Line Based Multi-range Asymmetric Conditional Random Field for Terrestrial Laser Scanning Data Classification

Chao Luo

A DISSERTATION SUBMITTED TO THE FACULTY OF GRADUATE STUDIES
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

GRADUATE PROGRAM IN EARTH AND SPACE SCIENCE
YORK UNIVERSITY
TORONTO, ONTARIO

DECEMBER, 2015

Abstract

Terrestrial Laser Scanning (TLS) is a ground-based, active imaging method that rapidly acquires accurate, highly dense three-dimensional point cloud of object surfaces by laser range finding. For fully utilizing its benefits, developing a robust method to classify many objects of interests from huge amounts of laser point clouds is urgently required. However, classifying massive TLS data faces many challenges, such as complex urban scene, partial data acquisition from occlusion. To make an automatic, accurate and robust TLS data classification, we present a line-based multi-range asymmetric Conditional Random Field algorithm.

The **first contribution** is to propose a line-base TLS data classification method. In this thesis, we are interested in seven classes: building, roof, pedestrian road (PR), tree, low man-made object (LMO), vehicle road (VR), and low vegetation (LV). The line-based classification is implemented in each scan profile, which follows the line profiling nature of laser scanning mechanism. It is rather straightforward to extract lines in each scan profile, and the appearance of scanned objects can be characterized using lines. Ten conventional local classifiers are tested, including popular generative and discriminative classifiers, and experimental results validate that the line-based method can achieve satisfying classification performance. However, local classifiers implement labeling task on individual line independently of its neighborhood, the inference of which often suffers from similar local appearance across different object classes. The **second contribution** is to propose a multi-range asymmetric Conditional Random Field (maCRF) model, which

uses object context as post-classification to improve the performance of a local generative classifier. The maCRF incorporates appearance, local smoothness constraint, and global scene layout regularity together into a probabilistic graphical model. The local smoothness enforces that lines in a local area to have the same class label, while scene layout favours an asymmetric regularity of spatial arrangement between different object classes within long-range, which is considered both in vertical ("above-bellow" relation) and horizontal ("front-behind") directions. The asymmetric regularity allows capturing directional spatial arrangement between pairwise objects (e.g. it allows ground is lower than building, not vice-versa). The **third contribution** is to extend the maCRF model by adding across scan profile context, which is called Across scan profile Multi-range Asymmetric Conditional Random Field (amaCRF) model. Due to the sweeping nature of laser scanning, the sequentially acquired TLS data has strong spatial dependency, and the across scan profile context can provide more contextual information. The **final contribution** is to propose a sequential classification strategy. Along the sweeping direction of laser scanning, amaCRF models were sequentially constructed. By dynamically updating posterior probability of common scan profiles, contextual information propagates through adjacent scan profiles.

The proposed methods are finally evaluated using datasets collected at two different sites, York Village and York Blvd. And the experimental results validated the advantage using multi-range contexts and sequential processing. As line extraction is implemented in each scan profile, the line-based method has great potential on real-time TLS data classification. Due to the limited hardware condition, implementing the

algorithm in a real-time environment is not available. Thus we simulate the line-based real-time classification using off-time TLS data.

Acknowledgements

I would like to extend my sincere gratitude to all of the people who supported and helped me in the past six and half years.

At first, a very special word of appreciation gives to Dr. Gunho Sohn, my supervisor, for his clear guidance, insightful suggestions, and challenging discussions that I finished my PhD research on Machine Learning topic with a total different academic background, Geographic Information System. His continuous encouragement and supports created a stimulating academic environment that I appreciated.

I would like to thank Dr. Costas Armenakis (York University, GeoICT), and Dr. James Elder (York University, GeoICT), the supervision committee members, for their scientific suggestions, valuable advices, and encouragements which significantly speed up my research progress. Particular gratitude to Dr. James Elder, who opened the door of statistical machine learning to me, and guided me to solve the real world problem using probabilistic models. I also thank my previous supervisors (Master degree), Dr. Huayi Wu (Wuhan Univeristy, China) and MS. Aihong Song (Wuhan Univeristy, China), for their continuous encouragements and helps.

