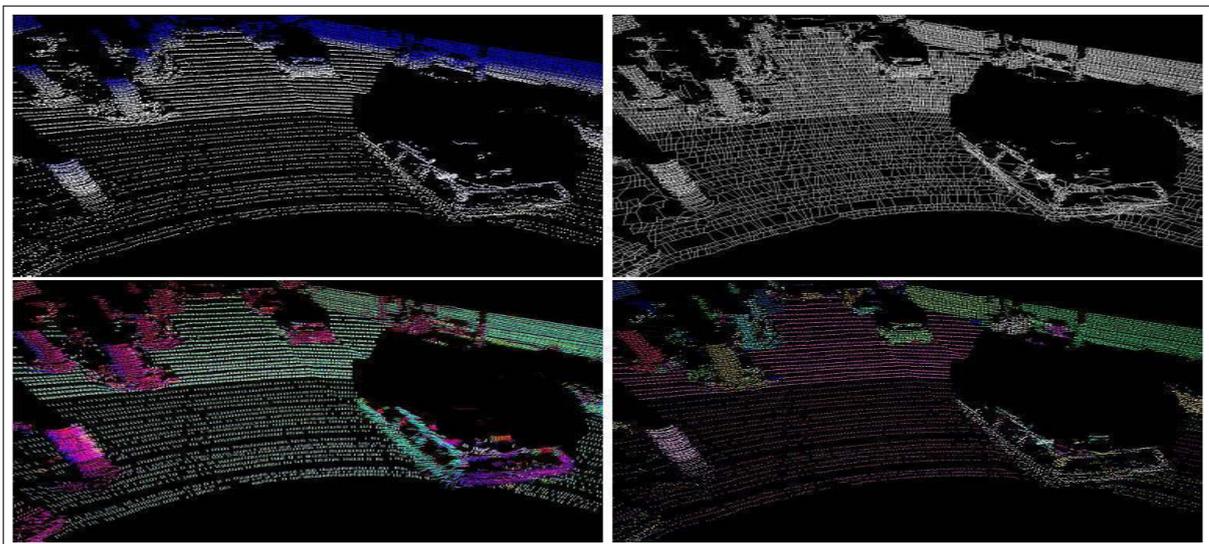


Figure 3.1: Illustration of the method: (top-left) 3D point cloud, blueness decodes height above ground. Street is in vertical direction with parked cars and a wall on the right and tree trunks on the left. (top-right) Neighborhood Graph. (bottom-left) Normal vectors, direction is indicated both by color and a small line. (bottom-right) Segmentation result (source [MPS09]).

3.1.1.2 Graph Clustering

Lately, segmentation has been commonly formulated as graph clustering. Instances of such approaches are Graph-Cuts including Normalized-Cuts and Min-Cuts. Golovinskiy and

Funkhouser [GF09] extended Graph-Cuts segmentation to 3D point clouds by using k-Nearest Neighbors (k-NN) to build a 3D graph. In this work, edge weights based on exponential decay in length were used. But the limitation of this method is that it requires prior knowledge of the location of the objects to be segmented. An illustration of the method is presented in Fig. 3.2.

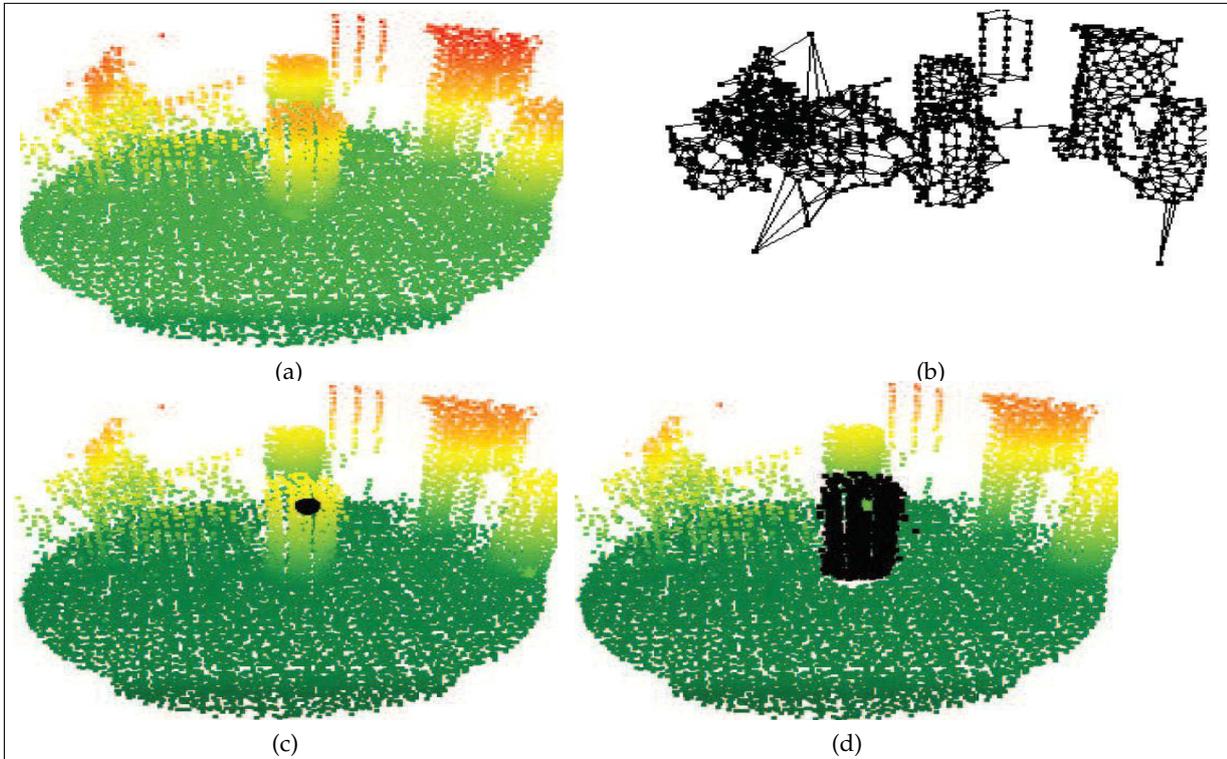


Figure 3.2: (a) The method takes as input a point cloud near an object location (in this case, a short post). (b) A k-nearest neighbors graph is constructed. (c) A foreground point is chosen as a hard constraint (in the interactive mode, the user chooses hard foreground and background constraints). (d) The resulting segmentation is created via a min-cut (source [GF09]).

Another segmentation algorithm for natural images, introduced by Felzenszwalb and Huttenlocher [FH04] (FH algorithm), has gained popularity for several robotic applications due to its efficiency. This algorithm makes simple greedy decisions, and yet produces segmentations that satisfy the global properties by using a particular region comparison function which measures the evidence for a boundary between pairs of regions. Zhu *et al.* [ZZL+10] presented a method in which a 3D graph is built with k-NN while assuming the ground to be flat for removal during pre-processing. 3D partitioning is then obtained with the FH algorithm. We also used the same assumption in our proposed method.

Triebel, Shin, and Siegwart [TSS10] modified the FH algorithm for range images to propose an unsupervised probabilistic segmentation technique. In this approach, the 3D data is first over-segmented during pre-processing. Schoenberg, Nathan, and Campbell [SNC10] have applied the FH algorithm to colored 3D data obtained from a co-registered camera laser pair. The edge weights are computed as a weighted combination of Euclidean distances, pixel intensity differences and angles between surface normals estimated at each 3D point. The FH algorithm is then run on the image graph to provide the final 3D partitioning. The evaluation of the algorithm is done on road segments only.

Strom, Richardson, and Olson [SRO10] proposed a similar approach but modified the FH algorithm to incorporate angle differences between surface normals in addition to the differences in color values. The improvement in results are shown in Fig. 3.3. However, the segmentation evaluation was done visually without ground truth data. Our approach differs from the above

mentioned methods as, instead of using the properties of each point for segmentation resulting in over segmentation, we have grouped the 3D points based on Euclidian distance into voxels and then assigned normalized properties to these voxels transforming them into super-voxels. This not only prevents over segmentation but in fact reduces the dataset by many fold thus reducing post-processing time.

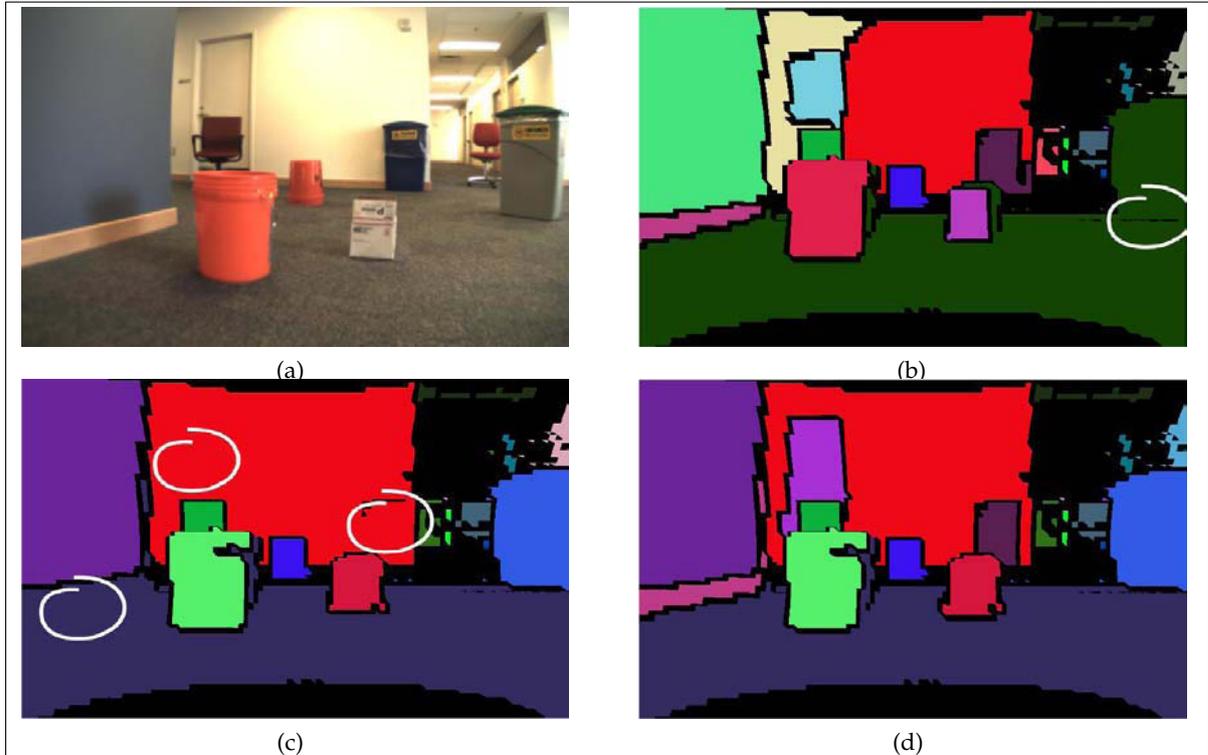


Figure 3.3: (a) The camera image for the scene of which the color coded 3D points were acquired. (b) Graph segments with only color difference as a cluster criterion. (c) Graph segments with only surface normals as a cluster criterion. (d) Graph segments with both color difference and surface normals as a cluster criterion. The white rings mark the errors observed in each case (source [SRO10]).

3.1.1.3 Geometrical Primitives

A spanning tree approach to the segmentation of 3D point clouds was proposed by Pauling, Bosse, and Zlot [PBZ09]. Graph nodes represent Gaussian ellipsoids as geometric primitives. These ellipsoids are then merged using a tree growing algorithm. The resulting segmentation (as shown in Fig. 3.4) is similar to a super-voxel type of partitioning with voxels of ellipsoidal shapes and various sizes. Unlike this method, our approach uses cuboids of different shapes and sizes as geometric primitives and a link-chain method to group them together.

3.1.1.4 Markov Random Fields

In the literature review, we also find some techniques, such as [ATC+05] and [LS07], that segment and label 3D points by employing Markov Random Fields to model their relationship in the local vicinity. These techniques proved to outperform classifiers based only on local features but at a cost of computational time. Different methods such as [MVH09] have been introduced to increase the efficiency.

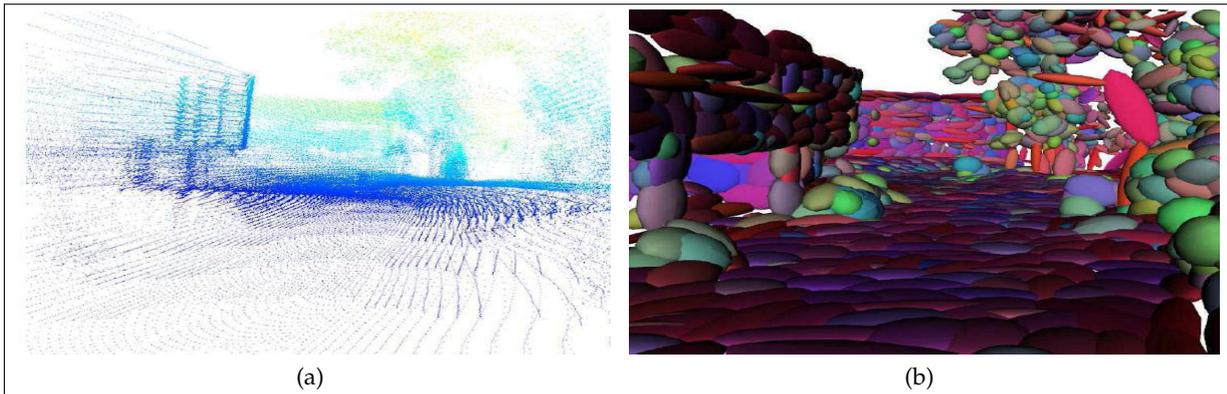


Figure 3.4: (a) The 3D points of an urban scene. (b) Segmentation results with ellipsoids as geometric primitives (source [PBZ09]).

3.1.1.5 Mathematical Morphology

A method for segmenting 3D urban point cloud based on mathematical morphology is presented by Serna and Marcotegui [SM14]. In this method, first, input point cloud is mapped to a range image. Second, the image is interpolated in order to avoid connectivity problems and the k -flat zones algorithm is used to segment the ground (road and sidewalk). Third, façades and objects are extracted using morphological transformations. The methods rely on façades being the highest vertical structures in the scene and objects are represented as bumps on the ground on the range image as shown in Fig. 3.5. Although the method is relatively fast, it fails to correctly segment out closely positioned or overlapping objects. Compared to this method, our proposed method is more efficient in segmenting out such objects as along with other information, the values of reflectance intensity and RGB color (if available) are also incorporated.

3.1.2 Classification of 3D Data

In the past, research related to 3D urban scene classification and analysis had been mostly performed using either 3D data collected by airborne LiDAR for extracting bare-earth and building structures [LOL09] [VKG05] or 3D data collected from static terrestrial laser scanners for extraction of building features such as walls and windows [PV09]. Recently, classification of urban environment using data obtained from mobile terrestrial platforms (such as [HS10]) has gained much interest in the scientific community due to the ever-increasing demand of realistic 3D models for different popular applications coupled with the recent advancements in the 3D data acquisition technology.

3.1.2.1 Discriminate Models and Model Fitting

In [LS08] a method of multi-scale Conditional Random Fields is proposed to classify 3D outdoor terrestrial laser scanned data by introducing regional edge potentials in addition to the local edge and node potentials in the multi-scale Conditional Random Fields. This is followed by fitting plane patches onto the labeled objects such as building terrain and floor data using the RANSAC algorithm as a post-processing step to geometrically model the scene as shown in Fig. 3.6. In [LKM+10] the authors extracted roads and objects just around the roads like road signs. They used a least squares fit plane and RANSAC method to first extract a plane from the points followed by a Kalman filter to extract roads in an urban environment. Douillard, Brooks, and Ramos [DBR09] presented a method in which 3D points are projected onto the image to find regions of interest for classification. Object classification is implemented using a rule based system to combine binary deterministic and probabilistic features. Figure 3.7 shows some of the results.

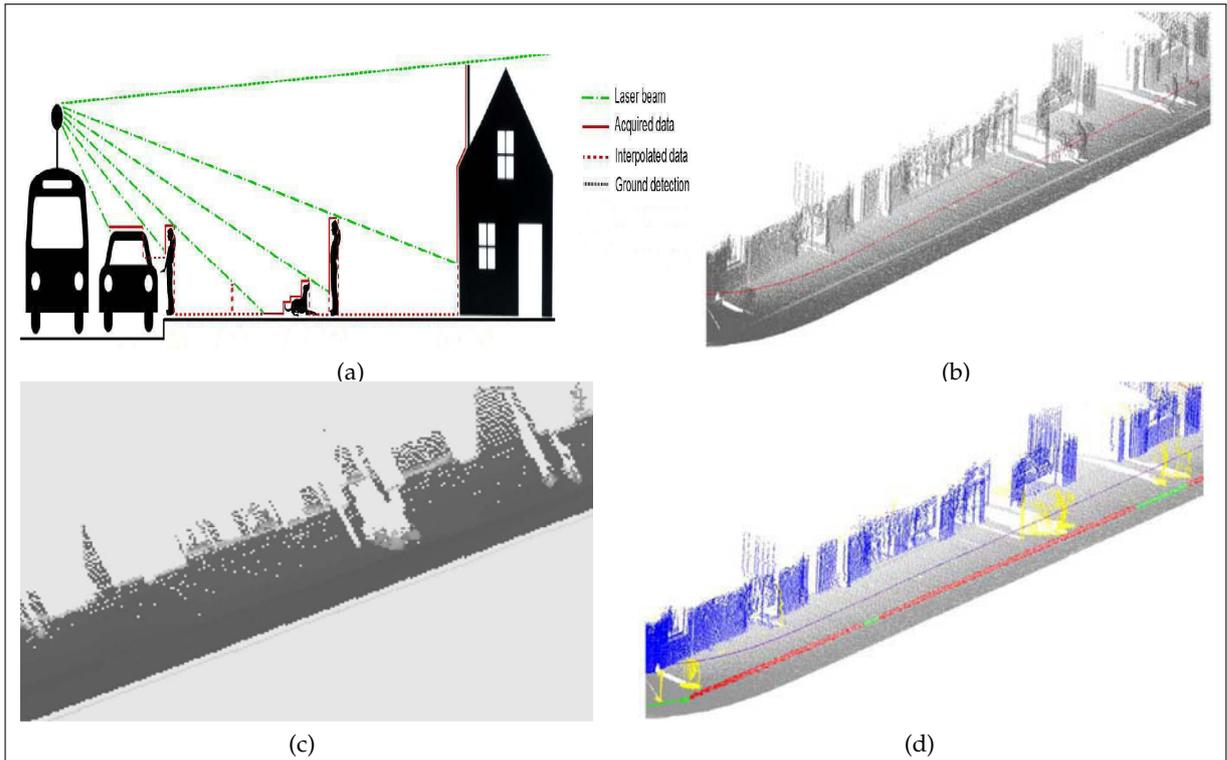


Figure 3.5: (a) Segmentation of 3D point clouds based on morphological modeling. Objects are segmented out as bumps on the ground. (b) Input point cloud. (c) Range image. (d) Segmentation results (source [SM14]).

3.1.2.2 Features based

The use of different characteristic features for discriminating and classifying objects is an effective technique. A method of classification based on global features is presented in [HHB+10] in which a single global spin image for every object is used to detect cars in the scene while in [RBT+10] a Fast Point Feature Histogram (FPFH) local feature is modified into global feature for simultaneous object identification and view-point detection. Classification using local features and descriptors such as Spin Image [Joh97], Spherical Harmonic Descriptors [KFR03], Heat Kernel Signatures [SOG09], Shape Distributions [OFC+02], 3D SURF feature [KPG10] is also found in the literature survey. In [PMD08], the authors use both local and global features in a combination of bottom-up and top-down processes. In this approach, spin images are used as local descriptors to classify cars in the scene in the bottom-up stage while Extended Gaussian Images are used as global descriptors for verification in the top-down stage. The work shows that the combination local and global descriptors provides a good trade-off between efficiency and accuracy. There is also a third type of classification based on Bag Of Features (BOF) as discussed in [LZQ06].

In our method, we use geometrical models based on local features and descriptors to successfully classify different segmented objects represented by groups of voxels in the urban scene. Ground is assumed to be flat and is used as an object separator. Our segmentation technique is discussed in Section 3.2. Section 3.3 deals with the classification of these segmented objects. In Section 3.4, a new evaluation metric is introduced to evaluate both segmentation and classification together while, in Section 3.5, we present the results of our work. Finally, we conclude in Section 3.7.

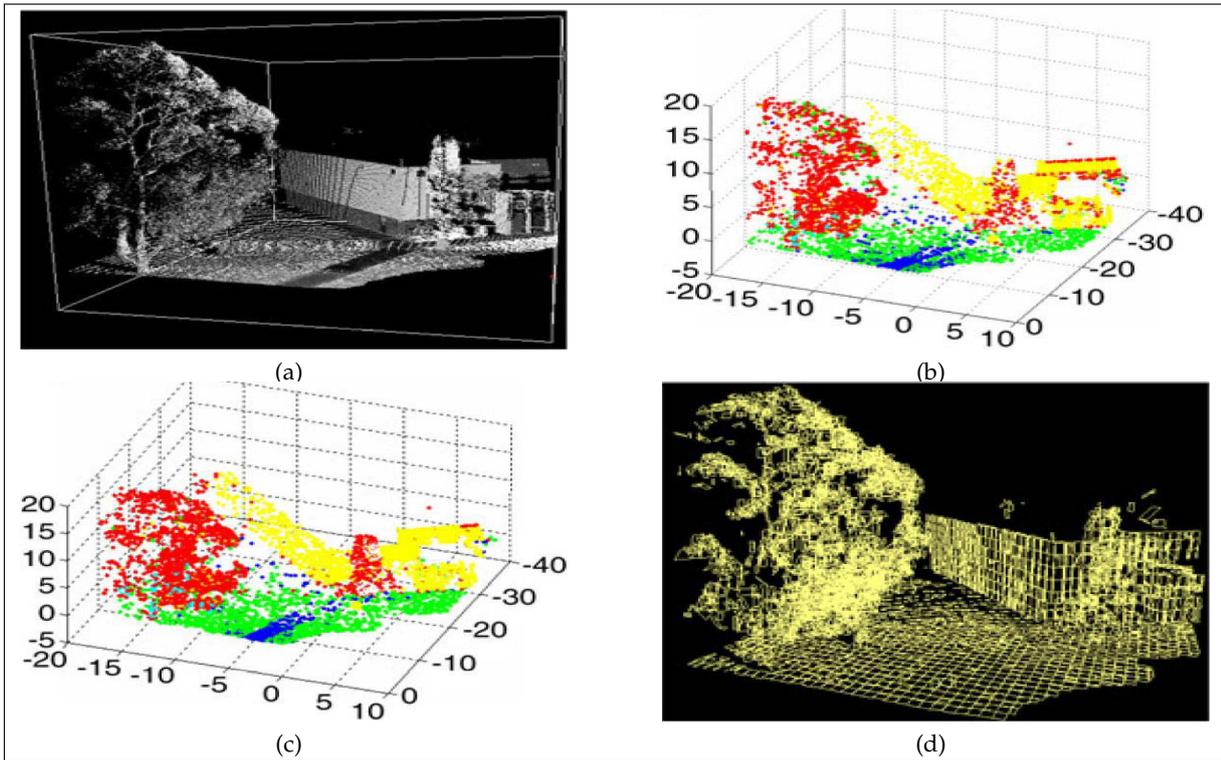


Figure 3.6: (a) 3D points of an urban scene shown in laser direct intensity. (b) Classification results with CRFs. (c) Classification results with mCRFs (Man-made objects in yellow), Vegetation in red, Tree trunks in blue, Terrain in green and Pathways in blue). (d) RANSAC fitted plane patches (source [LS08]).

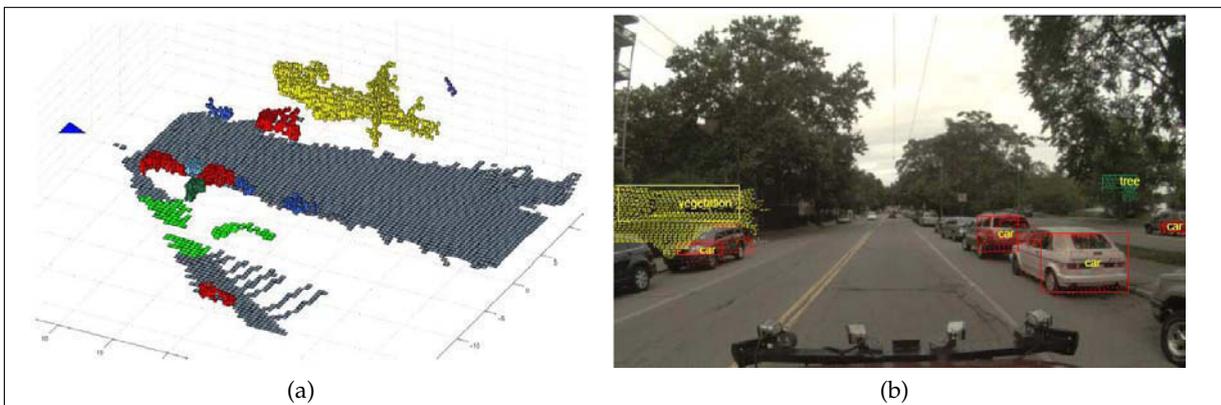


Figure 3.7: (a) 3D view of the inferred class labels. The blue triangle indicates the vehicle's position. (b) displays the inferred labels/classes as well as the ROIs and the projected laser returns. The color of each ROI matches the color of the associated object in the 3D plot (source [DBR09]).

3.2 Voxel-based Segmentation

The proposed voxel based segmentation method consists of three main parts which are the voxelisation of data, the transformation of voxels into super-voxels and the clustering by link-chain method.

3.2.1 Voxelisation of Data

When dealing with large 3D datasets, the computational cost of processing all the individual points is very high, making it impractical for real time applications. It is therefore sought to reduce these points by grouping together or removing redundant or un-useful points. Similarly, in our work the individual 3D points are clustered together to form a higher level representation or voxel as shown in Fig. 3.8.

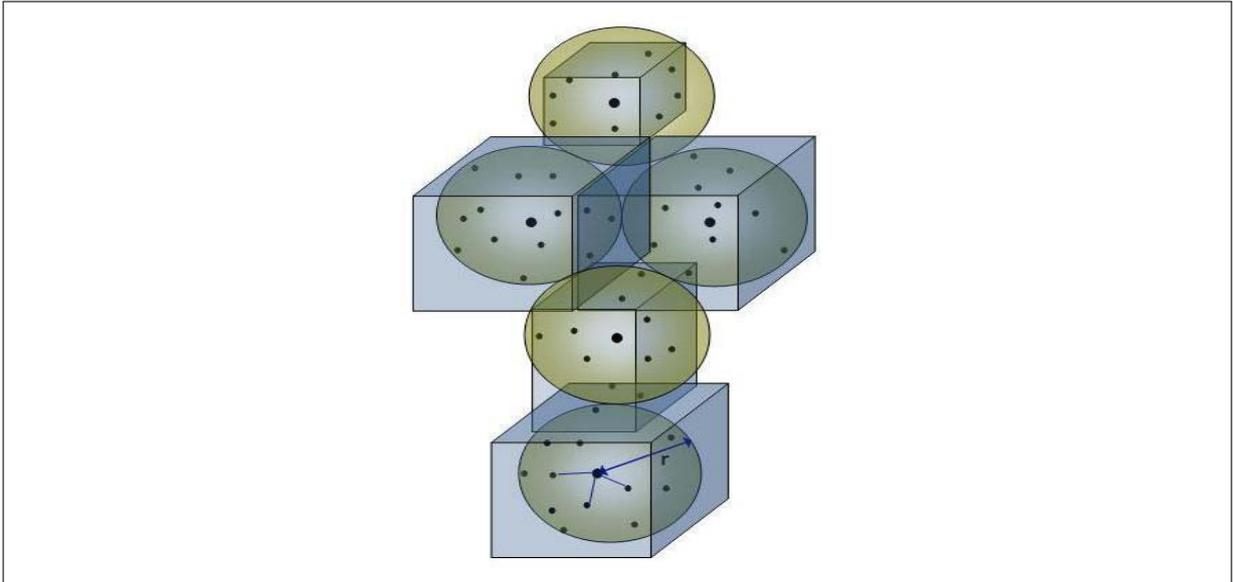


Figure 3.8: A number of points is grouped together to form cubical voxels of maximum size $2r$. The actual voxel sizes vary according to the maximum and minimum values of the neighboring points found along each axis to ensure the profile of the structure.

For p data points, a number of s voxels, where $s \ll p$, is computed based on r -NN (r -Nearest Neighbors), where r is the radius of ellipsoid. The maximum size of the voxel $2r$ depends upon the density of the 3D point cloud (the choice of this maximum voxel size is discussed in Section 3.5.5). In order to create the voxels, first a 3D point is selected as centre and using an r -NN with a fixed diameter (equal to maximum voxel size), all the 3D points in the vicinity are selected. All these 3D points now belong to this first voxel. Then, based on the maximum and minimum values of the 3D points contained in this voxel, the actual voxel size is determined. The same step is then repeated for other 3D points which are not part of the earlier voxel until all 3D points are considered (see Algorithm 1). In [LS08] color values are also added in this step but it is observed that for relatively smaller voxel sizes, the variation in properties such as color is not profound and just increases computational cost. For these reasons, we have only used distance as a parameter in this step. The other properties are used in the next step of clustering the voxels to form objects. Also we have ensured that each 3D point which belongs to a voxel is not considered for further voxelisation. This not only prevents over segmentation but also reduces processing time.

For the voxels we use a cuboid because of its symmetry, as it avoids fitting problems while grouping and also minimizes the effect of voxel shape during feature extraction. Although the maximum voxel size is predefined, the actual voxel sizes vary according to the maximum and minimum values of the neighboring points found along each axis to ensure the profile of the structure.

3.2.2 Transformation of Voxels into Super-Voxels

A voxel is transformed into a super-voxel when properties based on its constituting points are assigned to it. These properties include:

- ▷ $V_{X,Y,Z}$: geometrical center of the voxel;
- ▷ $V_{R,G,B}$: mean R, G, & B value of constituting 3D points;
- ▷ $Var(R, G, B)$: maximum of the variance of R, G & B values;
- ▷ V_I : mean laser reflectance intensity value of constituting 3D points;
- ▷ $Var(I)$: variance of laser reflectance intensity values;
- ▷ $s_{X,Y,Z}$ is the voxel size along each axis X, Y & Z ;
- ▷ Surface normals: A surface normal is calculated using PCA (Principal Component Analysis). The PCA method has been proved to perform better than the area averaging method [KAW+09] to estimate the surface normal. Given a point cloud dataset $\mathcal{D} = \{x_i\}_{i=1}^n$, the PCA surface normal approximation for a given data point $p \in \mathcal{D}$ is typically computed by first determining the k -Nearest Neighbors of the points contained inside the voxel, $x_k \in \mathcal{D}$, of p . Given the K neighbors, the approximate surface normal is then the eigenvector associated with the smallest eigenvalue of the symmetric positive semi-definite matrix

$$\mathbf{P} = \sum_{k=1}^K (x_k - \bar{p})^T (x_k - \bar{p}) \quad (3.1)$$

where \bar{p} is the local data centroid: $\bar{p} = \frac{1}{K} \sum_{j=1}^K x_j$.

The estimated surface normal is ambiguous in terms of sign (specially if the location of the sensor is unknown); to account for this ambiguity it is homogenized using the dot product. Yet for us the sign of the normal vector is not important as we are more interested in the orientation. A surface normal is estimated for all the points belonging to a voxel and is then associated with that particular voxel which then also defines its 3D orientation.

With the assignment of all these properties, a voxel is transformed into a super-voxel. All these properties would then be used in grouping these super-voxels (from now onwards referred to as s -voxels) into objects and then during the classification of these objects.

Instead of using thousands of points in the dataset, the advantage of this approach is that we can now use the reduced number of s -voxels to obtain similar results for classification and other algorithms.

3.2.3 Clustering by Link-Chain Method

When the 3D data is converted into s -voxels, the next step is to group these s -voxels to segment into distinct objects. Usually for such tasks, a region growing algorithm [VS05] is used in which the properties of the whole growing region may influence the boundary or edge conditions. This may sometimes lead to erroneous segmentation. Also common in such type of methods is a node based approach [MPS09] in which at every node, boundary conditions have to be checked in all 5 different possible directions. In our work, we have proposed a link-chain method instead to group these s -voxels together into segmented objects.

In this method each s -voxel is considered as a link of a chain. Unlike the classical region growing algorithm, where a region is progressively grown from a seed (carefully selected start point), in the proposed method any s -voxel can be taken as a principal link and all secondary links attached to this principal link are found. The same is repeated for all s -voxels until all s -voxels

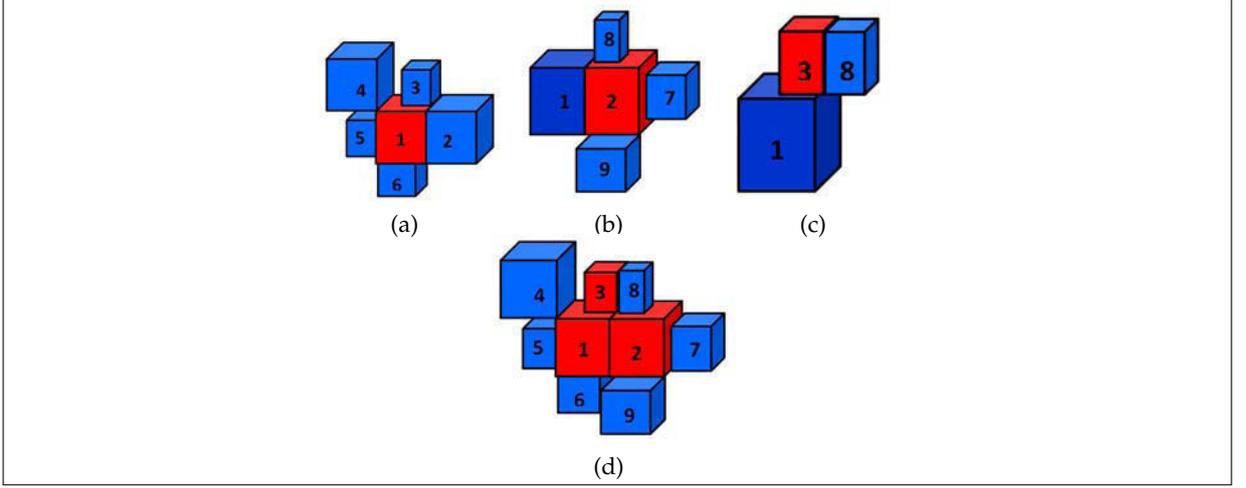


Figure 3.9: Clustering of s -voxels using a link-chain method is demonstrated. (a) shows s -voxel 1 taken as principal link in red and all secondary links attached to it in blue. (b) and (c) show the same for s -voxel 2 and 3 taken as principal links. (d) shows the linking of principal links (s -voxels 1, 2 & 3) to form a chain removing redundant secondary links.

are taken into account (see Algorithm 1 for details). Thus there is no need of a specific start point, no preference for choice of principal link nor any directional constraint, etc. In the final step, all the principal links are linked together to form a continuous chain removing redundant secondary links in the process as shown in Fig. 3.9. These clusters of s -voxels represent the segmented objects.

Let \mathbf{V}_P be a principal link and \mathbf{V}_n be the n^{th} secondary link. Each \mathbf{V}_n is linked to \mathbf{V}_P if and only if the following three conditions are fulfilled:

$$|\mathbf{V}_{P_{X,Y,Z}} - \mathbf{V}_{n_{X,Y,Z}}| \leq (w_D + c_D) \quad (3.2)$$

$$|\mathbf{V}_{P_{R,G,B}} - \mathbf{V}_{n_{R,G,B}}| \leq 3\sqrt{w_C} \quad (3.3)$$

$$|\mathbf{V}_{P_I} - \mathbf{V}_{n_I}| \leq 3\sqrt{w_I} \quad (3.4)$$

where, for the principal and secondary link s -voxels respectively:

- ▷ $\mathbf{V}_{P_{X,Y,Z}}, \mathbf{V}_{n_{X,Y,Z}}$ are the geometrical centers;
- ▷ $\mathbf{V}_{P_{R,G,B}}, \mathbf{V}_{n_{R,G,B}}$ are the mean R, G & B values;
- ▷ $\mathbf{V}_{P_I}, \mathbf{V}_{n_I}$ are the mean laser reflectance intensity values;
- ▷ w_C is the color weight equal to the maximum value of the two variances $Var(R, G, B)$, i.e. $\max(\mathbf{V}_{P_{Var(R,G,B)}}, \mathbf{V}_{n_{Var(R,G,B)}})$;
- ▷ w_I is the intensity weight equal to the maximum value of the two variances $Var(I)$.

w_D is the distance weight given as $\frac{(\mathbf{V}_{P_{s_{X,Y,Z}}} + \mathbf{V}_{n_{s_{X,Y,Z}}})}{2}$. c_D is the inter-distance constant (along the three dimensions) added depending upon the density of points and also to overcome measurement errors, holes and occlusions, etc. The value of c_D needs to be carefully selected depending upon the data (see Section 3.5.5 for more details on the selection of this value). The orientation of normals is not considered in this stage to allow the segmentation of complete 3D objects as one entity instead of just planar faces.

This segmentation method ensures that only the adjacent boundary conditions are considered for segmentation with no influence of a distant neighbor's properties. This may prove to be more adapted to sharp structural changes in the urban environment. An overview of the segmentation

method is presented in Algorithm 1. The proposed method does not put any constraint on the selection of voxels for segmentation as long as all the voxels are taken into account. They can even be taken randomly. However, in order to optimize the program code a certain selection process can be adopted. The programming structure adopted for implementation is based on standard graph-based algorithms [Wan08].

Algorithm 1 Segmentation

- 1: **repeat**
 - 2: Select a 3D point for voxelisation
 - 3: Find all neighboring points to be included in the voxel using r -NN within the specified maximum voxel length
 - 4: Transform voxel into s -voxel by first finding and then assigning all the properties including surface normal, found by using PCA, to it.
 - 5: **until** all 3D points are used in a voxel
 - 6: **repeat**
 - 7: Specify s -voxel as a principal link
 - 8: Find all secondary links attached to the principal link
 - 9: **until** all s -voxels are used
 - 10: Link all principal links to form a chain removing redundant links in the process
-

With this method 18,541, 6,928 and 7,924 s -voxels obtained from processing 3 different datasets were successfully segmented into 237, 75 and 41 distinct objects respectively.

3.3 Classification of Segmented Objects

In order to classify these segmented objects, we assume the ground to be flat and use it as separator between objects. For this purpose we first classify and segment out the ground from the scene and then the rest of the objects. This step leaves the remaining objects as if suspended in space, i.e. distinct and well-separated, making them easier to be classified as shown in Fig. 3.10. In order to classify these segmented objects a method is used which compares the geometrical models and local descriptors of these already segmented objects with a set of standard, predefined thresholds. As the object types are so distinctly different that a simple choice of values for these differentiating thresholds is sufficient.

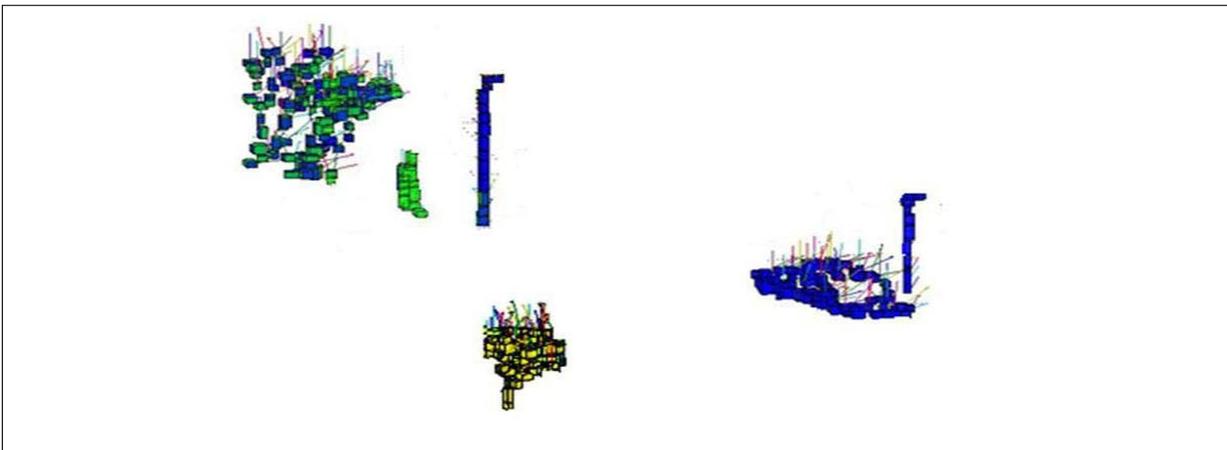


Figure 3.10: Segmented objects in a scene with prior ground removal.

The ground or roads followed by these objects are classified using geometrical and local descriptors based on the constituting super-voxels. These mainly include:

- a. **Surface normals:** The orientation of the surface normals is found essential for classification of ground and building faces. For ground object the surface normals are predominantly (threshold values greater than 80%) along Z-axis (height axis) whereas for building faces the surface normals are predominantly (threshold values greater than 80%) parallel to the X-Y axis (ground plane), see Fig. 3.11.
- b. **Geometrical center and barycenter:** The height difference between the geometrical center and the barycenter along with other properties is very useful in distinguishing objects like trees and vegetation, etc., where $h(\text{barycenter} - \text{geometrical center}) > 0$, with h being the height function.
- c. **Color and intensity:** Intensity and color are also an important discriminating factor for several objects.
- d. **Geometrical shape:** Along with the above mentioned descriptors, geometrical shape plays an important role in classifying objects. In 3D space, where pedestrians and poles are represented as long and thin with poles being longer, cars and vegetation are broad and short. Similarly, as roads represent a low flat plane, the buildings are represented as large (both in width and height) vertical blocks (as shown in Fig. 3.12). The values for these comparison threshold on the shape and size for each of the object types are set accordingly.

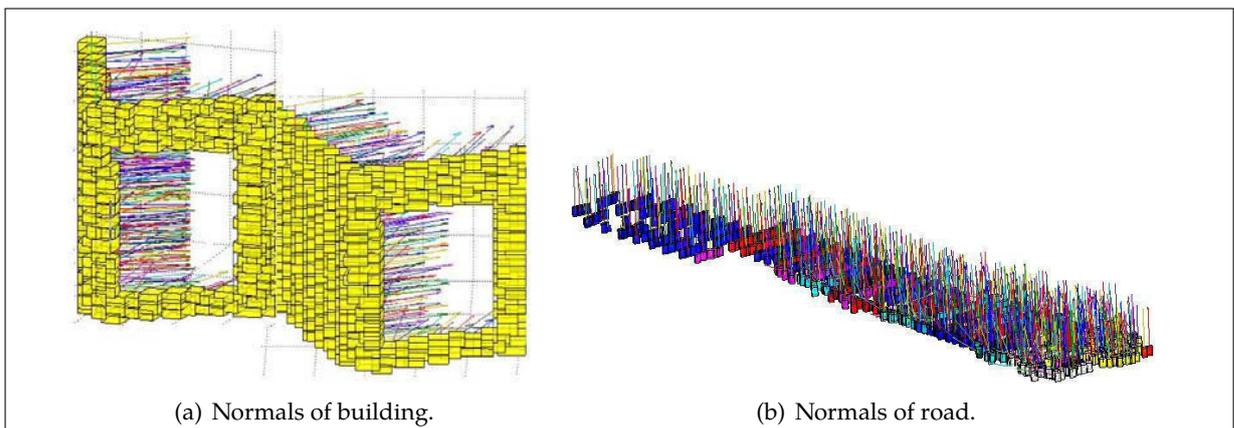


Figure 3.11: (a) shows surface normals of building s -voxels which are parallel to the ground plane. In (b) it can be clearly seen that the surface normals of road surface s -voxels are perpendicular to the ground plane.

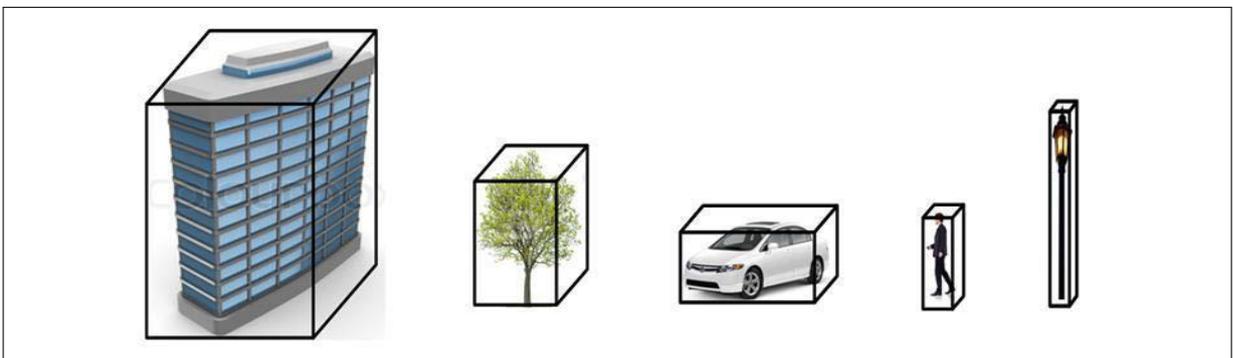


Figure 3.12: Bounding boxes for buildings, trees, cars, pedestrians and poles.

Using these descriptors we successfully classify urban scenes into 5 different classes (mostly present in our scenes) i.e. buildings, roads, cars, poles and trees. The object types chosen for classification are so distinctly different that if they are correctly segmented out, a simple

classification method like the one proposed may be sufficient. The classification results and a new evaluation metric are discussed in the following sections.

3.4 Evaluation Metrics

Over the years as new segmentation and classification methods have been introduced, different evaluation metrics have been proposed to evaluate their performances. In previous works, different evaluation metrics are introduced for both segmentation results and classifiers independently. Thus in our work we present a new evaluation metric which incorporates both segmentation and classification together.

The evaluation method is based on comparing the total percentage of s -voxels successfully classified as a particular object. Let $T_i, i \in \{1, \dots, N\}$, be the total number of s -voxels distributed into objects belonging to N number of different classes, i.e. this serves as the ground truth, and let $t_{j_i}, i \in \{1, \dots, N\}$, be the total number of s -voxels classified as a particular class of type- j and distributed into objects belonging to N different classes (for example a s -voxel classified as part of the building class may actually belong to a tree) then the ratio S_{jk} (j is the class type as well as the row number of the matrix and $k \in \{1, \dots, N\}$) is given as:

$$S_{jk} = \frac{t_{jk}}{T_k}$$

These values of S_{jk} are calculated for each type of class and are used to fill up each element of the confusion matrix, row by row (refer to Table 3.2 for instance). Each row of the matrix represents a particular class.

Thus, for a class of type-1 (i.e. first row of the matrix) the values of:

- ▷ **True Positive rate TP** = S_{11} (i.e. the diagonal of the matrix represents the TPs)
- ▷ **False Positive rate FP** = $\sum_{m=2}^N S_{1m}$
- ▷ **True Negative rate TN** = $(1 - \text{FP})$
- ▷ **False Negative rate FN** = $(1 - \text{TP})$

The diagonal of this matrix or TPs gives the Segmentation ACCuracy (SACC), similar to the voxel scores recently introduced by Douillard *et al.* [DUK+11]. The effects of unclassified s -voxels are automatically incorporated in the segmentation accuracy. Using the above values, the Classification ACCuracy (CACC) is given as:

$$\text{CACC} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \quad (3.5)$$

This value of CACC is calculated for all N types of classes of objects present in the scene. Overall Classification ACCuracy (OCACC) can then be calculated as

$$\text{OCACC} = \frac{1}{N} \sum_{i=1}^N \text{CACC}_i \quad (3.6)$$

where N is the total number of object classes present in the scene. Similarly, the Overall Segmentation ACCuracy (OSACC) can also be calculated. The values of T_i and t_{j_i} used above are laboriously evaluated by hand matching the voxelised data output and the final classified s -voxels and points. In case no voxels are available, the 3D points can be used to calculate the evaluation metrics.

Evaluating classification results is a common task as presented in [MVH09] that employs a standard confusion matrix using the number of classification results of an object type in different

classes which are used to calculate the precision and recall. These values are then used to calculate classification accuracy (which indeed is a standard feature in object classification). Whereas, the proposed method differs from the standard approach as it uses a new metric: S_{jk} to fill in each element of the matrix. This method is more suitable for work similar to ours as it provides more insight in segmentation results and directly gives the segmentation accuracy similar to Douillard *et al.* [DUK+11]. Also as compared to standard precision and recall evaluation, the use of this metric in the proposed method, also accommodates for the unclassified voxels in the results giving a more accurate result without incorporating the unclassified-objects as a class in the confusion matrix. Hence these make this approach unique. Although, the evaluation method is envisaged for voxel type representations, however, if voxels are not available then the corresponding 3D points can be used instead for calculating the evaluation metrics in the same manner.

3.5 Results

In order to test our algorithm two different datasets were used:

1. 3D datasets of Blaise Pascal University;
2. 3D Urban Data Challenge dataset [FS11].

The 3D Urban Data challenge dataset is not only one of the most recent dataset but it also contains the corresponding RGB and reflectance intensity values necessary to validate the proposed method. The proposed method is also suitable and well-adapted for directly geo-referenced 3D point clouds obtained from mobile data acquisition and mapping techniques [BMSV08].

3.5.1 3D Datasets of Blaise Pascal University

These datasets consist of 3D data acquired from different urban scenes on the Campus of Blaise Pascal University in Clermont-Ferrand, France, using a LEICA HDS-3000 3D laser scanner. The results of three such datasets are discussed here. The datasets consist of 27,396, 53,676 and 110,392 3D points respectively. These 3D points were coupled with corresponding RGB and reflectance intensity values. The results are summarized in Table 3.1 and shown in Fig. 3.13, 3.14 and 3.15 respectively. The evaluation results using the new evaluation metrics for the three datasets are presented in Tables 3.2, 3.3 and 3.4 respectively. These results are evaluated using a value of maximum voxel size equal to $0.3 m$ and $c_D = 0.25 m$.

Table 3.1: Segmentation results of 3D datasets of Blaise Pascal University.

Dataset #	Number of 3D data points	Number of segmented s -voxels	Number of segmented objects
# 1	27,396	7,924	41
# 2	53,676	6,928	75
# 3	110,392	18,541	237

3.5.2 3D Urban Data Challenge Dataset

The algorithm was further tested on the dataset of the recently concluded 3D Urban Data Challenge 2011, acquired and used by Friedman and Stamos [FS11]. This dataset contains a rich collection of 3D urban scenes of New York city mainly focusing on building façades and structures. These 3D points are coupled with corresponding RGB and reflectance intensity values. A value of maximum voxel size equal to $0.5 m$ and $c_D = 0.15 m$ were used for this dataset.

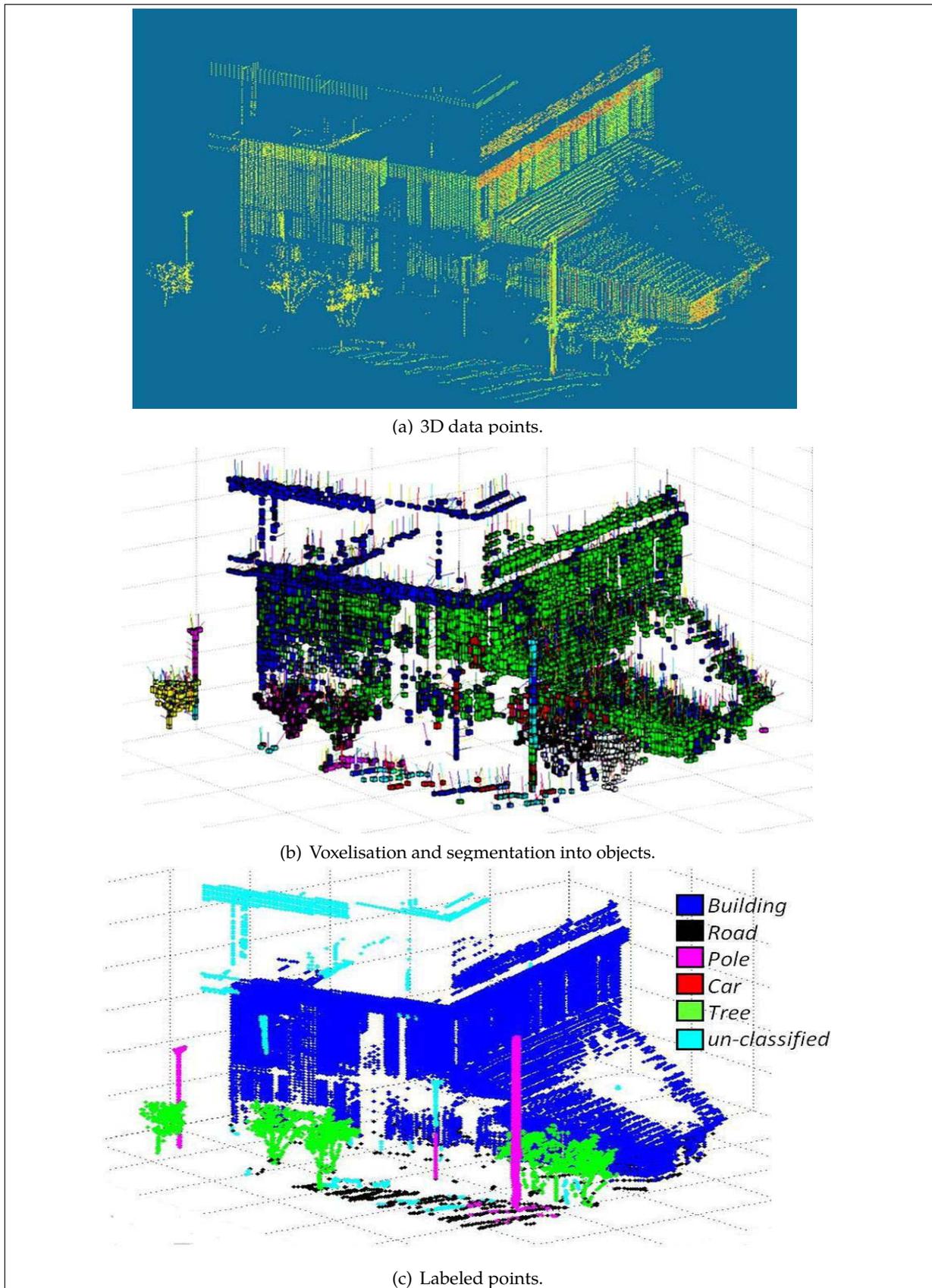


Figure 3.13: (a) shows 3D data points of dataset 1. (b) shows s -voxel segmentation of 3D points (along with orientation of normals). (c) shows classification results (labeled 3D points).

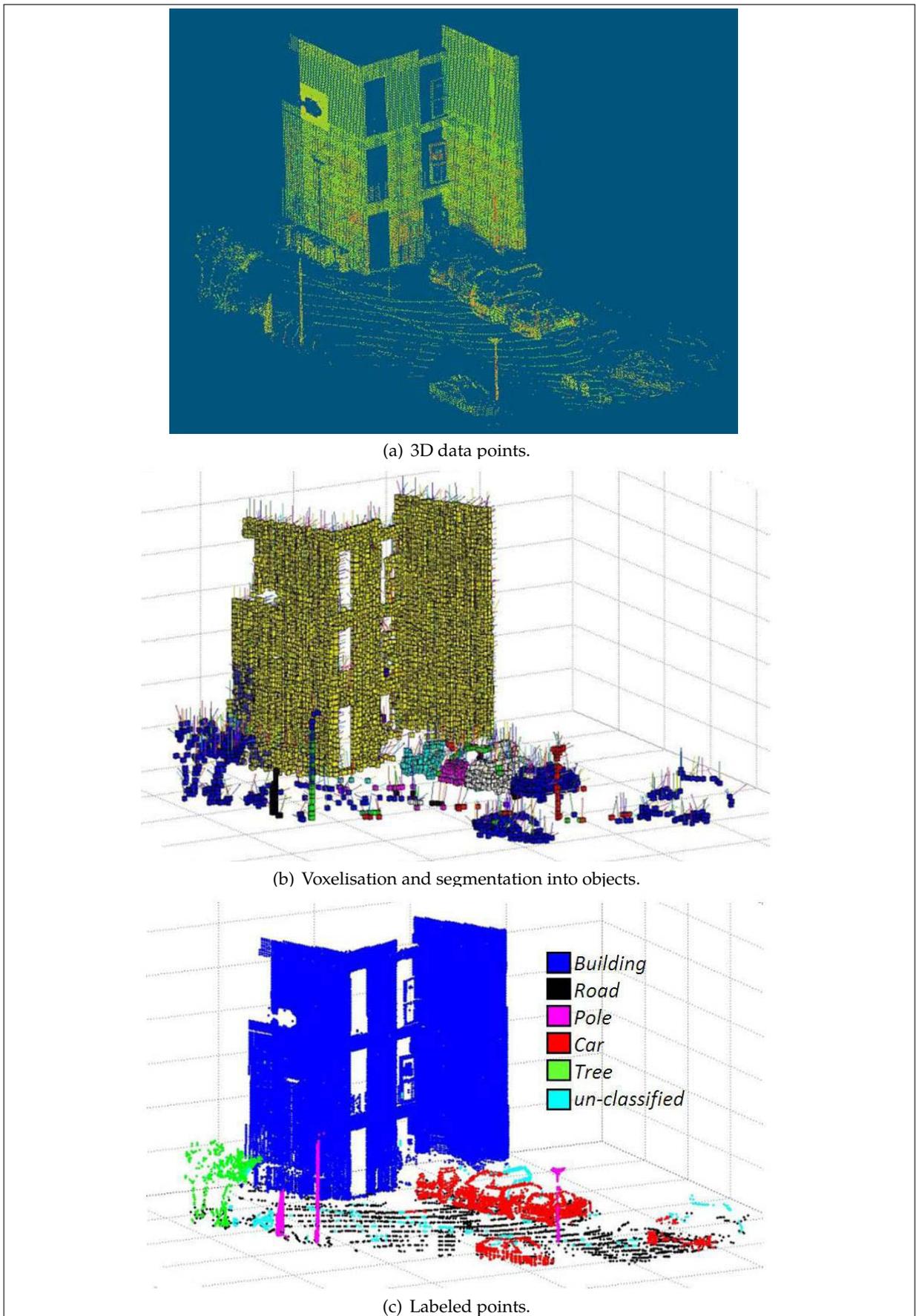
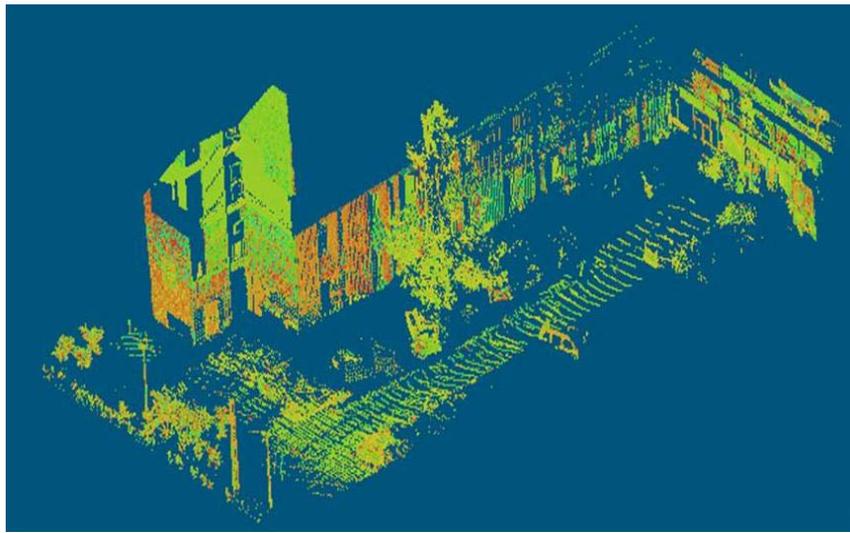
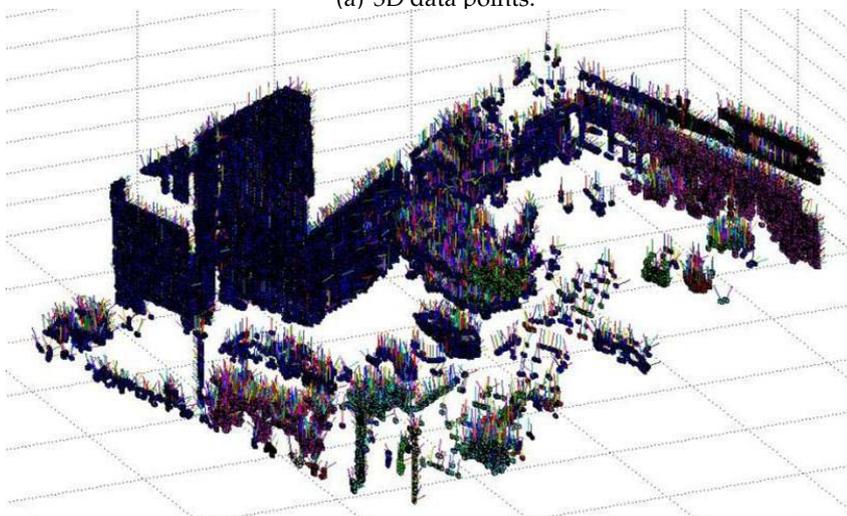


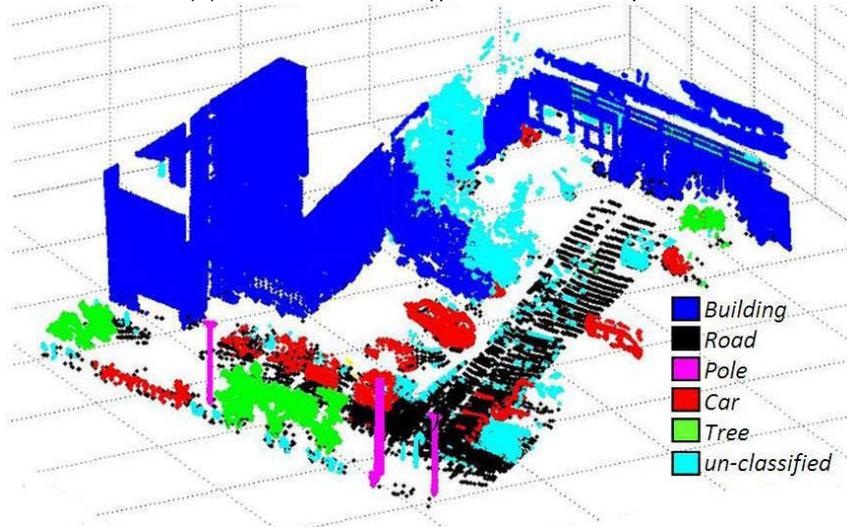
Figure 3.14: (a) shows 3D data points of dataset 2. (b) shows s -voxel segmentation of 3D points (along with orientation of normals). (c) shows classification results (labeled 3D points).



(a) 3D data points.



(b) Voxelisation and segmentation into objects.



(c) Labeled points.

Figure 3.15: (a) shows 3D data points of dataset 3. (b) shows s -voxel segmentation of 3D points (along with orientation of normals). (c) shows classification results (labeled 3D points).

Table 3.2: Classification results of Blaise Pascal University dataset 1 with the new evaluation metrics.

	Building	Road	Tree	Pole	Car	CACC
Building	0.943	0.073	0	0	0	0.935
Road	0.007	0.858	0.015	0.008	0	0.914
Tree	0	0.025	0.984	0	0	0.979
Pole	0	0.049	0	0.937	0	0.944
Car	-	-	-	-	-	-
Overall segmentation accuracy: OSACC					0.930	
Overall classification accuracy: OCACC						0.943

Table 3.3: Classification results of Blaise Pascal University dataset 2 with the new evaluation metrics.

	Building	Road	Tree	Pole	Car	CACC
Building	0.996	0.007	0	0	0	0.995
Road	0	0.906	0.028	0.023	0.012	0.921
Tree	0	0.045	0.922	0	0	0.938
Pole	0	0.012	0	0.964	0	0.976
Car	0	0.012	0	0	0.907	0.947
Overall segmentation accuracy: OSACC					0.939	
Overall classification accuracy: OCACC						0.955

Results of different scenes from this dataset are shown in Fig. 3.16, 3.17, 3.18, Tables 3.5, 3.6 and 3.7.

3.5.3 Comparison of Results with Existing Evaluation Methods

The classification results were also evaluated using already existing methods along with the proposed evaluation metrics for comparison purpose. Firstly, one of the more frequently used metrics: F-Measure which is based on the calculation of Recall and Precision as described in [FWE03] is used. Secondly, a conditional entropy based metrics: V-Measure which is based on the calculation of Homogeneity and Completeness as presented in [RH07] is used. The latter method overcomes the problem of matching suffered by the former and evaluates a solution independent of the algorithm, size of the dataset, number of classes and number of clusters as explained in [RH07]. Another advantage of using these two metrics is that just like the proposed metrics they have the same bounded score. For all three metrics the score varies from 0 to 1 and higher score signifies better classification results and vice versa. The results are summarized in Table 3.8.

From Table 3.8 it can be seen that the results evaluated by all the three evaluation metrics are consistent with dataset 2 receiving the highest scores and dataset 3 the lowest. The results not only validate the proposed metrics but also indicate that it can be used as an alternative evaluation method. The results evaluated using these standard existing evaluation methods also permit to compare the performance of the proposed algorithm with other published techniques evaluated using them.

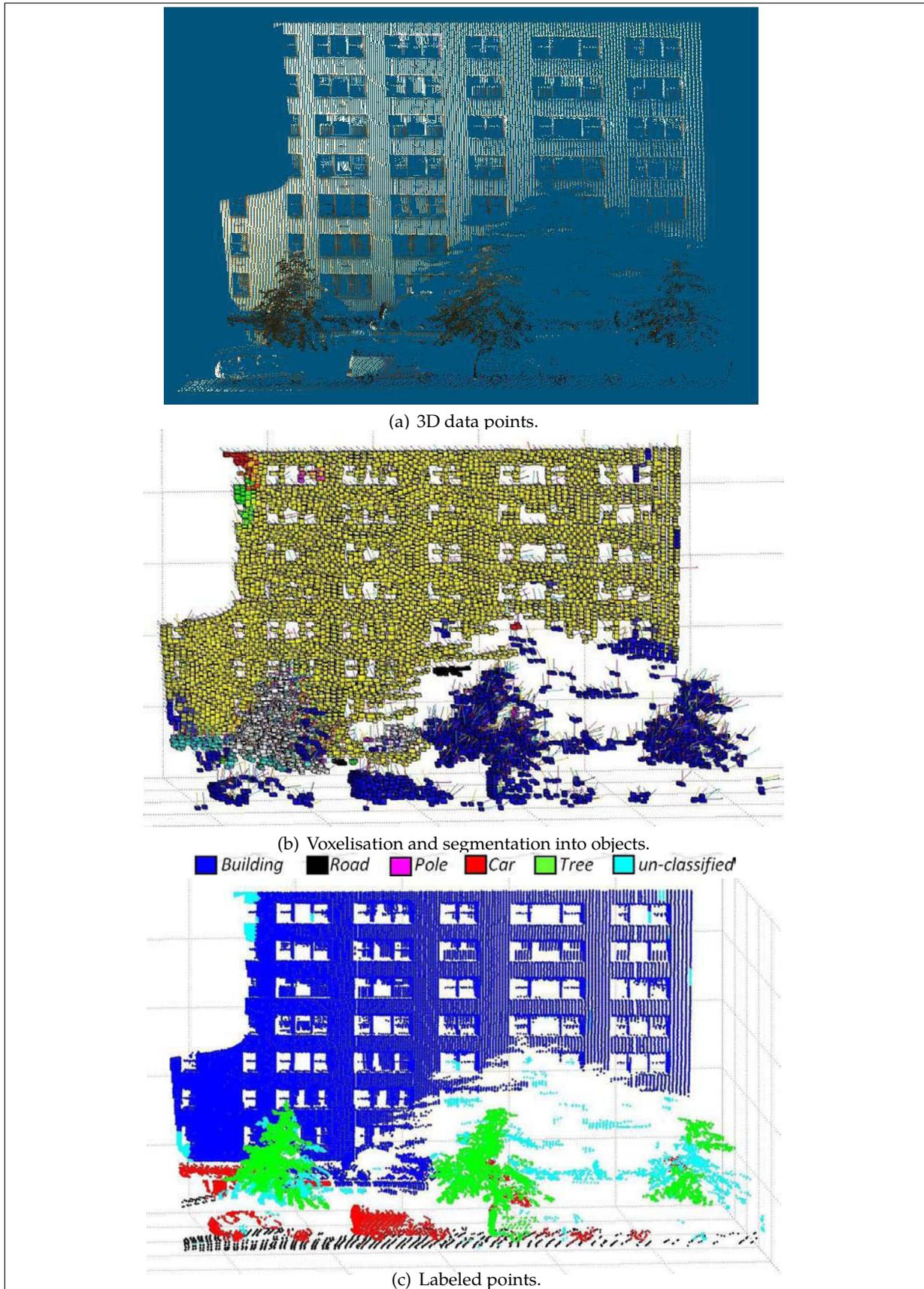


Figure 3.16: Segmentation and classification results for a particular scene-A of scenes from 3D Urban Data Challenge 2011, image # ParkAvenue_SW12_piece07 [FS11]. (a) shows 3D data points of scene-A. (b) shows s -voxel segmentation of 3D points (along with orientation of normals). (c) shows classification results (labeled 3D points).

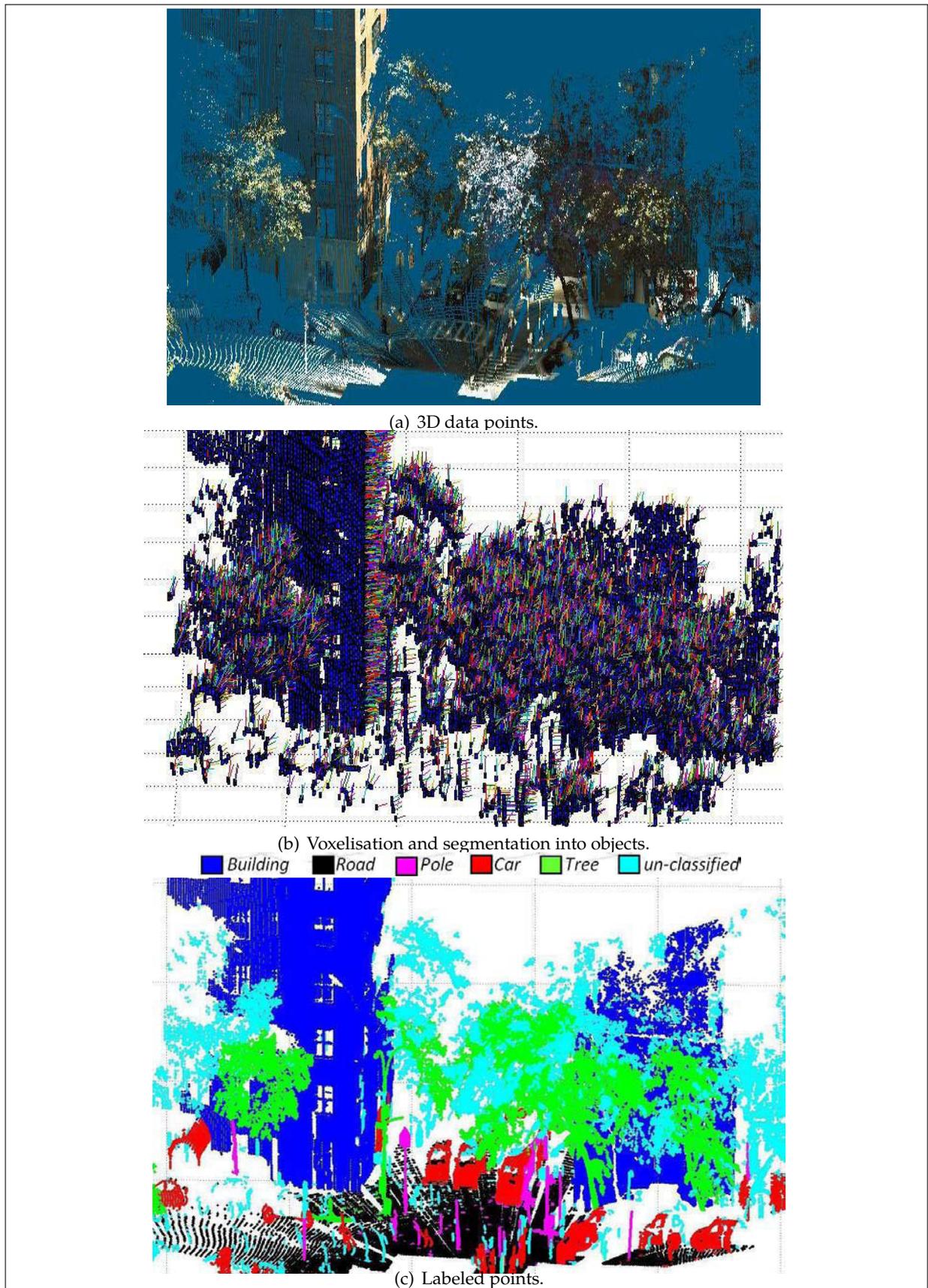


Figure 3.17: Segmentation and classification results for a particular scene-B of scenes from 3D Urban Data Challenge 2011, image # ParkAvenue_SW12_piece00 [FS11]. (a) shows 3D data points of scene-B. (b) shows s -voxel segmentation of 3D points (along with orientation of normals). (c) shows classification results (labeled 3D points).

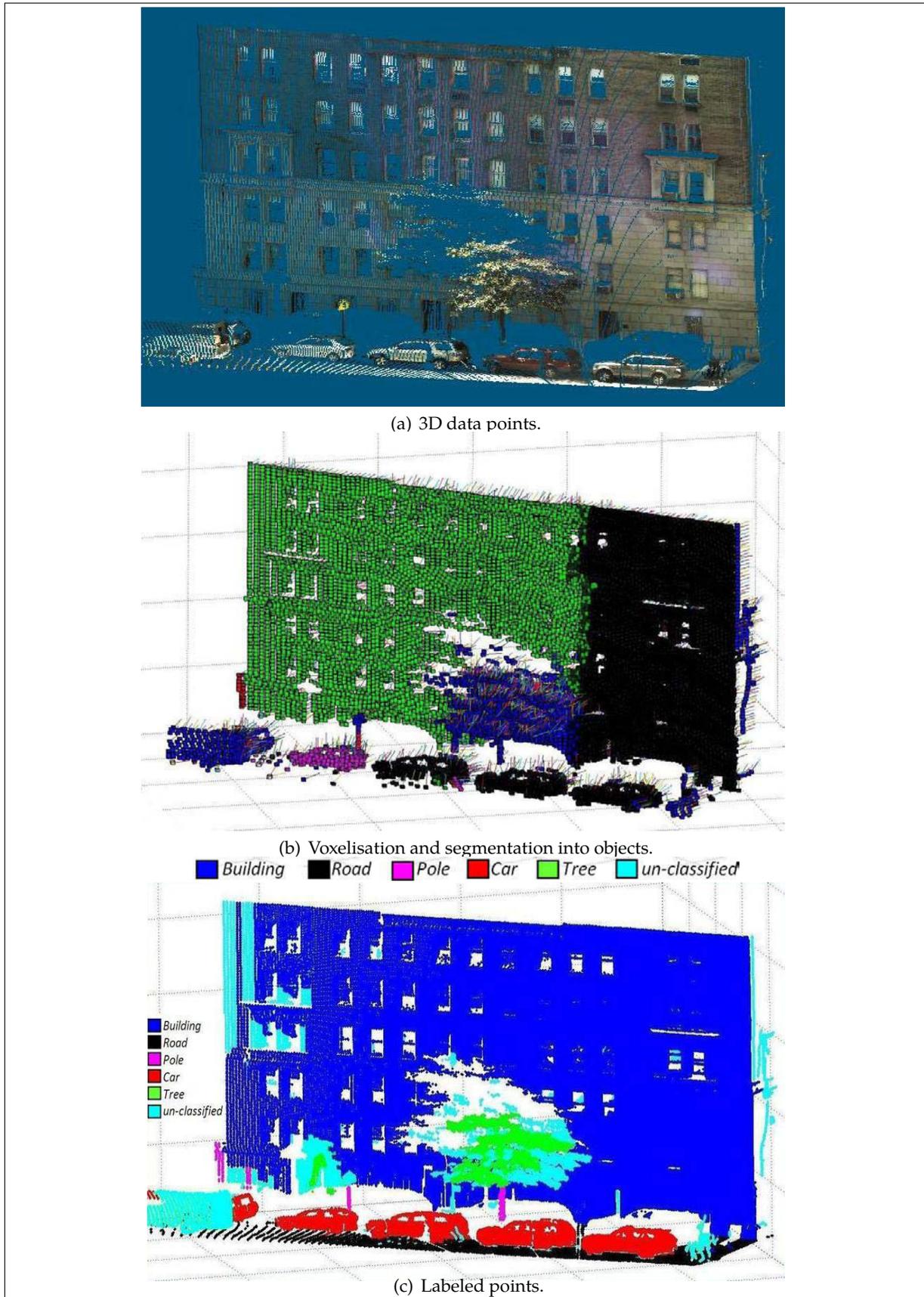


Figure 3.18: Segmentation and classification results for a particular scene-C of scenes from 3D Urban Data Challenge 2011, image # ParkAvenue_SW14_piece00 [FS11]. (a) shows 3D data points of scene-C. (b) shows *s*-voxel segmentation of 3D points (along with orientation of normals). (c) shows classification results (labeled 3D points).

Table 3.4: Classification results of Blaise Pascal University dataset 3 with the new evaluation metrics.

	Building	Road	Tree	Pole	Car	CACC
Building	0.901	0.005	0.148	0	0	0.874
Road	0.003	0.887	0.011	0.016	0.026	0.916
Tree	0.042	0.005	0.780	0	0	0.867
Pole	0	0.002	0	0.966	0	0.982
Car	0	0.016	0.12	0	0.862	0.863
Overall segmentation accuracy: OSACC					0.879	
Overall classification accuracy: OCACC						0.901

Table 3.5: Classification results of 3D Urban Data Challenge dataset (scene-A) with the new evaluation metrics.

	Building	Road	Tree	Pole	Car	CACC
Building	0.980	0.002	0	0	0	0.989
Road	0.002	0.950	0.002	0	0.080	0.933
Tree	0	0.040	0.890	0	0.080	0.885
Pole	0	0	0	0	0	-
Car	0.040	0.020	0.030	0	0.900	0.905
Overall segmentation accuracy: OSACC					0.930	
Overall classification accuracy: OCACC						0.928

3.5.4 Performance Evaluation and Discussion

The proposed method gives good (in terms of scores) segmentation and classification results in all three evaluation methods. In general, the classification accuracy (**OCACC**) was found to be slightly better than the segmentation accuracy (**OSACC**). Not taking anything away from the segmentation method, one of the main reasons for this being the fact that the 5 types of objects chosen for classification are distinctly different and that if the segmentation is good, classification becomes easier and a simple method like the one proposed is sufficient.

As compared to V-Measure, the proposed method of evaluation can provide more information regarding individual segmentation and classification results (**SACC** and **CACC**). These results show that in most of the cases, the buildings, roads and poles have been classified the best with a consistent scores of **SACC** and **CACC** higher than 90% except in the case of dataset 3 in which the building classification accuracy **CACC** is slightly deteriorated due to a large overlapping tree which is wrongly classified as a building rather than a tree. This is also reflected in the low Homogeneity value of 0.670 obtained when calculating V-Measure for this dataset. The classification of cars is generally good and the results are consistent but they are slightly hampered due to occlusions in some scenes (dataset 3: **CACC** 86.3%, Scene B: **CACC** 89.5%). In case of trees, the **SACC** and **CACC** are found to vary the most. This is mainly due to the fact that the proposed classification method is based on local descriptors and geometrical features which in the case of trees are very difficult to define (due to large variation of shapes, sizes and types). Thus, where the proposed algorithm succeeded in classifying smaller trees of more classical shapes with a higher **SACC** and **CACC** scores, it produced a low **SACC** and **CACC** score of 68% and 79.4% respectively for Scene B. The Recall and Precision scores obtained during the calculation of F-Measure for the tree class of this scene were similarly found low as well (0.682 and 0.614 respectively).

Table 3.6: Classification results of 3D Urban Data Challenge dataset (scene-B) with the new evaluation metrics.

	Building	Road	Tree	Pole	Car	CACC
Building	0.985	0.002	0	0	0	0.991
Road	0.002	0.950	0.002	0	0.080	0.933
Tree	0	0.012	0.680	0.080	0	0.794
Pole	0	0.006	0	0.860	0.016	0.919
Car	0.060	0.050	0.020	0.050	0.970	0.895
Overall segmentation accuracy: OSACC					0.889	
Overall classification accuracy: OCACC						0.906

Table 3.7: Classification results of 3D Urban Data Challenge dataset (scene-C) with the new evaluation metrics.

	Building	Road	Tree	Pole	Car	CACC
Building	0.955	0.002	0.005	0.001	0	0.976
Road	0.002	0.950	0	0	0.007	0.970
Tree	0	0	0.800	0.035	0	0.882
Pole	0	0	0	0.950	0	0.950
Car	0	0.003	0	0	0.900	0.948
Overall segmentation accuracy: OSACC					0.911	
Overall classification accuracy: OCACC						0.945

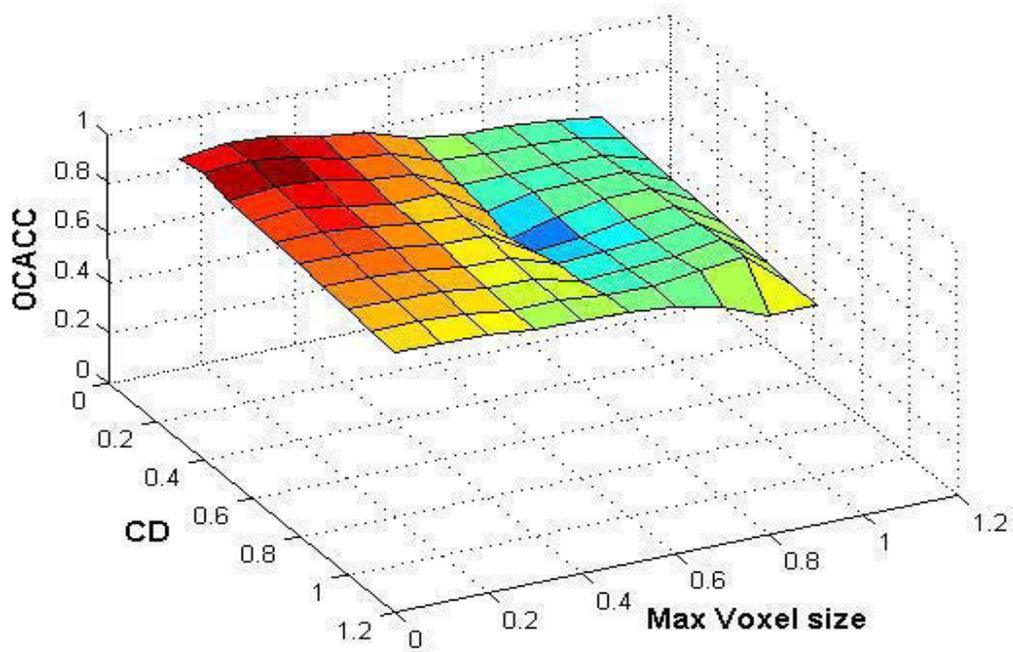
3.5.5 Effect of Voxel Size on Classification Accuracy and Choice of Optimal Values

As the properties of s -voxels are constant mainly over the whole voxel length and these properties are then used for segmentation and then classification, thus their size impacts the classification process. However as the voxel size changes, the inter-distance constant c_D also needs to be adjusted accordingly.

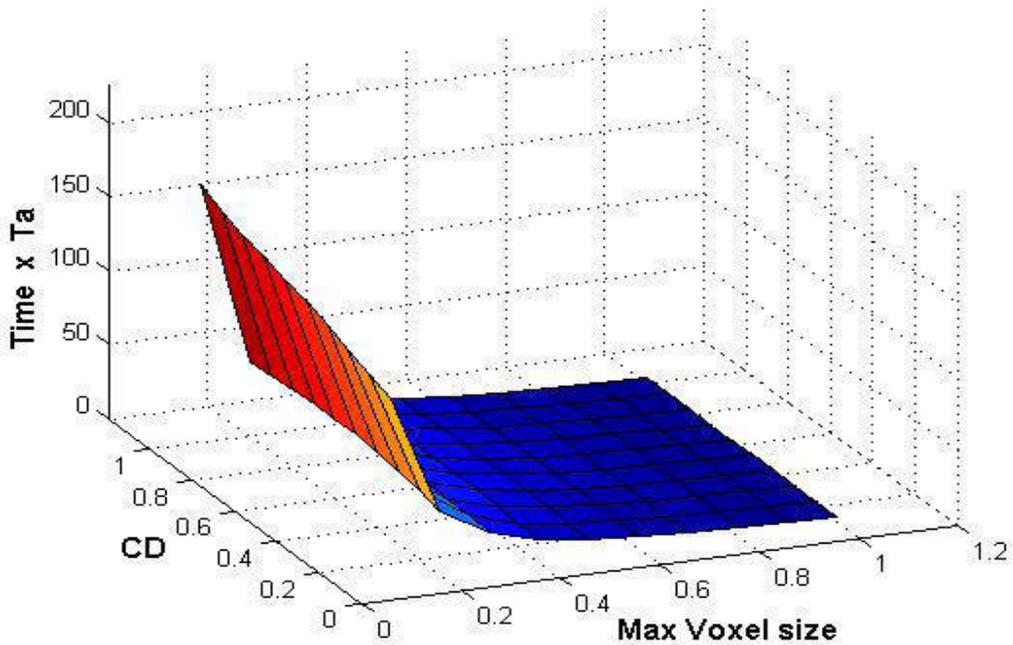
The effect of voxel size on the classification result was studied. The maximum voxel size and the value of c_D were varied from 0.1 m to 1.0 m on dataset 1 and corresponding classification accuracy was calculated. The results are shown in Fig. 3.19(a). Then for the same variation of maximum voxel size and c_D , the variation in processing time was studied as shown in Fig. 3.19(b).

An arbitrary value of time T_a is chosen for comparison purposes (along Z-axis time varies from 0 to $200T_a$). This makes the comparison results less dependent on the processor used, even though the same computer was used for all computations.

The results show that with smaller voxel size the segmentation and classification results are improved (with a suitable value of c_D) but the computational cost increases. It is also evident that variation in value of c_D has no significant impact on time t . It is also observed that after a certain reduction in voxel size the classification result does not improve much but the computational cost continues to increase manifold. As both OCACC and time (both plotted along Z-axis) are independent, thus using and combining the results of the two 3D plots in Fig. 3.19, we can find the optimal value (in terms of OCACC and time t) of maximum voxel size and c_D depending upon the final application requirements. For our work we have chosen a maximum voxel size of 0.3 m and $c_D = 0.25 m$.



(a) Influence of voxel size on OCACC.



(b) Influence of voxel size on processing time.

Figure 3.19: (a) is a 3D plot in which the effect of maximum voxel size and variation on OCACC is shown. In (b) the effect of maximum voxel size and variation on processing time is shown. Using the two plots we can easily find the optimal value for maximum voxel size and inter-distance constant c_D .

Table 3.8: Classification results evaluated using three different metrics. For the calculation of V-Measure the value $\beta = 1$ is used.

Dataset #	OCACC	F-Measure	V-Measure
# 1	0.943	0.922	0.745
# 2	0.955	0.942	0.826
# 3	0.901	0.831	0.733
# A	0.928	0.917	0.741
# B	0.906	0.860	0.734

Table 3.9: Overall segmentation and classification accuracies when using RGB-Color and reflectance intensity values.

Dataset #	Only RGB-Color		Intensity value with RGB-Color	
	OSACC	OCACC	OSACC	OCACC
# 1	0.660	0.772	0.930	0.943
# 2	0.701	0.830	0.939	0.955
# 3	0.658	0.766	0.879	0.901

3.5.6 Influence of RGB Color and Reflectance Intensity

The effect of incorporating RGB Color and reflectance intensity values on the segmentation and classification was also studied. The results are presented in Table 3.9.

It is observed that incorporating RGB color alone is not sufficient in an urban environment due to the fact that it is heavily affected by illumination variation (part of an object may be under shade or reflect bright sunlight) even in the same scene. This deteriorates the segmentation process and hence the classification. This is perhaps responsible for the lower classification accuracy as seen in first part of Table 3.9. It is the reason why intensity values are incorporated as they are more illumination invariant and found to be more consistent. The improved classification results are presented in second part of Table 3.9.

3.5.7 Limitations

The evaluated results of the proposed method on real datasets show great promise. However, the method has two obvious limitations (which are more or less common to all classification methods). Firstly, it classifies the urban environment into 5 basic object classes, where as sometimes in the urban environment there could be other object types present as well. Secondly, as the proposed method relies on geometrical models and local descriptors to classify segmented objects, it could be possible that an object observed by the sensor is different from its normal appearance for example due to partial occlusion as shown in Fig. 3.20 (in which a car is partially occluded by another car) or due to limited viewing angle of the 3D lidar sensor as shown in Fig. 3.21 (for this dataset the LiDAR sensor on the IGN STÉRÉOPOLIS II mapping system was mounted at such an angle that it acquired only a small part of the building façade and that too was hampered due to occlusions at ground level). As a result these objects either remain unclassified or are wrongly classified as part of another class.

Unlike other classification methods, we further study such objects in an attempt to better classify them using the notion of multiple passages (discussed in Chapter 4). Hence, for both these cases we place such unclassified objects under a special label/class called "unclassified" which will be further treated.

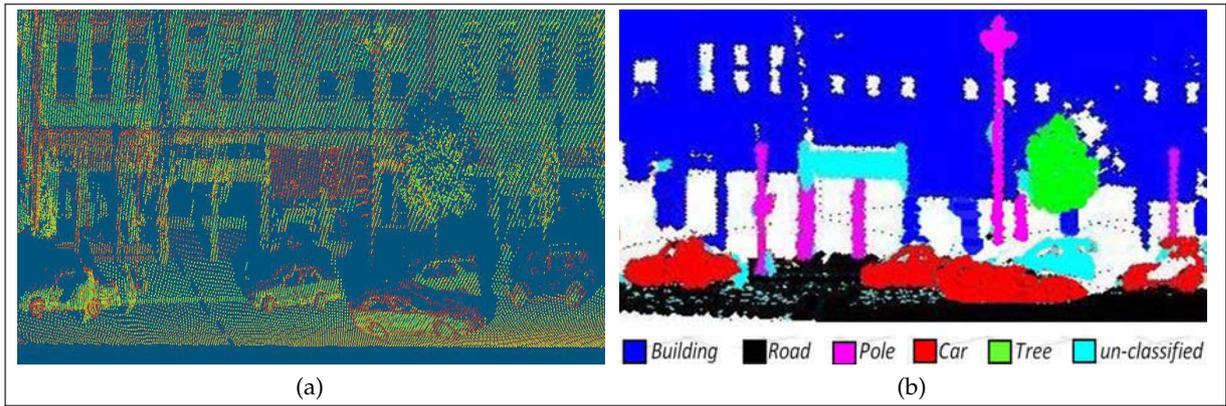


Figure 3.20: The figure shows the effects partial occlusion. A car partially occluded by another car remains unclassified.

3.6 Comparison with Existing Techniques

The performance of the proposed method was compared with existing techniques at two different levels: first, the proposed classification method with a standard classification technique, and then the whole segmentation-classification pipeline (i.e. the complete method) with another state of the art method.

3.6.1 Comparison with Standard SVM based Classification Method

The proposed classification method was compared with a classic SVM based classification method. The parameters used in the SVM classifier were the same as those used for the feature based method. They are summarized in Table 3.10. The classifier was trained using manually corrected ground truth pertaining to the different dataset used in the evaluation process. The ground was segmented out first and the remaining objects were then subjected to the two methods. The results for comparison are evaluated on two carefully selected datasets pertaining to two different urban scenes. The results for the two methods are presented in Table 3.11.

Table 3.10: Parameters/Features description.

Symbol	Description
H_{\min}	minimum height
H_{\max}	maximum height
d_x	width
d_y	breadth
d_z	height
d_c	barycenter - geometrical center
n_x	normal component (average) in x direction
n_y	normal component (average) in y direction
n_z	normal component (average) in z direction

The results show that the performance of the two methods is comparable. Although the classification results for the SVM method is slightly better, there is a stronger interaction between different classes as compared to the feature based method. In the feature based method, most of the objects that are not correctly classified are added to the list of unclassified objects that are then analyzed in successive passages to improve classification results as explained in the next chapter.

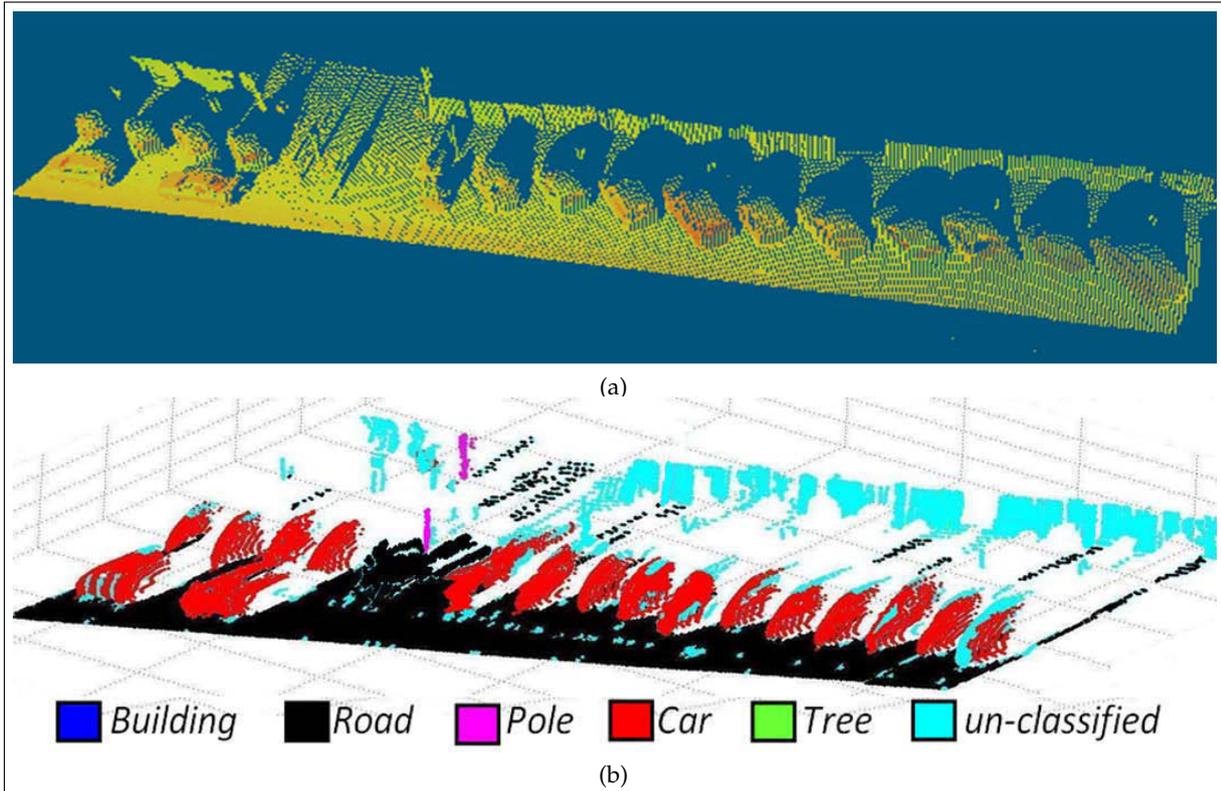


Figure 3.21: The figure shows the effects of strong variation in appearance. The building façade which is partially acquired, and is also incomplete due to occluding objects at ground level, remains unclassified.

Table 3.11: Comparison results for classification methods.

	Test Dataset 1 (CACC)		Test Dataset 2 (CACC)	
	Proposed Method	SVM	Proposed Method	SVM
Building	0.935	1.00	0.944	0.975
Car	0.825	0.860	0.865	0.900
Pole	0.800	0.860	0.880	0.895
Tree	0.750	0.790	0.795	0.840
OCACC	0.827	0.877	0.871	0.903

On the other hand, this also shows that depending upon the application and according to the requirements, feature based classification method can be easily replaced by other classical classification techniques like SVM using same/similar parameters after the super-voxel based segmentation to improve results.

3.6.2 Comparison with a State of the Art Method

The performance of the proposed method was also compared with that of another state of the art method [SM14] presented in Section 3.1.1.5. Although this method is relatively faster than the super-voxel based approach, both methods employ data reduction approaches (elevation images and voxels respectively) to improve efficiency in terms of result quality and computational time. Similarly, both methods use geometrical features for classification, however, mathematical morphology method also incorporates contextual features in its SVM (Support Vector Machine) based classifier while the super-voxel based approach uses local descriptors to classify the

$$\text{Re}(m) = \frac{\text{number_of_detected_objects_matched}}{\text{total_number_of_ground_truth_objects}} \quad (3.9)$$

These values of Pr and Re are then combined together to calculate the F-Measure as a function of m as expressed in Equation (3.10).

$$F(m) = (1 + \beta^2) \frac{\text{Pr}(m) \times \text{Re}(m)}{\beta^2 \text{Pr}(m) + \text{Re}(m)} \quad (3.10)$$

Figure 3.23 shows the values of F-Measure with the variation of m for the different object types using both methods. The value of β was taken as 1 to obtain the harmonic mean. The value of F-Measure decreases with the increasing value of m and this decay indicates the performance quality of the detection (good performance implies slower decay).

Although the voxel-based method does not classify motorcycles, they were detected and classified manually to analyze their segmentation quality (discussed in the next section).

The results show that the building and ground are much better detected by the morphological transformation method while the detection quality performance for cars, poles, and other road furniture is much more superior for the voxel-based method.

3.6.2.2 Classification Evaluation

The evaluation was conducted for five common classes: {Building, Road, Pole, Tree, Car} and also the motorcycle class (only segmentation results). The segmentation and classification results for both the methods are shown in Fig. 3.24 while the quantitative results are presented in Table 3.12 using the same evaluation metrics described in Section 3.4. As trees were not present in the dataset they were not considered for analysis.

Table 3.12: Comparison of classification results.

	Building	Road	Pole	Car	CACC
Building	0.914 (0.986)	0.013 (0.045)	0.000 (0.000)	0.000 (0.010)	0.950 (0.970)
Road	0.02 (0.002)	0.901 (0.940)	0.005 (0.000)	0.010 (0.002)	0.933 (0.968)
Pole	0.000 (0.000)	0.001 (0.001)	0.71 (0)	0 (0.01)	0.850 (0.495)
Car	0.000 (0.010)	0.005 (0.195)	0.000 (0.000)	0.900 (0.950)	0.950 (0.870)
Overall segmentation accuracy: OSACC				0.856 (0.720)	
Overall classification accuracy: OCACC					0.920 (0.825)

The Table 3.12 shows the results for both super-voxel based method and morphological transformation method (values inside braces). It can be seen that for the voxel-based method, some of the 3D points belonging to different object classes are found in the road class and vice versa. This was found evident at boundary regions of objects belonging to two different classes, as shown in Fig. 3.25, as sometimes in the voxelisation process, some of the 3D points belonging to adjacent objects are incorporated in the same voxel if they have similar color and reflectance intensity values.

Also, it was found that, for this method, one of the traffic sign post was wrongly classified as a tree resulting in a low SACC and CACC of 0.71 and 0.85 respectively. This was due to the fact that the particular sign post contained two traffic signs on the same post giving it a small tree like appearance (in 3D point cloud at least) as shown in Fig. 3.26. Compared to this method, the morphological transformation method failed to classify any of the poles correctly (as depicted in the table), confusing most of them with trees.

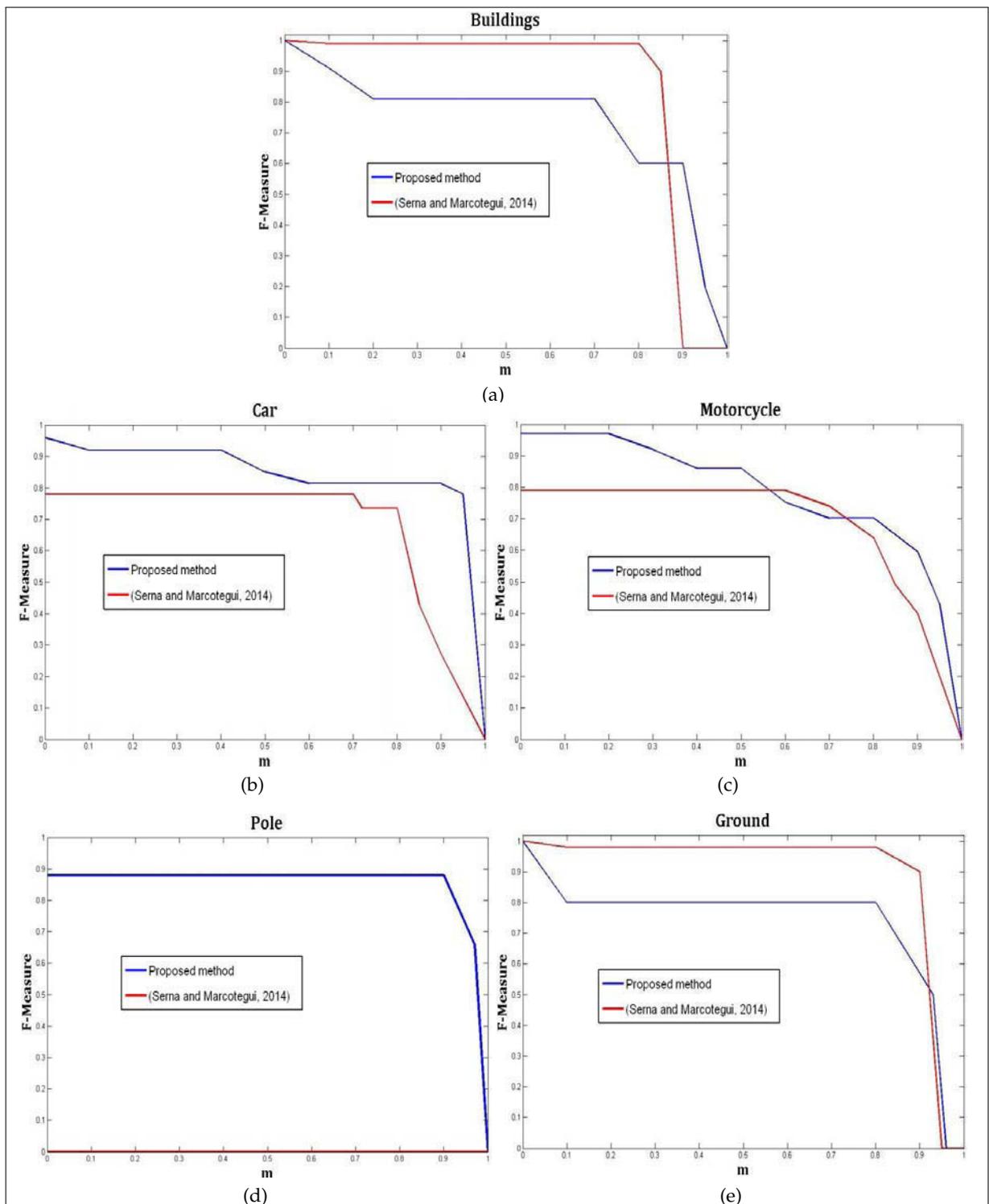


Figure 3.23: Detection results for both voxel-based and morphological transformation based methods are presented for 5 different classes in (a)-(e) respectively.

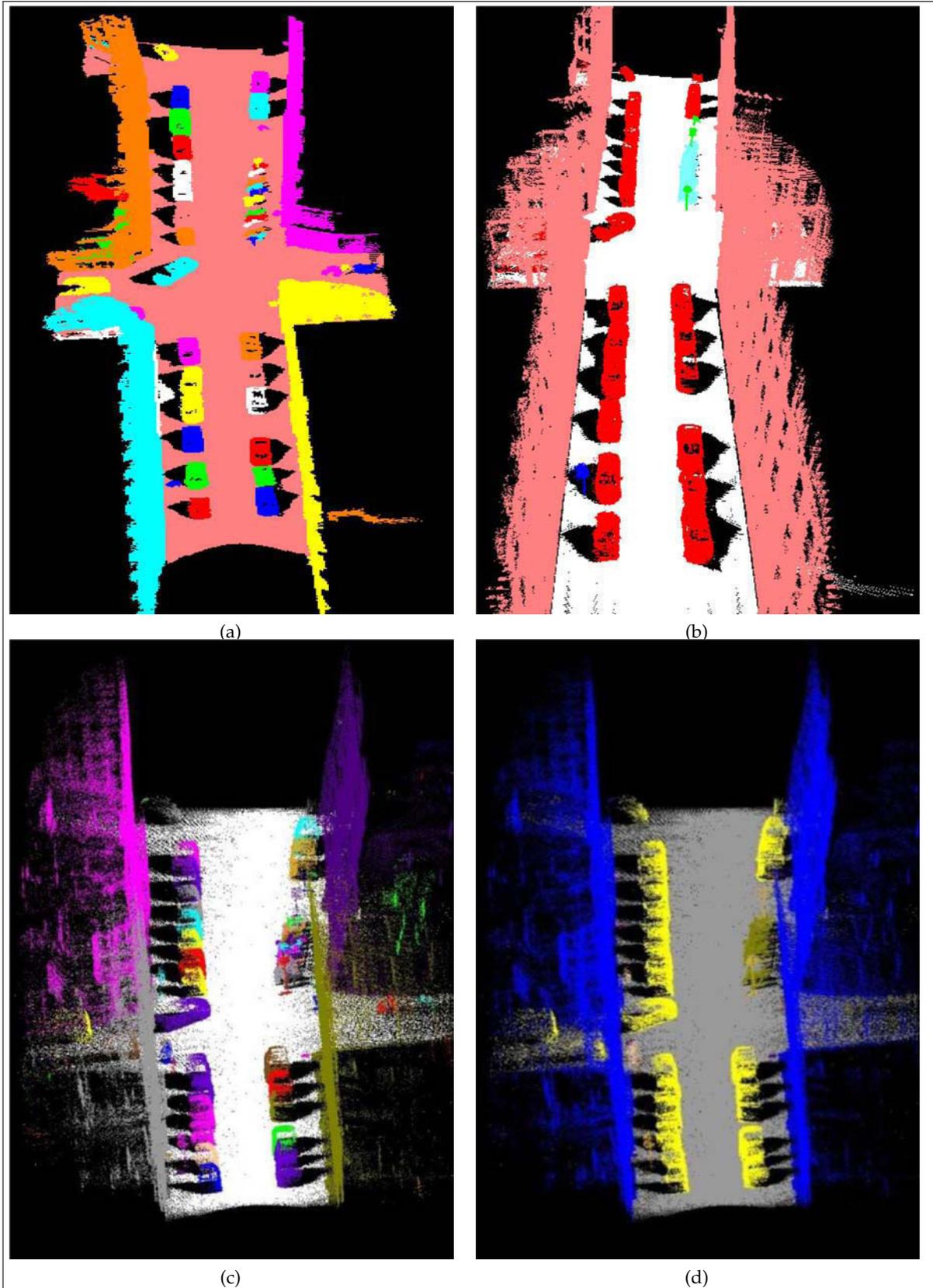


Figure 3.24: (a) & (b) show the segmentation and classification results for voxel-based while (c) & (d) show the segmentation and classification results for morphological transformation based method respectively. In (a) & (c) every segmented object is represented by a separate color (some colors are repeated) while in (b) & (d) each class is represented by a separate color.

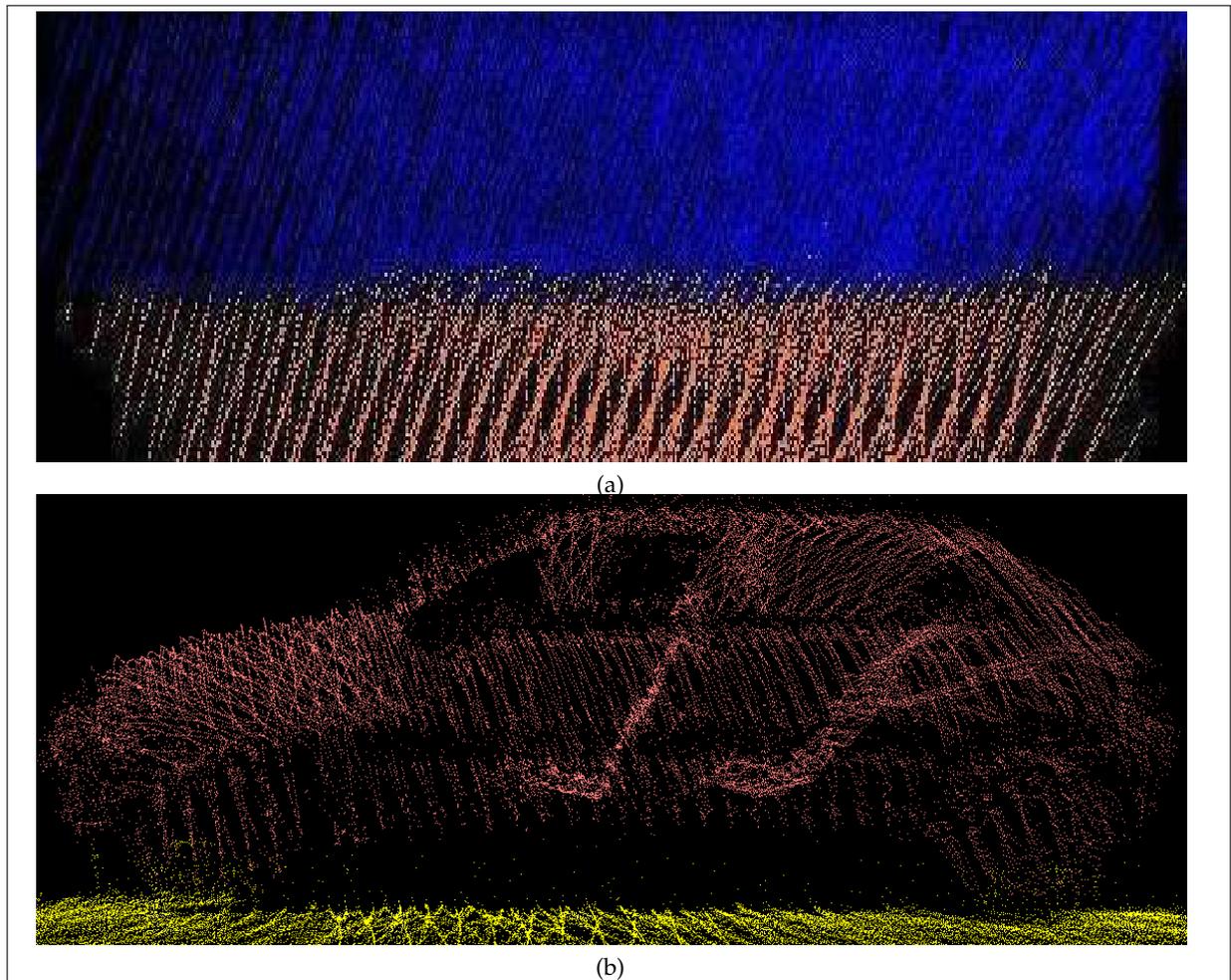


Figure 3.25: (a) & (b) show the misclassification of some 3D points at boundary regions of road surface with building and car respectively in the voxel-based method.

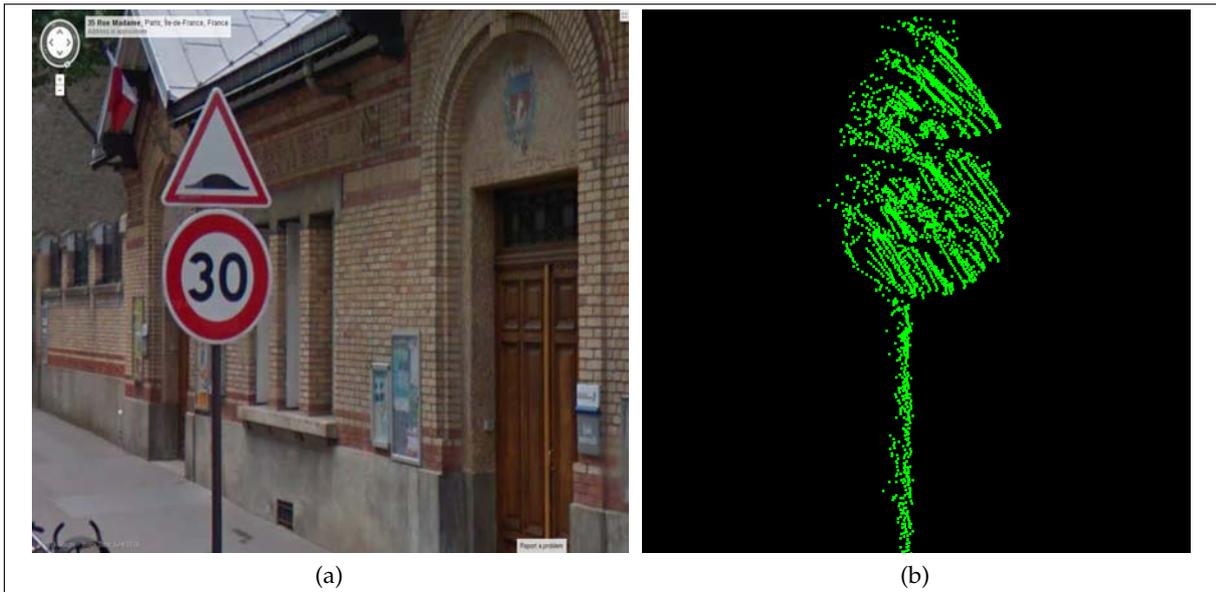


Figure 3.26: (a) Google Street View photo of the sign post with two traffic signs on Rue Madame. (b) Corresponding 3D points.

Also evident from the table, the interaction between classes is much more significant in the case of the morphological transformation method while on the other hand in the super-voxel based method the segmented objects belonging to a particular class instead of being distributed in other classes rather remain unclassified.

3.6.2.3 Segmentation Evaluation

In order to further assess the quality of segmentation, the ratio “ f ” of the total number of objects segmented by the applied method and the total number of segmented objects in the ground truth was plotted for each of the object class (see Fig. 3.27). A value of 1 represents overall best segmentation whereas a value greater than 1 denotes overall over-segmentation while a value less than 1 denotes overall under-segmentation. A value of 0 shows failure to detect or no detection.

The morphological transformation method seems to outperform the voxel-based method in terms of segmenting building and road surface. In the voxel-based segmentation method the road was over-segmented in 4 parts as they were found disconnected and also one of the buildings was over-segmented due to strong variation in color and reflectance intensity values (as shown in Fig. 3.28) while in case of another building small part found disjoint from the main building was segmented as a separate object (shown in Fig. 3.29).

However, compared to the morphological transformation method, the voxel-based method segments out cars and other road furniture better as apart from the adjacency of the 3D points it also uses color and reflectance intensity values in the segmentation phase. Figure 3.30 shows the segmentation results, for both methods, of some of the motorcycles parked in the scene.

For the voxel based method, we also find in one instance that two cars parked very close together having similar color and reflectance intensity values are segmented out as one single car.

The morphological transformation method, constrained by the generated profile, also fails to segment out 3D ground points directly under the motorcycles and car as shown in Fig. 3.31. These ground points are hence considered as part of the car (also expressed in the Table 3.12 i.e. value of 0.195). This is not an issue for the voxel-based method relying on local descriptors i.e. color, reflectance intensity and surface normals.

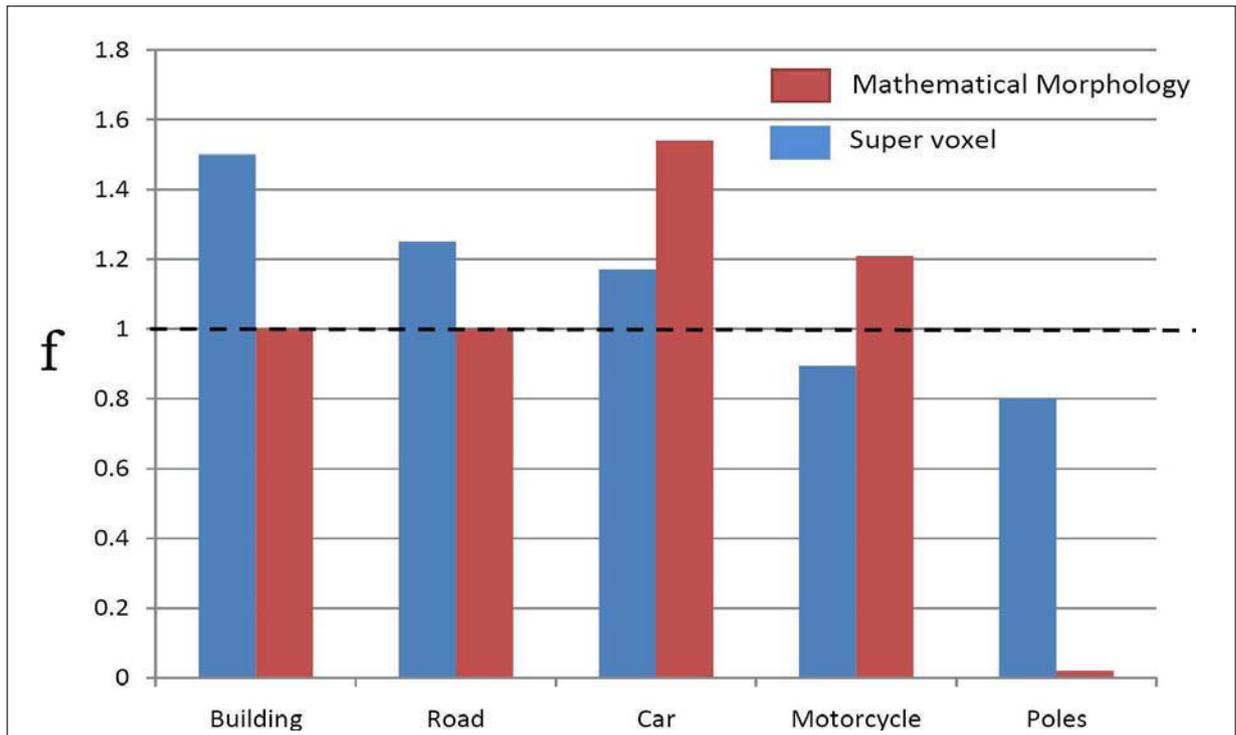


Figure 3.27: Overall segmentation quality of the two methods for different object classes is presented here. “ f ”, on the vertical axis, is the ratio of the total number of objects segmented by the applied method and the total number of segmented objects in the ground truth.

3.7 Conclusion

In this chapter, after discussing in detail the different existing techniques for segmenting and classifying urban environment, we have presented a super-voxel based segmentation and classification method for 3D urban scenes. For segmentation, a link-chain method is proposed. It is followed by the classification of objects using local descriptors and geometrical models. In order to evaluate our work, we have introduced a new evaluation metric which incorporates both segmentation and classification results. The results show an overall segmentation accuracy (OSACC) of 87% and an overall classification accuracy (OCACC) of about 90%. The results indicate that with good segmentation, a simplified classification method like the one proposed is sufficient.

Our study shows that the classification accuracy improves by reducing voxel size (with an appropriate value of c_D) but at the cost of processing time. Thus a choice of an optimal value, as discussed, is recommended. The study also demonstrates the importance of using laser reflectance intensity values along with RGB colors in the segmentation and classification of urban environment as they are more illumination invariant and more consistent. The method was also compared with other state-of-the-art-methods using different evaluation metrics to demonstrate its suitability and efficacy.

Segmentation and classification of the urban environment play an important role in effective 3D urban cartography. Whereas the proposed method can also be used as an add-on boost for other classification algorithms, the implication of this method for 3D urban cartography is discussed in the next chapter.

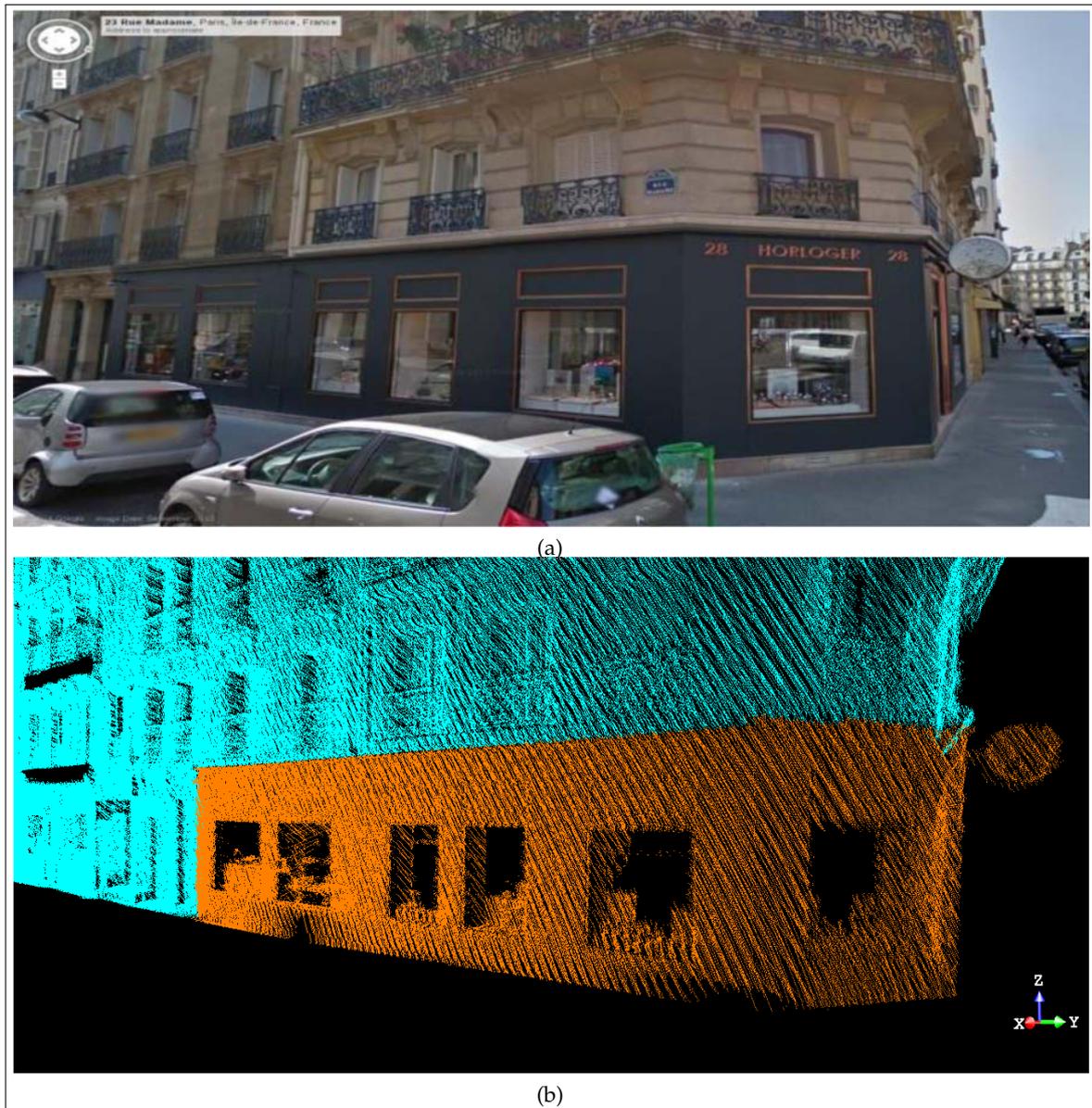


Figure 3.28: (a) Google Street View photo of the building on Rue Madame with a strong variation of paint color. (b) Segmentation results of the voxel-based method.

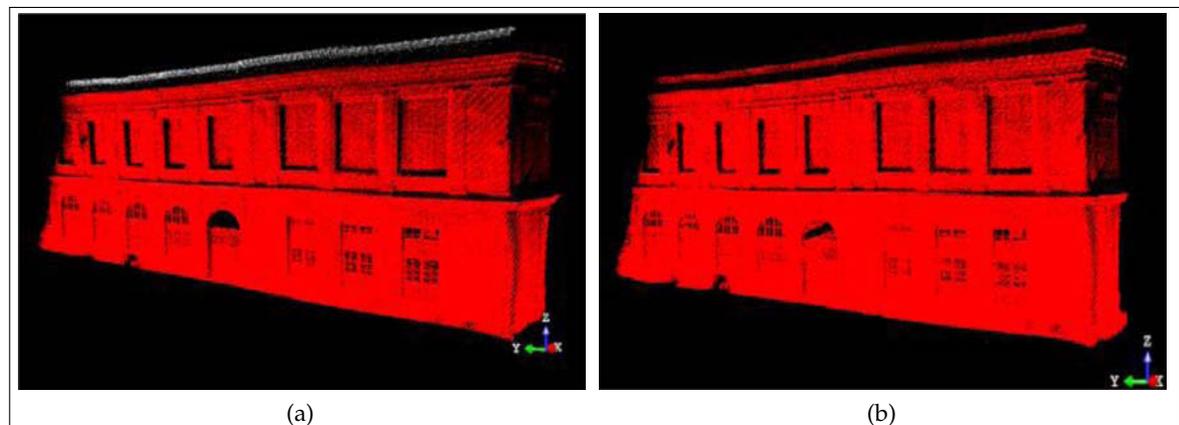


Figure 3.29: (a) & (b) show the segmentation results of a particular building in Rue Madame for voxel-based method and morphological transformation method respectively. In (a) it can be seen that part of the building that was disjoint was segmented as a separate object.

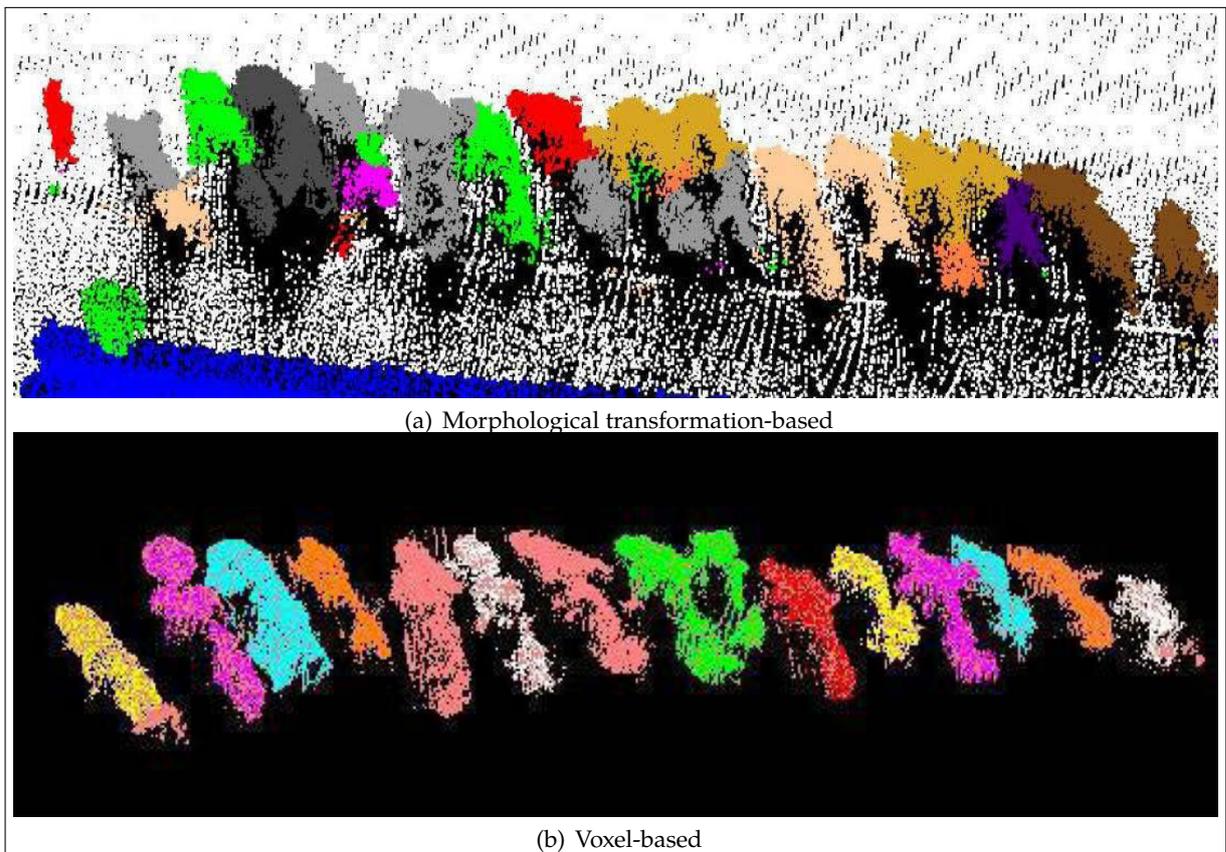


Figure 3.30: (a) & (b) show the segmentation results of some motorcycles parked in the street for morphological transformation and voxel-based method respectively.

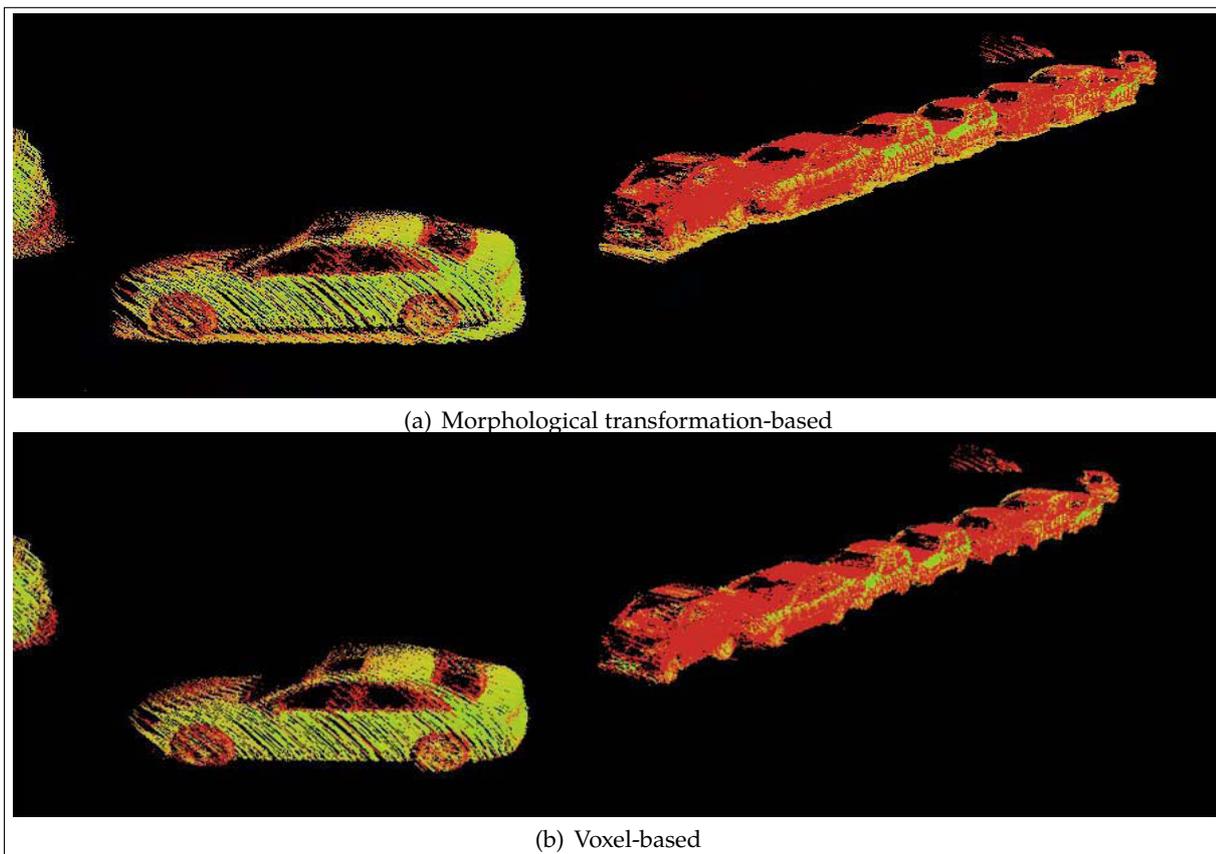


Figure 3.31: (a) & (b) show the segmentation results of some cars in the street for both morphological transformation and voxel-based method respectively. (a) shows some ground point directly underneath the cars, segmented as part of the cars.

3D URBAN CARTOGRAPHY INCORPORATING CLASSIFICATION AND INCREMENTAL UPDATING

In this chapter, we first give a detailed overview of the different state of the art techniques for 3D urban cartography. This is then followed by the presentation of a new method of automatic 3D urban cartography incorporating classification and incremental updating, exploiting the concept of multiple passages. In the proposed method, the 3D point clouds are first classified into 3 main object classes: Permanently Static, Temporarily Static and Mobile using a point matching technique. The Temporarily Static and Mobile objects are then removed from the 3D point clouds leaving behind a perforated 3D point cloud of the urban scene. These perforated 3D point clouds obtained from successive passages (in the same place) on different days and at different

times are then matched together to complete the 3D urban landscape. The changes occurring in the urban landscape over this period of time are detected and analyzed using cognitive functions of similarity and the resulting 3D cartography is progressively modified accordingly. Different specialized functions are also introduced, to help remove the different imperfections due to occlusions, misclassifications and different changes occurring in the environment over time, thus increasing the robustness of the method. The results, evaluated on real data, demonstrate that not only is the resulting 3D cartography accurate, containing only the exact permanent features free from imperfections, but the method is also suitable for handling large urban scenes.

4.1 Introduction

Automatic 3D urban cartography and modeling have become an issue of major interest in recent years, motivated by ambitious applications that aim to build realistic models, maps and digital representation of real cities and urban environment from data acquired by mobile terrestrial data acquisition techniques (as explained in Chapter 2). However, in urban environments, this task consisting in automatically generating accurate and reliable 3D cartography and models, without imperfections, using the data obtained from these hybrid terrestrial vehicles still remains a challenge. These imperfections mainly include missing features/regions due to occlusions caused by the presence of temporarily stationary and dynamic objects (pedestrians, cars, etc.) in the scenes [WTT+07], false features resulting from misclassifications of objects in the scene [VS04] and failure to effectively incorporate different changes occurring in the environment over time [CE09]. In this chapter, we present a new method for automatic 3D urban cartography that progressively removes these imperfections in an effective manner.

The work on handling such imperfections in automatic 3D urban cartography is heavily biased towards occlusion detection and management. Criminisi, Pérez, and Toyama [CPT04] present a technique of inpainting based on the patch exemplar-based method for completing occluded regions in images. Wang *et al.* [WJY+08] extended this approach to also infer depth from stereo pairs. Engels *et al.* [ETV+11] used a similar method for 3D data, but occlusions were found automatically using object-specific detectors, making it more suitable for larger datasets (see Fig. 4.1 for results). Several works have suggested that manual workload can be greatly reduced by using interactive methods that allow a user to quickly mark foreground and background areas, while exact segmentations are determined using graph cuts.

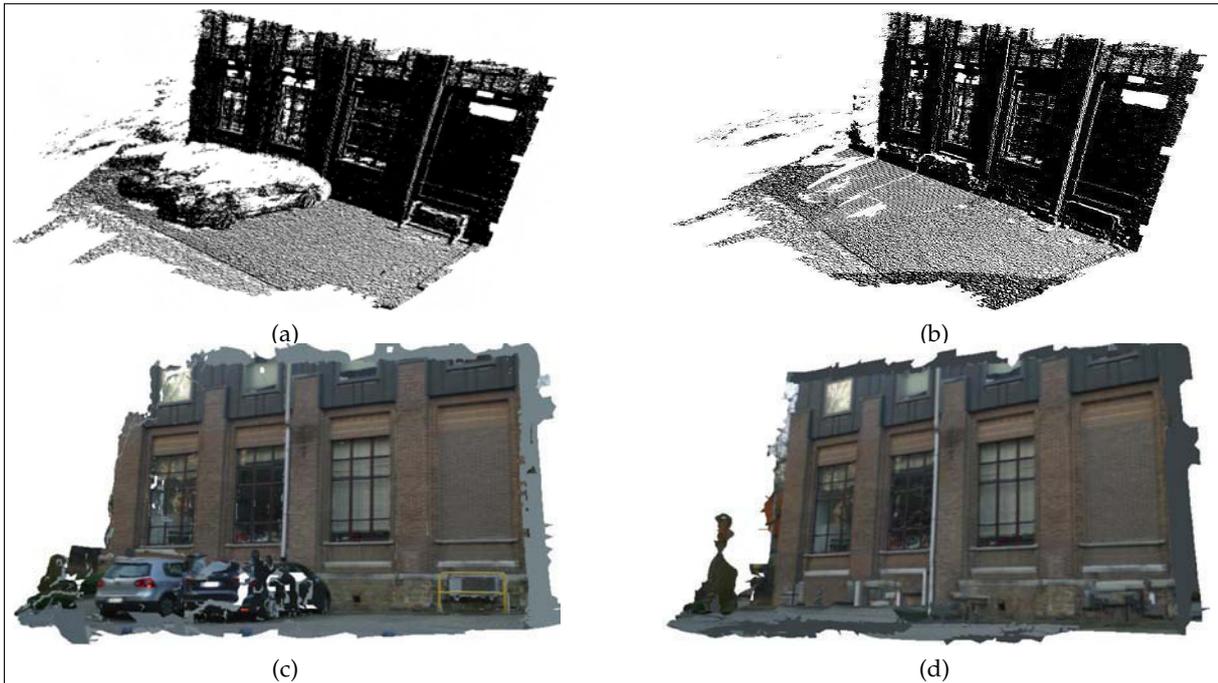


Figure 4.1: (a) Original point cloud. (b) The geometric occlusions have been removed by using object-specific detectors. (c) & (d) show reconstructed façade before and after occlusion removal respectively (source [ETV+11]).

In [BSF+09], the authors present a Patch-Match algorithm that allows for interactive rates for inpainting and reshuffling via simple user-defined constraints and efficient nearest neighbor search. In the context of building façades and urban reconstruction, increased contextual knowledge is available by assuming the planarity of the structure and the repetition of features, such as floors, windows, etc. Building models are reconstructed by detecting floors and estimating

building height in [KV07]. Occlusions are removed by cloning upper floors and propagating them downward. In [RKU05], multiple views and median fusion are used to remove most occlusions, requiring inpainting only for smaller regions. A method relying on LiDAR point cloud to find and remove occlusions by combining image fusion and inpainting is presented by Bénéitez, Denis, and Baillard [BDB10]. Xiao *et al.* [XFZ+09] semantically segmented street-side scenes into several classes, including vegetation and vehicles, but did not actively fill in missing data. Instead, they relied on the missing information being available from other views. In [OR05; Böh04], two methods for occlusion-free texture generation are presented, but a minimum of three overlapping images is necessary, which requires a high acquisition rate or a very slow driving speed in narrow streets. Some results are presented in Fig. 4.2.

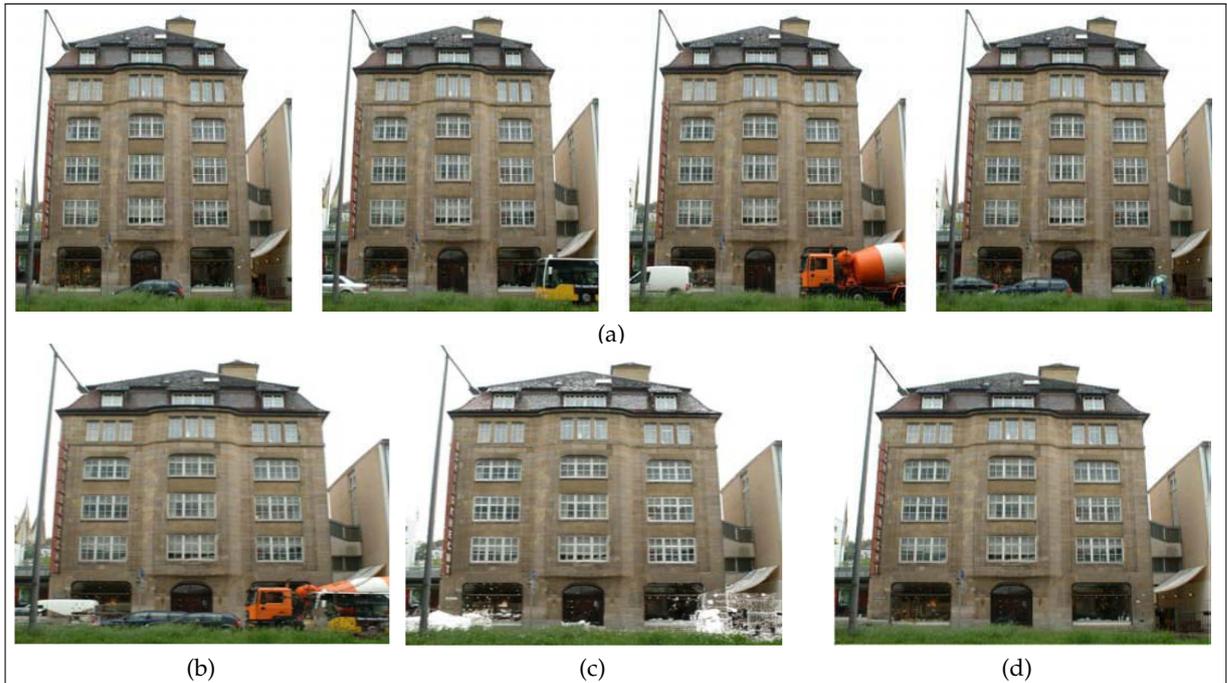


Figure 4.2: (a) Four images from a fixed viewpoint of a façade imaged from across a busy street. Traffic and pedestrians partially occlude the lower portion of the façade. (b) shows the combined pixels of all the images, containing all the occlusions. (c) pixels for which no unique decision could be made are marked white. (d) shows the final results after occlusion removal (source [Böh04]).

A method is discussed in [CPS10], which consists in aligning multiple scans from various viewpoints to ensure the 3D scene model completeness for complex and unstructured underground environments. A technique for extracting features from urban buildings by fusing camera and LiDAR data is discussed in [BH07a], but it does not specifically address this problem and relies on their simple geometrical modeling to complete partially occluded features. Frueh, Sammon, and Zakhor [FSZ04] propose a method in which the point cloud is used to generate a 3D mesh that is then classified as foreground or background. Large holes in the background layer, caused by occlusions from foreground layer objects, are then filled by planar or horizontal interpolation as shown in Fig. 4.3. However, such an approach may result in false features in case of insufficient repetitions or lack of symmetry [LZS+11]. In our work, we aim to resolve this problem by using a multi-session approach in which multiple scans of the same environment obtained on different days and at different times of the day are matched and used to complete the occluded regions.

Automatic detection of changes in urban environment for updating cartography and maps has lately gained some interest in the scientific community. Most of the proposed techniques detect changes in the urban environment from airborne data using Digital Surface Models (DSM),

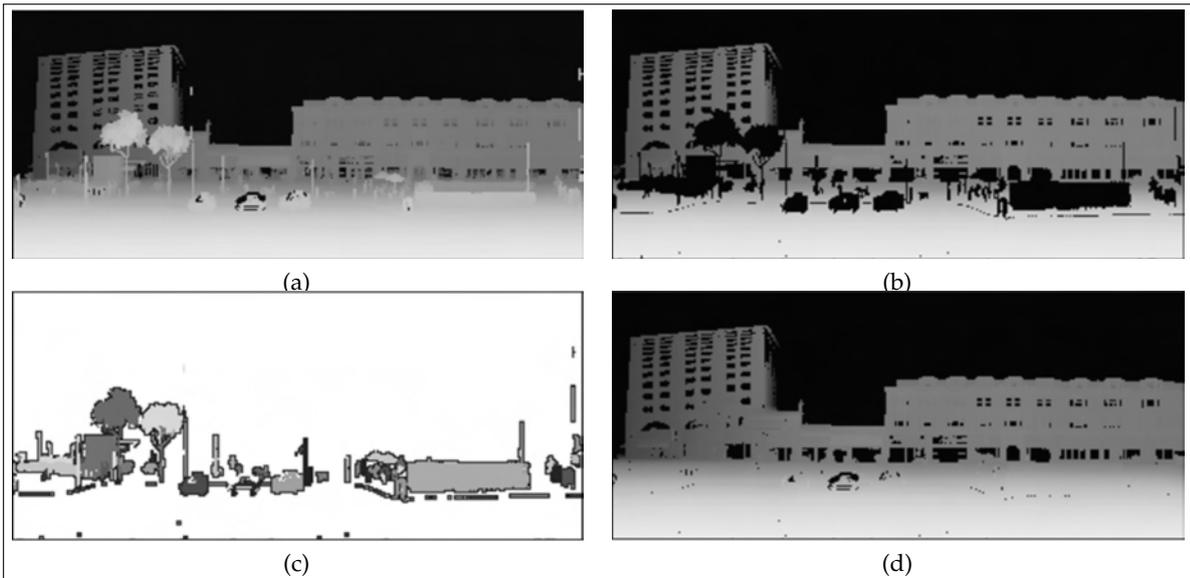


Figure 4.3: (a) Initial depth image. (b) shows the background layer segmentation after removing invalid scan points. (c) Foreground layer segmentation. (d) shows the results after filling holes due to occlusions (source [FSZ04]).

such as [MHK04; VMY04]. Vögtle and Steinle [VS04] propose a methodology for detecting changes in urban areas following disastrous events. Instead of solely computing the difference between the laser-based DSM, a region growing segmentation procedure is used to separate the objects and detect the buildings; only then, an object-based comparison is applied. However, this method remains susceptible to misclassifications. Bouziani, Goita, and He [BGH10] presented a knowledge-based change detection method for the detection of demolished and new buildings from very high resolution satellite images. Different object properties, including possible transitions and contextual relationships between object classes, were taken into account. Map data were used to determine processing parameters and to learn object properties. Matikainen *et al.* [MHA+10] also present a change detection method to update building maps, which compares buildings detected by a classification tree method with an existing building map. Some additional rules relying on existing map data are added to handle some likely misclassifications. Unlike these methods, in our work, we handle the 3D point cloud at the 3D grid level for change detection, instead of the object level, making it more robust against misclassifications.

Most of the work using terrestrial laser scans focuses on deformation analysis for designated objects. Change is detected by subtraction of a resampled set of the data [SWK+04] or adjustment to surface models, like planes [LP05] and cylinders [VGLP06]. In order to detect changes in large scenes, Hsiao *et al.* [HLY+04] combine terrestrial laser scanning and conventional surveying devices to acquire and register topographic data. The dataset is then transformed into a 2D grid and is compared with information obtained by the digitization of these existing topographic maps.

In [GMRM+05], changes are detected in the 3D Cartesian world, and the possibilities of scan comparison in point-to-point, point-to-model or model-to-model manners are discussed. The authors then use point-to-point comparison with some adaptations and make use of an octree as a data structure for accessing the 3D point cloud. Comparison is then carried out by using the Hausdorff distance as a measure for changes. Hyppä *et al.* [HJH+09] also use the method of point-to-point matching for detecting changes in 3D urban scenes. However, due to bad point registration, incorrect corresponding point pairs are likely to cause false change detection. Our method is more robust to such situations, as it not only analyzes the point cloud at the point level, but also at the grid level for this task. Zeibak and Filin [ZF07] extend this method by further characterizing the changes caused by occlusions as shown in Fig. 4.4. Kang

and Lu [KL11] not only detect changes in buildings in the urban environment, but also quantify the changed regions using a series of point cloud epochs over time and rebuilt building models. However, this work only focuses on disappearing changes. In our work, we not only detect, but also analyze both appearing and disappearing changes in the 3D urban scene by using 3D evidence grid and cognitive similarity functions.

To the best of our knowledge, no prior work has ever been presented that first effectively detects, then analyzes the changes by point-to-point comparison occurring in the 3D urban scene, in this manner, using terrestrial data and, eventually, updates the 3D cartography accordingly.

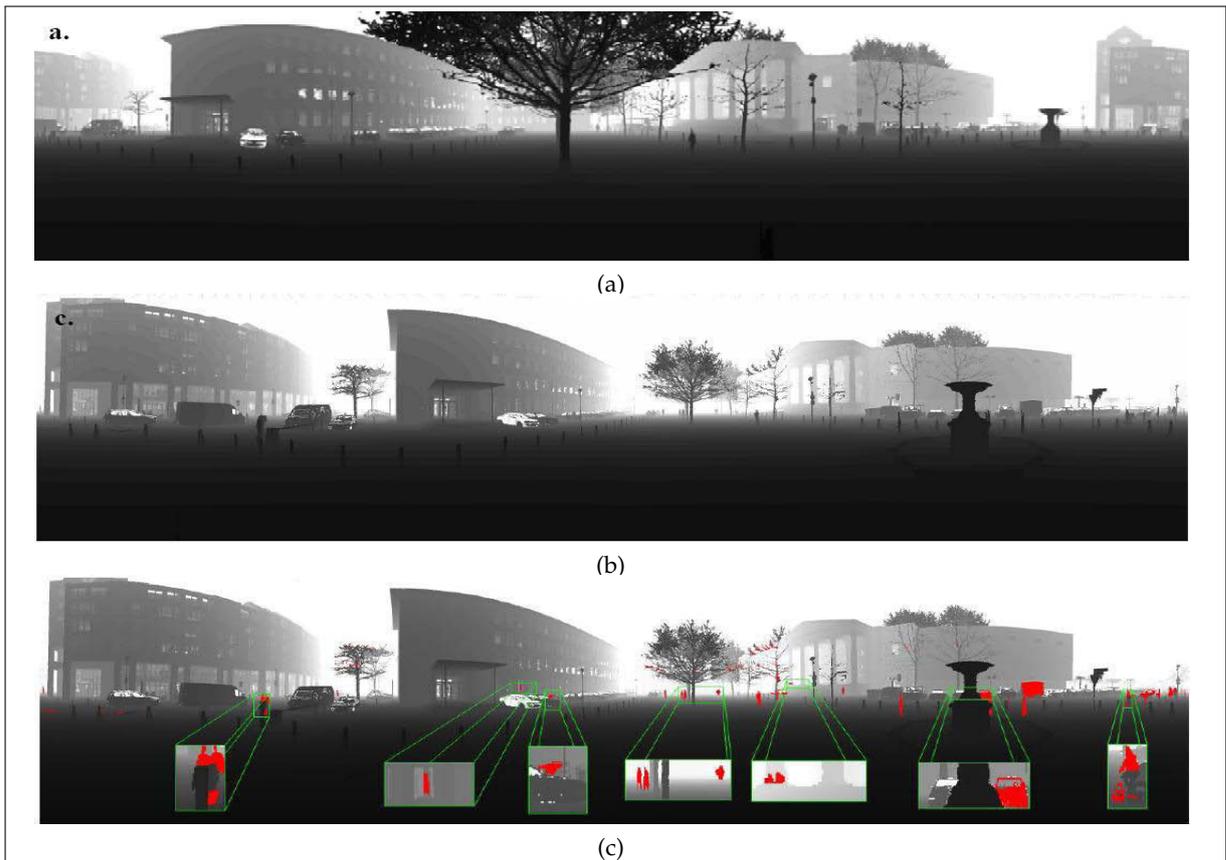


Figure 4.4: Occlusion removal by point-to-point comparison (a) shows the reference scan. (b) shows the analyzed scan in its original local frame. (c) presents the changes detected in the analyzed scan when compared to the reference one (source [ZF07]).

Another approach to detecting changes using mobile terrestrial LiDAR data is presented in [XVP13]. The method is based upon the physical scanning mechanism (scanning origin and geometry). The scan rays are reconstructed by the origin and the point and then the occupancy of space is derived from these rays. The Weighted Dempster-Shafer theory is used to combine the evidence from different rays and meanwhile minimizes the contradiction of evidence. The consistency between different scans is obtained by comparing the occupancy at points from reference scan with the fused occupancy of neighboring rays from the other scan. Although compared to other conventional methods, it helps to distinguish real changes from occlusions, etc., but the prerequisite of knowing the origin of each scanned point limits its use and applicability on different data sets. Some results are presented in Fig. 4.5.

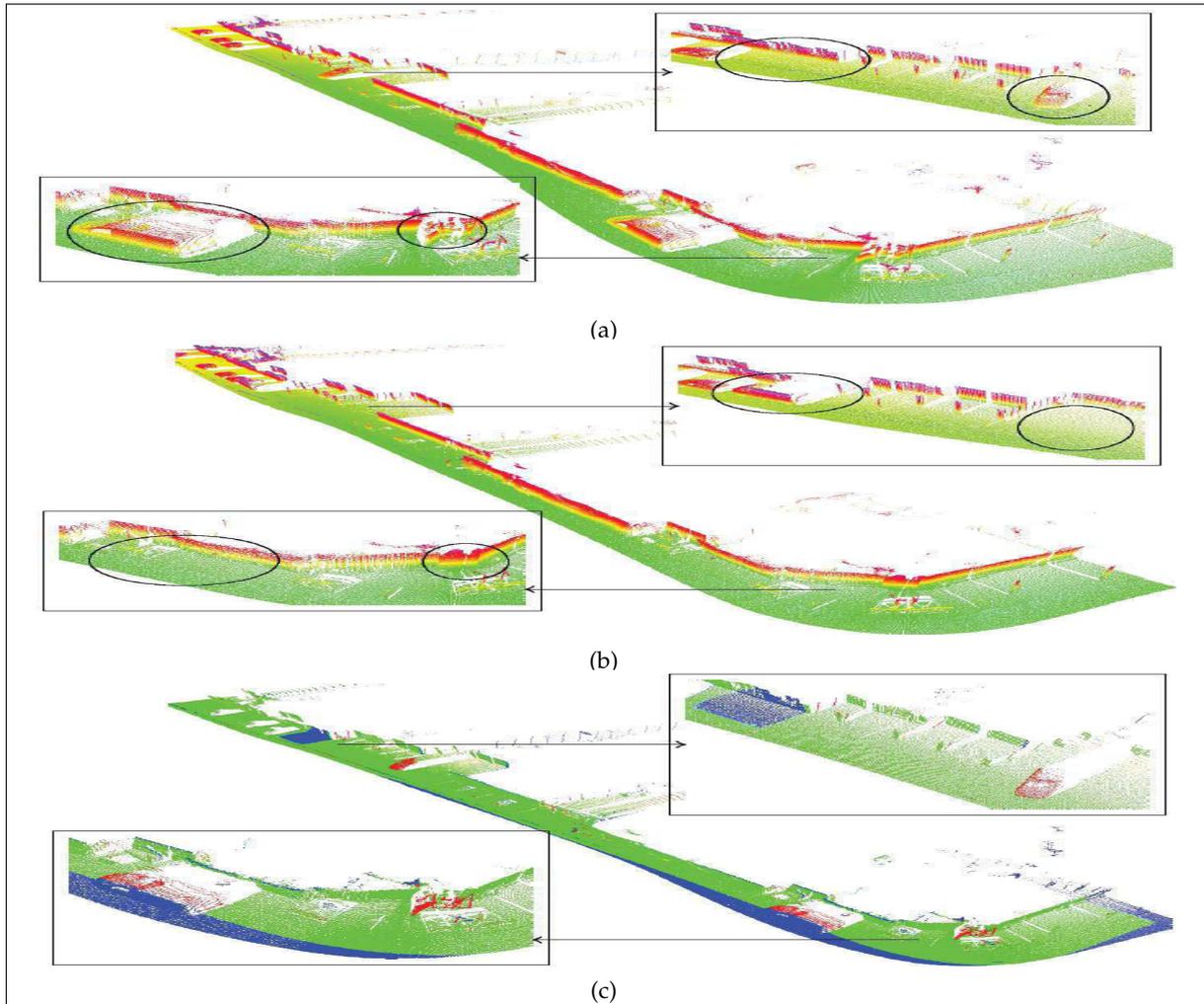


Figure 4.5: (a) & (b) show two scans of test sites (coloured by height). In scan one, there was a van and a man changing the poster on the billboard (lower right, site 1). None of these were present in scan two. On another site in scan one (upper left, site 2), there was a car coming out of the garage. In scan two, a van parked beside the pavement which was not there in scan one. (c) presents the changes detected in the two scan. Green color represents the points found consistent in the two scans while red represents the real changes. Blue colored points represent the changes due to other reasons such as occlusion, shadows, etc. (source [XVP13]).

Recently, the increasing demand for updated 3D maps and models for different popular applications has prompted frequent scanning of the same urban environment (i.e. multiple passages) using mobile terrestrial data acquisition vehicles. Although these applications utilize the recently acquired scans/data to simply refresh/update the resulting maps and models, they fail to exploit the redundant data available for further use, such as occlusion handling and accurate change detection and updating. Hence, our method successfully takes advantage of the already available redundant data, not only to detect, analyze and update changes in the urban environment, but also to complete occluded regions, so as to ensure that the resulting 3D cartography contains only the exact, actual and permanent features free from imperfections.

Another difficulty that arises when attempting to detect effective changes occurring in the urban environment is to differentiate the changes caused by the presence of transient objects i.e. frequently pedestrians and vehicles in the scene with the actual structural changes occurring in the cartography due to addition, removal or modification of certain parts of the environment. We note that this problem can also be mitigated via an application of change detection using different scales. Change detection involving the comparison of scans taken with large timescale

differences, and differences over a certain size can be classified as meaningful, structural changes. However, if we compare scans taken with a short time difference –i.e. a few days, or even a few hours– we can instead look for smaller changes, which can be interpreted as the presence or absence of transient objects. This method alone is not sufficient as it may not always be practical due to larger number of scans (i.e. both at smaller and longer intervals) required to converge towards the required results and also not very accurate (larger number of false positives and false negatives). This is why our method adopts a more pragmatic approach in which not only relying on this, we also use a classification method in each passage to extract these transient objects more effectively from the scene in order to not only obtain the desired results more quickly (in less number of passages) but also with higher accuracy.

The proposed method solves the above mentioned problems by handling the point cloud at three different levels using special dedicated functions: (1) the point level to accurately complete occluded regions; (2) the grid level to analyze and handle different changes occurring in the urban environment; and (3) the object level to accommodate for misclassifications and other unclassified objects. In this method, the 3D urban data obtained from mobile terrestrial LiDARs in each passage is directly geo-referenced. These geo-referenced 3D point clouds are segmented and, then, classified into three main object classes: Permanently Static, Temporarily Static and Mobile. The Temporarily Static and Mobile objects are then removed from the 3D point clouds, which are then merged together, leaving behind a perforated 3D point cloud of the urban scene. These perforated 3D point clouds obtained from different passages (in the same place) on different days and at different times are then matched together to complete the 3D urban landscape. Different changes occurring in the urban landscape over this period of time are studied using cognitive functions of similarity, and the resulting 3D cartography is progressively modified accordingly. An overview of the method is presented in Algorithm 2 and explained in the later sections.

4.2 3D Scan Registration

Different features and robust landmarks extracted from 3D images as points of interest and as references for image mapping and scan registration have commonly been used for different multi-session SLAM (Simultaneous Localization And Mapping) algorithms [MKC+12]. This approach works well in simple repetitive paths. However, some more complex situations can be found in urban environments, where the selected features/regions can be occluded. When the data-acquiring vehicle enters from different directions, then the path is not repetitive. As a result, the selected features/regions may not be readily visible. Thus, in order to cater to this problem, the method of direct geo-referencing of 3D LiDAR points is found most suitable in our case as discussed and explained in Section 2.3.2. The method uses integrated GPS/IMU data to directly orient laser data from its local reference frame to the mapping reference frame (WGS84). The advantage of using this method is that the transformation between the local laser reference frame and the mapping reference frame is known at any given moment (as long as the laser is synchronized), independently, if the laser is collecting data in a static mode or in kinematic mode. Thus, the laser can be used as a push-broom sensor sweeping the scene with profiles, while fixing the scan angles as the vehicle moves.

The data that we have used to evaluate our work are the dynamic data set of the 3D Urban Data Challenge 2011, which contains dynamic scenes from downtown Lexington, Kentucky, USA, obtained from the Vis Center’s (University of Kentucky) LiDAR Truck containing two Optech LiDAR sensor heads (high scan frequency up to 200 Hz), a GPS, an inertial measurement unit and a spherical digital camera, as shown in Fig. 4.6.

Algorithm 2 Automatic 3D Urban Cartography**Input:** 3D urban point clouds for passage number, n_p

- 1: Classify 3D urban point cloud obtained from each of the two sensors into six groups: {Road, Building, Car, Pole, Tree, Unclassified}
- 2: Compare the classified objects from the two point clouds to further classify them as: {Permanently Static, Temporarily Static, Mobile} and merge the two point clouds
- 3: Separate out Temporarily Static and Mobile objects, leaving behind a perforated 3D point cloud, $\mathbf{P}(n_p)$
- 4: Store Temporarily Static objects in a register, $\mathbf{R}(n_p)$

Sequential update function: {

- 5: Match and compare $\mathbf{P}(n_p)$ with $\mathbf{P}(n_p - 1)$ to fill in holes and complete 3D cartography }
- 6: Formulate 3D evidence grids for $\mathbf{P}(n_p)$ and $\mathbf{P}(n_p - 1)$ and compute 3D cell scores, C_S , for both
- 7: Calculate similarity scores, Sym and $ASyms$, and uncertainty measures for the 3D cells
- 8: Update similarity map, \mathbf{S}_{Map}

Automatic reset function: {

- 9: Compare the Sym , $ASyms$ and the uncertainty measures of the 3D cells in the \mathbf{S}_{Map} after n_{reset} number of passages
- 10: If there is low Sym and uncertainty measure, then reset those 3D cell(s) in $\mathbf{P}(n_p)$ with those in the recently acquired point cloud (perforated) }

Object update function: {

- 11: Compare Temporarily Static objects in $\mathbf{R}(n_p)$ with $\mathbf{R}(n_p - 1)$
- 12: Upgrade Temporarily Static objects in $\mathbf{R}(n_p)$ if they are repeated in n_{update} number of passages as permanently static and add in $\mathbf{P}(n_p)$ }
- 13: Delete 3D evidence grids for $\mathbf{P}(n_p)$ and $\mathbf{P}(n_p - 1)$
- 14: Store \mathbf{S}_{Map}
- 15: Update and store $\mathbf{R}(n_p)$
- 16: Store $\mathbf{P}(n_p)$
- 17: $\mathbf{R}(n_p - 1) \leftarrow \mathbf{R}(n_p)$ and $\mathbf{P}(n_p - 1) \leftarrow \mathbf{P}(n_p)$
- 18: **return** $\mathbf{P}(n_p)$

4.3 Classification of 3D Urban Environment

In urban environments, the quality of the data acquired by different mobile terrestrial data acquisition systems is widely hampered by the presence of temporary static and dynamic objects (pedestrians, cars, etc.) in the scene. As a result, there is a problem of occlusion of regions. Moving objects or certain temporary stationed objects (parked cars, traffic, pedestrian, etc.) present in the area hide certain zones of the urban landscape (buildings, road sides, etc.). Therefore, the first step for 3D urban cartography is to obtain the permanent cartography. This is achieved by removing/extracting the temporarily static and dynamic objects from the scene/point cloud, leaving behind only the permanent features.

In order to achieve this, we classify the urban environment into three main categories: Permanently Static objects, Temporarily Static objects and Mobile objects. In order to achieve this goal, the 3D point cloud is first segmented into objects, which are then classified into basic object classes using the method explained and presented in Chapter 3. Once classified into these basic classes, they are then grouped under one of the three mentioned categories. The method presents a super-voxel-based approach in which the 3D urban point cloud is first segmented into voxels and then converted into super-voxels by assigning properties to them based on the constituting 3D points. These are then clustered together using an efficient link-chain method to form objects. Using local descriptors and geometrical features, these objects are then classified into six main classes: {Road, Building, Car, Pole, Tree, Unclassified}, as discussed earlier (see Chapter 3). The

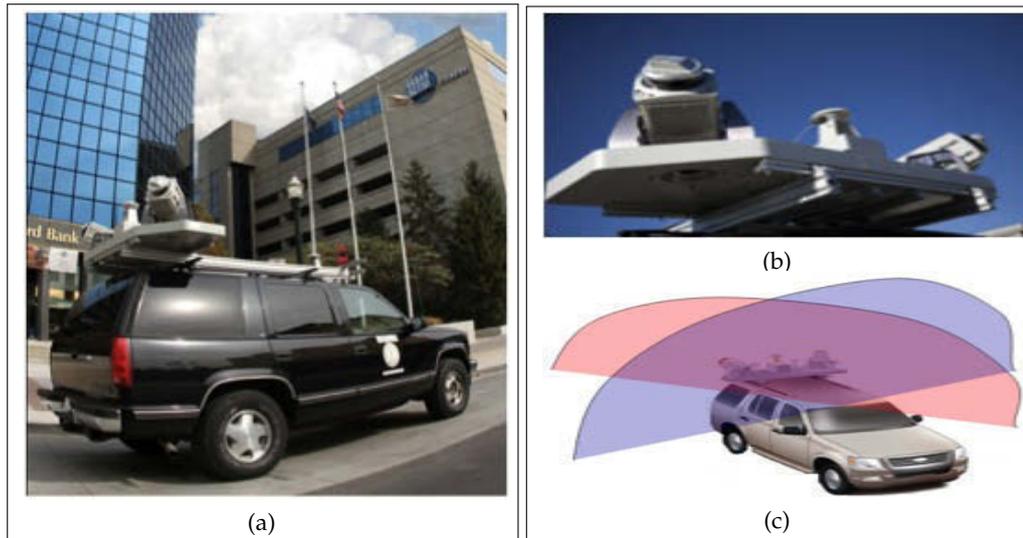


Figure 4.6: (a) The Vis Center’s LiDAR truck; (b) Optech LiDAR/GPS system along with IMU mounted on a rigid frame; (c) the different viewing angles of the mounted LiDAR systems.

salient features of this method which makes it suitable for this work are data reduction, efficiency and the simplicity of the approach. Some results of this method are again shown in Fig. 4.7.

The 3D point cloud obtained in a single passage from each of the two mounted LiDAR sensors is divided into the six object classes using this method. This step may be sufficient for characterizing static objects, but not for dynamic ones. In order to separate out moving objects from these classified objects, the two 3D point clouds obtained in the same passage are matched and merged together. This not only helps in distinguishing mobile objects, but also completes the 3D cartography (building façades, etc.) due to the different viewing angles of the two sensors (from now on referred to as S – 01 and S – 02), as shown in Fig. 4.6(b). This configuration of LiDAR is very common for this type of sensor and is used for acquiring detailed 3D data along both sides of the road. In addition to this, we exploit this difference in viewing angle to identify

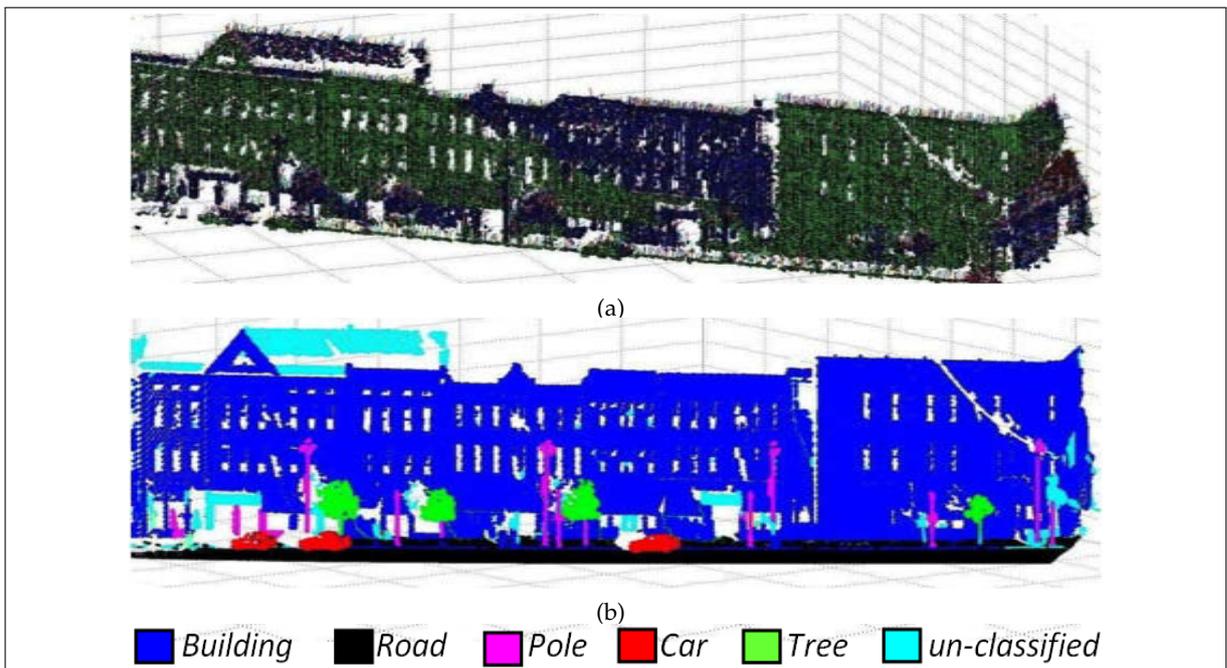


Figure 4.7: (a) Voxel-based segmentation of a particular scene; (b) classified 3D points.

mobile objects. Due to this difference in viewing angle, these sensors see the same point in the 3D scene with a slight time difference. We use this time difference to infer whether an object is static or moving by comparing the position of these classified objects at these two different times corresponding to the two different point clouds. Those objects are considered static if the constituting 3D points of the objects have the same position in the two point clouds, while those with different positions are considered mobile (see Fig. 4.8). As this time difference is very small, object/point association or matching in the two point clouds is not an issue. Simple matching based on the constituting points, color and intensity of all the objects in the two point clouds is sufficient. Let P and Q be the two point clouds obtained from sensor, $S - 01$ and $S - 02$, respectively, then an object, n , denoted by Obj_n in the two point clouds is given as P_{Obj_n} and Q_{Obj_n} . This Obj_n is considered static if and only if the following three conditions are satisfied:

$$\frac{\text{Card}(PQ_{Obj_n})}{\text{Card}(\min[P_{Obj_n}, Q_{Obj_n}])} \times 100 \geq w_p \quad (4.1)$$

$$\left| P_{Obj_nR,G,B} - Q_{Obj_nR,G,B} \right| \leq 3\sqrt{w_c} \quad (4.2)$$

$$\left| P_{Obj_nI} - Q_{Obj_nI} \right| \leq 3\sqrt{w_I} \quad (4.3)$$

where $PQ_{Obj_n} = P_{Obj_nX,Y,Z} \cap Q_{Obj_nX,Y,Z}$. PQ_{Obj_n} is the set containing the matched points obtained by point-wise intersection of the 3D points in the two sets if and only if the difference in the distance along the x -, y - and z -axes between two points in the $P_{Obj_nX,Y,Z}$ and $Q_{Obj_nX,Y,Z}$ is $\leq 2 \times P_e$. Here, P_e is taken as the measurement accuracy of the LiDAR sensor (the value can be obtained from the data sheet) and Card is the cardinal function. $P_{Obj_nX,Y,Z}$ and $Q_{Obj_nX,Y,Z}$ are the sets of the 3D coordinates of the points of the object, while $P_{Obj_nR,G,B}$ and $Q_{Obj_nR,G,B}$ are the mean R, G and B values of the object in P and Q point clouds, respectively. P_{Obj_nI} and Q_{Obj_nI} are the mean laser reflectance intensity values in P and Q point clouds, respectively. w_p is the matching weight equal to the allowable percentage of the object points whose position matches the two point clouds. w_c is the color weight equal to the maximum variance of R, G and B values for P_{Obj_n} and Q_{Obj_n} . w_I is the intensity weight equal to the maximum variance of intensity values for P_{Obj_n} and Q_{Obj_n} . It should be noted here that based on the basic characteristics of the classified objects (like roads, buildings, trees and poles that cannot move), we only compare (using Equations (4.1)–(4.3)) the objects classified as cars and unclassified to determine which of them are temporarily static and which ones are mobile. Here, it is observed that Equations (4.2) and (4.3) along with Equation (4.1) help to increase the robustness of the method in different precarious situations that often arise in urban environments; for example, two different vehicles coming from the opposite direction in the neighboring lane could be detected at the same place, i.e. Equation (4.1) would suggest it is just one static object, whereas Equations (4.2) and (4.3) would then differentiate the two objects (classifying them as two dynamic objects) based on their different appearance i.e. RGB color and laser reflectance values). Moreover, any apparition or disappearance of an object belonging to one of these two classes in any of the two point clouds is automatically considered mobile (i.e. a moving car detected by one of the sensors, which moves out of the field of view of the second sensor in the short time delay, or a moving car, which has just entered the scene after the first sensor has already scanned). Similarly, extending this basic reasoning, we infer the objects, classified as buildings, roads, trees and poles, as Permanently Static, whereas cars and pedestrians can be either Temporarily Static or Mobile. The classification chart as per our inference is presented in Table 4.1, whereas some of the results of this method are shown in Fig. 4.9.

Once the objects present in the urban scene are classified into these three main classes, in each passage, the objects classified as Temporarily Static and Mobile are extracted out from the scene, leaving behind a perforated point cloud for each passage, as shown in Fig. 4.10. This perforation is due to occlusions caused by the Temporarily Static and Mobile objects in the scene. These perforated 3D images/point clouds of the same place obtained via a single passage on

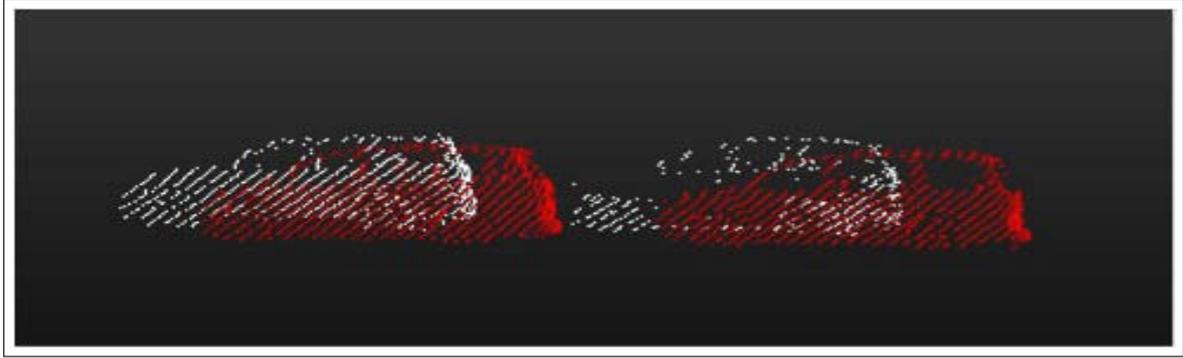


Figure 4.8: 3D points of moving vehicles in the urban scene. In red, the 3D points acquired from S – 01, and in white, the 3D points acquired from S – 02.

Object Type	Permanently Static	Temporarily Static	Mobile
Road	x		
Building	x		
Tree	x		
Pole	x		
Car		x	x
Pedestrian		x	x
Unclassified		x	x

Table 4.1: Object classification chart.

different days and at different times are then combined together to complete the 3D cartography, as discussed in the following section.

The unclassified objects found to be static are considered Temporarily Static by default, because all the objects classified as Temporarily Static are compared in the update phase. If the same objects belonging to this class are found in repeated passages, they are then upgraded as Permanently Static objects in the object update phase discussed in Section 4.5.4 and are considered part of the 3D cartography.

4.4 Combining Multiple Views for Scene Completion

In an attempt to fill some of the perforations and complete some of the occluded regions the two 3D point clouds obtained in the same passage from the two LiDAR sensors are matched and merged together. The different viewing angle of the two sensors S – 01 and S – 02 helps completes the 3D cartography (building façades, etc.) but also fill some of the holes (caused due to extraction of occluding objects) due to static objects present in the scene (see Fig. 4.11) but also those caused by the mobile objects as due to this difference in viewing angle these sensors see the same point in the 3D scene with a slight time difference. This time difference allows a moving object to move such that the occluded portion of the cartography due to this object in S – 01 becomes evident in S – 02 while the occluded portion in S – 02 was evident in S – 01. Thus, combining the two 3D point clouds help fill up such holes in an attempt to eliminate the effect of the moving object in the scene as shown in Fig. 4.11.

The two 3D point clouds are matched and merged together using the new matching method introduced in Section 4.5.4 to form a unified perforated 3D point cloud.

This method of fusing multiple view data to fill in holes may be effective in simple cases but in more complex scenarios offered in the urban environment relying on multiple views alone does not entirely solve the problem of occlusions and a number of holes remain due to several

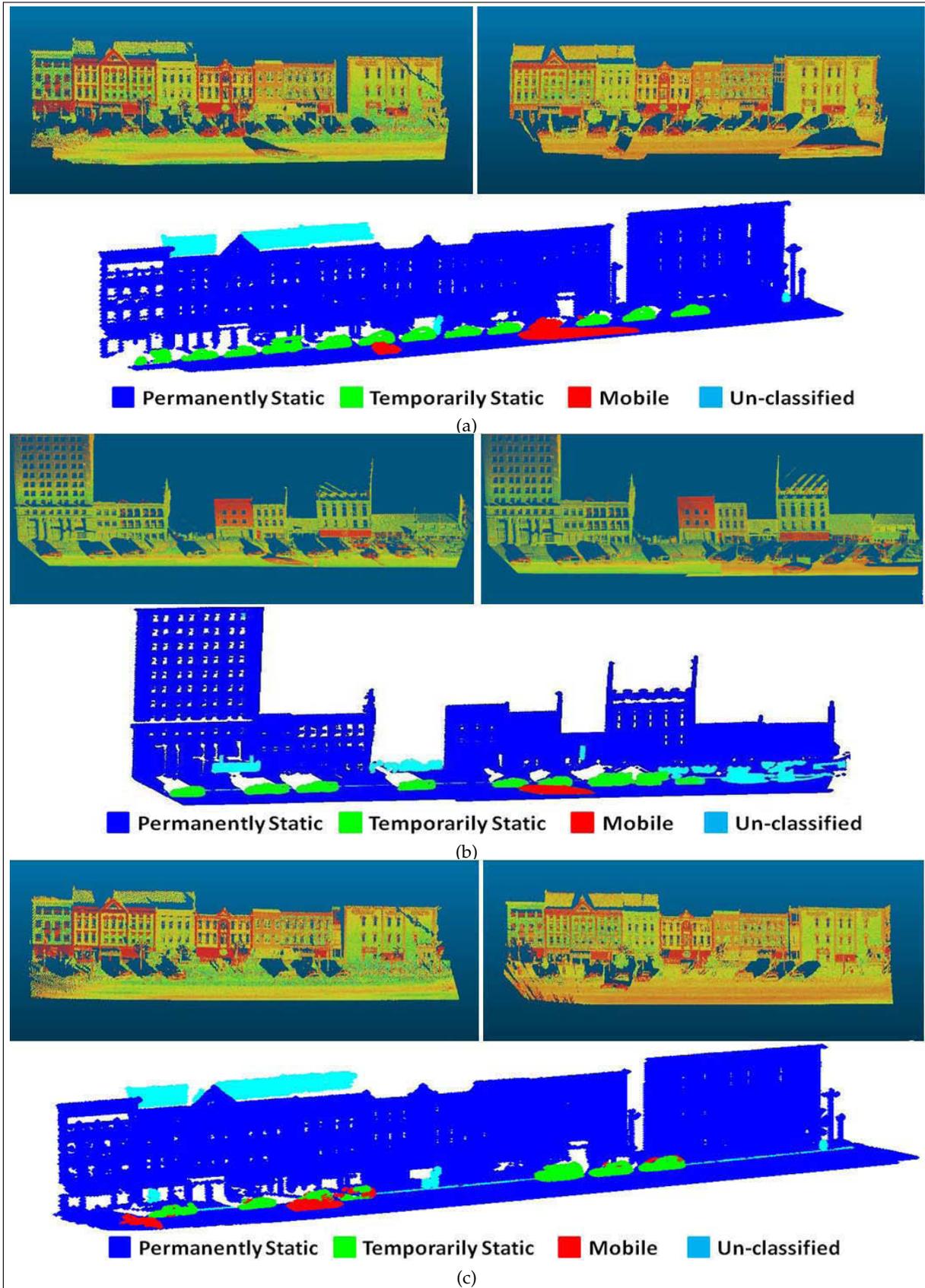


Figure 4.9: Characterization of the classified objects into three main classes. Raw LiDAR data obtained from S – 01 and S – 02 are presented in Top left and right while characterization results at the bottom figure respectively for each (a), (b) & (c).

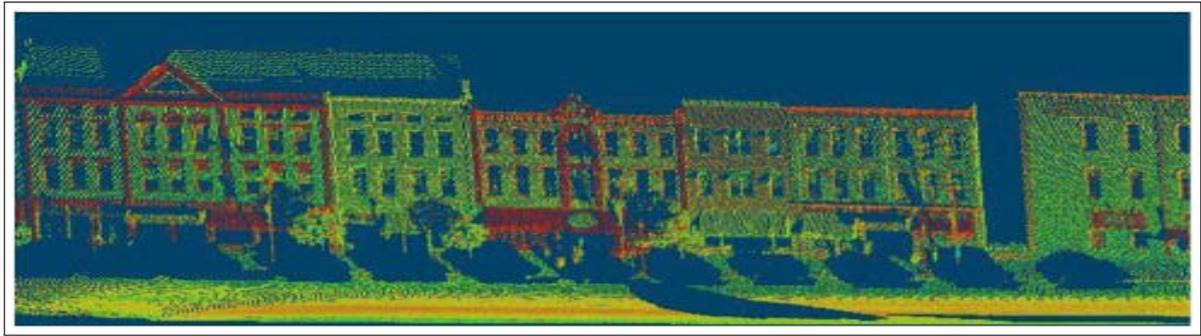


Figure 4.10: The objects classified as Temporarily Static and Mobile are extracted out from the scene, leaving behind a perforated point cloud for each passage.

blind spots as shown in Fig. 4.11.

In Fig. 4.11(c) (classic case of occlusion caused by a temporary static object, i.e. a parked car, present in the scene) it can be seen that even after combining the data from multiple views, some regions of the cartography remain occluded such as parts of shop's wall, lamp post, tree and road side.

Similarly for a temporary dynamic object, i.e. a moving car present in the urban scene, combining the data from multiple views does not fully complete all the occluded regions as seen in Fig. 4.11(f) (some parts of shop's wall and road side remain occluded). So, this technique of combining multiple views is helpful for completion of certain occluded regions but does not solve the problem completely.

Thus, in order to remove these imperfections and complete the remaining exact and actual occluded regions, we use a new method of 3D urban cartography exploiting the concept of multiple passages described in the next section.

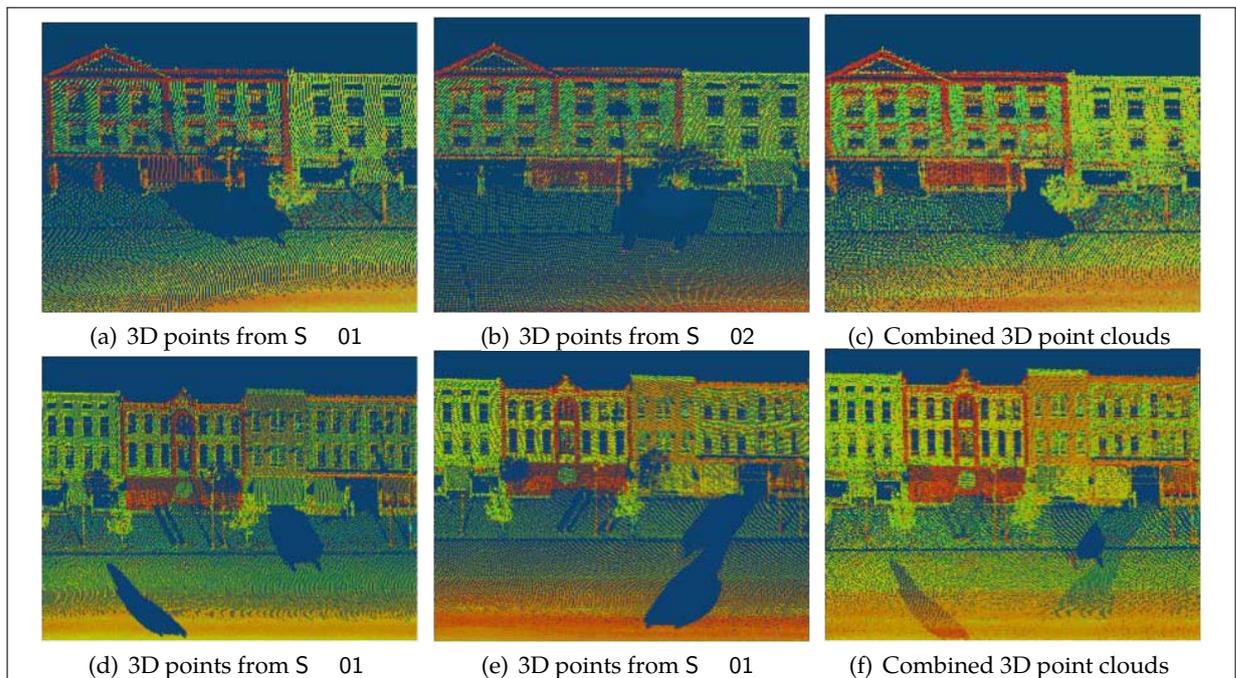


Figure 4.11: In (a), (b) & (c) the filling of holes due to the presence of a temporarily static (parked) car by multiple view combination is presented. In (d), (e) & (f) the filling of holes due to the presence of a temporary (dynamic) car by combining multiple views, is presented. It can be seen in (c) & (f) that even after combining data from multiple views, some holes/missing regions still remain due to blind spots.

4.5 3D Cartography and Removal of Imperfections Exploiting Multiple Passages

The perforated 3D point clouds obtained in subsequent passages of any particular place (on different days and/or at different times) are combined together to fill in the occluded regions and complete the 3D urban cartography. The perforated point cloud is first mapped onto a 3D evidence grid, and the corresponding 3D cell scores are calculated. A similarity map is generated in subsequent passages. Based on this similarity map and the associated uncertainty, different changes occurring in the urban environment are analyzed and appropriate actions are taken to cater for these changes. The details are provided below.

4.5.1 3D Evidence Grid Formulation

As the 3D point cloud is directly geo-referenced, the use of an occupancy grid for comparison in subsequent passages, as compared to the elaborate graph theory, is more logical and practical. The perforated 3D point cloud obtained in each passage is mapped onto a 3D evidence grid, as shown in Fig. 4.12. Each 3D cell or voxel of this grid occupies a volume, L^3 , and is assigned a cell score, C_S , based on certain attributes of the constituting 3D points using Equation (4.4). These attributes include the ratio of occupied volume V_{Occ} , surface normal components along x -, y - and z -axes, $N_{X,Y,Z}$, mean laser reflectance intensity and mean RGB color values i.e. R_I , R_c ($c \in \{\bar{R}, \bar{G}, \bar{B}\}$), respectively, and the number of the current passage, n_p . The number of current passage is added in calculation of the cell score to allow the comparison of/with empty 3D cells (when using the Minkowski distance is the Tversky's function as explained later on). However, it has no effect on the cell score as it just contributes a very small constant bias which helps make the method a bit more conservative. The normalized values of these attributes are used to compute the cell score which, as a result, is unitless:

$$C_S^j = \frac{w_{Occ}V_{Occ}^j + w_N N_{X,Y,Z}^j + w_{RI}R_I^j + w_{Rc}R_c^j + w_{np}n_p}{w_{Occ} + w_N + w_{RI} + w_{Rc} + w_{np}} \quad (4.4)$$

where j is the number of cells. $w_{Occ} = 1$ and $w_N = 0.5$ are occupation weight and normal weight, while $w_{RI} = 0.25$ and $w_{Rc} = 0.125$ are intensity and color weight, respectively. $w_{np} = 0.0625$ is the passage number weight. The values of these weights are chosen to bias the score more towards occupancy (magnitude and orientation) and less towards representation (intensity and color), as the former is more invariant in the urban environment and also keeps our approach closer to the classical occupancy grid method [SG12]. These 3D evidence grids are constructed in each passage for both the previous ($\mathbf{P}(n_p - 1)$) and latest perforated 3D point cloud acquired and are used to formulate and update the similarity map along with the associated uncertainty. They are then deleted at the end of the process in each passage (Algorithm 2). This makes this approach most suitable for analyzing large mapping areas.

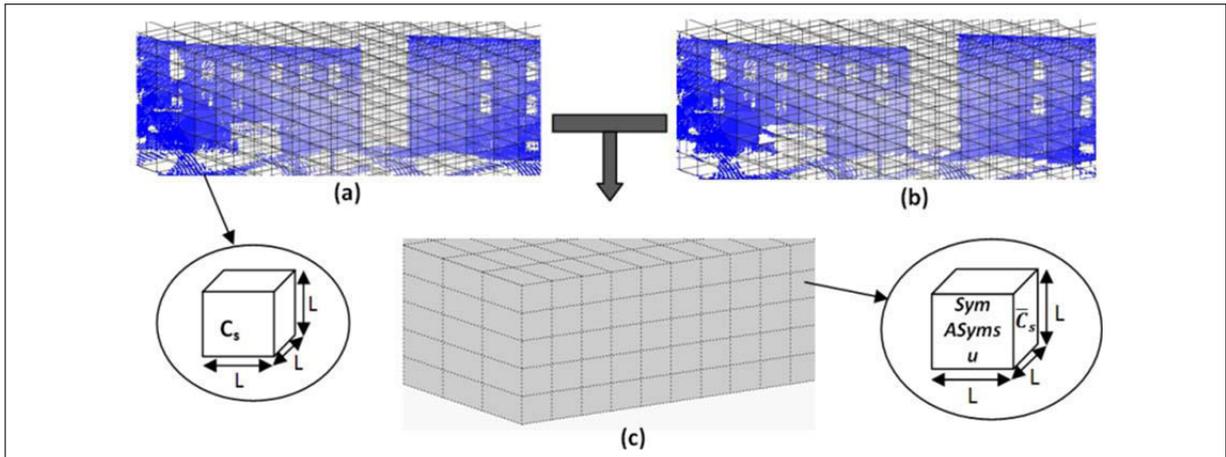


Figure 4.12: Formulation of 3D evidence grids and similarity map. (a) & (b) show the 3D point clouds ($P(n_p)$ and $P(n_p - 1)$) mapped onto an evidence grid of cell size, L^3 , respectively. In (c), the similarity map obtained from the two evidence grids.

4.5.2 Similarity Map Construction

The 3D evidence grid in successive passages is compared to obtain a similarity map in each passage. Instead of finding the overall graph/grid similarity, we are more interested in measuring the similarity of each cell, as this indicates exactly which part of the 3D cartography has changed. Currently, many distances have been developed to compare two objects (in this case, cell) according to the type of attributes, such as χ^2 or Mahalanobis. However, when working with real values, the most widely used (and simplest) metric is the Minkowski measure, dp :

$$dp(x, y) = \left[\sum_{i=1}^k W_i |x_i - y_i|^p \right]^{\frac{1}{p}} \quad \text{with } p > 0 \quad (4.5)$$

In this measure, x_i and y_i are the values of the i^{th} attribute describing the individuals, x and y . W_i is the numerical weight correlated with this attribute. k is the total number of attributes. In order to transform Minkowski distance Equation (4.5) into a similarity measure, sp , a value, D_i , is introduced, which corresponds to the difference between the upper and the lower bounds of the range of the i^{th} attribute:

$$sp(x, y) = \left[\sum_{i=1}^k W_i \frac{D_i - |x_i - y_i|^p}{D_i} \right]^{\frac{1}{p}} \quad \text{with } p > 0 \quad (4.6)$$

This similarity function may provide a measure indicating the amount of changes occurring in a 3D grid cell in subsequent passages, but it remains silent on the type of change taking place. Now, this information could be useful when deciding how to handle these changes in the 3D cartography. Thus, in order to get more insight into the type of changes, we incorporate the notion of distance between individuals or objects studied in the cognitive sciences. For this purpose, we use the method proposed by Tversky [Tve77] to evaluate the degree of similarity, $S_{x,y}$, between two individuals, x and y , respectively, described by a set of attributes, A and B , by combining the four terms, $A \cup B$, $A \cap B$, $A - B$ and $B - A$, into the formula (as explained in Fig. 4.13):

$$S_{x,y} = \frac{f(A \cap B)}{f(A \cup B) + \alpha f(A - B) + \beta f(B - A)} \quad (4.7)$$

As we want to compare a pair of individuals (in this case, cells in successive passages) described by a set of numerical attributes, we combine the definitions proposed by Tversky and

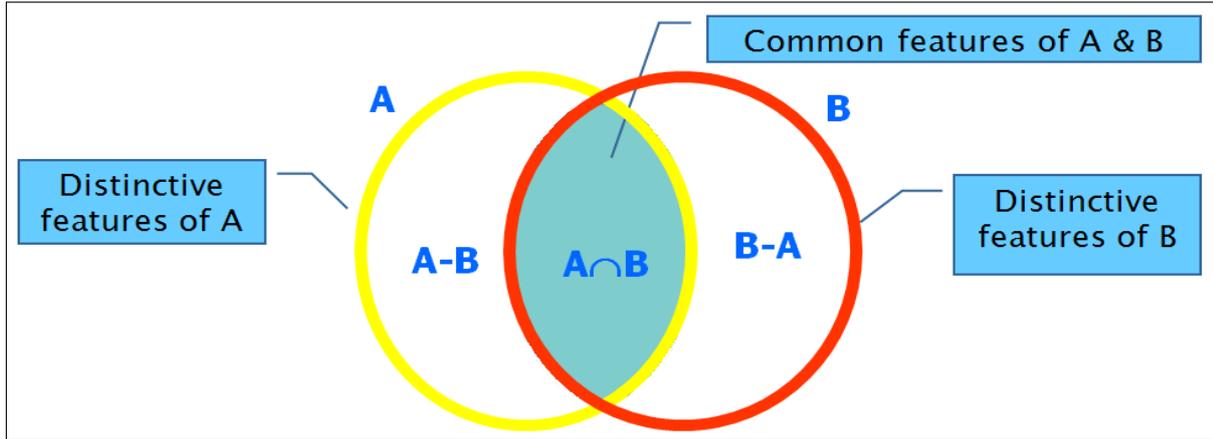


Figure 4.13: Representation of two objects that each contains its own unique features and also contains common features. An important aspect of Tversky's model is that similarity depends not only on the proportion of features common to the two objects but also on their unique features.

Minkowski. In these measures, we use Tversky's model to compare the two sets of attributes describing the individuals; the function, f , of this model is the Minkowski's formula, as rewritten in Equation (4.6). The parameter, p , of this formula equals one, since in Tversky's model, the function, f , corresponds to a linear combination of the features. Now, depending upon the way the parameters, α and β , are instantiated, different kinds of cognitive models of similarity can be expressed. By instantiating $\alpha = \beta = 0$, we obtain the symmetric similarity measure, Sym , while by instantiating $\alpha = 0$ and $\beta = -1$, we obtain the asymmetric similarity measure, $ASym$. Now, we use these values to fill up similarity map, S_{Map} , as shown in Fig. 4.12; the values of $ASym$ allow us to evaluate the degree of inclusion between the first cell (reference) into the second cell (target). Hence, with the attributes (along with their corresponding weights) assigned to these cells (discussed in Section 4.5.1), the value of $ASym$ can be used to assume the type of changes occurring in the 3D grid cell, as summarized in Table 4.2, where x and y are the same 3D grid cell in different passages. These values of Sym and $ASym$ s help in ascertaining the most suitable function required for that particular 3D grid cell. Condition 1 is automatically handled by the default sequential update function, whereas for Condition 2 and 3, the automatic reset function is called into action.

Table 4.2: Type of changes.

#	Condition	Possible Assumption
1	$ASym_{x,y} < ASym_{y,x}$	Addition of structure (could be new construction)
2	$ASym_{x,y} > ASym_{y,x}$	Removal of structure (could be demolition)
3	$ASym_{x,y} = ASym_{y,x}$	Modification of structure (depending on the value of Sym)

The similarity map, S_{Map} , is updated in each passage, and only the different cells (with $Sym < Similarity_{threshold}$) along with their associated uncertainty values are kept in the map, whereas the remaining cells considered as identical cells (with a high level of similarity) are deleted from the map. Hence, this not only reduces the size of the map progressively in subsequent passages, but also avoids possible storage memory issues for large point clouds in the case of large mapping areas. See Section 4.6.2 (Fig. 4.23) for more details.

4.5.3 Associated Uncertainty

Let the cell scores of a particular cell, j , in n number of passages, $C_{S_1}^j, C_{S_2}^j, \dots, C_{S_n}^j$, be an i.i.d. (independent and identically distributed) sequence of random variables; then the n^{th} sample variance $s_n^{j^2}$ is given as:

$$s_n^{j^2} = \frac{\sum_{k=1}^n (C_{S_k}^j - \bar{C}_{S_n}^j)^2}{n-1}$$

then, by adding and subtracting $\bar{C}_{S_{n-1}}^j$:

$$s_n^{j^2} = \frac{1}{n-1} \left[\sum_{k=1}^n (C_{S_k}^j - \bar{C}_{S_{n-1}}^j + \bar{C}_{S_{n-1}}^j - \bar{C}_{S_n}^j)^2 \right]$$

expanding and solving this to get:

$$\begin{aligned} s_n^{j^2} &= \frac{1}{n-1} \left[(n-2)s_{n-1}^{j^2} + (n-1)(\bar{C}_{S_{n-1}}^j - \bar{C}_{S_n}^j)^2 \right. \\ &\quad \left. + 2 \sum_{k=1}^{n-1} [(C_{S_k}^j - \bar{C}_{S_{n-1}}^j)(\bar{C}_{S_{n-1}}^j - \bar{C}_{S_n}^j)] + (C_{S_n}^j - \bar{C}_{S_n}^j)^2 \right] \end{aligned} \quad (4.8)$$

Using the standard mean, $(\sum_{k=1}^{n-1} C_{S_k}^j = (n-1)\bar{C}_{S_{n-1}}^j)$, the sum-term simplifies to zero as shown below:

$$2 \sum_{k=1}^{n-1} [(C_{S_k}^j - \bar{C}_{S_{n-1}}^j)(\bar{C}_{S_{n-1}}^j - \bar{C}_{S_n}^j)] = 2 \sum_{k=1}^{n-1} [C_{S_k}^j \bar{C}_{S_{n-1}}^j - C_{S_k}^j \bar{C}_{S_n}^j - \bar{C}_{S_{n-1}}^{j^2} + \bar{C}_{S_{n-1}}^j \bar{C}_{S_n}^j]$$

Further simplification yields:

$$2 \sum_{k=1}^{n-1} [(C_{S_k}^j - \bar{C}_{S_{n-1}}^j)(\bar{C}_{S_{n-1}}^j - \bar{C}_{S_n}^j)] = 2 \left(\sum_{k=1}^{n-1} C_{S_k}^j \bar{C}_{S_{n-1}}^j - \sum_{k=1}^{n-1} C_{S_k}^j \bar{C}_{S_n}^j - \sum_{k=1}^{n-1} \bar{C}_{S_{n-1}}^{j^2} + \sum_{k=1}^{n-1} \bar{C}_{S_{n-1}}^j \bar{C}_{S_n}^j \right)$$

Substituting $\sum_{k=1}^{n-1} C_{S_k}^j = (n-1)\bar{C}_{S_{n-1}}^j$ and solving:

$$\begin{aligned} 2 \sum_{k=1}^{n-1} [(C_{S_k}^j - \bar{C}_{S_{n-1}}^j)(\bar{C}_{S_{n-1}}^j - \bar{C}_{S_n}^j)] &= 2(n\bar{C}_{S_{n-1}}^{j^2} - \bar{C}_{S_{n-1}}^{j^2} - n\bar{C}_{S_{n-1}}^j \bar{C}_{S_n}^j + \bar{C}_{S_{n-1}}^j \bar{C}_{S_n}^j - n\bar{C}_{S_{n-1}}^{j^2} \\ &\quad + \bar{C}_{S_{n-1}}^{j^2} + n\bar{C}_{S_{n-1}}^j \bar{C}_{S_n}^j - \bar{C}_{S_{n-1}}^j \bar{C}_{S_n}^j) \end{aligned}$$

Further simplification yields:

$$2 \sum_{k=1}^{n-1} [(C_{S_k}^j - \bar{C}_{S_{n-1}}^j)(\bar{C}_{S_{n-1}}^j - \bar{C}_{S_n}^j)] = 2(0) = 0$$

Substituting the sum-term equals to zero in Equation (4.8) we get:

$$s_n^{j^2} = \frac{\left[(n-2)s_{n-1}^{j^2} + (n-1)(\bar{C}_{S_{n-1}}^j - \bar{C}_{S_n}^j)^2 + (C_{S_n}^j - \bar{C}_{S_n}^j)^2 \right]}{n-1}$$

Further simplification yields:

$$s_n^{j2} = \left[\left(\frac{n-2}{n-1} \right) s_{n-1}^{j2} + \frac{(C_{S_n}^j - \bar{C}_{S_{n-1}}^j)^2}{n} \right]$$

The uncertainty associated with each cell in the map, w^j , is, hence, estimated and updated in each passage ($n > 1$) using the following relations:

$$w_n^j = \left[\left(\frac{n-2}{n-1} \right) (w_{n-1}^j)^2 + \frac{(C_{S_n}^j - \bar{C}_{S_{n-1}}^j)^2}{n} \right]^{\frac{1}{2}} \quad (4.9)$$

where $\bar{C}_{S_n}^j$ is the moving or running average given as:

$$\bar{C}_{S_n}^j = \left(\frac{C_{S_n}^j + (n-1)\bar{C}_{S_{n-1}}^j}{n} \right) \quad (4.10)$$

There is no need to initialize Equations (4.9) and (4.10), as for $n = 2$, the first term in Equation (4.9) equals zero and in Equation (4.10), $\bar{C}_{S_1}^j = C_{S_1}^j$. This uncertainty measure, updated using Equations (4.9) and (4.10), for each cell, in each passage, sheds light on the reliability of the state of the mapped cells as high uncertainty value means that the contents of the 3D cell are changing quite frequently; that could suggest high traffic circulation and other movements in the urban area. On the other hand, low uncertainty means that the contents of the 3D cell are fairly stable, indicating an established change (for example, permanent modifications in buildings or any other part of the cartography). This measure can also be added as a criterion for selecting the different specialized functions for the changing 3D cells; see Section 4.5.4 for more details. The uncertainty distribution of neighborhood-1 can be seen in Fig. 4.14. The figure clearly shows a high level of uncertainty in the bottom part of the environment (close to the ground), due to different temporarily static and mobile objects present in the scene.

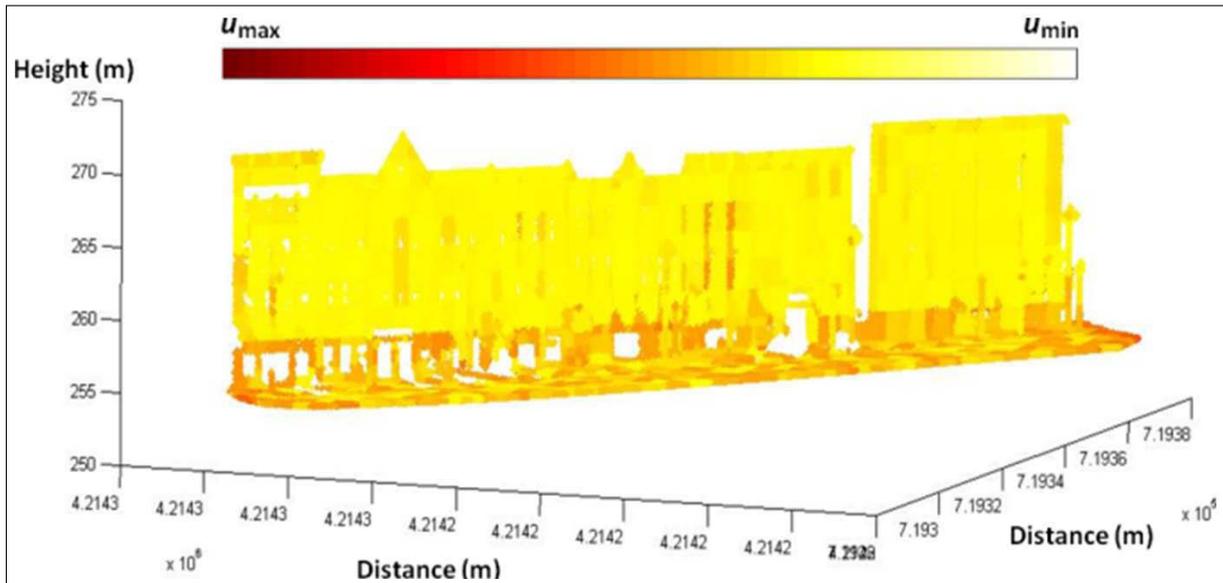


Figure 4.14: Uncertainty distribution map of the busy neighborhood-1 presented in the form of a color-coded heat map (with yellow being the lowest and red the highest level of uncertainty). The map clearly shows that the bottom part of the environment (close to the ground) is highly uncertain, whereas the top part is generally stable.

4.5.4 Specialized Functions

There are three specialized functions that are introduced in the method to make it more robust to different changes occurring in the urban environment and to remove different imperfections in the resulting cartography. These functions are described below.

4.5.4.1 Sequential Update Function

This is the default function (see Algorithm 2). It has two main purposes. Firstly, it offers a fine registration of the various 3D point clouds in subsequent passages. Secondly, it not only enriches the 3D point cloud by carefully adding 3D points in subsequent passages and, hence, completing the occluded regions in the process, but also automatically caters to the first type of change (Table 4.2) occurring in the urban environment.

Each subsequent 3D point cloud is registered with the former point cloud by using the ICP (Iterative Closest Point) method [BM92]. This method is most suitable for this task, as the 3D point clouds are already geo-referenced, hence lying in close proximity. It is observed that the major part of the 3D urban point clouds is composed of building points, which are also found to be most consistent. Thus, instead of applying the ICP method to complete 3D point clouds, only the building points are taken into account (see Algorithm 3). First, the profile/envelope of the buildings is extracted, and then, the ICP method is applied, matching these boundaries to obtain the transformation matrix. The outlines/envelopes of the buildings are extracted using a sweep scan method. As the bottom part of the building outline close to the ground is often occluded and, hence, inconsistent, due to the presence of different objects in the scene (see Fig. 4.14), only the boundary of the top half of the building outline is subjected to ICP, as shown in Fig. 4.15.

Algorithm 3 Extraction of outlines of buildings

Input: 3D points of building objects

- 1: **for** minimum x and y values of 3D points to maximum x and y values of 3D points **do**
 - 2: Scan building points in the $x - y$ plane
 - 3: Find the maximum and minimum value in the z -axis
 - 4: **end for**
 - 5: **for** minimum z values of 3D points to maximum z values of 3D points **do**
 - 6: Scan building points in the z -axis
 - 7: Find maximum and minimum value in the $x - y$ plane
 - 8: **end for**
 - 9: **return** envelope/profile of building objects
-

Once the transformation matrix (rotation matrix \mathbf{R}_m and translation matrix \mathbf{T}_m) is found, the whole 3D point cloud, $\mathbf{P}(n_p)$, is transformed into $\mathbf{P}'(n_p)$ and, then, registered with the former $\mathbf{P}(n_p - 1)$ using (4.11).

$$\mathbf{P}'(n_p) = \mathbf{R}_m(\mathbf{P}(n_p)) + \mathbf{T}_m \quad (4.11)$$

In order to avoid redundant points, a union of 3D points belonging to the two registered images is performed. Each 3D point of the first point cloud is matched with that of the second, if Equation (4.12) is satisfied:

$$\mathbf{p}_{0a} - \mathbf{p}_{0b} \leq \sqrt[3]{\mathbf{e}_{\text{tol}}} \quad (4.12)$$

Where \mathbf{p}_{0a} and \mathbf{p}_{0b} (3×1 matrix) are point positions in two point clouds along the x -, y - and z -axes. \mathbf{e}_{tol} (3×1 matrix) is equal to the inverse of the maximum number of 3D points per cubic meter that is desired in the 3D cartographic point cloud/image. The matched 3D points are considered as one point, i.e. the earlier point is retained, whereas the latter point is ignored. This ensures that only the missing points are added completing the perforated 3D point cloud/image.

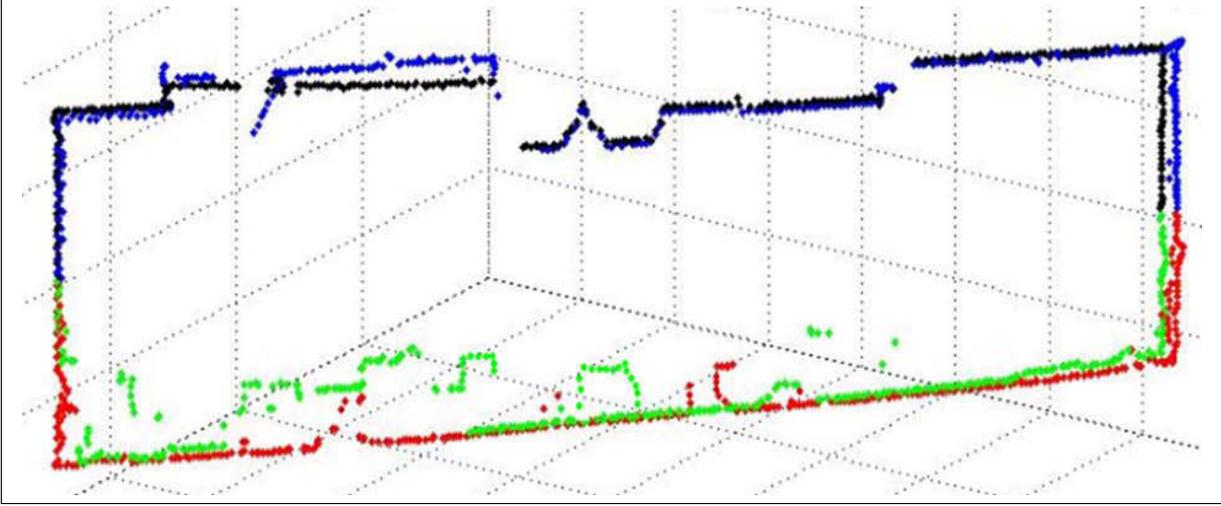


Figure 4.15: In red and blue, the building outline obtained from the first passage. In green and black, the outline obtained from the second passage. Only the top half of the two outlines in blue and black, respectively, is subjected to Iterative Closest Point (ICP).

Now, this specialized function also ensures that even if there is some new construction or addition of certain 3D points in the scene in subsequent passages, they are automatically added, hence catering for the first type of change (Table 4.2).

4.5.4.2 Automatic Reset Function

The main purpose of this function is to detect and analyze different changes occurring in the urban environment, over a number of passages, before incorporating them in the resulting 3D cartography. The function analyzes the similarity map, S_{Map} , and compares the Sym , $ASym$ s and the uncertainty measures, u , of the 3D cells after every n_{reset} number of passages. Those 3D cells, C_n^j , which satisfy Conditions 2 and 3 (Table 4.2) along with a low value of uncertainty measure, u , Equation (4.13) are reset with those in the recently acquired image/point cloud (perforated), i.e. their contents are replaced by the contents of the same 3D cells in the recently acquired point cloud/image. This ensures that any changes that occur in the urban environment are automatically incorporated in the resulting 3D cartography in a very smooth manner without affecting the remaining part of the 3D cartography.

$$Reset(C_n^j) \xrightarrow{\text{if}} \forall m \left(ASym(C_{m-1}^j, C_m^j) \geq ASym(C_m^j, C_{m-1}^j) \right) \wedge (u_n^j < u_{\text{threshold}}) \quad (4.13)$$

where $m = \{(n - n_{reset}) \dots n\}$, with $n > n_{reset}$. The proposed reset method was verified by synthetically modifying and demolishing different parts of the urban environment, including parts of buildings, roads and poles, in the datasets (for details, see Fig. 4.16).

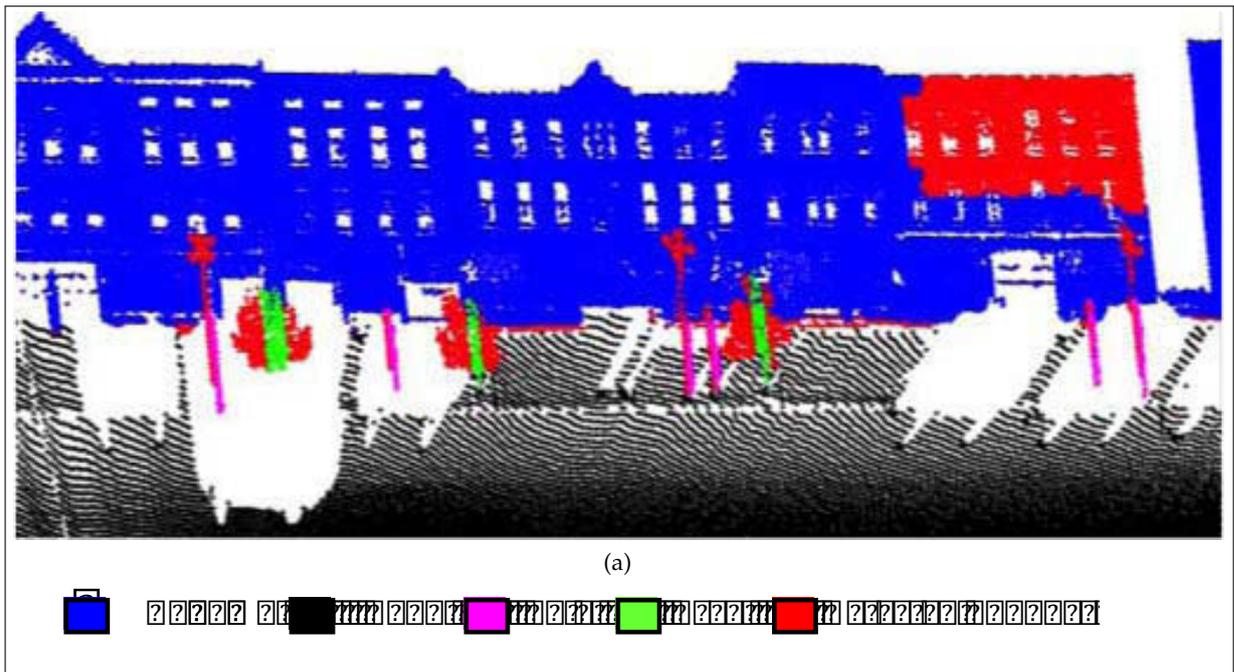


Figure 4.16: The different changes detected in one of the urban scenes are presented in red. These changes were due to a demolished building wall, larger poles cut in half and trimming/cutting of bushes/trees in the scene.

Now, Equation (4.13) ensures that a detected change is incorporated in the resulting cartography, only if we are reasonably certain about it. If the change is well-established, the uncertainty value of changing cells will progressively decrease, and when it falls below the $u_{\text{threshold}}$, these changes are incorporated in the 3D cartography via the reset function (as shown in Fig. 4.17(a)). On the contrary, detected change with increasing or high uncertainty indicates the occurrence of rapid continuing changes, which could be either due to ongoing construction/reconstruction or rapid traffic movement in the scene (as shown in Fig. 4.17(b)). In such situations, the detected changes are not incorporated at once, but in fact, we wait for the u to progressively decrease below $u_{\text{threshold}}$ as the number of passages, n_p , increases. This ensures the reliability and accuracy of our method.

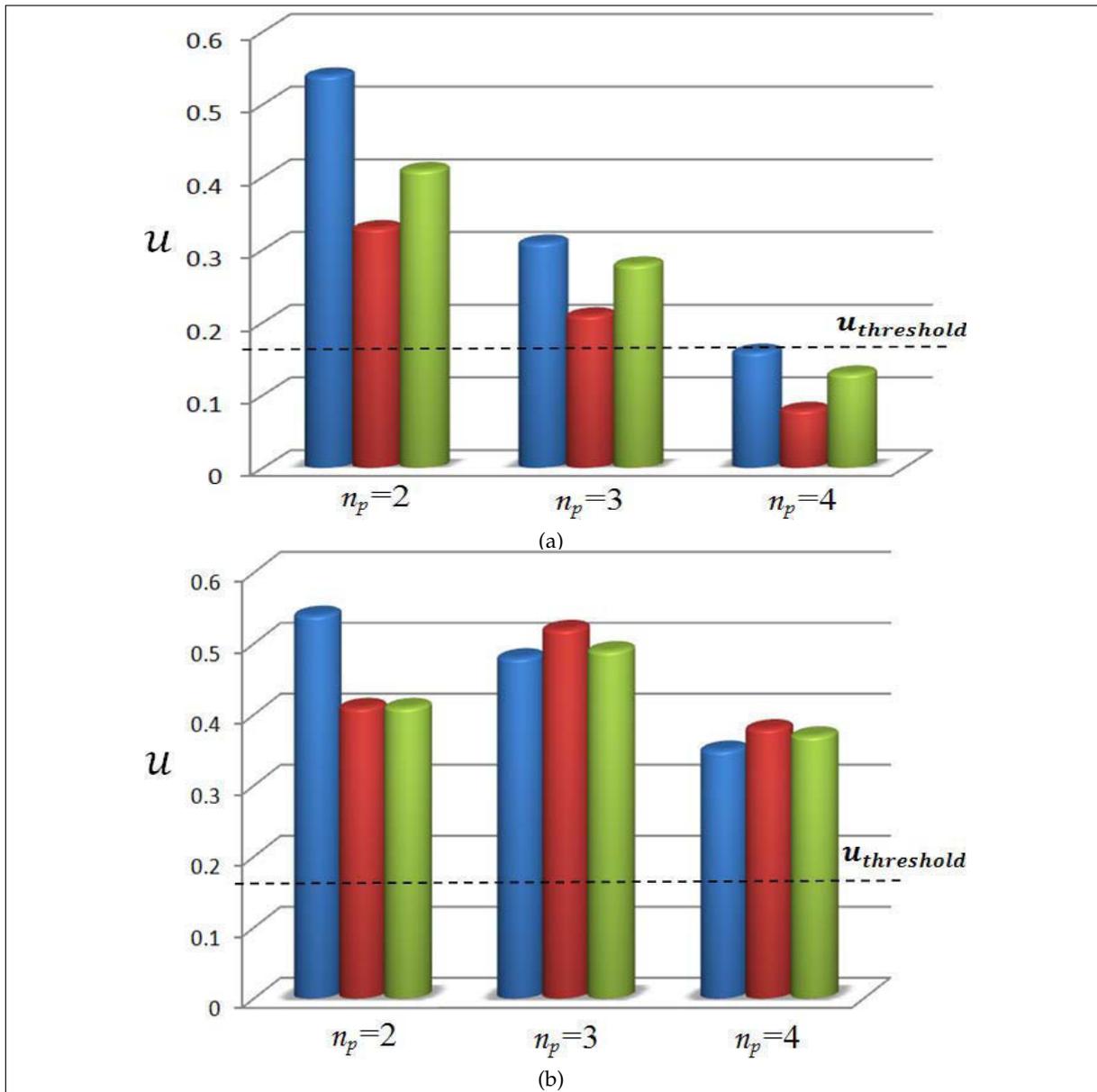


Figure 4.17: The uncertainty, u , variation of a group of three adjacent cells in which change was detected. (a) represents the case of established changes, while (b) represents the case of rapid or continuing changes.

4.5.4.3 Object Update Function

This function caters to different misclassifications and other unclassified objects. With every new passage, once the urban cartography is completed, the objects classified in each passage as Temporarily Static are also analyzed using Equations (4.1)–(4.3). If the same objects belonging to this class are found at the same place in repeated passages, they are then upgraded as Permanently Static objects and are considered part of the 3D cartography (Algorithm 2). They are then added to the 3D cartography. Otherwise, non-repetitive objects are deleted from the update register, $\mathbf{R}(n_p)$. This number of allowable repetitions, n_{update} , can be fixed, based upon the frequency of repetition, time of repetition, etc. This not only allows gradual update of the 3D cartography, but also accommodates for the unclassified objects in the scene; for example, certain parts of building walls in neighborhood-1 (Fig. 4.18(e)) and a few roadside trashcans in neighborhood-2 (Fig. 4.19(e)), which were unclassified, were added to the cartography, after repetition in the successive passages, during the update phase.

4.5.5 Automatic Checks and Balances

If certain 3D points or objects are wrongly added in the 3D cartography, due to either misclassification or certain repetitions in the object update function, these are then detected in subsequent passages as changes and, after analysis, are progressively removed by the automatic reset function. Hence, this ensures that the resulting 3D cartography contains only the exact, actual and permanent features.

4.6 Results, Evaluation and Discussion

In order to validate our method, the dynamic dataset of the 3D Urban Data Challenge 2011 [Cha11] was used. This dataset contains four sets of the same dynamic scenes of downtown Lexington, Kentucky, USA, obtained on different days and at different times. The dataset consists of 3D points coupled with corresponding laser reflectance intensity values. As the corresponding RGB values are not readily available, Equation (4.2) was not used. This did not have much impact on the results, as laser reflectance values are found to be more consistent than RGB values in an urban environment, which is more illumination invariant (see Section 3.5.6). The results for two different neighborhoods are discussed in this paper. In Fig. 4.18–4.20, the detailed results for neighborhood-1, neighborhood-2 and neighborhood-1(modified) are presented, respectively. The figures clearly show how different imperfections are progressively removed using the specialized functions, while the different changes occurring in the urban scene are updated after successful detection and analysis. Figure 4.21 shows the successful completion of occluded features along with the imperfection removal of a particular street corner using this method.

4.6.1 Change Detection and Reset Function

Now, in order to evaluate the reset function, we assessed the ability of the proposed method to successfully detect changes occurring in the urban environment before they can be updated in the cartography. We first constructed an ROC (Receiver Operating Characteristic) curve. The value of $Similarity_{\text{threshold}}$ (used as a discriminatory threshold) was varied from zero to one, and corresponding true positive rates (TPR) and false positive rates (FPR) were calculated for neighborhood-1 (modified). The variation of this ROC curve with respect to $u_{\text{threshold}}$ (varying from zero to 0.5) is presented by a 3D ROC curve (surface, in this case) in Fig. 4.22. From the figure, it could be observed that the best/optimal results (A in the figure) can be obtained from $Similarity_{\text{threshold}} = 66\%$ and $u_{\text{threshold}} = 0.15$. This analysis was conducted at the 3D cell level. Figures 4.16 and 4.20(f) show some of these detected changes in the urban scene.

These values of $Similarity_{\text{threshold}}$ and $u_{\text{threshold}}$ along with $L = 2 m$, $\mathbf{e}_{\text{tol}} = (0.000125 \ 0.000125 \ 0.000125)^T$ (in m^3) and $n_{\text{reset}} = 3$ (the maximum number of passages possible in our case is four) were then used to evaluate the performance of our method for all three neighborhoods.

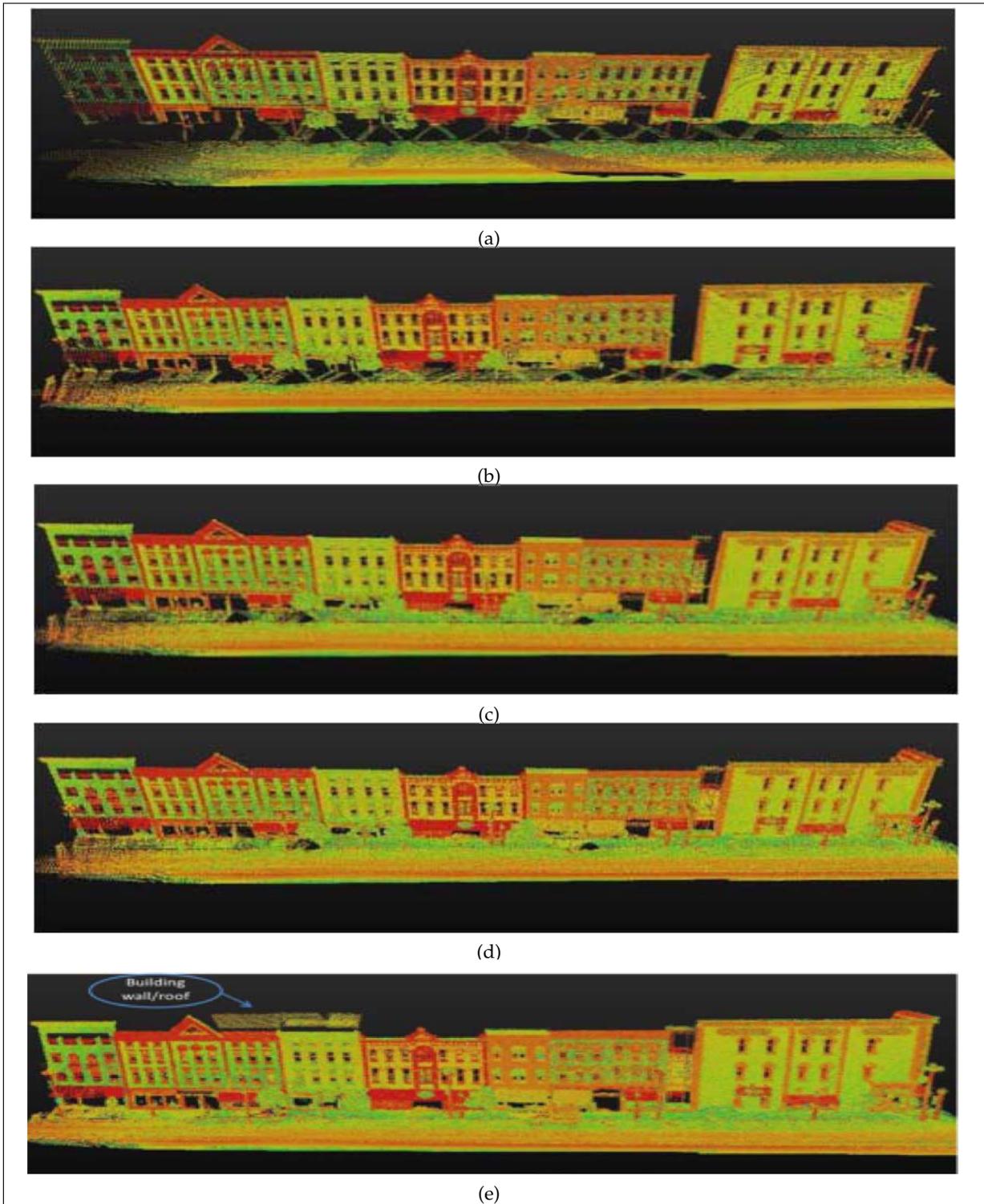


Figure 4.18: (a) Shows the initial point cloud related to urban cartography full of imperfections. In (b), (c) & (d), completion of occluded and missing features in the urban cartography, by incremental updating, using the default sequential update function as a change of type-1 is detected. (e) shows the point cloud after the object update function; certain parts of the roadside and building walls are added to the 3D cartography after repetition in subsequent passages.

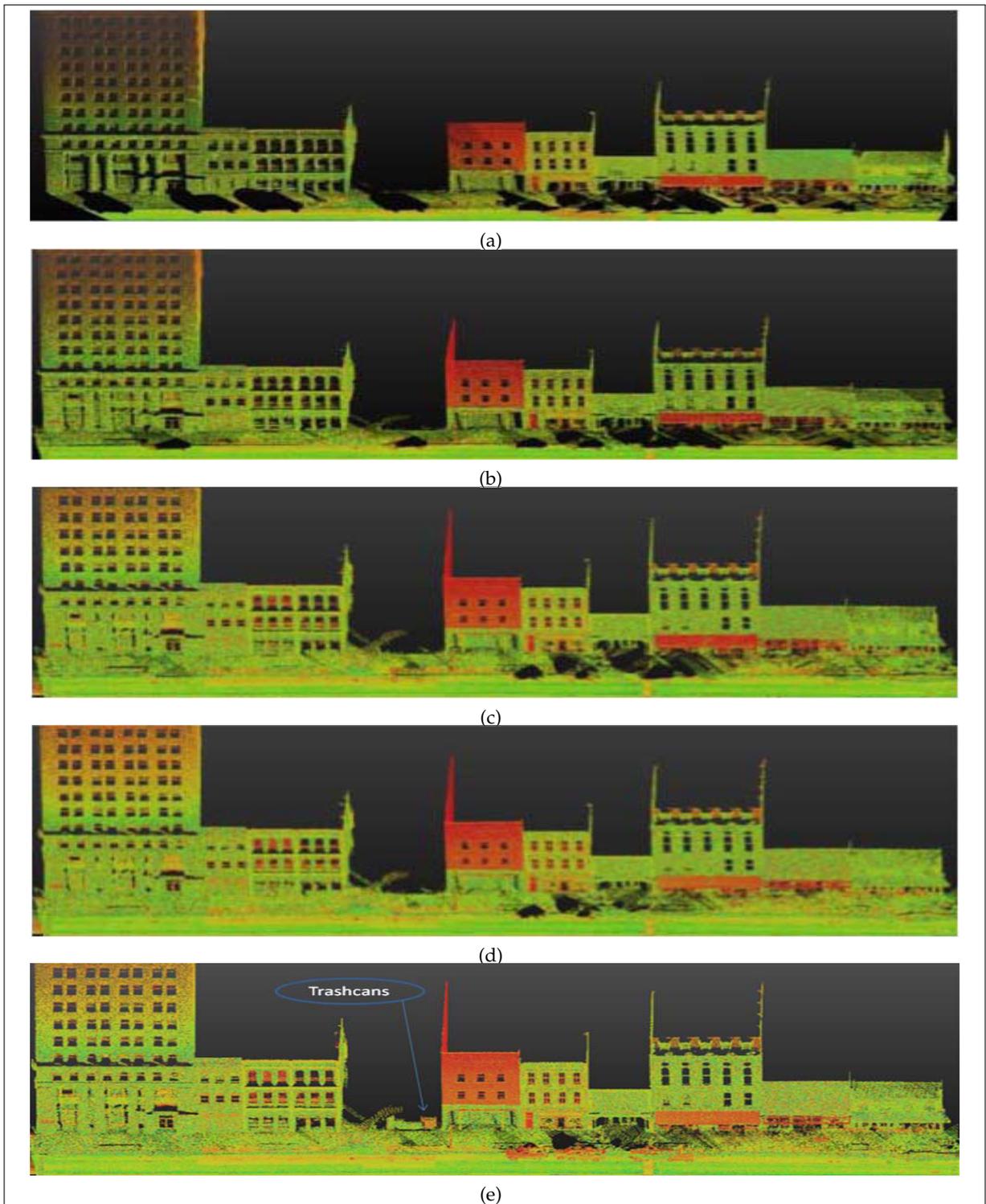


Figure 4.19: (a) shows the initial point cloud related to urban cartography full of imperfections. In (b), (c) & (d), completion of occluded and missing features in the urban cartography, by incremental updating, using the default sequential update function as a change of type-1 is detected. (e) shows the point cloud after the object update function; certain parts of the roadside and building walls along are added to the 3D cartography after repetition in subsequent passages.

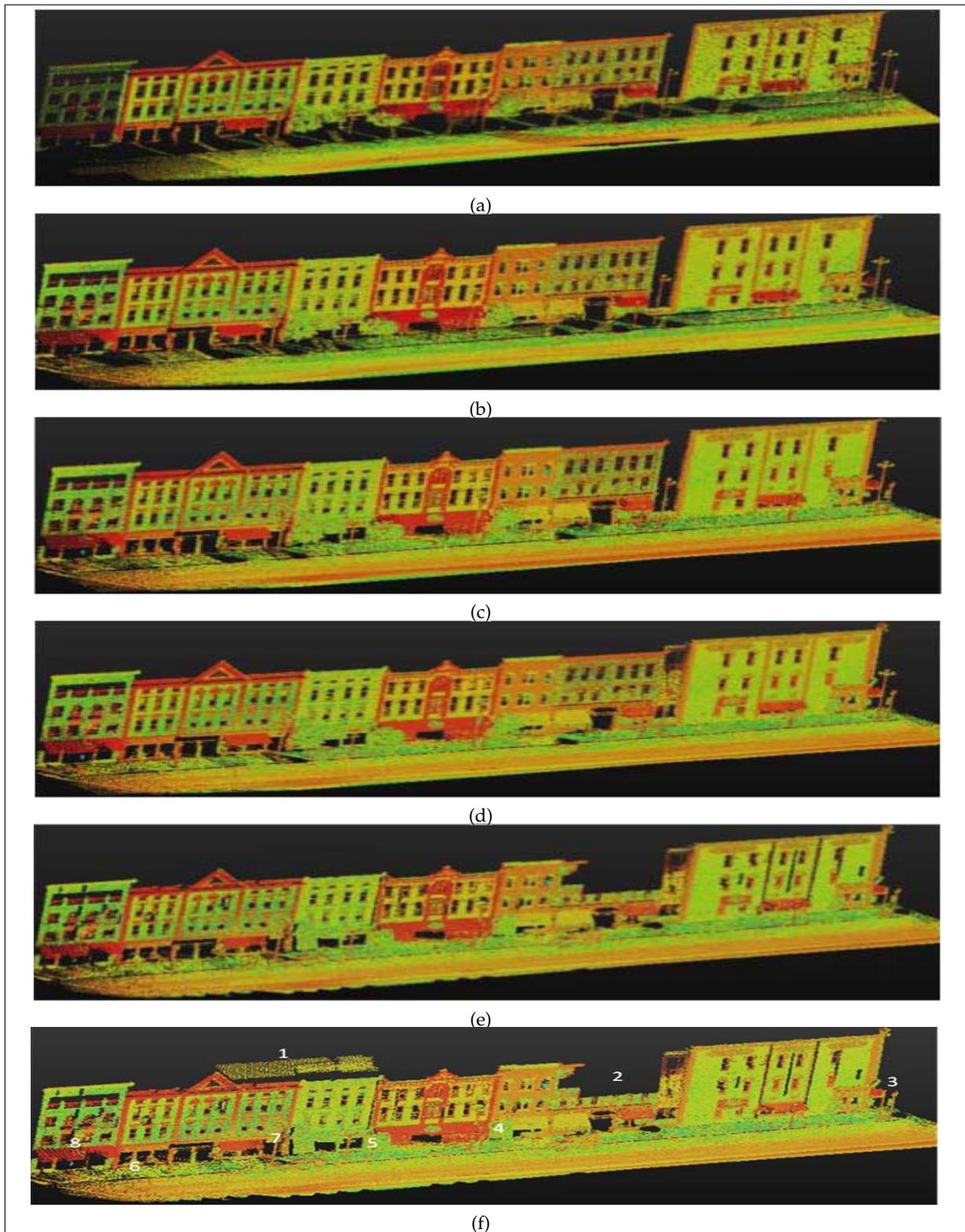


Figure 4.20: (a) shows the initial point cloud related to urban cartography full of imperfections. In (b), (c) & (d), completion of occluded and missing features in the urban cartography, by incremental updating, using the default sequential update function as a change of type-1 is detected. In (e), the automatic reset function came into action after changes of type 2&3 were detected, to update the modifications in the 3D cartography. (f) shows the point cloud after the object update function; certain parts of the roadside and building walls are added to the 3D cartography after repetition in subsequent passages. On (f) are also marked the changes successfully detected and updated after four passages. These include: 1—building wall/roof added, due to initial misclassification; 2—part of building demolished; 4, 5 and 6—trimmed or cut trees/bushes; 3, 7 and 8—longer poles cut in half.

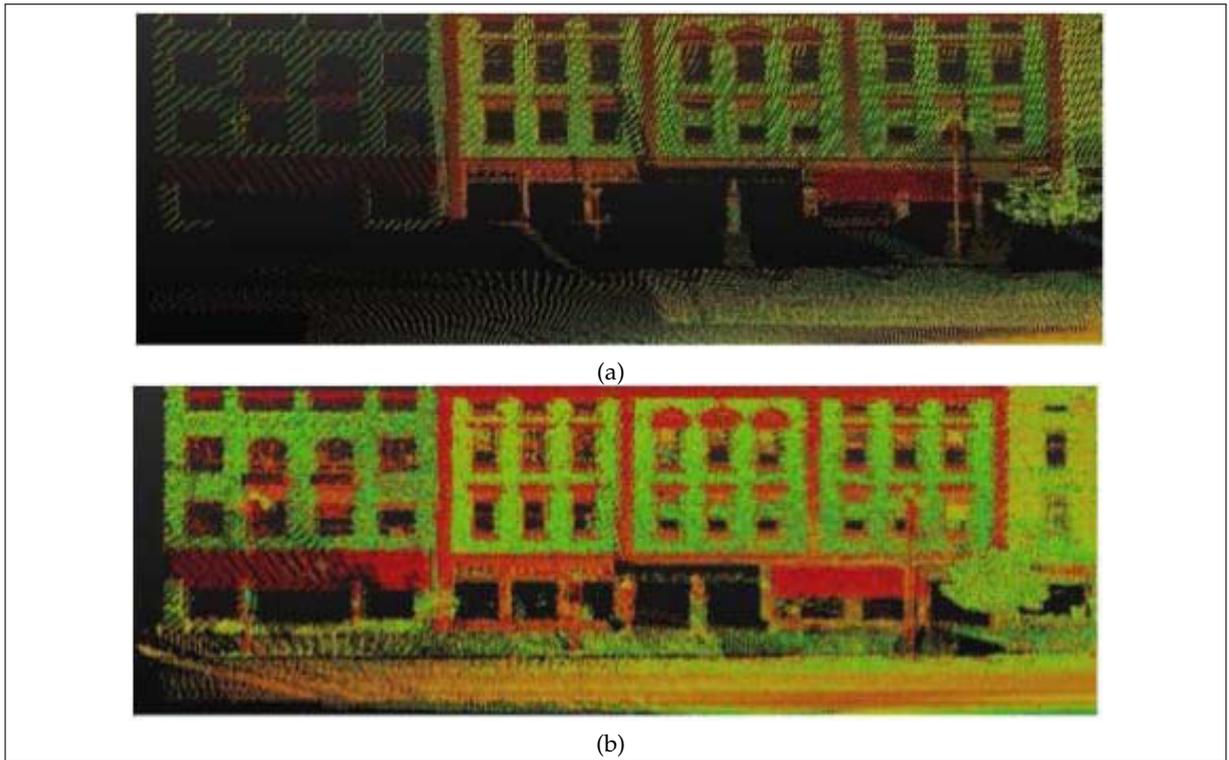


Figure 4.21: Different missing features and imperfections belonging to shop windows, walls, pole and road in (a) are successfully completed and removed in (b), respectively, using this method for neighborhood-1. (a) 3D point cloud, $\mathbf{P}(n_p)$, after first passage; (b) 3D point cloud, $\mathbf{P}(n_p)$, after fourth passage.

Change detection for the automatic reset function was evaluated using different standard evaluation metrics, as described in [Vih12] (see Table 4.3). Although all these metrics are commonly used to evaluate such algorithms, MCC (Matthews Correlation Coefficient) is regarded as the most balanced measure, as it is insensitive to different class sizes (as is the case with our application, the number of changed cells (changes) is generally quite inferior as compared to unchanged cells in the urban environment). The MCC, like the other measures, is calculated based on the count of the true positives (i.e. correct detection of changed 3D cells), false positives, true negatives and false negatives. A coefficient of +1 represents a perfect prediction, zero is no better than random prediction and -1 indicates total disagreement. The detailed results, including the overall accuracy, ACC, and MCC greater than 85% and +0.6, respectively, clearly demonstrate the efficacy of the proposed method.

Table 4.3: Change detection evaluation.

		Neighborhood 1 (modified)	Neighborhood 1	Neighborhood 2
ACC	Accuracy	0.864	0.891	0.903
PPV	Positive Predictive Value	0.842	0.850	0.900
NPV	Negative Predictive Value	0.864	0.902	0.901
FDR	False Discovery Rate	0.158	0.150	0.100
F₁	F ₁ measure	0.695	0.650	0.782
MCC	Matthews Correlation Coefficient	+0.624	+0.692	+0.729

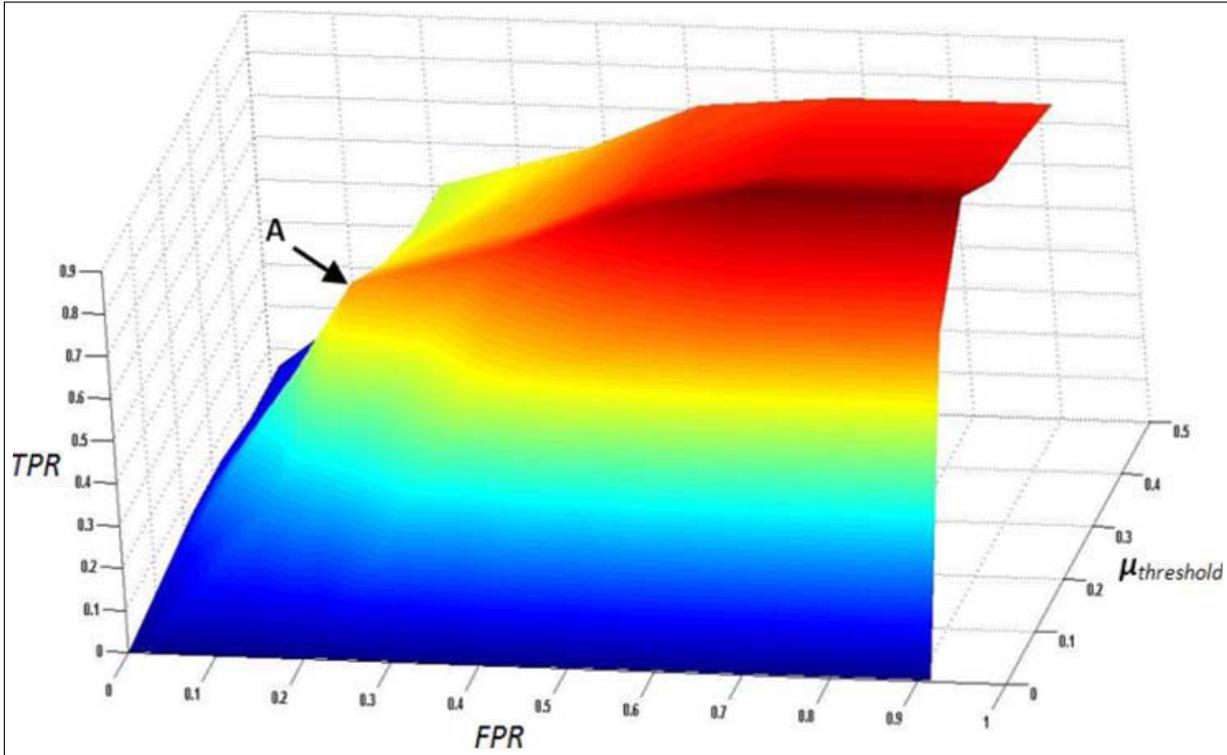


Figure 4.22: True positive rate (TPR) is plotted against false positive rate (FPR) to obtain a 3D Receiver Operating Characteristic (ROC) curve for neighborhood-1 (modified). Point A on the surface represents the optimal/best results.

4.6.2 Evolution of Similarity Map Size

Figure 4.23 shows the progressive reduction of the size of the similarity map in subsequent passages. This is due to the fact that in subsequent passages, as occluded regions are completed, the number of low similarities caused by the missing occluded features decreases, and also, the different changes occurring in the cartography are catered to by means of the automatic reset function. In the case of neighborhood-1 (modified), Fig. 4.23, the large size of S_{Map} is due to large parts of the building, poles, trees, etc., being demolished in neighborhood-1 (see Fig. 4.16), whereas a sharp decrease in size occurs at $n_p = 4$, due to the automatic reset function ($n_{reset} = 3$). We see that some non-repeating cells or those with high associated uncertainty remain even after the automatic reset. This number of cells in the S_{Map} should ideally reduce to zero as the number of n_p increases.

4.6.3 Handling Misclassifications and Improvement in Classification Results

To assess the removal of imperfections due to misclassification, the improvement in classification results by the proposed method as compared to single passage classification (standard practice) was evaluated using standard F-Measure as described in [Vih12; HR05]:

$$F_\beta = (1 + \beta^2) \frac{p \cdot r}{(\beta^2 \cdot p + r)} \quad (4.14)$$

where p and r are the precision and recall, respectively, and β is the weight constant. Table 4.4 shows that the proposed method improves the classification results (Permanently Static objects are analyzed here, as they constitute the permanent cartography, and, hence, are of most interest to us). The value of $\beta = 1$ was used to obtain a balanced F_1 -Score. The classified objects were considered as a percentage of their constituting points in the 3D scene. This improvement is due

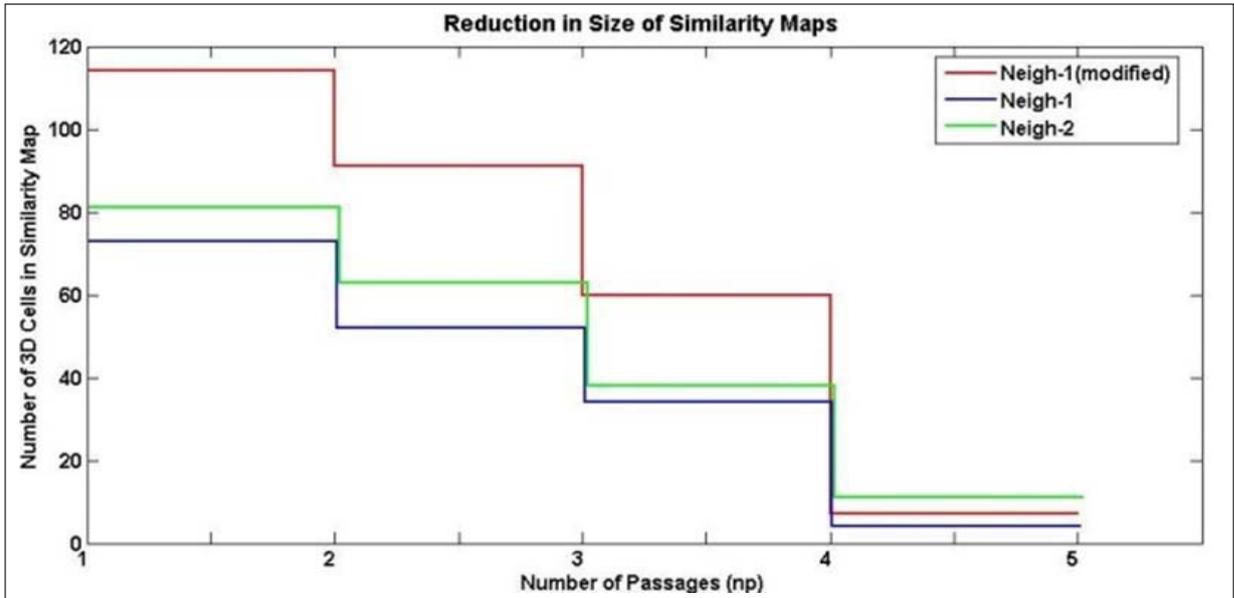


Figure 4.23: The progressive reduction of the size of similarity map, S_{Map} .

to different misclassifications being progressively corrected, hence reducing the imperfections caused by them in the resulting 3D cartography.

Table 4.4: Classification results for Permanently Static objects (F_1 -Score).

	Passage 1	Passage 2	Passage 3	Passage 4	Proposed Method
Neighbor-1	0.943	0.977	0.958	0.959	0.981
Neighbor-2	0.979	0.975	0.981	0.984	0.992

4.6.4 Accuracy Evaluation

In order to evaluate the accuracy of the completed permanent features/regions and, hence, the effectiveness of the imperfection removal, we selected a corner building in neighborhood-1. As no ground truth was readily available, we generated the ground truth by creating a simplified 3D model of the building using standard CAD software, as shown in Fig. 4.24. The 3D points from the initial acquisition were used for this purpose, while the missing features were completed by horizontal and vertical interpolations, exploiting the symmetry of the building design; these features were confirmed/matched with the images of the building, from different viewing angles, acquired by the digital spherical camera mounted on the vehicle (see Fig. 4.24). A number of features, including the occluded ones, were selected for comparison. The dimensions of these selected features extracted in the 3D image obtained at each passage, using our method, were then compared with their corresponding dimensions in the ground truth.

The average absolute errors in x , y and z -axes (height) of the available dimensions obtained in each passage are presented in Fig. 4.25. These error values include both registration and sensor measurement errors. Low average absolute error values obtained for passage-1 are due to the fact that part of these 3D points was used for ground truth generation. These generally low and fairly constant error values make this method suitable for most applications.



Figure 4.24: (a) shows the generated 3D model of the buildings according to dimensions; In (b), one of the camera images of the building is presented.

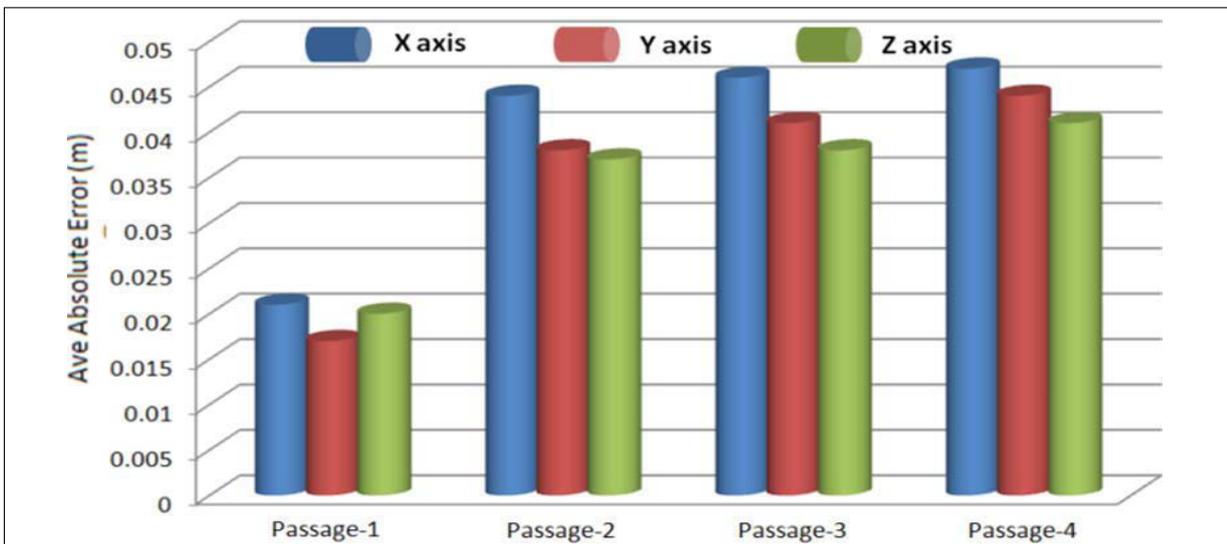


Figure 4.25: Average absolute errors in x , y and z (height) directions.

4.7 Conclusion

In this chapter, after discussing in detail the different existing techniques of handling imperfections in 3D urban cartography, we have presented a new method for automatic 3D urban cartography that progressively removes different imperfections caused by occlusions, misclassifications of objects in the scene and ineffective incorporation of changes occurring in the environment over time, by taking advantage of incremental updating using specialized functions. Different changes occurring in the urban landscape are automatically detected and analyzed using cognitive functions of similarity, and the resulting 3D cartography is modified accordingly. The proposed method ensures that the resulting 3D point cloud of the cartography is quite reliable, and it contains only the exact and actual permanent features, free from imperfections.

The results evaluated on real data using different evaluation metrics demonstrate the technical prowess of the method. The thorough analysis exhibits accurate change detection (including values of overall accuracy, ACC, and a Matthew's Correlation Coefficient, MCC, greater than 85% and +0.6, respectively), improved classification results (with a standard F_1 -Score > 0.98) and high accuracy of the completed permanent features/regions of the urban cartography. The

results also show that the method is well scalable and that it can both be easily integrated and ideally suited for removing various imperfections for different commercial or non-commercial applications pertaining to urban landscape modeling and cartography requiring frequent database updating.

In the next chapter, we show that the proposed method of temporal integration also helps in improved semantic analysis of the urban environment, specially building façades.

DETECTION AND FEATURE ESTIMATION OF WINDOWS FOR REFINING BUILDING FAÇADES

Due to the ever increasing demand for more realistic 3D urban models coupled with recent advancements in ground based LiDAR technologies, recovering details of building façade structures, such as windows, has gained considerable attention. However, fewer laser points are usually available for windows as window frames occupy only small parts of the building façades while the window glass also offers limited reflectivity. This insufficient raw laser information makes it very difficult to detect and recover reliable geometry of windows without human interaction. So, in this chapter, we discuss the different state of the art practices in this domain and then present a new method that automatically detects windows of different shapes in 3D LiDAR point clouds obtained from mobile terrestrial data acquisition systems in the urban environment. The proposed method first seg-

ments out 3D points belonging to the building façade from the 3D urban point cloud and then projects them onto a 2D plane parallel to the building façade. After point inversion within a watertight boundary, windows are segmented out based on geometrical information. The window features/parameters are then estimated exploiting both symmetrically corresponding windows in the façade as well as temporally corresponding windows in successive passages, based on ANOVA (ANalysis Of VAriance) measurements. This unique fusion of information not only accommodates for lack of symmetry but also helps complete missing features due to occlusions. The estimated windows are then used to refine the 3D point cloud of the building façade. The results, evaluated on real data using different standard evaluation metrics, not only demonstrate the efficacy but also the technical strength of the proposed method.

5.1 Introduction

Recently, analysis of building façades for 3D building reconstruction and realistic geometrical modeling of the urban environment has gained some interest in the scientific community. Such realistic models play a pivotal role in several applications pertaining to different perception tasks such as driving assistance, autonomous navigation through the urban environment, fly-through rendering, virtual tourism and urban mission planning, etc. Driven by an increasing demand to improve the quality of such applications, some work has been done lately, focusing on the semantic analysis of building façades including detecting and modeling geometric structures like windows and doors to provide higher Level of Details (LoD) as presented in [LBG+13] and shown in Fig. 5.1. Semantically enriched by features like windows, along with geometrically accurate outer dimensions, this LoD 3 information further increases the number of applications for example analysis of access ways to buildings for evacuation scenarios or police operations, sampling of windows for coefficient of heat transmission (thermal analysis) and area estimation for restructuring requirements, etc. This, in the context of the thesis, being a logical continuation of our work, not only helps to refine the quality but also increases the semantic richness of the resulting 3D urban cartography extending its suitability for broader range of applications (as mentioned above).



Figure 5.1: Representation of a building using LoD0 - LoD4 [LBG+13].

Usually, such applications mostly rely on data acquired from mobile terrestrial data acquisition systems as discussed in Chapter 2 and shown in Fig. 5.2. These ground-based or vehicle-borne laser scanning techniques help acquire highly accurate geo-referenced 3D points with sufficient details due to their close-range data collection. In addition, the capability of capturing millions of 3D points directly in a short time offers a great potential for a fast generation of detailed 3D models and maps. However, this task is made significantly harder by complex representations in such a highly cluttered urban environment (cars, poles, trees, vegetation, etc.), variability of appearances (e.g. open/closed windows and doors, etc.) and missing data due to physical properties (for instance window glass does not provide any measurements). The problem is further complicated by the fact that some data are missing due to occlusions caused

by different temporarily static objects (like cars, pedestrians etc.), self-occlusion of the façade due to an oblique viewing angle (i.e. protruding balconies or indented windows casting shadows along the direction of measurement, etc.) and variability in data resolution (farther objects are scanned in lower resolution than closer ones). So, in this chapter we present a new method that automatically detects and estimates windows on building façades while successfully addressing all these challenges. We apply our method on real data from large urban scenes.

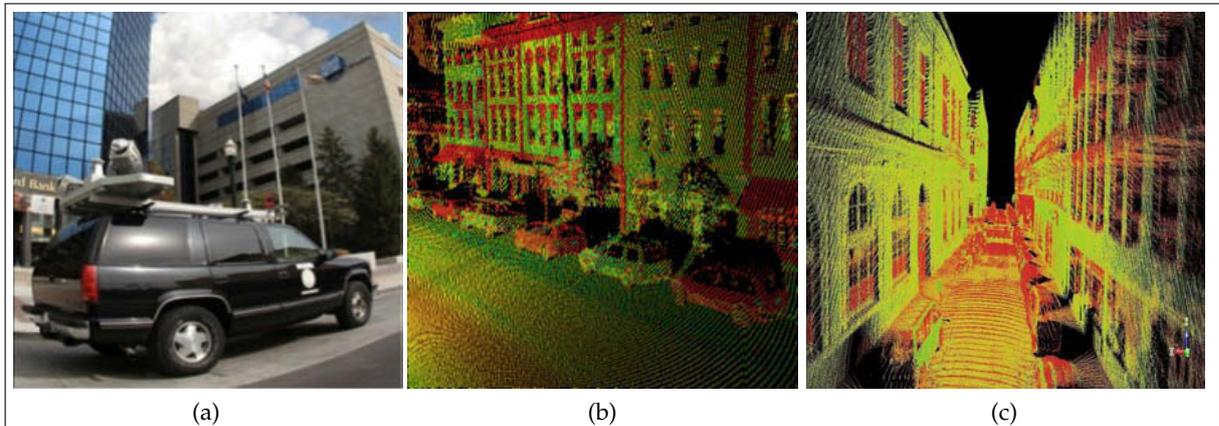


Figure 5.2: (a) The Vis Center’s LiDAR Truck. (b) & (c) shows 3D point cloud (in color-coded intensity) acquired by the truck, containing building façades in the urban environment.

The problem of detecting windows on building façades has lately received considerable attention. Several previous works have attempted to tackle this problem of detecting window structures using 2D images [MZW+07; MR07; ASJ+07; LN04]. These methods either use explicit window models [LN04] along with horizontal and vertical analysis of images to discover repetitive patterns, or use an implicit window model and a learning approach to recognize windows [MR07; ASJ+07]. The works [LN04; MZW+07] assume rectilinear window patterns with features relying on RGB data; however, these methods could not work for multiple structures. General approaches [JP09] detect windows as blobs in a color image but do not exploit the underlying structure as shown in Fig. 5.3.

Grammar-based methods, first proposed in the field of architecture [SG72], have been successfully used in [WWS+03]. The generation of shape grammar from façade images can be automated by image similarity detection. Typical intensity-based similarity measures include cross-correlation [LN04] and mutual information [SWK07; RTR+04]. The common assumption in these methods is that windows are a rectangular structure, which is fair for most buildings, although there are always some exceptions. In addition to this fact, in an image of a building, there often exist too many edges, luminance, color and texture variations, etc., which makes image-based window detection a challenging task.

Compared to detection in 2D images, lesser work has been done on detecting windows from LiDAR data. One reason for this may be due to the generally noisy and sparse nature of the LiDAR data which make the small structures like windows hard to differentiate. In addition, windows usually do not return any signals, however sometimes signals are returned from the interior of the buildings, which makes the task even more complicated. In [PMW+08], regularities of substructures are derived from a 3D model or range scan of a scene. This general approach may be used for extracting regularities but it is sensitive in the calculation of curvatures. In [AAP08], a method is presented that converts LiDAR data into distance images, and then employs image processing techniques like morphological operations and contour analysis to segment windows. This 3D-to-2D conversion causes information loss. A Markov Network approach that requires training is used to label points as windows in [TKB06]. In [ZSW+10], 3D repetitive elements are manually selected and automatically consolidated. In [SHF+11], façades

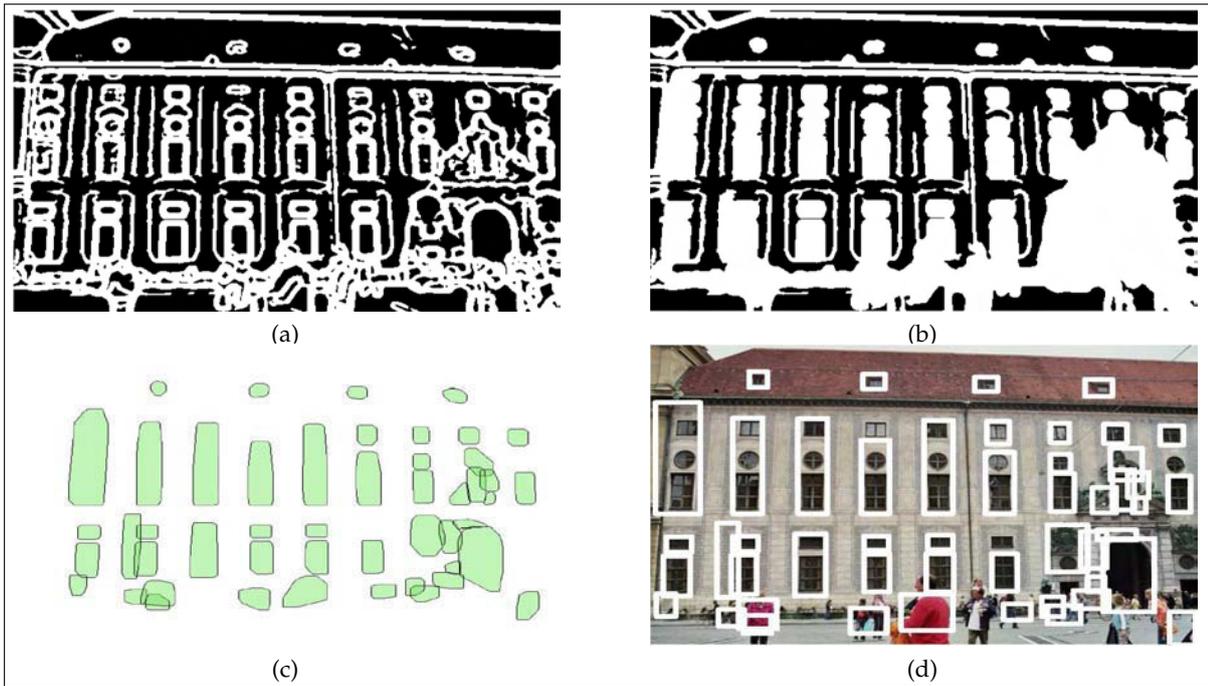


Figure 5.3: [JP09] shows the detection of windows as blobs in a color image. The main steps are: (a) Binarisation of image. (b) Filled contours. (c) Blob extraction. (d) Bounding boxes around all the blobs extracted.

are adaptively partitioned in horizontal and vertical planes based on the boundary features of the planar regions. However, the method works for data with uniform resolution, which is not always possible. Hence, it can be adversely affected by the variation in the resolution as well as the window appearance, which can both exist inside a single scan. On the other hand, our method remains unaffected by the variation in the resolution of the acquired data. This problem of varying resolution is also addressed in [MS12] in which the 3D points are first projected onto a 2D binary orthographic point occupancy map, followed by the extraction and iterative refinement of the 2D window centers using a local lattice fitting and lattice voting scheme as explained in Fig. 5.4.

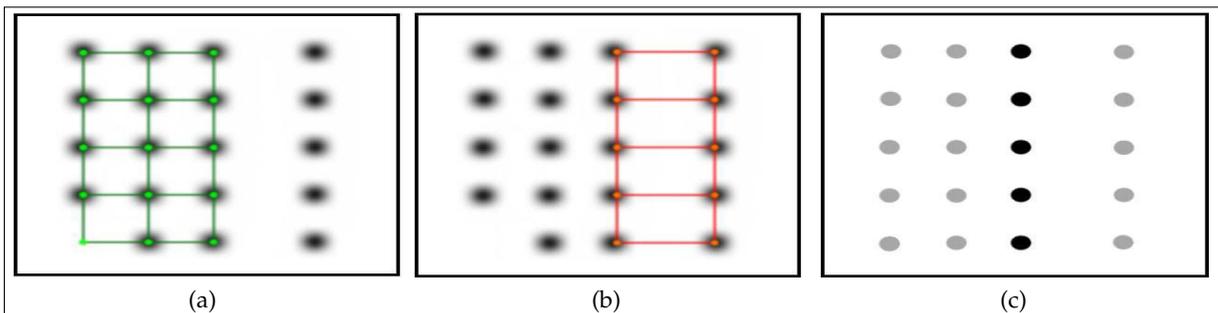


Figure 5.4: Illustration of lattice voting as presented in [MS12]. (a) & (b) are different extracted lattices overlaid on projected image and (c) is the resulting histogram. The darker discs accumulate votes from both lattices. Also seen is the automatic symmetrical reconstruction of the lower left corner.

These 2D centers are then projected back to 3D. This method, like most of the others, assumes regular symmetrical window patterns on the façade. However, these methods are less effective in case of lack of symmetry or large variations in symmetrical patterns. This is very common in case of interface of different façades joined together (as shown in Fig. 5.5) as discussed by

Mesolongitis and Stamos [MS12] in the results of the paper. In contrast to these methods, our method does not make any assumptions about the symmetry of the façade but in fact it also combines information from multiple successive passages to not only accommodate for lack of symmetry but also help to complete missing features due to occlusions. The proposed method uses the property of windows not returning any signal and hence these windows are represented by holes on the 3D façade. A hole-based extraction method is also presented in [PV07]. The method searches edges along the Triangular Irregular Network (TIN) of the façade to identify holes which are then fitted to rectangles. However, this bottom-up triangular meshing-based method suffers from noisy LiDAR data and as it does not detect regular groups it is more prone to occlusions.

The proposed method first segments out 3D points belonging to the building façade $B(n_p)$ from the 3D urban point cloud. These 3D points are then projected onto a 2D plane parallel to the building façade. After point inversion within the watertight boundary, windows are segmented out. The parameters/features of these segmented windows are then estimated exploiting symmetrically corresponding windows in the façade as well as temporally corresponding windows in successive passages, based on ANOVA (ANalysis Of VAriance) measurements.

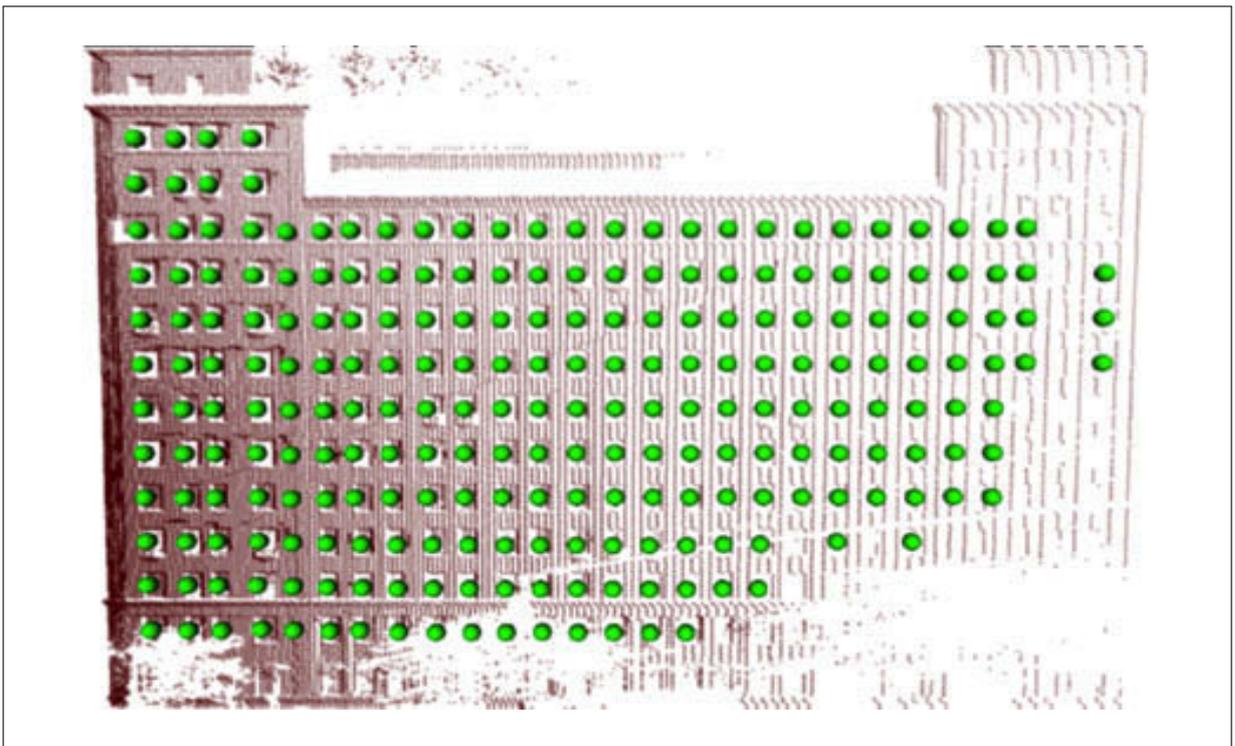


Figure 5.5: This figure is taken from the results of [MS12]. It shows the detected windows represented by green-filled circles. The method gives wrong detections at the interface of the two façades (i.e. 5th column from the left).

According to the best of our knowledge no previous work has ever been presented that fuses information from both symmetrical and temporal correspondences to detect and estimate windows. The estimated windows are then used to refine the 3D point cloud of the building façade via a reference table.

5.2 Segmentation of Building Façades and Road Surface from 3D Urban Point Clouds

In order to segment out the building façades for window detection, we first classify the 3D urban point clouds into basic object types using the method explained and presented in Chapter 3. These object types including roads and buildings are: {Road, Building, Car, Pole, Tree, Un-classified}.

The occluded regions in each passage due to temporarily static and dynamic objects in the scene are completed with the help of similar 3D point clouds obtained from successive passages in the same place on different days and at different times of the day as explained and presented in Chapter 4. In this method, the 3D point clouds in each successive passage are first registered with the former and the missing regions are completed using a point-to-point matching technique. The proposed work uses this framework of automatic 3D urban cartography. Using the same 3D urban point clouds as input, the integration of our method in this framework enables its output/results to be directly employed in 3D urban cartography and modeling applications. This not only helps in the validation but also in the better evaluation of the method.

After classification, the 3D points pertaining to the segmented buildings as well as the road surface are then projected onto a plane, parallel to the building façade in order to detect and segment windows in the building façades as explained in the following section. An overview of the proposed method is presented in Algorithm 4 and explained in the later sections.

5.3 Window Segmentation

In the 3D LiDAR point clouds of the building façades, windows are usually represented by holes due to limited reflectivity [MS12]. They may, sometimes, also contain few 3D points with low intensity values due to reflections from objects inside [TS11]. Our method for automatic detection of windows in such 3D façade point clouds is based on this property. The 3D points pertaining to the building façade and road surface are first projected onto an arbitrary plane parallel to the building façade. A watertight boundary of this projected 2D façade is then determined. Point inversion is done within this boundary and windows are then segmented. The details of this method are as follows.

5.3.1 3D Point Projection

As a mobile street mapper, mentioned in Section 5.7, is used for data acquisition, the building class consists of 3D points predominantly belonging to building façades due to the particular viewing angle. These 3D points belonging to the building façade are then projected onto a known arbitrary plane parallel to the building façade in every passage n_p , as shown in Fig. 5.6(b) to obtain $Proj(B(n_p))$. The projection plane is determined by calculating the best-fit plane through the 3D points of the building façade using planar regression of data as presented in [Fer05]. A best-fit plane can be defined with the equation:

$$(x_i - \bar{x}) = B(y_i - \bar{y}) + C(z_i - \bar{z}) \quad (5.1)$$

where \bar{x} , \bar{y} , and \bar{z} are the respective mean values of X , Y , and Z coordinates of all points. To find the equation of the best-fit plane for a given set of points, Press *et al.* [PFT+07] present the following equations that have to be solved for B and C :

$$\sum ((x_i - \bar{x})(y_i - \bar{y})) = B \sum (y_i - \bar{y})^2 + C \sum ((y_i - \bar{y})(z_i - \bar{z})) \quad (5.2)$$

$$\sum ((x_i - \bar{x})(z_i - \bar{z})) = B \sum ((y_i - \bar{y})(z_i - \bar{z})) + C \sum (z_i - \bar{z})^2 \quad (5.3)$$

The result of the regression is a plane that passes through a point with coordinates $(\bar{x}, \bar{y}, \bar{z})$ and is returned in the form of a vector normal to the best-fit plane. The equations in [PFT+07]

Algorithm 4 Automatic window detection and façade refinement.

Input: 3D urban point clouds $P(n_p)$ for passage number n_p

- 1: Segmentation-based classification of $P(n_p)$ to obtain building façades and road surfaces
- 2: Register, match and compare point cloud $P(n_p)$ with $P(n_p - 1)$ to try to complete missing features on the building façade due to occlusions
- 3: Project the 3D points pertaining to building façades and road surfaces in $P(n_p)$ onto an arbitrary plane parallel to the building façade to obtain $Proj(B(n_p))$ and reference table T_{proj}
- 4: Extract a watertight boundary of the projected building façade in $Proj(B(n_p))$
- 5: Conduct point inversion within the watertight boundary to segment out holes to obtain an inverted projection $Proj(B(n_p))_{INV}$
- 6: Cluster 2D points pertaining to holes in $Proj(B(n_p))_{INV}$ using a Nearest Neighbor Search (NNS)
- 7: Classify these segmented clusters as windows based on geometrical features and assign feature space {Height, Width, Profile, Geometrical center}
- 8: **for all** detected windows in the building façade **do**
- 9: Find symmetrically corresponding windows in rows and columns on the façade
- 10: Find temporally corresponding windows for the same façade from the previous passages in $Wc(n_p - 1)$
- 11: Estimate window features using these correspondences
- 12: **end for**
- 13: Update the window features along with temporal correspondences in $Wc(n_p)$
- 14: **if** $n_p = n_{reset}$ **then**
- 15: Reset Wc
- 16: **end if**
- 17: Correct $Proj(B(n_p))$ for estimated windows in $Wc(n_p)$
- 18: Refine and update 3D point cloud $P(n_p)$ using corrected $Proj(B(n_p))$ and T_{proj}
- 19: Delete T_{proj}
- 20: Delete $Proj(B(n_p))$ and $Proj(B(n_p))_{INV}$
- 21: Store $Wc(n_p)$
- 22: $Wc(n_p - 1) \leftarrow Wc(n_p)$
- 23: Store $P(n_p)$
- 24: $P(n_p - 1) \leftarrow P(n_p)$
- 25: **return** $P(n_p)$

need to be corrected to deal with traces/residue, by replacing Equation (5.1) with the following definition:

$$(y_i - \bar{y}) = A(x_i - \bar{x}) + C(z_i - \bar{z}) \quad (5.4)$$

and modifying Equations (5.2) and (5.3) accordingly.

Although these 3D points are projected in 2D, a simple reference table T_{proj} is also generated that stores the index of the 3D points corresponding to the 2D points in the projection $Proj(B(n_p))$. This table T_{proj} only helps to correct and update the actual 3D point cloud once final processing is done (see Section 5.5 for details). It is finally deleted.

5.3.2 Watertight Boundary Estimation

Now, in order to obtain a watertight boundary of the projected building façades, we first extract the outline/envelope of the façade using a sweep scan method summarized in Algorithm 5. As the bottom part of the building outline close to the ground is often occluded and hence inconsistent due to the presence of different temporarily static or dynamic objects such as cars

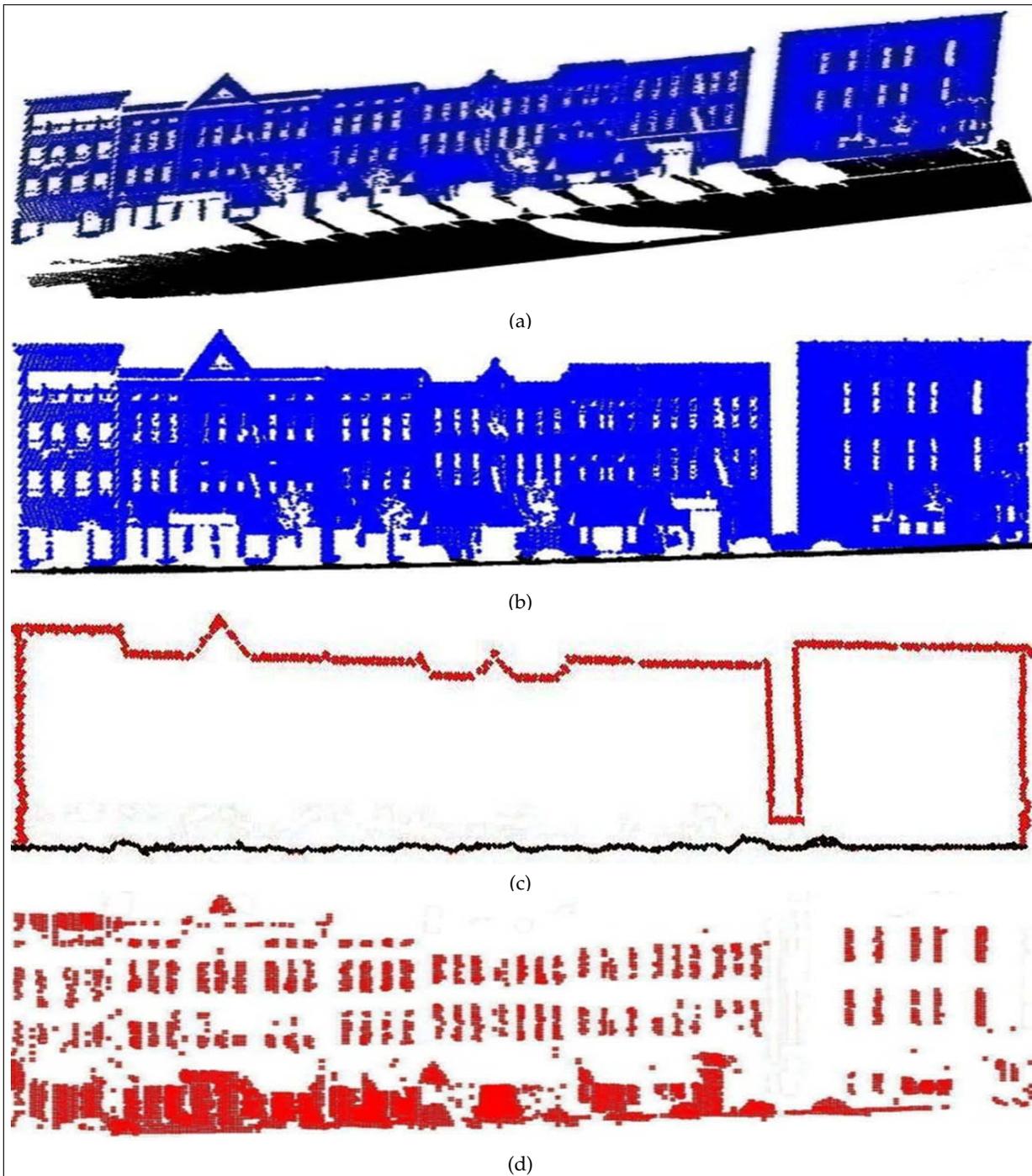


Figure 5.6: (a) 3D points belonging to building (in blue) and road surfaces (in black). (b) 2D projection of the 3D points on an arbitrary plane. (c) Watertight boundary. In red building outline closed by road projection (black) from below. (d) After point inversion within the watertight boundary.

or pedestrians in the scene (see Chapter 4, Section 4.5.3), we use the 2D projection of the road surface to close the watertight boundary from below as shown in Fig. 5.6(c).

Algorithm 5 Extraction of outlines.

Input: 3D points of object

- 1: **for** minimum x and y values of 3D points to maximum x and y values of 3D points **do**
 - 2: Scan building points in $x - y$ plane
 - 3: Find maximum and minimum values in z axis at small interval
 - 4: **end for**
 - 5: **for** minimum z values of 3D points to maximum z values of 3D points **do**
 - 6: Scan building points in z axis
 - 7: Find maximum and minimum value in $x - y$ plane at small interval
 - 8: **end for**
 - 9: **return** envelope/profile of object (set of all the minimum and maximum points)
-

The 3D road points are projected onto the same arbitrary plane along with the building façade points as explained in Section 5.3.1. However, no reference table is generated. Now, once the watertight boundary is extracted/obtained, we then perform a point inversion inside this boundary.

5.3.3 Point Inversion and Window Segmentation

As we are more interested in analyzing the holes in the building façade, we perform a point inversion within the watertight boundary as summarized in Algorithm 6. As a result of this point inversion, all holes including window holes surface out (a part of the segmentation process) as shown in Fig. 5.6(d) and we obtain an inverted projection $Proj(B(n_p))_{INV}$.

The 3D points belonging to each of these holes are then grouped together by means of a standard agglomerative clustering algorithm using a distance based Nearest Neighbor Search (NNS) to obtain segmented clusters. This distance S is taken as $2 \times D_i$, where D_i is the size of the small regular interval used in the point inversion stage. The value of D_i depends upon the resolution of the acquired data. A variation is usually detected in the density of data acquired by mobile terrestrial data acquisition systems (as shown in Fig. 5.7) first due to the varying speed of the LiDAR carrying vehicle itself as it sweeps the urban environment; and second due to farther objects (mainly in our case higher objects) that are scanned in lower resolution than the closer ones at/near ground level. Thus, in order to select an appropriate value of D_i , the points of the watertight boundary are analysed. In order to determine the variation along the height axis D_z , we analyse the points belonging to the two vertical edges of the boundary as shown in Fig. 5.6(c) while concerning the variation in the horizontal axis D_x , we consider the points belonging to the top horizontal edge of the boundary. The bottom edge is not used as it belongs to the road surface and not to the building façade. The values of D_z and D_x are found using the relations: $D_z = \frac{3}{2} \sqrt{\delta_{d_z}^2}$ and $D_x = \frac{3}{2} \sqrt{\delta_{d_x}^2}$. Here, d_x and d_z are the sets of values of the difference between two successive points (only height for D_z and width for D_x) in the selected vertical and horizontal edges respectively while δ^2 is the standard variance. The difference values in both sets are first sorted in increasing order before being used in the relations. The value of D_i is taken as D_z and D_x for vertical and horizontal axes respectively for point inversion (see Algorithm 6) and the calculation of S . The choice of using the watertight boundary compared to the whole façade to determine an appropriate value of S not only reduces computational burden but also gives a fairly good estimation due to the absence of window holes that could heavily bias the value.

The segmented clusters are then classified as windows based on the geometrical sizes. Very small clusters (max width and height less than $0.5 \text{ m} \times 0.5 \text{ m}$ respectively) corresponding to small holes due to acquisition errors and imperfections are automatically removed. Similarly very large clusters (max width and height greater than $5 \text{ m} \times 5 \text{ m}$ respectively) corresponding to

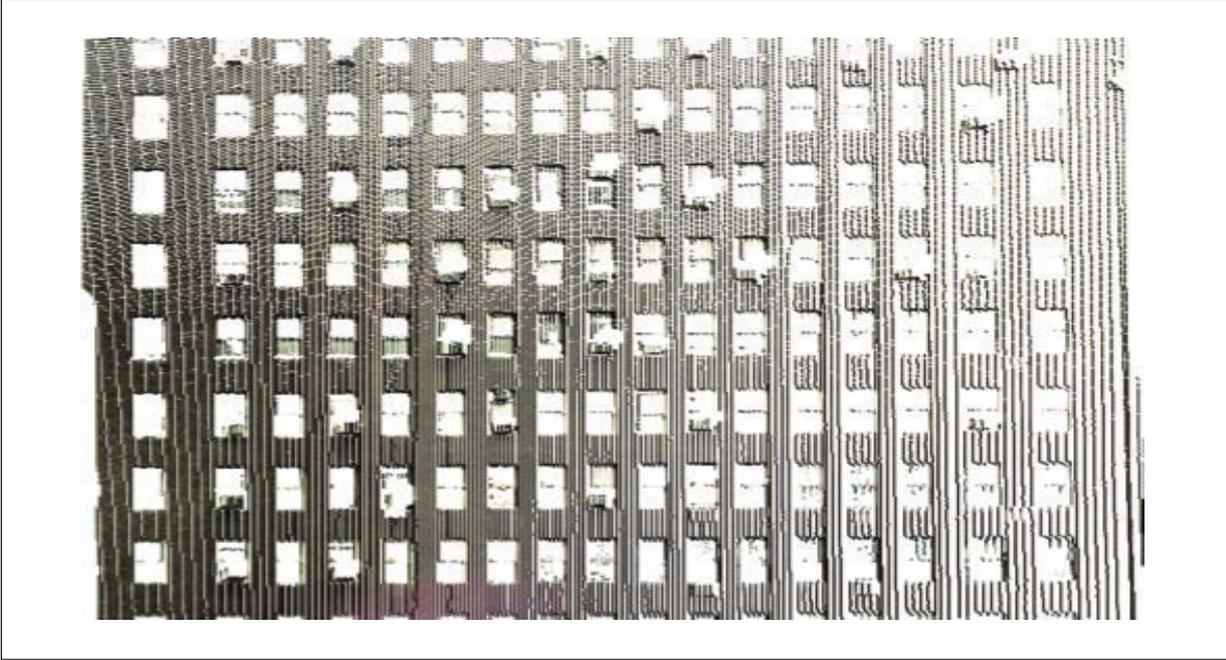


Figure 5.7: 3D point cloud of the building façade showing varying resolution. The point resolution decreases from left to right and from bottom to top.

holes mainly due to occlusions, are also removed. The remaining clusters are then considered as windows as shown in Fig. 5.8(c). Each detected window is then defined by a feature space: $W_F \in \{H, W, P, GC\}$. Here H and W are the maximum height and width of the cluster respectively while GC is the geometrical center. P is a set of points that define the outline/profile of the window. These sets of points are calculated by using the same sweep scan algorithm (now in 2D) presented in Algorithm 5 on each window.

This step of point inversion not only reduces significantly the data size for analysis (few points corresponding to holes rather than those of the whole façade) but also allows easy segmentation of windows. Also, in case of any reflected points from within the window region, represented by small empty spaces between the window points after point inversion, they are also catered for as the whole region is merged to form a window cluster.

Algorithm 6 Point inversion.

Input: 2D projected points of building façade, watertight boundary

- 1: **for** minimum x values to maximum x values of 2D points defined by the watertight boundary **do**
 - 2: Scan at small regular intervals D_x in horizontal x axis
 - 3: **for** minimum z values to maximum z values of 2D points defined by the watertight boundary **do**
 - 4: Scan at small regular intervals D_z in vertical z axis
 - 5: **if** 2D point found **then**
 - 6: Delete 2D point
 - 7: **else**
 - 8: Add a 2D point at the center of the interval
 - 9: **end if**
 - 10: **end for**
 - 11: **end for**
 - 12: **return** inverted point projection
-

5.4 Window Features Estimation

The windows detected in each passage are prone to different errors due to measurement errors, occlusions, etc. Hence, in order to estimate more accurately the detected windows along with their features/parameters, we then try to exploit both façade symmetry as well as multiple passages. In order to achieve this, we apply the ANOVA method and try to estimate the window features using both symmetrical windows in the façade in each passage as well as the corresponding windows in the same façade in successive passages.

If no symmetrical correspondences (similar windows in symmetrical pattern) are found for a window in the façade, then its features are only estimated by the corresponding window in successive passages. In this way, we are able to detect and estimate windows and their features even for non-symmetric building façades.

5.4.1 Exploiting Façade Symmetry

It is observed that usually building façades exhibit a structured arrangement consisting of repetitive patterns and self similarities. However, this is not always true due to rich architectural variations depending upon the type of buildings, countries and cultures.

In order to exploit the symmetrical properties of the façade (if possible), in each passage, we try to find symmetric correspondences (similar windows) on the building façade for each detected window. Symmetrical windows are found by simply analyzing the position of the windows on the façades by comparing the distance between their centers in both row and column. Windows at equidistant intervals in both row and column are considered symmetrical correspondences (at least 3 windows) and are grouped together as shown in Fig. 5.8(c). Symmetrical correspondences are determined for each window on the façade. In case of more than one type of symmetrical correspondences found for a particular window on the façade, the symmetrical correspondences with the smallest interval distances are considered.

Although symmetrically situated on the façade, their degree of similarity and how much these correspondences should contribute to estimate the features/parameters of a particular window are only ascertained after the ANOVA analysis.

5.4.2 Exploiting Multiple Passages

Once symmetrical correspondences for each window in the same façade are found in each passage, we then find corresponding windows in successive passages. As a result, we are able to estimate the window features more accurately, especially in case if we are not able to find any symmetrical correspondence due to either non-symmetry or occlusions. Similarly, the redundant data obtained via several passages are also used to complete occluded regions of the façade (including windows) as explained in Chapter 4. These passages (in the same place) could be on different days and at different hours of the day.

As the acquired 3D points are directly geo-referenced and then further co-registered in successive passages by applying Iterative Closest Point (ICP) method on the skylines of the building as presented in Section 4.5.4, there is no problem of finding corresponding windows for the same façade in successive passages. Windows found in the same position on the façade in successive scans are considered as temporal correspondences and are grouped together as shown in Fig. 5.9. Temporal correspondences are determined for each window on the façade.

Although these temporal correspondences usually represent the same window, only after the ANOVA analyses we are able to ascertain how much these correspondences should contribute to estimate the features/parameters of a particular window.

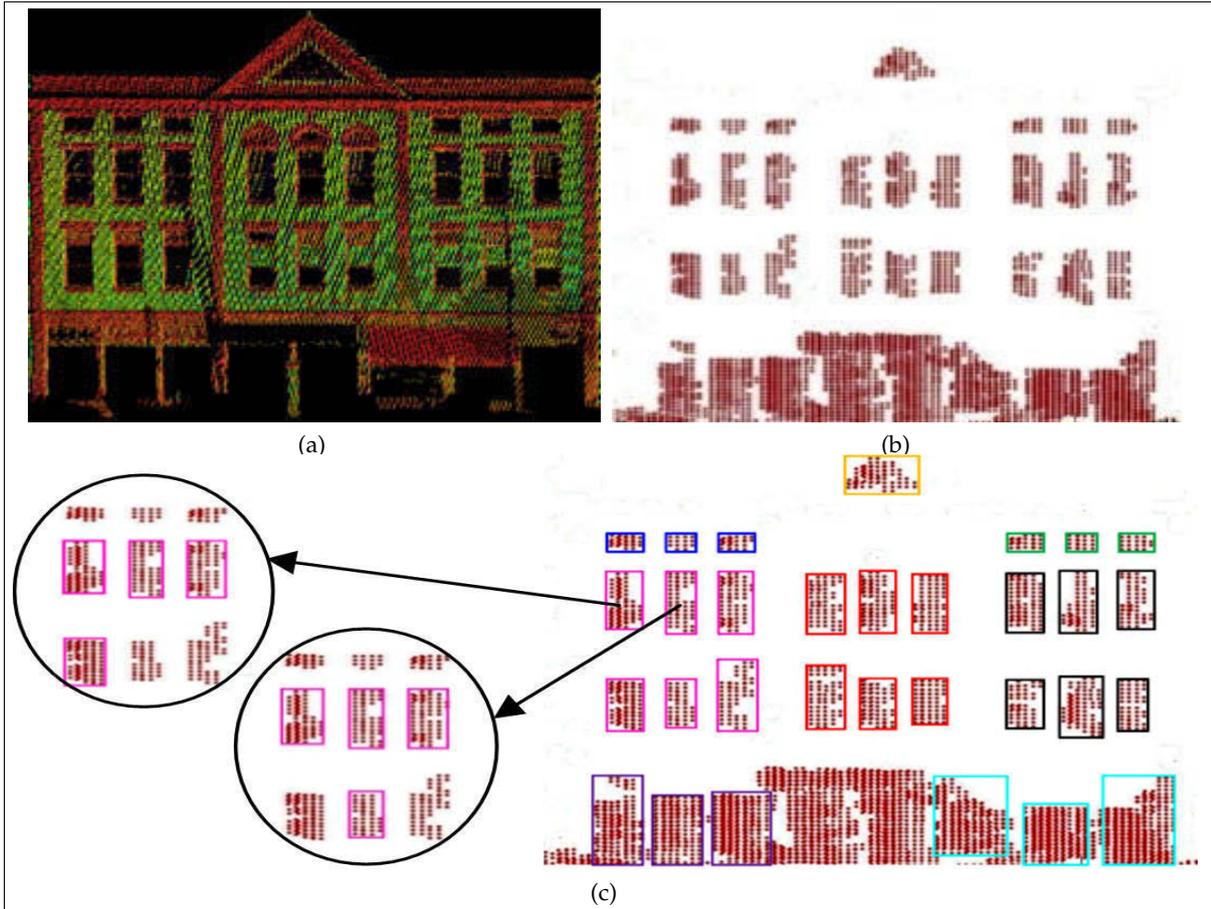


Figure 5.8: (a) 3D points of building façade. (b) Potential window candidates after point inversion and window segmentation. (c) Symmetrical correspondences in both row and column for first and second window respectively. Same colors (of the bounding box) represent similar windows.

5.4.3 ANOVA-based Estimation

In order to determine which correspondences should be used to estimate features/parameters for any particular detected window, we conduct a one-way ANOVA analysis [TF06] after adapting it to our requirements.

The statistic used in ANOVA partitions the variance into two components: 1) the between treatment variability, 2) the within treatment variability. Adapting them to our requirements, the within treatment variability SSW is used for symmetrical correspondences while between treatment variability SSB is used for temporal correspondences. The term “ SS ” in SSW and SSB is derived from the Sum of Squares method used in their calculation. It may be noted here that symmetrical correspondences are only considered for the current passage.

If n_w is the total number of detected windows (including both symmetric and temporal correspondences) in the total number of passages so far, G is the total number of passages so far, Y_{ip} is the value of Y^{th} feature for the i^{th} window in passage p , n_{wp} is the number of windows (only symmetrical correspondences) in passage p , \bar{Y}_p is the mean of the feature values for all the windows (only symmetrical correspondences) in passage p , s_p^2 is the sample variance of the feature values for all the windows (only symmetrical correspondences) in passage p , and \bar{Y} is the average of the feature values for all the windows (including both symmetric and temporal correspondences) in the total number of passages so far, then the total variabilities for the Y^{th}

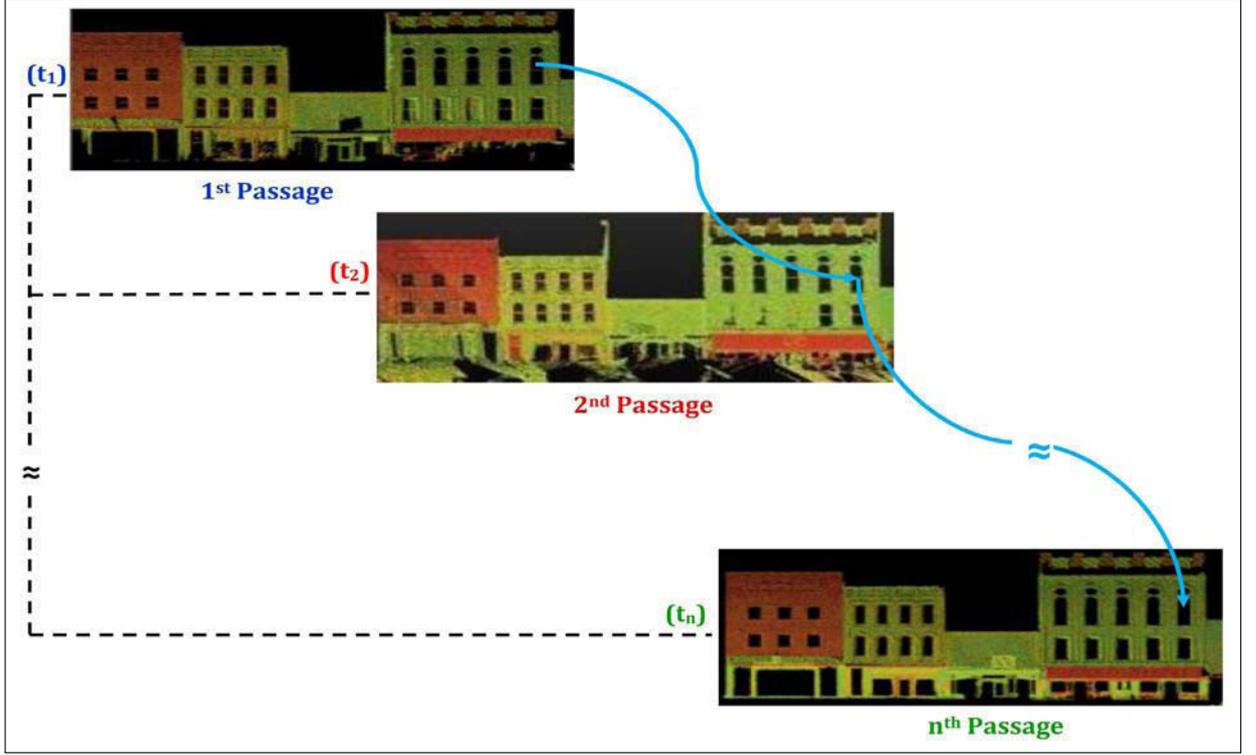


Figure 5.9: Finding Temporal correspondences for each window in multiple passages.

feature of the i^{th} window in passage p are given as:

$$SSW_{ip} = \sum_{p=1}^G \sum_{i=1}^{n_{wp}} (Y_{ip} - \bar{Y}_p)^2 = \sum_{p=1}^G (n_{wp} - 1) s_p^2 \quad (5.5)$$

Similarly,

$$SSB_{ip} = \sum_{p=1}^G \sum_{i=1}^{n_{wp}} (\bar{Y}_p - \bar{Y})^2 = \sum_{p=1}^G n_{wp} (\bar{Y}_p - \bar{Y})^2 = \sum_{p=1}^G n_{wp} \bar{Y}_p^2 - \frac{(\sum_{p=1}^G \sum_{i=1}^{n_{wp}} Y_{ip})^2}{n_w}$$

Further simplification yields:

$$SSB_{ip} = \sum_{p=1}^G n_{wp} \bar{Y}_p^2 - n_w \bar{Y}^2 \quad (5.6)$$

Using Equations (5.5) and (5.6), we calculate the average amount of variability within each passage MSW and the average amount of variability between successive passages MSB as:

$$MSW_{ip} = \frac{SSW_{ip}}{df_{within}} \quad (5.7)$$

$$MSB_{ip} = \frac{SSB_{ip}}{df_{total}} \quad (5.8)$$

where df_{within} and df_{total} are the degrees of freedom given as $(n - G)$ and $(G - 1)$ respectively. The F-statistic [TF06] is then calculated by computing the ratio of MSW and MSB :

$$F_{ip} = \frac{MSB_{ip}}{MSW_{ip}} \quad (5.9)$$

These values of F_{ip} are calculated independently for both width and height features of each detected window in each passage. The maximum value of F_{ip} , among the values obtained for both features, are then used for this analysis.

Now using df_{within} and df_{total} for each window, we determine the critical value F_c from the standard F-tables (see [Appendix B](#)) using $\alpha = 0.05$. Now using standard F-statistics, analyzing F_{ip} value of each window and comparing it with the corresponding F_c value help us to infer different weights to be used for the two types of correspondences (w_s and w_t for symmetrical and temporal correspondences, respectively) for window feature estimation. The choices of these weights are summarized in [Table 5.1](#).

As the null hypothesis being tested here is that windows in the symmetrical and temporal correspondences are similar, case 1 validates this hypothesis and hence both Symmetrical & Temporal correspondences are used to estimate the window features. In this table, case 2 suggests that the symmetrical correspondences do not contain a lot of similar windows (due to occlusions or lack of symmetry in façade). Thus, in such a case, temporal correspondences are given more preference (higher weights) over symmetrical correspondences to estimate window features. Case 3 suggests that there are strong inconsistencies in the windows detected in successive passages. As a result, the variability of features of the detected window is much higher in successive passages as compared to the symmetrical correspondences found on the façade. This could be due to false detections, occlusions in certain passages or even changes occurring in the building façade (see [Chapter 4](#)). Hence in this case, symmetrical correspondences are given more preference (higher weights) over temporal correspondences.

Table 5.1: Choice of weights for estimation.

Case	F_{ip}	Weights
1	$1 \leq F_{ip} \leq F_c$	$w_s = w_t = 1$
2	$F_{ip} < 1$	$w_t > w_s$
3	$F_{ip} > F_c$	$w_s > w_t$

Now in order to determine the weights associated with each correspondence for case 2 and case 3, we calculate the effect size measure ω^2 for ANOVA calculations. This measure, as expressed in [Equations \(5.10\) and \(5.11\)](#), gives us the contribution of each of the two correspondences in the total variability observed.

$$\omega_{t_{ip}}^2 = \frac{SSB_{ip} - (G - 1)MSW_{ip}}{SSB_{ip} + SSW_{ip} + MSW_{ip}} \quad (5.10)$$

$$\omega_{s_{ip}}^2 = 1 - \omega_{t_{ip}}^2 \quad (5.11)$$

Here $\omega_{t_{ip}}^2$ and $\omega_{s_{ip}}^2$ are the contribution of the temporal and symmetrical correspondences, respectively.

Once the weights for both correspondences are determined, the window feature value Y_{ip} (i.e. the Y^{th} feature for the i^{th} window in passage p) is estimated as \hat{Y}_{ip} :

$$\hat{Y}_{ip} = \frac{n_s \omega_{s_{ip}}^2 (\sum_{j=1}^{n_t} Y_{t_j}) + n_t \omega_{t_{ip}}^2 (\sum_{j=1}^{n_s} Y_{s_j})}{(n_s n_t) (\omega_{s_{ip}}^2 + \omega_{t_{ip}}^2)} \quad (5.12)$$

Here Y_t and Y_s are the values of the feature of the windows in temporal and symmetrical correspondences and n_t and n_s are the total number of windows considered as temporal and symmetrical correspondences respectively. Now using [Equation \(5.12\)](#), both width and height features of each detected window in each passage are estimated independently.

Now, in order to estimate the profile P of the window in the feature space, we first find a set of common profile points for each of the two correspondences by sampling both height and width dimensions at small regular intervals as shown in [Fig. 5.10](#).

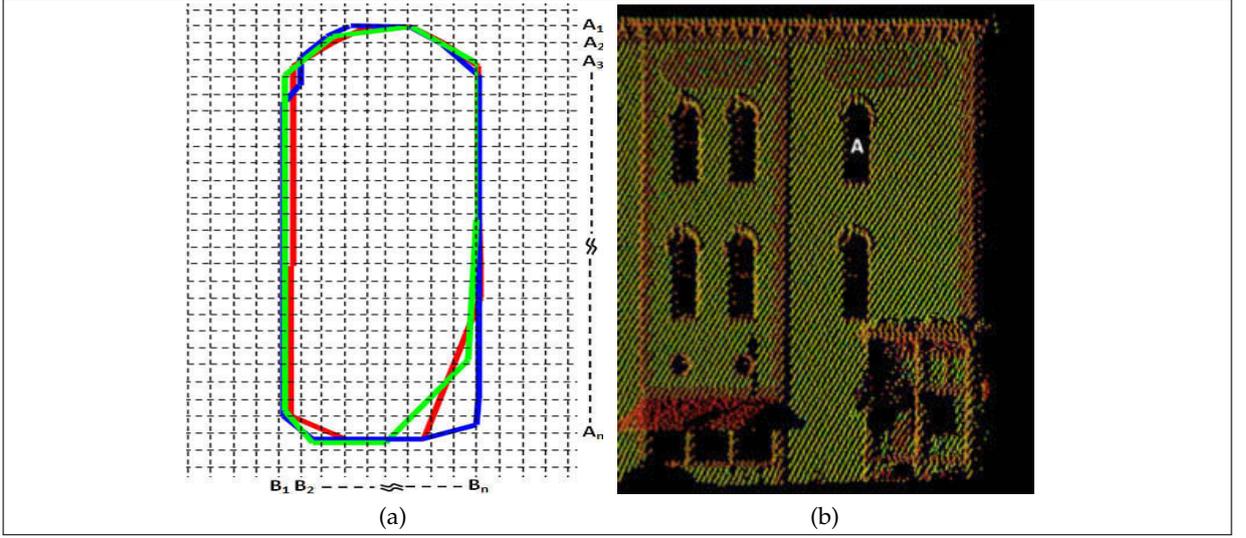


Figure 5.10: (a) shows the profile points (connected with straight lines) of temporal correspondences for window “A” marked on façade in (b). In (a) red, blue and green color represent the profiles obtained in 1st, 2nd and 3rd passage respectively. A_n and B_n are the set of common profile points (along the two axis) obtained for the 3 profiles at n^{th} regular interval (grid cell space). These points are then stored in pt_t , as these correspond to profiles of temporal correspondences.

It should be noted that for symmetrical correspondences we first align the windows by shifting their centers.

Each point in the set of common profile points is then estimated by using Equation (5.13) to obtain $\hat{P}_{ip} \in \{\hat{p}_{t_{ip_1}} \dots \hat{p}_{t_{ip_k}}\}$ such that:

$$\hat{p}_{t_{ip_k}} = \frac{n_s \omega_{s_{ip}}^2 (\sum_{j=1}^{n_t} pt_{t_{k_j}}) + n_t \omega_{t_{ip}}^2 (\sum_{j=1}^{n_s} pt_{s_{k_j}})}{(n_s n_t) (\omega_{s_{ip}}^2 + \omega_{t_{ip}}^2)} \quad (5.13)$$

where pt_t and pt_s are the sets of common profile points corresponding to temporal and symmetrical correspondences and k is the total number of points in \hat{P}_{ip} , i.e. the estimated profile of the i^{th} window in passage p .

Now using Equations (5.12) and (5.13), the features of the window are estimated and then updated for that window along with the list of temporal correspondences in Wc as described in Algorithm 4.

It must be noted here that in order to avoid initialization issues, for the first passage only symmetrical correspondences are considered whereas for all successive passages both symmetrical as well as temporal correspondences were considered in the equations for estimation.

5.5 Refining and Updating the Building Façades in the 3D Point Clouds

Once the feature space of the detected windows is estimated and updated in Wc , the next step is to correct, refine and update the 3D façade in the actual 3D point cloud. In order to do this, we first correct the 2D projection $Proj(B(n_p))$ by using the estimated profile of all the windows. These estimated profiles define the window space and hence all points within these profiles are considered as window points.

In many cases, the 3D point clouds obtained using LiDARs mounted on mobile terrestrial vehicles offer limited viewing angles and relatively sparse 3D points for distant windows. As a

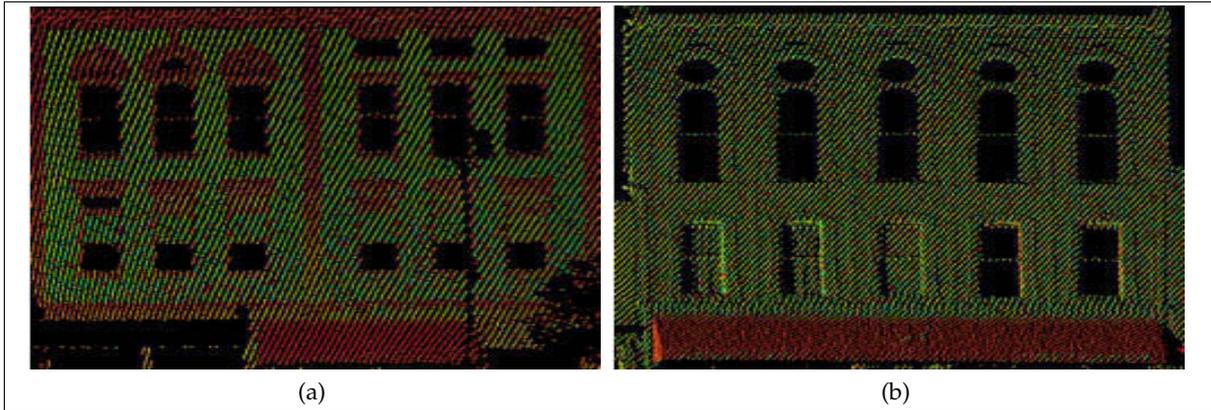


Figure 5.11: (a) & (b) show partial and completely shut/closed windows on the façade in one of the passages respectively.

result, fine details pertaining to window panes, etc., within the window regions remain missing (see Fig. 5.10(b)). Hence, we prefer to delete the within window points to obtain well-defined window holes in the 3D façades. On the other hand, if co-registered camera images are also available, then a 2D image of the window can be added to the window region (defined by the estimated profile boundary) to provide a more realistic/detailed view [BH07b]. However, for most applications pertaining to accurate 3D urban modeling, geographical navigators, etc., well-defined window-hole features on the building façade are more than sufficient.

Once the 2D points within the window region defined by the estimated window profile are deleted in $Proj(B(n_p))$, corresponding 3D points in the actual point cloud are also deleted using the reference table T_{proj} as shown in the results in Section 5.7. This processing ensures that the resulting building façade in the 3D urban point clouds contains accurate, well-defined window regions.

5.6 Automatic Checks and Balances

In case of any false detection, normally no symmetrical correspondences should be found on the building façade. However, in order to confirm whether this really is a false detection and not a non-symmetric feature of the façade, we also check the temporal correspondences. In case no temporal correspondences are found, this detection is deleted from the list of windows W_c and is no longer used for refining the building façade. On the other hand if a certain detected window was found missing in any of the passages due to temporary occlusion, etc., it is still considered, as correspondences from other passages are used to estimate the window parameters and hence it remains in W_c and is also used for refining the building façades.

Similarly, we see that although effects of occlusions (both complete or partial) are greatly reduced due to the exploitation of multiple passages as explained in Section 5.2, sometimes certain window features remain partially or fully obscured due to window shutters or curtains, etc., in some passages, as shown in Fig. 5.11. This may result in an inaccurate estimation of window parameters. However in our method, the increase in the variability of any of the correspondences, in such cases, results in its lower weighting in the parameter estimation of that window according to Equations (5.12) and (5.13). This makes our method more robust and ensures a more accurate estimation of window parameters.

These checks and balances ensure that with the number of passages, window detection accuracy improves and the final building façade only contains the exact and accurate window features.

5.7 Results, Evaluation and Discussion

In order to validate our method, we used two different datasets:

1. New York City dataset Mesolongitis and Stamos [MS12]: to evaluate and compare our window detection method with a state of the art method.
2. 3D Urban Data Challenge 2011 dataset [Cha11]: to fully evaluate the performance of our method.

5.7.1 Evaluation

We first evaluated the window detection part of the proposed method using the New York City dataset presented and used by Mesolongitis and Stamos [MS12]. The dataset contains a rich collection of 3D façades from the New York City area obtained from a single passage. Several building façades were used to evaluate our method. Some of the results are shown in Fig. 5.12. The results obtained (overall Precision Pr and Recall Re values) are comparable to the state of the art method [MS12] evaluated on the façades of the same dataset as presented in Table 5.2. Our method received lower Re scores compared to the state of the art method on façades with occlusions (as shown in Fig. 5.12(a)-(c)) given the fact that the latter predicts windows in the occluded regions based on the symmetry of the façade. As for our method, it exploits multiple passages (which were not available in this data set) to manage occlusions. Our own scores improve considerably with multiple passages, exploiting temporal correspondences as discussed later on for the second dataset. Besides, contrary to our method, the state of the art method generated some false positives on certain façades with varying symmetry or in cases of interfacing façades as shown in Fig. 5.5. On the other hand, the relatively fewer false positives generated by our method are easily corrected exploiting temporal correspondences.

Table 5.2: Comparison of detection result.

Method	Pr	Re
Mesolongitis and Stamos [MS12]	0.894	0.810
Proposed Method	0.92	0.73

In order to fully evaluate our method, the dynamic dataset of the 3D Urban Data Challenge 2011 was used. This dataset contains 4 sets of the same dynamic scenes in downtown Lexington, Kentucky, USA obtained on different days and at different times. The data was acquired by the Vis Center’s (University of Kentucky) LiDAR Truck [Vis11] containing two Optech LiDAR sensor heads (high scan frequency up to 200 Hz), a GPS, inertial measurement unit and a spherical digital camera.

Figures 5.13, 5.14 and 5.15 show some qualitative results of window detection and refinement of different building façades in this dataset. The method successfully detects and estimates windows of different types and sizes. Figure 5.13 illustrates how the method is able to successfully estimate window parameters in case of heavy occlusions (especially shop windows at ground level), where most other methods would probably fail [LN04]. Similarly the method works equally well in case of windows high above the ground (represented by sparser 3D points) as well as repeated window features (see Fig. 5.14). Figure 5.15 demonstrates that the proposed method successfully detects and estimates windows parameters pertaining to building façades with different non-symmetric window patterns. As the method is non-parametric (no assumption on window shapes) and does not rely solely on façade symmetry, it is able to estimate window parameters more accurately than conventional methods relying on façade symmetry only.

For quantitative analysis, we evaluated the accuracy of the window detection. The n_{reset} value was fixed at 4 as the maximum available number of passages in the dataset was 4. However,

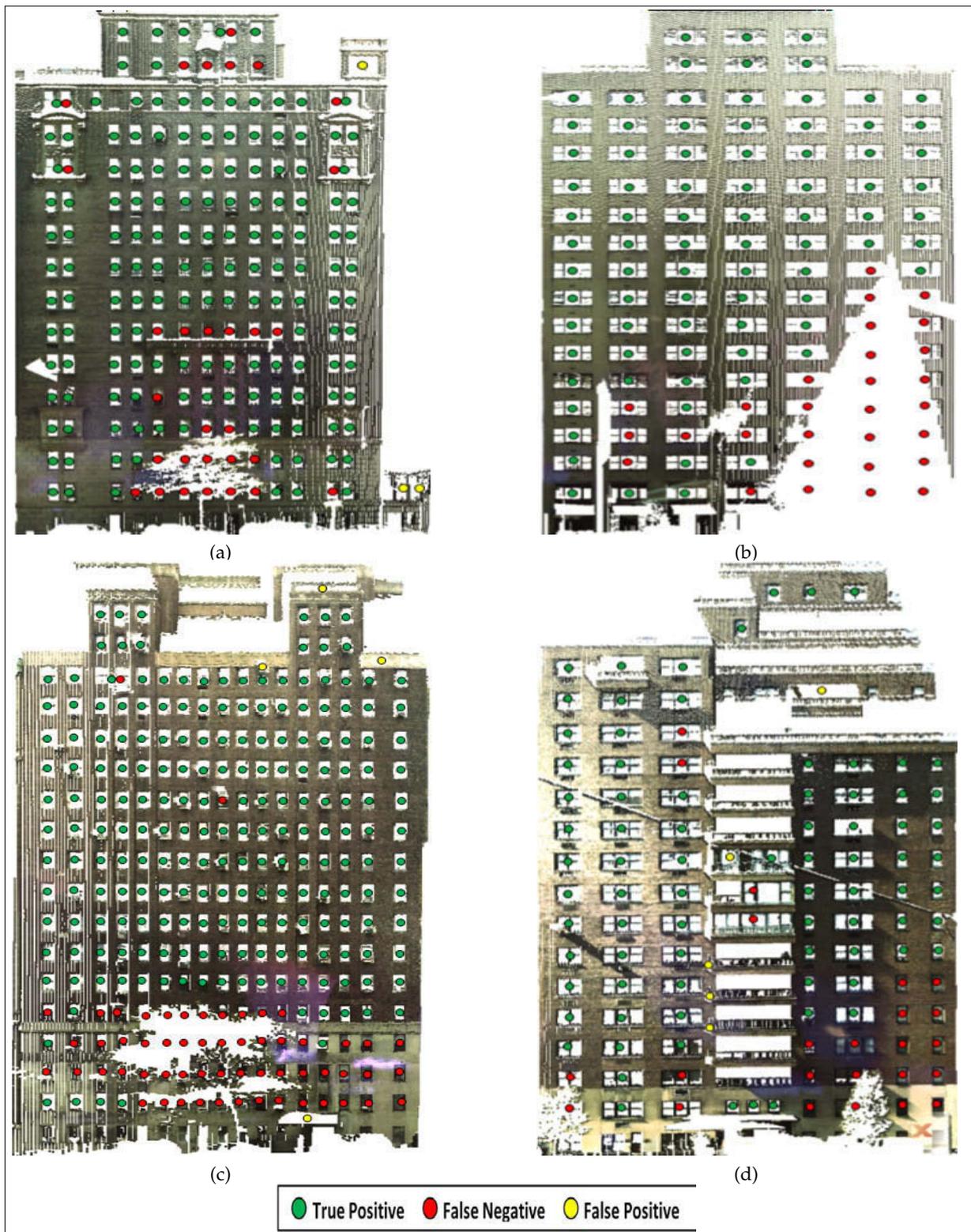


Figure 5.12: Window detection results are shown for some façades in the New York City dataset Mesolongitis and Stamos [MS12]. In (a), (b) & (c) we can see a number of false negatives due to occlusions. In (c) & (d) we also find some false negatives due to closed windows because of window shutters or curtains.

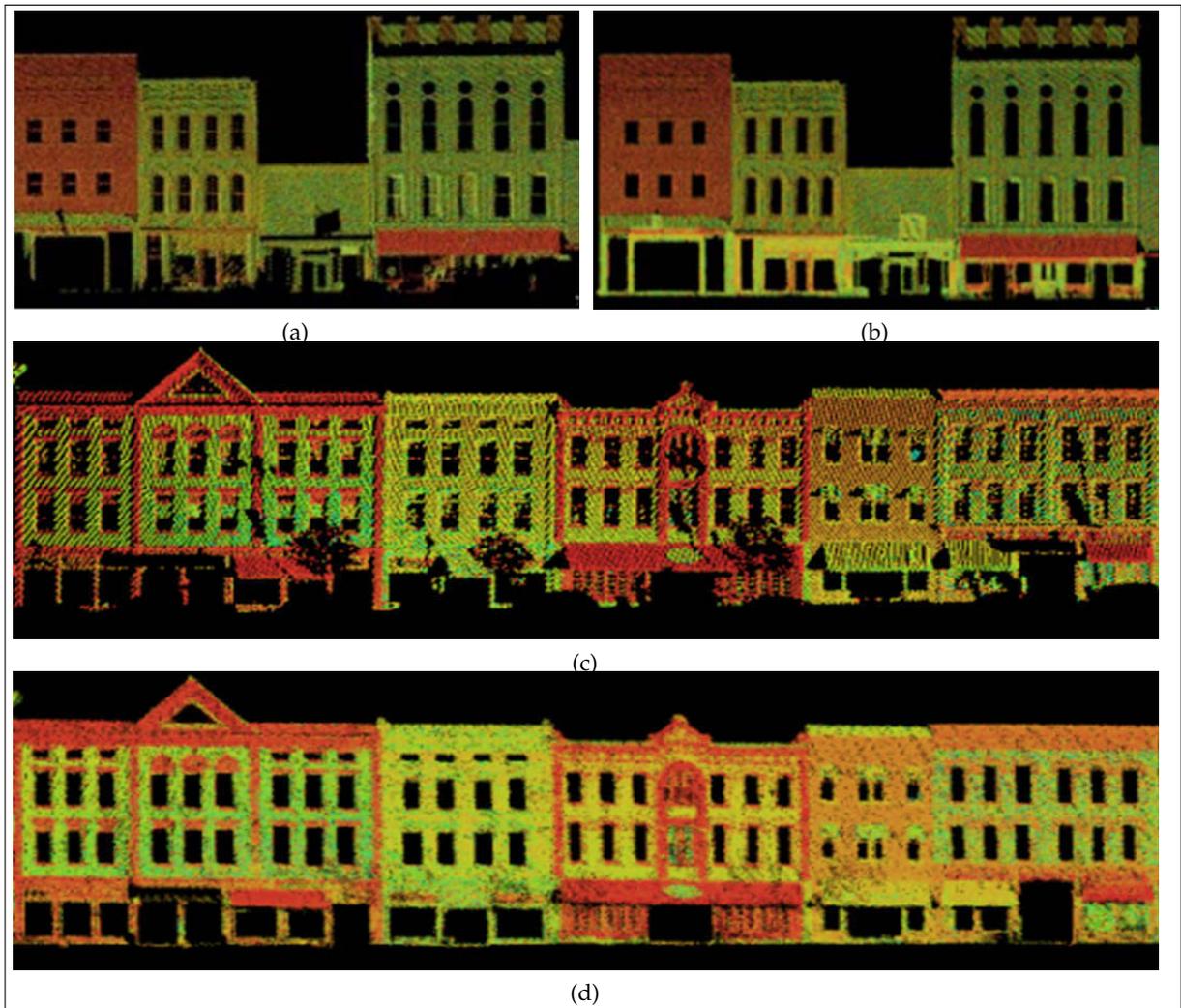


Figure 5.13: (a) & (c) show raw 3D points (color coded intensity) of different building façades from the dataset containing windows of different types and sizes in the 3D Urban Data Challenge 2011 dataset [Cha11] while (b) & (d) are the corresponding 3D point clouds after window detection and refinement. The results clearly demonstrate the efficacy of the method in case of heavy occlusions especially at ground level shop windows.

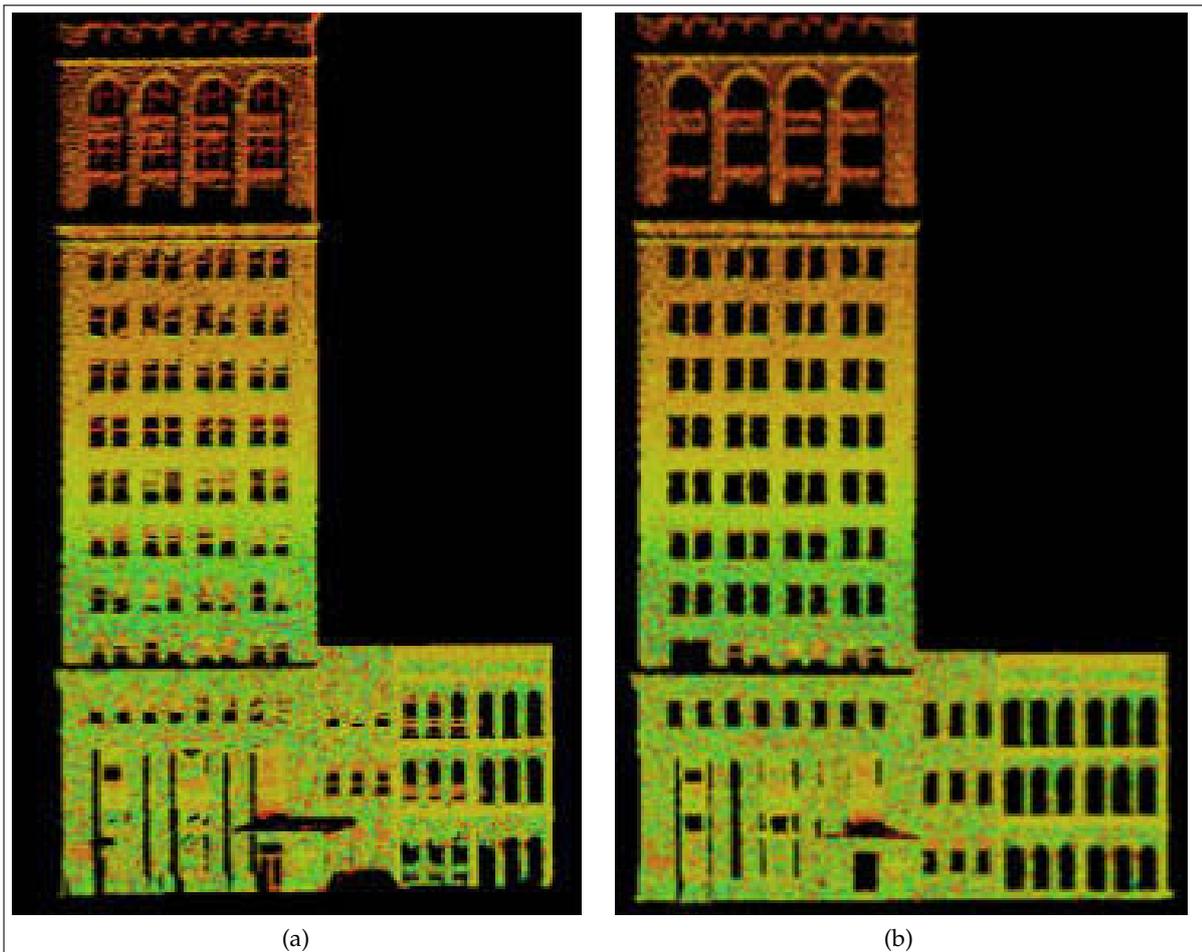


Figure 5.14: (a) show raw 3D points (color coded intensity) of building façade of a long multi-story building containing windows of different types and sizes in the 3D Urban Data Challenge 2011 dataset [Cha11] while (b) is the corresponding 3D point clouds after window detection and refinement. The results clearly shows that the method works equally well for repeated window structures.

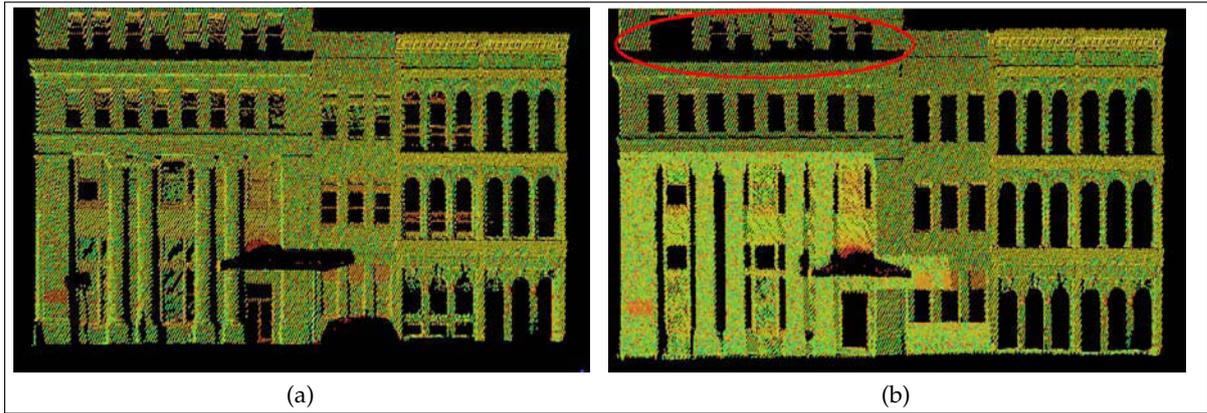


Figure 5.15: Successful detection and estimation of window parameters pertaining to non-symmetric building façades with different non-symmetric window patterns. (a) shows the raw 3D points of a building façade with different non-symmetric window patterns. In (b) the same façade after successful window detection and refinement can be seen. The red ellipse marks a permanently occluded region discussed in § 5.7.2.

where this reset can be done by human intervention, or manually set (as in our case) it could also be done automatically as presented in Section 4.5.4. The reset function only helps to avoid unnecessary memory storage issues, once the desired results are obtained. The window detection method was evaluated using different standard evaluation metrics as described in [Vih12] (see Table 5.3). Ground truth was obtained by visual inspection of both raw 3D point clouds as well as the available 2D camera images of the building façades. The results clearly demonstrate the efficacy of the proposed method.

Table 5.3: Window detection evaluation using metrics defined in [Vih12].

Evaluation Metrics		Results
Accuracy	ACC	0.917
Positive Predictive Value	PPV	0.971
Negative Predictive Value	NPV	0.908
False Discovery Rate	FDR	0.029
F_1 measure	F₁	0.912

Figure 5.16 shows the evolution of window detection performance expressed in terms of the different evaluation metrics, mentioned above, with the increasing number of passages. For analysis, one particular street was selected (about 1000 meters long with over 200 distinct windows of different shapes and sizes). All windows including occluded ones were taken into consideration. For the first passage only, symmetrical correspondences were considered whereas for all successive passages both symmetrical as well as temporal correspondences were considered. The initial low scores and high **FDR** (False Discovery Rate) value are due to the large amount of occluded windows, especially the shop windows at ground level as shown in Fig. 5.13(a) and 5.13(c). With every new passage, different occluded regions were completed and hence the window parameters were better estimated, resulting in an improvement in the window detection performance. Also contributing to the low score in the first passage were some unique or non-symmetric windows present on some façades. These were only added to the detected window list W_c , after their temporal correspondences were found in the successive passages.

It is also observed that the **PPV** (Positive Predictive Value) is generally found higher than **NPV** (Negative Predictive Value), which suggests that the method is more conservative and is more likely to detect and estimate windows once highly certain, otherwise it does not. Hence,

false positives usually occur less often than false negatives (also evident in Fig. 5.12).

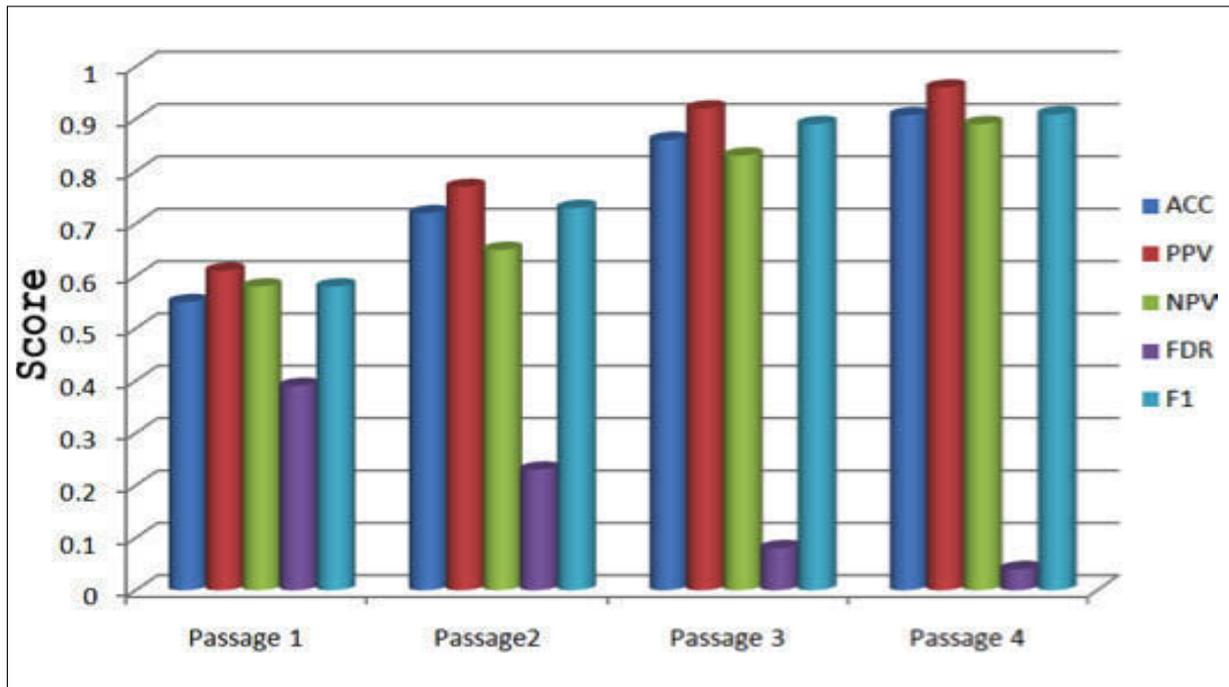


Figure 5.16: Evolution of window detection performance expressed in terms of the different evaluation metrics with the increasing number of passages.

5.7.2 Limitations

Although the method successfully detects windows of different shapes and sizes, and estimates their parameters even in difficult circumstances such as temporary occlusions and non-symmetrical façade structures, it has some limitations in case of permanent occlusions resulting in permanently missing 3D points, for example part of the building façade (zone marked in red) in Fig. 5.15(b). This permanently occluded region, due to the limited viewing angle of the data acquisition vehicle, results in large, deformed and interconnected window segments after point inversion and hence are not considered as a window nor estimated. Sometimes smaller permanent occlusions could also result in false positives. In such conditions, contrary to our algorithm, a classical symmetry-based window detection algorithm can assume/predict (which may be wrong) windows in the region based on the overall symmetry of the façade.

5.8 Conclusion

In this chapter a new method has been presented that automatically detects windows and estimates their parameters in 3D LiDAR point clouds obtained from mobile terrestrial data acquisition systems in the urban environment. The main feature of this method is that it combines information from both symmetrical and temporal correspondences using ANOVA measurements. This unique fusion of information not only accommodates for lack of symmetry but also helps complete missing features due to occlusions. This not only increases its robustness but also ensures a more accurate estimation of window parameters. These estimated windows are then used to refine 3D building façades for different applications pertaining to accurate 3D urban modeling or geographical navigators that frequently update their database. The described method also ensures that window detection and estimation accuracy improve with a growing number of passages and the final building façade contains only the exact and accurate window features.

The method applied to real data represents a significant advance in the state of the art and proves its effectiveness in case of non-symmetrical façades and temporary occlusions in the urban environment. The results evaluated using different standard evaluation metrics clearly demonstrate the efficacy (with values such as Accuracy **ACC** and F_1 measure **F₁** greater than 90%) as well as the technical asset of the method.

This approach is also an essential step towards the creation of façade descriptors which can be used in 3D registration, reconstruction, compression and scene understanding.

CONCLUSION AND FUTURE WORK

This thesis presents a scalable framework for automatic 3D urban cartography which incorporates recognition and temporal integration. It first discusses in detail the current practices in the domain of 3D urban cartography along with the different methods, applications, recent data acquisition and mapping technologies as well as the different problems and challenges associated with them. The work presented addresses many of these challenges mainly pertaining to classification of urban environment, automatic change detection, efficient updating

of 3D urban cartography and semantic analysis of the urban environment, etc. In this chapter we conclude the thesis. The chapter reiterates the basic motivation of the work presented along with its context. It is divided into two main sections. In the first section, we summarize all our main contributions while in the second section we give some directions for future research. These perspectives include several research tracks that could not only be investigated for the extension of these works but also open doors to different important applications and prospective research projects.

In this thesis, we have presented a scalable framework for automatic 3D urban cartography which incorporates recognition and temporal integration. We have first discussed in detail the current practices in the domain 3D urban cartography along with the different methods, applications, recent data acquisition and mapping technologies as well as the different problems and challenges associated with them. The work presented addresses many of these challenges mainly pertaining to classification of urban environment, automatic change detection, efficient updating of 3D urban cartography and semantic analysis of the urban environment, etc. The proposed methods ensure that the resulting 3D cartography contains only the exact, accurate and well-updated, permanent features of the urban environment. These methods are validated on real data obtained from different sources in different environments. After extensive evaluation at different stages, the results not only demonstrate the efficiency, scalability and technical prowess of the method that can both be easily integrated and ideally suited for different commercial or non-commercial applications pertaining to urban landscape modeling and cartography requiring frequent database updating.

To conclude, we summarize our main contributions and give directions for future research work.

6.1 Summary

The main aim of this PhD thesis is to address the different problems currently faced in 3D urban cartography by developing new methods of 3D cartography using 3D LiDAR data acquired from terrestrial mobile mapping systems. As discussed in this thesis, in the urban environments the quality of the data obtained from Mobile Mapping Systems (MMS) are widely hampered due to the presence of temporarily static and dynamic objects (pedestrians, cars, etc.) in the scene. So, the first step for 3D urban cartography is to obtain the permanent part of the urban environment. Thus, our first main contribution is a new method to classify the urban environment into permanent and temporary object classes. The proposed method first uses a super-voxel based segmentation employing a new link-chain method followed by classification of these segmented objects into basic object types using local descriptors and geometrical models. To help evaluate such methods a new evaluation metric is also introduced which incorporates both segmentation and classification results together. The results show an overall segmentation accuracy (OSACC) of 87% and overall classification accuracy (OCACC) about 90%. The method is not only extensively evaluated but also compared with existing state of the art methods. The results indicate that with good segmentation, a simplified classification method like the one proposed is sufficient. Our study shows that the classification accuracy improves by reducing voxel size but at the cost of processing time. Thus, a choice of an optimal value, as discussed, is recommended. The study also demonstrates the importance of using laser reflectance intensity values along with RGB colors in the segmentation and classification of urban environment as they are more illumination invariant and more consistent.

Once classified into these basic object types, these objects are then further characterized into permanent and temporary based on their basic characteristics and using a point matching technique. The objects classified as temporary are then removed from the 3D point cloud leaving behind a perforated 3D point cloud of the urban environment. These perforations along with other imperfections are then analyzed and progressively removed by incremental updating exploiting the concept of multiple passages, which is our second main contribution. The proposed method for automatic 3D urban cartography progressively removes different imperfections caused by occlusions, misclassifications of objects in the scene and ineffective incorporation of changes occurring in the environment over time, by taking advantage of incremental updating using specialized functions. Different changes occurring in the urban landscape are automatically detected and analyzed using cognitive functions of similarity, and the resulting 3D cartography is modified accordingly. The proposed method ensures that the resulting 3D point cloud of the

cartography is quite reliable, and it contains only the exact and actual permanent features, free from imperfections. The results evaluated on a real dataset using different evaluation metrics demonstrate the technical prowess of the method. The thorough analysis exhibits accurate change detection (including values of overall accuracy, ACC, and a Matthew's Correlation Coefficient, MCC, greater than 85% and +0.6, respectively), improved classification results (with a standard F_1 -Score > 0.98) and high accuracy of the completed permanent features/regions of the urban cartography. The results also show that the method is scalable and that it can both be easily integrated and ideally suited for removing various imperfections for different commercial or non-commercial applications pertaining to urban landscape modeling and cartography requiring frequent database updating.

Our third main contribution is that the proposed method of temporal integration also helps in improved semantic analysis of the urban environment, especially building façades. This, in the context of the thesis, being a logical continuation of our work, not only helps to refine the quality but also increases the semantic richness of the resulting 3D urban cartography extending its suitability for broader range of applications. The proposed method automatically detects windows and estimates their parameters in 3D LiDAR point clouds obtained from mobile terrestrial data acquisition systems in the urban environment. The main feature of this method is that it combines information from both symmetrical and temporal correspondences using ANOVA measurements. This unique fusion of information not only accommodates for lack of symmetry but also helps complete missing features due to occlusions. This not only increases its robustness but also ensures a more accurate estimation of window parameters. These estimated windows are then used to refine 3D building façades for different applications pertaining to accurate 3D urban modeling or geographical navigators that frequently update their database. The described method also ensures that window detection and estimation accuracy improve with a growing number of passages and the final building façade contains only the exact and accurate window features. The results evaluated using different standard evaluation metrics clearly demonstrate the efficacy as well as the technical asset of the method. The results also prove the methods effectiveness in case of non-symmetrical façades and temporary occlusions in the urban environment.

6.2 Directions for Future Research

In this thesis, we have presented a variety of methods to overcome different problems of the current state of the art in automatic 3D urban cartography. However, the challenging task of generating and updating 3D maps and models with increased effectiveness and improved efficiency is by far not solved yet. In the following paragraphs, we thus conclude our work with suggestions for future research directions.

The proposed method of super-voxel based segmentation helps in intelligent data reduction, which is an effective step for handling large 3D point clouds. However, at the moment, the method is implemented in sequential manner whereas it has the potential to be implemented as parallel processes, especially the link-chain step which can further reduce the processing time.

Another interesting aspect that needs to be explored is the effect of incorporating different intelligent data reduction/subsampling methods as presented by Puttonen *et al.* [PLK+13] in the voxelization process. This might help to not only further reduce the effective data size along with the over-segmentation but also help to better estimate the features of each voxel for super-voxel transformation. This could allow the use of smaller voxel sizes to increase the segmentation and classification accuracy without significantly increasing the processing/computational cost. In the proposed method, the values of certain parameters used such as maximum voxel size and inter-distance constant are determined using a semi-supervised approach. The choice of these parameters can be automatically optimized using metaheuristic algorithms as presented by Madic, Markovic, and Radovanovic [MMR13] and Lee and El-Sharkawi [LES08]. Further-

more, for the classification of segmented objects, contextual features/information could also be added in addition to the geometrical features and local descriptors to further improve upon the classification results.

The common practice of using terrestrial mobile mapping systems for 3D urban cartography allows the efficient and effective construction of 3D city maps and realistic models. However, due to the limited access of such vehicle-borne mobile mapping systems certain areas of the urban environment remain un-scanned such as adjoining alleys, small streets, stepped or narrow passages as shown in Fig. 6.1. One way to tackle this problem could be to use smaller portable mapping systems [BZF12; Lei14] to scan such areas and then merge the 3D point clouds together to complete the cartography. The problem associated with this is that most of such areas are GPS devoid and hence we would have to rely on fairly robust and accurate SLAM (Simultaneous Localisation and Mapping) techniques.



Figure 6.1: (a), (b) & (c) show examples of adjoining alleys, small streets, stepped or narrow passages that are difficult for vehicle-borne mobile mapping systems to access.

The change detection method for the urban environment should be tested and adapted to also detect changes due to climatic and seasonal changes. This is very important for long-term processing as one needs to reliably differentiate between structural changes from the climatic and seasonal changes for effective update of 3D maps and models. One aspect that would be useful in this regard is to incorporate image information with the 3D point cloud to obtain a more detailed feature space.

Beyond the purpose of the updating, the change itself is also of much interest. The presented method can thus also be adapted for not only monitoring of visible damage to public infrastructures such as bus stops, street benches, dustbins, light and electricity poles, but also to determine whether any of these has malfunctioned and needs to be repaired as essentially required by different public facility departments for maintaining large urban sites [LTA14]. In this case, the 3D model or point clouds can be built right after these infrastructures are installed and then change detection is performed periodically to monitor these infrastructures.

Similarly, the proposed method can also be extended to indoor environment. The cognitive models of similarity employed can provide insight on the type of changes taking place. If tailored properly, this could be quite useful for different applications pertaining to surveillance of airports, offices and factory floors, etc. With the rapid success of handheld 3D mapping systems like Zebedee [BZF12], this seems increasingly viable nowadays. The automatic window detection and feature estimation method successfully handles non symmetric façades and partial occlusions very well. However, there are two aspects that still need to be addressed: i.e. limitation due to permanent occlusions and realistic rendering of window regions. One practical solution for the first issue is to incorporate predictions based on symmetrical patterns [FS13] along with the ANOVA-based fusion. Another option is to use co-registered images to obtain insight into these regions. This would be more than useful, especially in regions with relatively sparse 3D points.

This would also help in completing/determining the missing finer details of the window panes, etc., within the window regions (which are usually not possible with only LiDAR data) or a 2D image of window can be directly added to the window region (defined by the estimated profile boundary) to provide a more realistic/detailed view [BH07b].

The proposed method can also be adapted to indoor environment. This could be essential for 3D indoor semantic mapping that can not only be used for different SLAM applications but also for construction of building plans with different access points (including doors and windows) and escape plans to help people with reduced mobility.

Incorporating information from thermal camera into the proposed method can help obtain a detailed thermal analysis of building structures to analyze heat losses, etc. On the other hand, the thermal patterns can also be used to better estimate the window detections and dimensions (windows allow more heat loss as compared to wall structures, etc., as shown in Fig. 6.2).

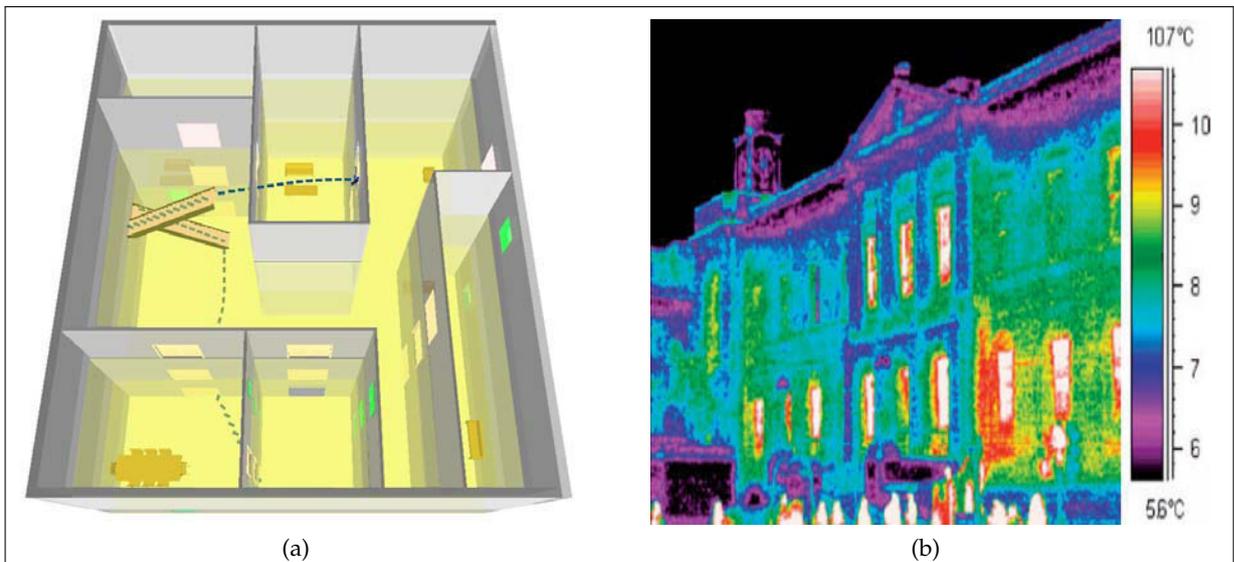


Figure 6.2: (a) shows semantic mapping of the interior of a building. Different access points including doors (in white) and windows (in green) are also shown. (b) presents the thermal image of a building (from outside). It can be clearly seen that there is maximum heat loss from the window regions.

As we have discussed in the thesis, the idea of multiple passages is essential for effective updating of databases for different geographic navigators and other urban analysis and modeling applications. We have also shown that this not only improves the quality of the cartography but also helps in the semantic analysis of the urban environment. However, the process of multiple passages can be made more efficient if the data acquisition sensors are mounted on top of public transport like buses, tramways, etc., as shown in Fig. 6.3. This would allow easy repetitive data acquisition of the different urban city environments as, usually, such modes of public transport move (both directions) on the same prescribed routes, on very frequent basis. This would eliminate the need for planning and executing special missions to acquire the requisite data. This could not only facilitate the process but also make it more cost effective.

The idea of using sensor-borne public transport to update databases of different geographic navigators and city models in near real-time, both 3D and 4D (as in the case of Project iSpace&Time), is a new idea and being a logical continuation of Project iSpace&Time (see Chapter 1) merits serious consideration. Such models would be made accessible to a wide public who can not only enthusiastically view the evolving real-like representation of the city centers but also use different associated information (as presented in Fig. 6.4) such as traffic flow, available parking spaces and pedestrian movements to plan their different daily tasks.

The work presented in this thesis also lays the foundation/basis of the concept of collaborative



Figure 6.3: (a) & (b) show different sensors mounted on a bus and tramway respectively. These sensors usually include GPS, IMU, cameras and LiDAR sensors.

3D urban cartography. The concept of collaborative mapping has recently gained some interest in the scientific community. The applications like OpenStreetMap (OSM) and Google Map Maker allow the grand public to propose the creation, edition and updating of maps which after verification, are updated in the map database. This offers relatively less technical difficulties as the maps are in 2D, with lesser associated accuracies and limited allowable input data formats. Recently, with the advancements in Structure-from-Motion (SfM) technique, some work is being done by using large number of camera images to reconstruct large urban scenes and more realistic representations of the cities [USM+13]. The amount of images required to reconstruct a whole city and not only its landmark buildings necessitate collaborative data acquisition. In case of 2D images, this task is facilitated by the widespread adoption of modern smartphones equipped with high quality cameras and GPS sensor. Similarly, Google also uses its specialized cars equipped with panoramic cameras in a similar manner to generate and update their Google street view application as presented in Chapter 2. The concept of using 3D sensors for collaborative 3D urban cartography is new and according to the best of our knowledge no work on this is presented

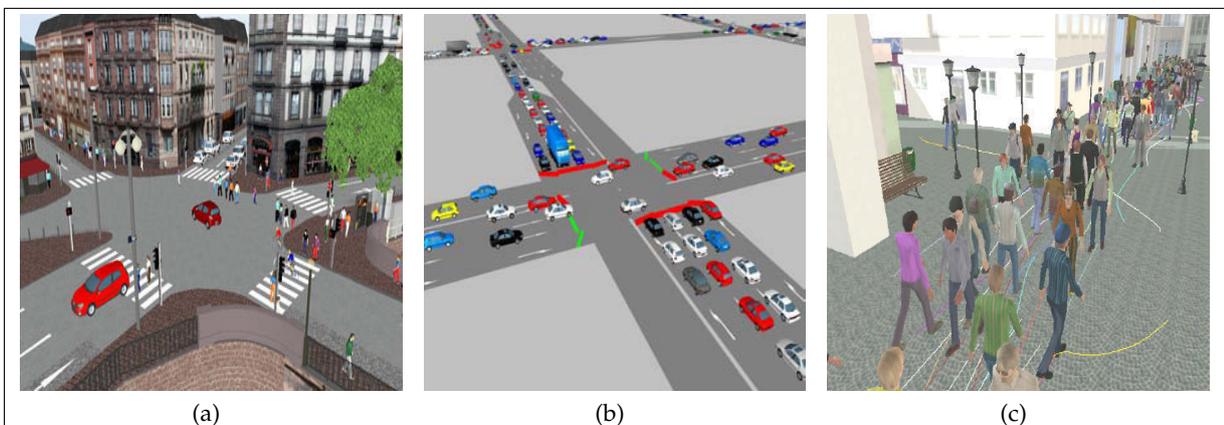


Figure 6.4: (a) shows real-like representation of a city center while (b) & (c) present the traffic flow and pedestrian movement information respectively for given time of the day. Such information would be updated on regular basis throughout the day.

so far. With the recent advancements in 3D data acquisition technologies, the performance is steadily improving whereas the manufacturing costs are decreasing. This balance between the cost and high level performance capabilities (higher level of detail with easy and large range coverage), that are now imperative for many modern day applications, would soon make the use of these sensors more and more widespread. In this thesis, we have used the data obtained from the same sensors mounted on the same vehicles. However the proposed methods can be adapted and easily extended to configurations of different vehicles with similar sensors. Main challenges associated, and also addressed in this thesis, include variation in sensor parameters, 3D point resolution, calibration errors, etc., along with rapid transfer of data. The use of ordinary household vehicles equipped with these sensors would replace the need of multiple passages by the same specialized vehicle, passing through an area again and again, with the use of collaborative means; i.e. any such vehicle, whenever it passes through an area, helps in enriching and improving the 3D urban cartography as presented in this thesis. This approach would not only allow more frequent updating of the centralized database but also make it more cost effective.

The next step would then be to explore the possibility of using the configuration of different vehicles and different sensors for collaborative 3D urban cartography. These sensors could be either 2D cameras or 3D range sensors commonly mounted (only one or both) on different vehicles for diverse perception tasks. Apart from the earlier mentioned problems, the biggest challenge in this case would be to fuse information from different types of sensors (for e.g. information from a geo-referenced smart-phone image or a camera mounted on a vehicle with that of a LiDAR mounted on another vehicle) to improve the 3D urban cartography. A suitable approach in this case would be to decompose the urban scenes into common semantic features that could then be easily fused together to enrich and update the 3D cartography.

Another problem that we came across during this thesis was the availability of relevant datasets. Although we were extremely lucky to have obtained different datasets from our collaborators, i.e. IGN (Institut Géographique National), Universities of New York and Kentucky during the thesis, we felt that the availability of such type of datasets was very difficult. While there is a large amount of data sets and benchmarks available for 2D sensors due to a large community working in this domain, the number of datasets with ground truths and benchmarks pertaining to 3D sensors and specially multi-passages is quite limited and not easily accessible. It is therefore requested that different groups working in this field should make available their datasets and results on a common website/platform easy accessible to everyone. International societies, like ISPRS (International Society of Photogrammetry and Remote Sensing), can use their influence and play an important role in the development of such platforms for the collection and diffusion of data.

Appendix A

Articles in International Journals

- [ACT14c] A. K. Aijazi, P. Checchin, and L. Trassoudaine, "Automatic Detection and Feature Estimation of Windows in 3D Urban Point Clouds Exploiting Façade Symmetry and Temporal Correspondences," *Int. Journal of Remote Sensing*, (In press), 2014. doi: [10.1080/01431161.2014.975420](https://doi.org/10.1080/01431161.2014.975420).
- [ACT13b] A. K. Aijazi, P. Checchin, and L. Trassoudaine, "Automatic Removal of Imperfections and Change Detection for Accurate 3D Urban Cartography by Classification and Incremental Updating," *Remote Sensing*, vol. 5, no. 8, pp. 3701–3728, 2013, issn: 2072-4292. doi: [10.3390/rs5083701](https://doi.org/10.3390/rs5083701). [Online]. Available: <http://www.mdpi.com/2072-4292/5/8/3701>.
- [ACT13e] A. K. Aijazi, P. Checchin, and L. Trassoudaine, "Segmentation Based Classification of 3D Urban Point Clouds: A Super-Voxel Based Approach," *Remote Sensing*, vol. 5, no. 4, pp. 1624–1650, 2013. doi: [10.3390/rs5041624](https://doi.org/10.3390/rs5041624). [Online]. Available: <http://www.mdpi.com/2072-4292/5/4/1624>.

Book Chapters

- [ACT15] A. K. Aijazi, P. Checchin, and L. Trassoudaine, "Intelligent Autonomous Systems – Springer," in Springer (In press), 2015, ch. Automatic Detection and Feature Estimation of Windows from Mobile Terrestrial LiDAR Data, isbn: 978-3-319-08337-7.
- [ACT14e] A. K. Aijazi, P. Checchin, and L. Trassoudaine, "Field and Service Robotics – Springer Tracts in Advanced Robotics," in K. Yoshida and S. Tadokoro, Eds. Springer, Feb. 2014, vol. 92, ch. A Super-Voxel Based Segmentation and Classification Method for 3D Urban Landscapes with Evaluation and Comparison, pp. 511–526, isbn: 978-3-642-40685-0. doi: [10.1007/978-3-642-13408-1_14](https://doi.org/10.1007/978-3-642-13408-1_14).

Articles in International Conferences

- [ACT14a] A. K. Aijazi, P. Checchin, and L. Trassoudaine, "Automatic Detection and Feature Estimation of Windows for Refining Building Façades in 3D Urban Point Clouds," in *Inter. Conf. on Photogrammetry Computer Vision (PCV)*, ETH Zürich, Switzerland, Sep. 2014.
- [ACT14b] A. K. Aijazi, P. Checchin, and L. Trassoudaine, "Automatic Detection and Feature Estimation of Windows from Mobile Terrestrial LiDAR Data," in *The 13th International Conference on Intelligent Autonomous Systems (IAS-13)*, Padova, Italy, Jul. 2014.
- [ACT13d] A. K. Aijazi, P. Checchin, and L. Trassoudaine, "Detecting and Updating Changes in LiDAR Point Clouds for Automatic 3D Urban Cartography," in *Laser Scanning 2013 ISPRS Workshop*, vol. II (Part 5/W2), Antalya, Turkey: ISPRS Annals PRS&SIS, Nov. 2013, 7–12. **Best Poster Award** sponsored by IAPR –International Association of Pattern Recognition.

- [**ACT13a**] A. K. Aijazi, P. Checchin, and L. Trassoudaine, "Automatic Change Detection and Incremental Updating for Accurate 3D Urban Cartography," in *The 16th IEEE Inter. Conf. on Intelligent Transportation Systems (ITSC)*, 2013, 61–70. **Best Student Paper Award (runner up)**.
- [**ACT13c**] A. K. Aijazi, P. Checchin, and L. Trassoudaine, "Automatic Update and Completion of Occluded Regions for Accurate 3D Urban Cartography by Combining Multiple Views and Multiple Passages," in *Int. Conf. on Pattern Recognition, Applications and Methods*, Barcelona, Spain, Feb. 2013, pp. 61–70.
- [**ACT12c**] A. K. Aijazi, P. Checchin, and L. Trassoudaine, "Handling Occlusions for Accurate 3D Urban Cartography: A new Approach based on Characterization and Multiple Passages," in *International Conference on 3D Imaging, Modeling, Processing, Visualization, Transmissions (3DIMPVT)*, ETH Zürich, Switzerland, Oct. 2012.
- [**ACT12a**] A. K. Aijazi, P. Checchin, and L. Trassoudaine, "A Super-Voxel Based Segmentation and Classification Method for 3D Urban Landscapes with Evaluation and Comparison," in *The 8th Inter. Conf. on Field and Service Robots (FSR)*, Matsushima, Japan, Jul. 2012.
- [**ACT12b**] A. K. Aijazi, P. Checchin, and L. Trassoudaine, "Classification of 3D Urban Scenes – A Voxel Based Approach," in *Int. Conf. on Pattern Recognition, Applications and Methods*, Vilamoura, Portugal, Feb. 2012, pp. 61–70. doi: [10.5220/0003733100610070](https://doi.org/10.5220/0003733100610070).

Articles in National Conferences (peer reviewed):

- [**ACT14d**] A. K. AIJAZI, P. CHECCHIN et L. TRASSOUDAINE, "Cartographie urbaine 3D : détection et analyse des évolutions de l'environnement pour la mise à jour automatique", in *Actes du dix-neuvième congrès francophone AFRIF-RFIA*, Rouen, France, juin 2014.
- [**ACT12d**] A. K. AIJAZI, P. CHECCHIN et L. TRASSOUDAINE, "Segmentation et classification de points 3D obtenus à partir de relevés laser terrestres : une approche par super-voxels", in *Actes du dix-huitième congrès francophone AFRIF-RFIA*, Lyon, France, jan. 2012.

Appendix B

Table of Probabilities for the F Distribution

Alpha = 0.05

D/N	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	20	24	30	40	60	120
1	161.45	199.50	215.71	224.58	230.16	233.99	236.77	238.88	240.54	241.88	242.98	243.91	244.69	245.36	245.95	248.01	249.05	250.10	251.14	252.20	253.25
2	18.51	19.00	19.16	19.25	19.30	19.33	19.35	19.37	19.38	19.40	19.40	19.41	19.42	19.42	19.43	19.45	19.45	19.46	19.47	19.48	19.49
3	10.13	9.55	9.28	9.12	9.01	8.94	8.89	8.85	8.81	8.79	8.76	8.74	8.73	8.71	8.70	8.66	8.64	8.62	8.59	8.57	8.55
4	7.71	6.94	6.59	6.39	6.26	6.16	6.09	6.04	6.00	5.96	5.94	5.91	5.89	5.87	5.86	5.80	5.77	5.75	5.72	5.69	5.66
5	6.61	5.79	5.41	5.19	5.05	4.95	4.88	4.82	4.77	4.74	4.70	4.68	4.66	4.64	4.62	4.56	4.53	4.50	4.46	4.43	4.40
6	5.99	5.14	4.76	4.53	4.39	4.28	4.21	4.15	4.10	4.06	4.03	4.00	3.98	3.96	3.94	3.87	3.84	3.81	3.77	3.74	3.70
7	5.59	4.74	4.35	4.12	3.97	3.87	3.79	3.73	3.68	3.64	3.60	3.57	3.55	3.53	3.51	3.44	3.41	3.38	3.34	3.30	3.27
8	5.32	4.46	4.07	3.84	3.69	3.58	3.50	3.44	3.39	3.35	3.31	3.28	3.26	3.24	3.22	3.15	3.12	3.08	3.04	3.01	2.97
9	5.12	4.26	3.86	3.63	3.48	3.37	3.29	3.23	3.18	3.14	3.10	3.07	3.05	3.03	3.01	2.94	2.90	2.86	2.83	2.79	2.75
10	4.96	4.10	3.71	3.48	3.33	3.22	3.14	3.07	3.02	2.98	2.94	2.91	2.89	2.86	2.85	2.77	2.74	2.70	2.66	2.62	2.58
11	4.84	3.98	3.59	3.36	3.20	3.09	3.01	2.95	2.90	2.85	2.82	2.79	2.76	2.74	2.72	2.65	2.61	2.57	2.53	2.49	2.45
12	4.75	3.89	3.49	3.26	3.11	3.00	2.91	2.85	2.80	2.75	2.72	2.69	2.66	2.64	2.62	2.54	2.51	2.47	2.43	2.38	2.34
13	4.67	3.81	3.41	3.18	3.03	2.92	2.83	2.77	2.71	2.67	2.63	2.60	2.58	2.55	2.53	2.46	2.42	2.38	2.34	2.30	2.25
14	4.60	3.74	3.34	3.11	2.96	2.85	2.76	2.70	2.65	2.60	2.57	2.53	2.51	2.48	2.46	2.39	2.35	2.31	2.27	2.22	2.18
15	4.54	3.68	3.29	3.06	2.90	2.79	2.71	2.64	2.59	2.54	2.51	2.48	2.45	2.42	2.40	2.33	2.29	2.25	2.20	2.16	2.11
16	4.49	3.63	3.24	3.01	2.85	2.74	2.66	2.59	2.54	2.49	2.46	2.42	2.40	2.37	2.35	2.28	2.24	2.19	2.15	2.11	2.06
17	4.45	3.59	3.20	2.96	2.81	2.70	2.61	2.55	2.49	2.45	2.41	2.38	2.35	2.33	2.31	2.23	2.19	2.15	2.10	2.06	2.01
18	4.41	3.55	3.16	2.93	2.77	2.66	2.58	2.51	2.46	2.41	2.37	2.34	2.31	2.29	2.27	2.19	2.15	2.11	2.06	2.02	1.97
19	4.38	3.52	3.13	2.90	2.74	2.63	2.54	2.48	2.42	2.38	2.34	2.31	2.28	2.26	2.23	2.16	2.11	2.07	2.03	1.98	1.93
20	4.35	3.49	3.10	2.87	2.71	2.60	2.51	2.45	2.39	2.35	2.31	2.28	2.25	2.22	2.20	2.12	2.08	2.04	1.99	1.95	1.90
21	4.32	3.47	3.07	2.84	2.68	2.57	2.49	2.42	2.37	2.32	2.28	2.25	2.22	2.20	2.18	2.10	2.05	2.01	1.96	1.92	1.87
22	4.30	3.44	3.05	2.82	2.66	2.55	2.46	2.40	2.34	2.30	2.26	2.23	2.20	2.17	2.15	2.07	2.03	1.98	1.94	1.89	1.84
23	4.28	3.42	3.03	2.80	2.64	2.53	2.44	2.37	2.32	2.27	2.24	2.20	2.18	2.15	2.13	2.05	2.01	1.96	1.91	1.86	1.81
24	4.26	3.40	3.01	2.78	2.62	2.51	2.42	2.36	2.30	2.25	2.22	2.18	2.15	2.13	2.11	2.03	1.98	1.94	1.89	1.84	1.79
25	4.24	3.39	2.99	2.76	2.60	2.49	2.40	2.34	2.28	2.24	2.20	2.16	2.14	2.11	2.09	2.01	1.96	1.92	1.87	1.82	1.77
26	4.23	3.37	2.98	2.74	2.59	2.47	2.39	2.32	2.27	2.22	2.18	2.15	2.12	2.09	2.07	1.99	1.95	1.90	1.85	1.80	1.75
27	4.21	3.35	2.96	2.73	2.57	2.46	2.37	2.31	2.25	2.20	2.17	2.13	2.10	2.08	2.06	1.97	1.93	1.88	1.84	1.79	1.73
28	4.20	3.34	2.95	2.71	2.56	2.45	2.36	2.29	2.24	2.19	2.15	2.12	2.09	2.06	2.04	1.96	1.91	1.87	1.82	1.77	1.71
29	4.18	3.33	2.93	2.70	2.55	2.43	2.35	2.28	2.22	2.18	2.14	2.10	2.08	2.05	2.03	1.94	1.90	1.85	1.81	1.75	1.70
30	4.17	3.32	2.92	2.69	2.53	2.42	2.33	2.27	2.21	2.16	2.13	2.09	2.06	2.04	2.01	1.93	1.89	1.84	1.79	1.74	1.68
40	4.08	3.23	2.84	2.61	2.45	2.34	2.25	2.18	2.12	2.08	2.04	2.00	1.97	1.95	1.92	1.84	1.79	1.74	1.69	1.64	1.58
60	4.00	3.15	2.76	2.53	2.37	2.25	2.17	2.10	2.04	1.99	1.95	1.92	1.89	1.86	1.84	1.75	1.70	1.65	1.59	1.53	1.47
120	3.92	3.07	2.68	2.45	2.29	2.18	2.09	2.02	1.96	1.91	1.87	1.83	1.80	1.78	1.75	1.66	1.61	1.55	1.50	1.43	1.35

Right Tailed, $D/N = df$ in denominator = down the rows, df in numerator = across the columns
 Table of Probabilities for F Distribution
 Note: Table is for an alpha of 0.05
 ©Copyright Lean Sigma Corporation 2013

Bibliography

- [AAP08] H. Ali, B. Ahmed, and G. Paar, "Robust Window Detection from 3D Laser Scanner Data," in *CISP 2008 Congress on Image and Signal Processing*, genaue Daten + Seitenzahlen nachtragen, vol. 2, Sayna/Hainan, China, May 2008, pp. 115–118.
- [ACT12a] A. K. Aijazi, P. Checchin, and L. Trassoudaine, "A Super-Voxel Based Segmentation and Classification Method for 3D Urban Landscapes with Evaluation and Comparison," in *The 8th Inter. Conf. on Field and Service Robots (FSR)*, Matsushima, Japan, Jul. 2012.
- [ACT12b] —, "Classification of 3D Urban Scenes – A Voxel Based Approach," in *Int. Conf. on Pattern Recognition, Applications and Methods*, Vilamoura, Portugal, Feb. 2012, pp. 61–70. DOI: [10.5220/0003733100610070](https://doi.org/10.5220/0003733100610070).
- [ACT12c] —, "Handling Occlusions for Accurate 3D Urban Cartography: A new Approach based on Characterization and Multiple Passages," in *International Conference on 3D Imaging, Modeling, Processing, Visualization, Transmissions (3DIMPVT)*, ETH Zürich, Switzerland, Oct. 2012.
- [ACT12d] —, "Segmentation et classification de points 3D obtenus à partir de relevés laser terrestres : une approche par super-voxels," French, in *Actes du dix-huitième congrès francophone AFRIF-RFIA*, Lyon, France, Jan. 2012.
- [ACT13a] —, "Automatic Change Detection and Incremental Updating for Accurate 3D Urban Cartography," in *The 16th IEEE Inter. Conf. on Intelligent Transportation Systems (ITSC)*, 2013, 61–70. **Best Student Paper Award (runner up)**.
- [ACT13b] —, "Automatic Removal of Imperfections and Change Detection for Accurate 3D Urban Cartography by Classification and Incremental Updating," *Remote Sensing*, vol. 5, no. 8, pp. 3701–3728, 2013, ISSN: 2072-4292. DOI: [10.3390/rs5083701](https://doi.org/10.3390/rs5083701). [Online]. Available: <http://www.mdpi.com/2072-4292/5/8/3701>.
- [ACT13c] —, "Automatic Update and Completion of Occluded Regions for Accurate 3D Urban Cartography by Combining Multiple Views and Multiple Passages," in *Int. Conf. on Pattern Recognition, Applications and Methods*, Barcelona, Spain, Feb. 2013, pp. 61–70.
- [ACT13d] —, "Detecting and Updating Changes in LiDAR Point Clouds for Automatic 3D Urban Cartography," in *Laser Scanning 2013 ISPRS Workshop*, vol. II (Part 5/W2), Antalya, Turkey: ISPRS Annals PRS&SIS, Nov. 2013, 7–12. **Best Poster Award** sponsored by IAPR –International Association of Pattern Recognition.
- [ACT13e] —, "Segmentation Based Classification of 3D Urban Point Clouds: A Super-Voxel Based Approach," *Remote Sensing*, vol. 5, no. 4, pp. 1624–1650, 2013. DOI: [10.3390/rs5041624](https://doi.org/10.3390/rs5041624). [Online]. Available: <http://www.mdpi.com/2072-4292/5/4/1624>.
- [ACT14a] —, "Automatic Detection and Feature Estimation of Windows for Refining Building Façades in 3D Urban Point Clouds," in *Inter. Conf. on Photogrammetry Computer Vision (PCV)*, ETH Zürich, Switzerland, Sep. 2014.

- [ACT14b] —, “Automatic Detection and Feature Estimation of Windows from Mobile Terrestrial LiDAR Data,” in *The 13th International Conference on Intelligent Autonomous Systems (IAS-13)*, Padova, Italy, Jul. 2014.
- [ACT14c] —, “Automatic Detection and Feature Estimation of Windows in 3D Urban Point Clouds Exploiting Façade Symmetry and Temporal Correspondences,” *Int. Journal of Remote Sensing*, (In press), 2014. DOI: [10.1080/01431161.2014.975420](https://doi.org/10.1080/01431161.2014.975420).
- [ACT14d] —, “Cartographie urbaine 3D : détection et analyse des évolutions de l’environnement pour la mise à jour automatique,” French, in *Actes du dix-neuvième congrès francophone AFRIF-RFIA*, Rouen, France, Jun. 2014.
- [ACT14e] —, “Field and Service Robotics – Springer Tracts in Advanced Robotics,” in K. Yoshida and S. Tadokoro, Eds. Springer, Feb. 2014, vol. 92, ch. A Super-Voxel Based Segmentation and Classification Method for 3D Urban Landscapes with Evaluation and Comparison, pp. 511–526, ISBN: 978-3-642-40685-0. DOI: [10.1007/978-3-642-13408-1_14](https://doi.org/10.1007/978-3-642-13408-1_14).
- [ACT15] —, “Intelligent Autonomous Systems – Springer,” in Springer (In press), 2015, ch. Automatic Detection and Feature Estimation of Windows from Mobile Terrestrial LiDAR Data, ISBN: 978-3-319-08337-7.
- [AKY05] T. Asai, M. Kanbara, and N. Yokoya, “3D modeling of outdoor scenes by integrating stop-and-go and continuous scanning of rangefinder,” *CD-ROM Proc. ISPRS Working Group*, vol. 4, 2005.
- [ASJ+07] H. Ali, C. Seifert, N. Jindal, L. Paletta, and G. Paar, “Window Detection in Façades,” in *14th International Conference on Image Analysis and Processing*, Modena, Italy, Sep. 2007, pp. 837–842. DOI: [10.1109/ICIAP.2007.4362880](https://doi.org/10.1109/ICIAP.2007.4362880).
- [ATC+05] D. Anguelov, B. Taskar, V. Chatalbashev, D. Koller, D. Gupta, G. Heitz, and A. Ng, “Discriminative Learning of Markov Random Fields for Segmentation of 3D Scan Data,” in *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on*, vol. 2, Los Alamitos, CA, USA: IEEE Computer Society, 2005, pp. 169–176. DOI: <http://doi.ieeecomputersociety.org/10.1109/CVPR.2005.133>.
- [BDB10] S. Bénitez, E. Denis, and C. Baillard, “Automatic production of occlusion-free rectified façade textures using vehicle-based imagery,” *Internat. Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences, Commission XXXVIII, Saint-Mande, France*, 2010.
- [BGH10] M. Bouziani, K. Goïta, and D.-C. He, “Automatic change detection of buildings in urban environment from very high spatial resolution images using existing geodatabase and prior knowledge,” *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 65, no. 1, pp. 143–153, 2010.
- [BH07a] S. Becker and N. Haala, “Combined Feature Extraction for Façade Reconstruction,” in *ISPRS Workshop on Laser Scanning and SilviLaser*, Espoo, Finland: ISPRS, 2007, p. 44.
- [BH07b] S. Becker and N. Haala, “Refinement of Building Façades by Integrated Processing of LiDAR and Image Data,” in *PIA07 - Photogrammetric Image Analysis - Proceedings of ISPRS Technical Commission III Symposium*, U. Stilla, H. Mayer, F. Rottensteiner, C. Heipke, and S. Hinz, Eds., International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences, vol. XXXVI (Part 3/W49A), Munich, Germany, Sep. 2007, pp. 7–12.
- [BM92] P. Besl and N. McKay, “A Method for Registration of 3-D Shapes,” in *IEEE Transaction on Pattern Analysis and Machine Intelligence*, vol. 14(2), USA, 1992, pp. 239–256.

- [BMSV08] D. Barber, J. Mills, and S. Smith-Voysey, "Geometric validation of a ground-based mobile laser scanning system," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 63, no. 1, pp. 128–141, 2008, ISSN: 0924-2716. DOI: [10.1016/j.isprsjprs.2007.07.005](https://doi.org/10.1016/j.isprsjprs.2007.07.005). [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0924271607000834>.
- [BSF+09] C. Barnes, E. Shechtman, A. Finkelstein, and D. Goldman, "PatchMatch: a randomized correspondence algorithm for structural image editing," *ACM Trans. Graph.*, vol. 28, 24:1–24:11, 3 Jul. 2009, ISSN: 0730-0301. DOI: <http://doi.acm.org/10.1145/1531326.1531330>. [Online]. Available: <http://doi.acm.org/10.1145/1531326.1531330>.
- [BVS+14] M. Brédif, B. Vallet, A. Serna, B. Marcotegui, and N. Paparoditis, "Terramobilita/IQmulus Urban Point Cloud Analysis Benchmark," in *IQmulus workshop in conjunction with SGP 14*, Cardiff, UK, Jul. 2014.
- [BZF12] M. Bosse, R. Zlot, and P. Flick, "Zebedee: Design of a spring-mounted 3-d range sensor with application to mobile mapping," *Robotics, IEEE Transactions on*, vol. 28, no. 5, pp. 1104–1119, 2012.
- [Bré13] M. Brédif, "Image-Based Rendering of LOD1 3D City Models for traffic-augmented Immersive Street-view Navigation," *ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. 1, no. 3, pp. 7–11, 2013.
- [Böh04] J. Böhm, "A Multi-Image Fusion for Occlusion-Free Façade Texturing," in *ISPRS Kongress 2004*, IAPRS, Ed., vol. XXXV, Part B, Istanbul, Turkey: ISPRS, 2004, pp. 867–873.
- [CE09] N. Champion and J. Everaerts, "Detection of unregistered buildings for updating 2d databases," *EuroSDR Official Publication*, vol. 56, pp. 7–54, 2009.
- [CPS10] D. Craciun, N. Paparoditis, and F. Schmitt, "Multi-view scans alignment for 3D spherical mosaicing in large-scale unstructured environments," *Computer Vision and Image Understanding*, vol. 114, no. 11, pp. 1248–1263, 2010, Special issue on Embedded Vision, ISSN: 1077-3142. DOI: [10.1016/j.cviu.2010.03.019](https://doi.org/10.1016/j.cviu.2010.03.019). [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S1077314210000962>.
- [CPT04] A. Criminisi, P. Pérez, and K. Toyama, "Region filling and object removal by exemplar-based inpainting," *IEEE Trans. Image Processing*, vol. 13, no. 9, pp. 1200–1212, 2004.
- [CZ13] N. Corso and A. Zakhor, "Indoor Localization Algorithms for an Ambulatory Human Operated 3D Mobile Mapping System," *Remote Sensing*, vol. 5, no. 12, pp. 6611–6646, 2013.
- [Cha11] D. U. D. Challenge. (2011), [Online]. Available: www.vis.uky.edu/3dimpvt/challenge.php.
- [DBR09] B. Douillard, A. Brooks, and F. Ramos, "A 3D Laser and Vision Based Classifier," in *5th Int. Conf. on Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP)*, Melbourne, Australia, Dec. 2009, p. 6.
- [DUK+11] B. Douillard, J. Underwood, N. Kuntz, V. Vlaskine, A. Quadros, P. Morton, and A. Frenkel, "On the Segmentation of 3D LIDAR Point Clouds," in *IEEE Int. Conf. on Robotics and Automation (ICRA)*, Shanghai, China, May 2011, p. 8.
- [EES02] C. Ellum and N. El-Sheimy, "Land-based mobile mapping systems," *Photogrammetric Engineering & Remote Sensing*, vol. 68, no. 1, pp. 13–28, 2002.

- [ETV+11] C. Engels, D. Tingdahl, M. Vercauysse, T. Tuytelaars, H. Sahli, and L. Van Gool, "Automatic occlusion removal from façades for 3D urban reconstruction," *Advanced Concepts for Intellig. Vision Syst.*, pp. 681–692, 2011.
- [FH04] P. Felzenszwalb and D. Huttenlocher, "Efficient Graph-Based Image Segmentation," *International Journal of Computer Vision*, vol. 59, pp. 167–181, 2004, 10.1023/B:VISI.0000022288.19776.77, ISSN: 0920-5691. [Online]. Available: <http://dx.doi.org/10.1023/B:VISI.0000022288.19776.77>.
- [FS11] S. Friedman and I. Stamos, "Real Time Detection of Repeated Structures in Point Clouds of Urban Scenes," in *The First Joint 3DIM/3DPVT (3DIMPVT) Conference*, Hangzhou, China, May 2011, p. 8.
- [FS13] S. Friedman and I. Stamos, "Online detection of repeated structures in point clouds of urban scenes for compression and registration," *International Journal of Computer Vision*, vol. 102, no. 1-3, pp. 112–128, 2013.
- [FSZ04] C. Frueh, R. Sammon, and A. Zakhor, "Automated texture mapping of 3D city models with oblique aerial imagery," in *Proc. of 2nd Int. Symp. on 3D Data Processing, Visualization and Transmission (3DPVT)*, Thessaloniki, Greece: IEEE, 2004, pp. 396–403.
- [FWE03] B. Fung, K. Wang, and M. Ester, "Hierarchical document clustering using frequent itemsets," in *Proceedings of the SIAM International Conference on Data Mining*, 5, vol. 30, SIAM, 2003, pp. 59–70.
- [Fer05] O. Fernandez, "Obtaining a best fitting plane through 3D georeferenced data," *Journal of Structural Geology*, vol. 27, no. 5, pp. 855–858, 2005, ISSN: 0191-8141. DOI: <http://dx.doi.org/10.1016/j.jsg.2004.12.004>. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0191814105000143>.
- [GBR+08] S. Gandolfi, M. Barbarella, E. Ronci, and A. Burchi, "Close photogrammetry and laser scanning using a mobile mapping system for the high detailed survey of a high density urban area," *Proceedings of The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, Beijing, China, vol. 37, pp. 909–914, 2008.
- [GCM+12] A. Gressin, B. Cannelle, C. Mallet, and J.-P. Papellard, "Trajectory-based Registration of 3D Lidar Point Clouds Acquired with a Mobile Mapping System," *ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. 1, pp. 117–122, 2012.
- [GF09] A. Golovinskiy and T. Funkhouser, "Min-Cut Based Segmentation of Point Clouds," in *IEEE Workshop on Search in 3D and Video (S3DV) at ICCV*, Sep. 2009, pp. 39–46.
- [GMRM+05] D. Girardeau-Montaut, M. Roux, R. Marc, and G. Thibault, "Change detection on points cloud data acquired with a ground laser scanner," *Internat. Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. 36, pp. 30–35, 2005.
- [GNA+06] F. Goulette, F. Nashashibi, I. Abuhadrous, S. Ammoun, and C. Lourceau, "An integrated on-board laser range sensing system for on-the-way city and road modelling," *ISPRS REPT*, 2006.
- [GS10] C. Gao and J. R. Spletzer, "On-line calibration of multiple LIDARs on a mobile vehicle platform," in *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, IEEE, 2010, pp. 279–284.

- [Goa91] C. C. Goad, "The Ohio State University highway mapping project: the positioning component," in *Proceedings of the 47th Annual Meeting of The Institute of Navigation*, 1991, pp. 117–120.
- [Gór09] R. Góralski, "Three-dimensional interactive maps," PhD thesis, University of Glamorgan, England, 2009.
- [HCK06] G. Hunter, C. Cox, and J. Kremer, "Development of a commercial laser scanning mobile mapping system—StreetMapper," *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.*, vol. 36, 2006.
- [HHB+10] A. Halma, F. ter Haar, E. Bovenkamp, P. Eendebak, and A. van Eekeren, "Single spin image-ICP matching for efficient 3D object recognition," in *Proceedings of the ACM workshop on 3D object retrieval*, ser. 3DOR '10, Firenze, Italy: ACM, 2010, pp. 21–26, ISBN: 978-1-4503-0160-2. DOI: <http://doi.acm.org/10.1145/1877808.1877814>. [Online]. Available: <http://doi.acm.org/10.1145/1877808.1877814>.
- [HJH+09] J. Hyypä, A. Jaakkola, H. Hyypä, H. Kaartinen, A. Kukko, M. Holopainen, *et al.*, "Map updating and change detection using vehicle-based laser scanning," in *Urban Remote Sensing Event, 2009 Joint*, IEEE, Shanghai, China, 2009, pp. 1–6.
- [HLY+04] K. Hsiao, J. Liu, M. Yu, and Y. Tseng, "Change detection of landslide terrains using ground-based lidar data," in *XXth ISPRS Congress, Commission VII, WG VII/5*, Istanbul, Turkey: ISPRS, 2004.
- [HR05] G. Hripcsak and A. S. Rothschild, "Agreement, the F-Measure, and Reliability in Information Retrieval," *Journal of American Medical Informatics Association*, vol. 12, no. 3, pp. 296–298, 2005.
- [HS10] O. Hadjiliadis and I. Stamos, "Sequential classification in point clouds of urban scenes," in *Proc. 3DPVT*, 2010.
- [Har07] A. Haring, *Die Orientierung von Laserscanner-und Bilddaten bei der fahrzeuggestützten Objekterfassung*. na, 2007.
- [JP09] M. Jahangiri and M. Petrou, "An attention model for extracting components that merit identification," in *16th IEEE International Conference on Image Processing (ICIP), 2009*, IEEE, Cairo, Egypt, Nov. 2009, pp. 965–968.
- [Joh97] A. Johnson, "Spin-Images: A Representation for 3-D Surface Matching," PhD thesis, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, Aug. 1997.
- [KAW+09] K. Klasing, D. Althoff, D. Wollherr, and M. Buss, "Comparison of Surface Normal Estimation Methods for Range Sensing Applications," in *IEEE Int. Conf. on Robotics and Automation*, Kobe, Japan, May 2009, pp. 3206–3211.
- [KFR03] M. Kazhdan, T. Funkhouser, and S. Rusinkiewicz, "Rotation Invariant Spherical Harmonic Representation of 3D Shape Descriptors," in *Proceedings of the 2003 Eurographics/ACM SIGGRAPH symposium on Geometry processing*, ser. SGP '03, Aachen, Germany: Eurographics Association, Jun. 2003, pp. 156–164, ISBN: 1-58113-687-0. [Online]. Available: <http://dl.acm.org/citation.cfm?id=882370.882392>.
- [KL11] Z. Kang and Z. Lu, "The change detection of building models using epochs of terrestrial point clouds," in *Multi-Platform/Multi-Sensor Remote Sensing and Mapping (M2RSM), 2011 International Workshop on*, Xiamen, China: IEEE, 2011, pp. 1–6.
- [KPG10] J. Knopp, M. Prasad, and L. V. Gool, "Orientation invariant 3D object classification using hough transform based methods," in *Proceedings of the ACM workshop on 3D object retrieval*, ser. 3DOR '10, Firenze, Italy: ACM, 2010, pp. 15–20, ISBN: 978-1-4503-0160-2. DOI: <http://doi.acm.org/10.1145/1877808.1877813>. [Online]. Available: <http://doi.acm.org/10.1145/1877808.1877813>.

- [KV07] V. Konushin and V. Vezhnevets, "Automatic building texture completion," in *Proc. Graphicon'2007*, Moscow, Russia, 2007, pp. 174–177.
- [LBG+13] M.-O. Lowner, J. Benner, G. Groger, and K.-H. Hafele, "New Concepts for Structuring 3D City Models – An Extended Level of Detail Concept for CityGML Buildings," in *Computational Science and Its Applications–ICCSA 2013*, Springer, 2013, pp. 466–480.
- [LC05] J. Li and M. A. Chapman, "Introduction to MMS special issue," 2005.
- [LES08] K. Y. Lee and M. A. El-Sharkawi, *Modern heuristic optimization techniques: theory and applications to power systems*. John Wiley & Sons, 2008, vol. 39.
- [LKM+10] J. Lam, K. Kusevic, P. Mrstik, R. Harrap, and M. Greenspan, "Urban Scene Extraction from Mobile Ground Based LiDAR Data," in *International Symposium on 3D Data Processing Visualization and Transmission*, Paris, France, 2010, p. 8. [Online]. Available: <http://campwww.informatik.tu-muenchen.de/3DPVT2010/data/media/e-proceeding/papers/paper058.pdf>.
- [LN04] S. C. Lee and R. Nevatia, "Extraction and Integration of Window in a 3D Building Model from Ground View Images," in *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, IEEE, vol. 2, Washington, DC, USA, Jun. 2004, pp. II–113 –II–120. DOI: [10.1109/CVPR.2004.1315152](https://doi.org/10.1109/CVPR.2004.1315152).
- [LOL09] W. L. Lu, K. Okuma, and J. J. Little, "A Hybrid Conditional Random Field for Estimating the Underlying Ground Surface from Airborne LiDAR Data," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 47, no. 8, pp. 2913–2922, 2009.
- [LP05] R. Lindenbergh and N. Pfeifer, "A statistical deformation analysis of two epochs of terrestrial laser data of a lock," in *Proc. of 7th Conf. on Optical 3D Measurement Techniques*, vol. 2, Vienna, Austria: ISPRS, 2005, pp. 61–70.
- [LS07] E. Lim and D. Suter, "Conditional Random Field for 3D Point Clouds with Adaptive Data Reduction," in *International Conference on Cyberworlds*, Hannover, Oct. 2007, pp. 404–408.
- [LS08] E. H. Lim and D. Suter, "Multi-scale Conditional Random Fields for Over-Segmented Irregular 3D Point Clouds Classification," in *Computer Vision and Pattern Recognition Workshop*, vol. 0, Anchorage, AK, USA: IEEE Computer Society, Jun. 2008, pp. 1–7, ISBN: 978-1-4244-2339-2. DOI: <http://doi.ieeecomputersociety.org/10.1109/CVPRW.2008.4563064>.
- [LZQ06] Y. Liu, H. Zha, and H. Qin, "Shape Topics-A Compact Representation and New Algorithms for 3D Partial Shape Retrieval," in *IEEE Conf. on Computer Vision and Pattern Recognition*, vol. 2, New York, NY, USA: IEEE Computer Society, Jun. 2006, pp. 2025–2032.
- [LZS+11] Y. Li, Q. Zheng, A. Sharf, D. Cohen-Or, B. Chen, and N. J. Mitra, "2D-3D Fusion for Layer Decomposition of Urban Facades," in *IEEE Int. Conf. on Computer Vision (ICCV)*, Barcelona, Spain, Nov. 2011.
- [ME07] T. Malisiewicz and A. A. Efros, "Improving Spatial Support for Objects via Multiple Segmentations," in *British Machine Vision Conference (BMVC)*, Sep. 2007.
- [MHA+10] L. Matikainen, J. Hyyppä, E. Ahokas, L. Markelin, and H. Kaartinen, "Automatic detection of buildings and changes in buildings for updating of maps," *Remote Sensing*, vol. 2, no. 5, pp. 1217–1248, 2010, ISSN: 2072-4292. DOI: [10.3390/rs2051217](https://doi.org/10.3390/rs2051217). [Online]. Available: <http://www.mdpi.com/2072-4292/2/5/1217>.

- [MHK04] L. Matikainen, J. Hyyppä, and H. Kaartinen, "Automatic detection of changes from laser scanner and aerial image data for updating building maps," in *Internat. Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences (IAPRS & SIS)*, O. Altan, Ed., vol. XXXV Part B2, Istanbul, Turkey: ISPRS, Jul. 2004, pp. 434–439. [Online]. Available: <http://www.isprs.org/proceedings/XXXV/congress/comm2/comm2.aspx>.
- [MKC+12] J. McDonald, M. Kaess, C. Cadena, J. Neira, and J. Leonard, "Real-time 6-DOF multi-session visual SLAM over large-scale environments," *Robotics and Autonomous Systems*, pp. –, 2012, Online, ISSN: 0921-8890. DOI: [10.1016/j.robot.2012.08.008](https://doi.org/10.1016/j.robot.2012.08.008). [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0921889012001406>.
- [MMR13] M. Madic, D. Markovic, and M. Radovanovic, "Comparison of meta-heuristic algorithms for solving machining optimization problems," *Facta universitatis-series: Mechanical Engineering*, vol. 11, no. 1, pp. 29–44, 2013.
- [MPS09] F. Moosmann, O. Pink, and C. Stiller, "Segmentation of 3D Lidar Data in non-flat Urban Environments using a Local Convexity Criterion," in *Proc. of the IEEE Intelligent Vehicles Symposium (IV)*, Nashville, Tennessee, USA, Jun. 2009, pp. 215–220.
- [MR07] H. Mayer and S. Reznik, "Building facade interpretation from uncalibrated wide-baseline image sequences," *{ISPRS} Journal of Photogrammetry and Remote Sensing*, vol. 61, no. 6, pp. 371–380, 2007, ISSN: 0924-2716. DOI: <http://dx.doi.org/10.1016/j.isprsjprs.2006.10.007>. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0924271606001419>.
- [MS12] A. Mesolongitis and I. Stamos, "Detection of windows in point clouds of urban scenes," in *2012 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, Providence, RI, USA: IEEE, Jun. 2012, pp. 17–24, ISBN: 978-1-4673-1611-8.
- [MVH09] D. Munoz, N Vandapel, and M. Hebert, "Onboard contextual classification of 3-D point clouds with learned high-order Markov Random Fields," in *IEEE Int. Conf. on Robotics and Automation (ICRA)*, Kobe, Japan, May 2009, pp. 2009–2016.
- [MZW+07] P. Muller, G. Zeng, P. Wonka, and L. V. Gool, "Image-based procedural modeling of façades," *ACM Transactions on Graphics*, vol. 26, no. 3, p. 85, 2007.
- [Nov91] K. Novak, "The Ohio State University highway mapping system: The stereo vision system component," in *Proceedings of the 47th Annual Meeting of The Institute of Navigation*, 1991, pp. 121–124.
- [OFC+02] R. Osada, T. Funkhouser, B. Chazelle, and D. Dobkin, "Shape distributions," *ACM Trans. Graph.*, vol. 21, pp. 807–832, 4 Oct. 2002, ISSN: 0730-0301. DOI: <http://doi.acm.org/10.1145/571647.571648>. [Online]. Available: <http://doi.acm.org/10.1145/571647.571648>.
- [OR05] D. Ortin and F. Remondino, "Occlusion-free Image Generation for Realistic Texture Mapping," in *IAPRS & SIS*, vol. XXXVI (Part 5/W17), Venice, Italy: ISPRS, 2005.
- [PBZ09] F. Pauling, M. Bosse, and R. Zlot, "Automatic Segmentation of 3D Laser Point Clouds by Ellipsoidal Region Growing," in *Australasian Conference on Robotics & Automation*, Sydney, Australia, 2009, p. 10.
- [PFT+07] W. H. Press, B. P. Flannery, S. A. Teukolsky, and W. T. Vetterling, *Numerical recipes 3rd Edition: The Art of Scientific Computing*, 3rd. New York, NY, USA: Cambridge University Press, 2007.

- [PGJMS+13] I. Puente, H. González-Jorge, J. Martínez-Sánchez, and P. Arias, "Review of mobile mapping and surveying technologies," *Measurement*, vol. 46, no. 7, pp. 2127–2145, 2013.
- [PLK+13] E. Puttonen, M. Lehtomaki, H. Kaartinen, L. Zhu, A. Kukko, and A. Jaakkola, "Improved sampling for terrestrial and mobile laser scanner point cloud data," *Remote Sensing*, vol. 5, no. 4, pp. 1754–1773, 2013.
- [PMD08] A. Patterson, P. Mordohai, and K. Daniilidis, "Object Detection from Large-Scale 3D Datasets Using Bottom-Up and Top-Down Descriptors," in *ECCV (4)*, D. A. Forsyth, P. H. S. Torr, and A. Zisserman, Eds., ser. Lecture Notes in Computer Science, vol. 5305, Springer, 2008, pp. 553–566, ISBN: 978-3-540-88692-1.
- [PME11] G. Pandey, J. R. McBride, and R. M. Eustice, "Ford campus vision and lidar data set," *The International Journal of Robotics Research*, vol. 30, no. 13, pp. 1543–1552, 2011.
- [PMW+08] M. Pauly, N. J. Mitra, J. Wallner, H. Pottmann, and L. J. Guibas, "Discovering Structural Regularity in 3D Geometry," *ACM Transactions on Graphics*, vol. 27, no. 3, #43, 1–11, 2008.
- [PPC+12] N. Paparoditis, J.-P. Papelard, B. Cannelle, A. Devaux, B. Soheilian, N. David, and E. Houzay, "Stéréopolis II: A multi-purpose and multi-sensor 3D mobile mapping system for street visualisation and 3D metrology," *Revue française de photogrammétrie et de télédétection*, no. 200, pp. 69–79, 2012.
- [PV07] S. Pu and G. Vosselman, "Extracting windows from terrestrial laser scanning," in *ISPRS Workshop on Laser Scanning 2007 and SilviLaser 2007*, International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences, vol. XXXVI (Part 3/W52), Espoo, Finland, Sep. 2007, pp. 320–325. [Online]. Available: http://www.isprs.org/proceedings/XXXVI/3-W52/final_papers/Pu_2007.pdf.
- [PV09] —, "Building Facade Reconstruction by Fusing Terrestrial Laser Points and Images," *Sensors*, vol. 9, no. 6, pp. 4525–4542, 2009, ISSN: 1424-8220. DOI: [10.3390/s90604525](https://doi.org/10.3390/s90604525). [Online]. Available: <http://www.mdpi.com/1424-8220/9/6/4525/>.
- [Per10] G. Perie, "Mobile mapping systems: An introduction to the technology," *GeoInformatics*, vol. 13, pp. 32–43, 2010.
- [RBT+10] R. Rusu, G. Bradski, R. Thibaux, and J. Hsu, "Fast 3D Recognition and Pose Using the Viewpoint Feature Histogram," in *IEEE/RSJ Int. Conf. on Intellig. Robots and Systems (IROS)*, Taipei, Taiwan, Oct. 2010, pp. 2155–2162.
- [RDHV06] T. Rabbani, F. van Den Heuvel, and G. Vosselmann, "Segmentation of point clouds using smoothness constraint," *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. 36, no. 5, pp. 248–253, 2006.
- [RH06] M. Redstall and G. Hunter, "Accurate terrestrial laser scanning from a moving platform," *Geomatics World*, pp. 28–30, 2006.
- [RH07] A. Rosenberg and J. Hirschberg, "V-Measure: A Conditional Entropy-Based External Cluster Evaluation Measure," in *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, 2007, pp. 410–420. [Online]. Available: <http://www.aclweb.org/anthology/D/D07/D07-1043>.
- [RKU05] C. Rasmussen, T. Korah, and W. Ulrich, "Randomized view planning and occlusion removal for mosaicing building façades," in *Proceedings of IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS 2005)*, Edmonton, Alberta, Canada: IEEE, 2005, pp. 4137–4142.

- [RTR+04] D. B. Russakoff, C. Tomasi, T. Rohlving, and C. Maurer, "Image similarity using mutual information of regions," in *Computer Vision - ECCV 2004*, ser. Lecture Notes in Computer Science, T. Pajdla and J. Matas, Eds., vol. 3023, Springer Berlin Heidelberg, 2004, pp. 596–607, ISBN: 978-3-540-21982-8. DOI: [10.1007/978-3-540-24672-5_47](https://doi.org/10.1007/978-3-540-24672-5_47). [Online]. Available: http://dx.doi.org/10.1007/978-3-540-24672-5_47.
- [SBV+11] A. Swart, J. Broere, R. Veltkamp, and R. Tan, "Refined non-rigid registration of a panoramic image sequence to a LiDAR point cloud," in *Photogrammetric Image Analysis*, Springer, 2011, pp. 73–84.
- [SG12] A. A. Souza and L. M. Gonçalves, "3D Robotic Mapping with Probabilistic Occupancy Grids," *Internat. Journal of Engineering Sciences & Emerging Technologies*, vol. 4, no. 1, pp. 15–25, Dec. 2012, ISSN: 2231-6604.
- [SG72] G. Stiny and J. Gips, "Best computer papers of 1971," in. Petrocelli OR (ed), 1972, ch. Shape Grammars and the Generative Specification, pp. 125–135.
- [SHF+11] C.-H. Shen, S.-S. Huang, H. Fu, and S.-M. Hu, "Adaptive Partitioning of Urban Façades," *ACM Transactions on Graphics*, vol. 30, no. 6, 184:1–184:10, Dec. 2011, ISSN: 0730-0301. DOI: [10.1145/2070781.2024218](https://doi.org/10.1145/2070781.2024218). [Online]. Available: <http://doi.acm.org/10.1145/2070781.2024218>.
- [SM14] A. Serna and B. Marcotegui, "Detection, segmentation and classification of 3D urban objects using mathematical morphology and supervised learning," *ISPRS Journal of Photogrammetry and Remote Sensing*, 2014.
- [SMG+14] A. Serna, B. Marcotegui, F. Goulette, J.-E. Jean-Emmanuel, et al., "Paris-Rue-Madame database: a 3D mobile laser scanner dataset for benchmarking urban detection, segmentation and classification methods," in *4th International Conference on Pattern Recognition, Applications and Methods ICPRAM 2014*, 2014.
- [SNC10] J. Schoenberg, A. Nathan, and M. Campbell, "Segmentation of dense range information in complex urban scenes," in *IEEE/RSJ Int. Conf. on Intellig. Robots and Systems (IROS)*, Taipei, Taiwan, Oct. 2010, pp. 2033–2038.
- [SOG09] J. Sun, M. Ovsjanikov, and L. Guibas, "A Concise and Provably Informative Multi-Scale Signature Based on Heat Diffusion," in *Proceedings of the Symposium on Geometry Processing*, ser. SGP '09, Berlin, Germany: Eurographics Association, 2009, pp. 1383–1392. [Online]. Available: <http://dl.acm.org/citation.cfm?id=1735603.1735621>.
- [SRO10] J. Strom, A. Richardson, and E. Olson, "Graph-based Segmentation for Colored 3D Laser Point Clouds," in *Proceedings of the IEEE/RSJ Int. Conf. on Intellig. Robots and Systems (IROS)*, Oct. 2010, pp. 2131–2136.
- [SV03] G. Sithole and G. Vosselman, "Automatic structure detection in a point-cloud of an urban landscape," in *Remote Sensing and Data Fusion over Urban Areas, 2003. 2nd GRSS/ISPRS Joint Workshop on*, IEEE, 2003, pp. 67–71.
- [SWK+04] T. Schafer, T. Weber, P. Kyrinovic, and M. Zamecnikova, "Deformation measurement using terrestrial laser scanning at the Hydropower Station of Gabčíkovo," in *INGEO 2004 and FIG Regional Central and Eastern European Conference on Engineering Surveying*, Bratislava, Slovakia: FIG, Nov. 2004.
- [SWK07] R. Schnabel, R. Wahl, and R. Klein, "Efficient ransac for point-cloud shape detection," *Computer Graphics Forum*, vol. 26, no. 2, pp. 214–226, 2007, ISSN: 1467-8659. DOI: [10.1111/j.1467-8659.2007.01016.x](https://doi.org/10.1111/j.1467-8659.2007.01016.x). [Online]. Available: <http://dx.doi.org/10.1111/j.1467-8659.2007.01016.x>.

- [TF06] B. G. Tabachnick and L. S. Fidell, *Using Multivariate Statistics (5th Edition)*. Needham Heights, MA, USA: Allyn & Bacon, Inc., 2006, ISBN: 0205459382.
- [TKB06] R. Triebel, K. Kersting, and W. Burgard, "Robust 3D scan point classification using associative Markov networks," in *Proceedings of IEEE International Conference on Robotics and Automation (ICRA)*, Orlando, Florida, USA, May 2006, pp. 2603–2608. DOI: [10.1109/ROBOT.2006.1642094](https://doi.org/10.1109/ROBOT.2006.1642094).
- [TS11] S. Tutas and U. Stilla, "Window detection in sparse point clouds using indoor points," in *PIA11 - Photogrammetric Image Analysis*, U. Stilla, F. Rottensteiner, H. Mayer, B. Jutzi, and M. Butenuth, Eds., International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences, vol. XXXVIII (Part 3/W22), Munich, Germany, Oct. 2011, pp. 131–136. [Online]. Available: <http://www.pf.bgu.tum.de/stilla/pub.html#2011>.
- [TSS10] R. Triebel, J. Shin, and R. Siegwart, "Segmentation and Unsupervised Part-based Discovery of Repetitive Objects," in *Proceedings of Robotics: Science and Systems*, Zaragoza, Spain, Jun. 2010, p. 8.
- [Tve77] A. Tversky, "Features of similarity," *Psychological review*, vol. 84, no. 4, p. 327, 1977.
- [USM+13] O. Untzelmann, T. Sattler, S. Middelberg, and L. Kobbelt, "A Scalable Collaborative Online System for City Reconstruction," in *Computer Vision Workshops (ICCVW), 2013 IEEE International Conference on*, IEEE, 2013, pp. 644–651.
- [VGLP06] R. Van Gosliga, R. Lindenbergh, and N. Pfeifer, "Deformation analysis of a bored tunnel by means of terrestrial laser scanning," *Image Engineering and Vision Metrology. ISPRS Commission*, vol. 36, pp. 167–172, 2006.
- [VKG05] G. Vosselman, P. Kessels, and B. Gorte, "The utilisation of airborne laser scanning for mapping," *International Journal of Applied Earth Observation and Geoinformation*, vol. 6, no. 3-4, pp. 177–186, 2005, Data Quality in Earth Observation Techniques, ISSN: 0303-2434. DOI: [DOI: 10.1016/j.jag.2004.10.005](https://doi.org/10.1016/j.jag.2004.10.005). [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0303243404000868>.
- [VM10] G. Vosselman and H.-G. Maas, *Airborne and terrestrial laser scanning*. Whittles Publishing, 2010.
- [VMY04] T. T. Vu, M. Matsuoka, and F. Yamazaki, "Lidar-based change detection of buildings in dense urban areas," in *Geoscience and Remote Sensing Symposium, 2004. IGARSS'04. Proceedings. 2004 IEEE International*, vol. 5, Anchorage, Alaska, USA: IEEE, 2004, pp. 3413–3416.
- [VS04] T. Vögtle and E. Steinle, "Detection and Recognition of Changes in Building Geometry Derived from Multitemporal Laserscanning Data," in *Internat. Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences (IAPRS & SIS)*, O. Altan, Ed., vol. XXXV Part B2, Istanbul, Turkey: ISPRS, Jul. 2004, pp. 428–433. [Online]. Available: <http://www.isprs.org/proceedings/XXXV/congress/comm2/comm2.aspx>.
- [VS05] M. Vieira and K. Shimada, "Surface mesh segmentation and smooth surface extraction through region growing," *Computer Aided Geometric Design*, vol. 22, no. 8, pp. 771–792, 2005, ISSN: 0167-8396. DOI: [DOI: 10.1016/j.cagd.2005.03.006](https://doi.org/10.1016/j.cagd.2005.03.006). [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0167839605000282>.
- [Vih12] M. Vihinen, "How to evaluate performance of prediction methods? Measures and their interpretation in variation effect analysis," *BMC Genomics*, vol. 13(Suppl 4), S2, 2012.

- [WJY+08] L. Wang, H. Jin, R. Yang, and M. Gong, "Stereoscopic inpainting: Joint color and depth completion from stereo images," in *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, IEEE, Anchorage, AK, USA: IEEE Computer Society, 2008, pp. 1–8.
- [WTT+07] C. Wang, C. Thorpe, S. Thrun, M. Hebert, and H. Durrant-Whyte, "Simultaneous Localization, Mapping and Moving Object Tracking," *The International Journal of Robotics Research*, vol. 26, no. 9, pp. 889–916, 2007. DOI: [10.1177/0278364907081229](https://doi.org/10.1177/0278364907081229). eprint: <http://ijr.sagepub.com/content/26/9/889.full.pdf+html>. [Online]. Available: <http://ijr.sagepub.com/content/26/9/889.abstract>.
- [WWS+03] P. Wonka, M. Wimmer, F. Sillion, and W. Ribarsky, "Instant Architecture," *ACM Transactions on Graphics*, vol. 22, no. 3, pp. 669–677, Jul. 2003, ISSN: 0730-0301. DOI: [10.1145/882262.882324](https://doi.org/10.1145/882262.882324). [Online]. Available: <http://doi.acm.org/10.1145/882262.882324>.
- [Wan08] J. Wang, *Graph Based Image Segmentation: A Modern Approach*. VDM Verlag Dr. Müller Aktiengesellschaft & Co, Dec. 2008, ISBN-10: 3639110749, ISBN-13: 978-3639110746.
- [XFZ+09] J. Xiao, T. Fang, P. Zhao, M. Lhuillier, and L. Quan, "Image-based street-side city modeling," *ACM Trans. Graph.*, vol. 28, 114:1–114:12, 5 Dec. 2009, ISSN: 0730-0301. DOI: [http://doi.acm.org/10.1145/1618452.1618460](https://doi.org/10.1145/1618452.1618460). [Online]. Available: <http://doi.acm.org/10.1145/1618452.1618460>.
- [XVP13] W. Xiao, B. Vallet, and N. Paparoditis, "Change Detection in 3D Point Clouds Acquired by a Mobile Mapping System," *ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. 1, no. 2, pp. 331–336, 2013.
- [YMP+09] B. Yersin, J. Maim, J. Pettré, and D. Thalmann, "Crowd patches: populating large-scale virtual environments for real-time applications," in *Proceedings of the 2009 symposium on Interactive 3D graphics and games*, ACM, 2009, pp. 207–214.
- [ZF07] R. Zeibak and S. Filin, "Change detection via terrestrial laser scanning," *Internat. Archives of Photogrammetry and Remote Sensing*, vol. 36, no. 3/W52, pp. 430–435, 2007.
- [ZSW+10] Q. Zheng, A. Sharf, G. Wan, Y. Li, N. J. Mitra, D. Cohen-Or, and B. Chen, "Non-local Scan Consolidation for 3D Urban Scenes," *ACM Transactions on Graphics*, vol. 29, no. 4, 94:1–94:9, Jul. 2010, ISSN: 0730-0301. DOI: [10.1145/1778765.1778831](https://doi.org/10.1145/1778765.1778831). [Online]. Available: <http://doi.acm.org/10.1145/1778765.1778831>.
- [ZZL+10] X. Zhu, H. Zhao, Y. Liu, Y. Zhao, and H. Zha, "Segmentation and classification of range image from an intelligent vehicle in urban environment," in *IEEE/RSJ Int. Conf. on Intellig. Robots and Systems (IROS)*, Taipei, Taiwan, Oct. 2010, pp. 1457–1462.
- [3D 14] 3D data acquisition of Capitol Hill. (2014), [Online]. Available: <https://www.flickr.com/photos/72128514@N04/6604642639/sizes/l/in/photostream/>.
- [3Dl14] 3Dlasermapping Inc. (2014), [Online]. Available: <http://www.3dlasermapping.com/applications>.
- [CyA14] CyArk. (2014), [Online]. Available: <http://archive.cyark.org>.
- [Goo14] Google mobile mapping fleet. (2014), [Online]. Available: <https://www.google.com/intl/en-US/maps/about/behind-the-scenes/streetview/>.
- [LTA14] LTA. (2014). Maintaining Our Roads and Facilities, [Online]. Available: <http://www.lta.gov.sg/content/ltaweb/en/roads-and-motoring/road-safety-and-regulations/maintaining-our-roads-and-facilities.html>.

- [Lei14] Leica Geosystems. (2014), [Online]. Available: <http://www.prweb.com/releases/2014/10/prweb12259559.htm>.
- [Nav14] Navteq navigation. (2014), [Online]. Available: <http://navteq.navigation.com>.
- [Opt14] Optech Inc. (2014), [Online]. Available: <http://www.optech.com>.
- [Poi14] Point Grey Inc. (2014), [Online]. Available: <http://www.ptgrey.com>.
- [Rie14] Riegl Inc. (2014), [Online]. Available: <http://www.riegl.com>.
- [Ste14] Stereopolis. (2014), [Online]. Available: <http://recherche.ign.fr/labos/matis/arSYMPA.php>.
- [Ter14] Terranumerica. (2014), [Online]. Available: <http://terranumerica.com>.
- [The14] The American Heritage Science Dictionary. (2014), [Online]. Available: <http://dictionary.reference.com/browse/cartography>.
- [Top14] Topcon Inc. (2014), [Online]. Available: <http://www.topcon.com>.
- [Urb14] Urbandive. (2014), [Online]. Available: <http://www.urbandive.net>.
- [Vis11] Vis center's LiDAR truck. (2011), [Online]. Available: <http://viscenter.wordpress.com/2011/01/06/taking-google-street-view-technology-to-the-next-level>.
- [iTo14] iTowns. (2014), [Online]. Available: <http://www.itowns.fr>.