Many individuals have contributed in different ways to my PhD research. I would like to appreciate Dr. Yooseok Jwa (GeoICT, York University) for helping me to design the framework of object recognition from mobile railway laser scanning data; Dr. Heungsik Kim (GeoICT, York University) for helping me to explore potential application of the line-based classification method on mobile urban laser scanning data, including

line extraction, feature extraction; Mr. Jaewook Jung (GeoICT, York University) for helping me to develop features and providing me visualization tool of TLS data; and Dr. Mojgan Jadidi (GeoICT, York University), who gave me many suggestions on thesis writing.

I wish to thank GEOmaticsfor Informed DEcisions (GEOIDE), Ontario Centres of Excellence (OCE), Natural Sciences and Engineering Research Council of Canada (NSERC) Discovery, Hyundai, and China Scholarship Council (CSC) for arranging the financial support for this PhD research. Thank all my research partners from GeoICT, Langyue Wang and Ravi Persad, and partners from CVR at York University, Eduardo Corral-Soto and Ron Tal. Warm thanks to all other student members and visiting scholars at GeoICT of York University, who helped me in my research, such as Dr. Yawen Liu, Dr. Jili Li, Dr. Connie Ko, Dr. Junjie Zhang, Julien Lee-Chee-Ming.

Finally, I deeply thank my parents, Guixian Chen and Fuhua Luo, for their endless love and continuous encouragement. They gave me support through their understanding for years at such long distance.

Table of Contents

Abstract ii

Acknowledgements v

Table of Contents vii

List of Tables xii

List of Figures xiv

List of Abbreviations xvi

Chapter 1: Introduction 1

1.1 Problem Domain 1

1.1.1 Research Context 1

1.1.2 Problem Statement 4

1.2 Research Objectives 9

1.3 Methodology Overview 10

1.4 Outline 13

Chapter 2: Background 15

2.1 Terrestrial Laser Scanning Technology 15

2.1.1 Laser Scanning Mapping 15

2.1.2 Introduction of Terrestrial Laser Scanning 17

2.1.3 Terrestrial Laser Scanning Data Classification 19
2.1.3.1 Rule Based Classification 21
2.1.3.2 Machine Learning 22
2.2 Context Based Object Recognition 25
2.2.1 Object Context 26
2.2.2 Scene Layout Prior 28
2.3 Probabilistic Graphical Model 30
2.3.1 Probabilistic Graphical Model 30
2.3.2 Markov Random Field 31
2.3.3 Conditional Random Field 34
2.4 Chapter Summary 36

Chapter 3: Line-based TLS Data Classification 37
3.1 Line-based Classification 38
3.1.1 Motivation of Line-based Classification 38
3.1.2 Scan Profile Generation 40
3.1.3 Line Segment Extraction 41
3.2 Linear Feature Extraction 44
3.2.1 Local Features 44
3.2.2 Contextual Features 45
3.2.3 Feature Selection 46
3.3 Generative Classifiers 47

3.3.1 Naïve Bayes（NB） 49
3.3.2 Multivariate Gaussian (MG) 52
3.3.3 Gaussian Mixture Model (GMM) 53
3.4 Discriminative Classifiers 56
3.4.1 K-Nearest Neighbors (KNN) 57
3.4.2 Logistic Regression (LR) 59
3.4.3 Support Vector Machine (SVM) 61
3.4.4 Artificial Neural Networks (ANN) 64
3.4.5 Decision Tree (DT) 66
3.4.6 Random Forest (RF) 68
3.4.7 Adaptive Boosting (AdaBoost) 70
3.5 Experiment Results 72
3.5.1 Experimental Data 73
3.5.2 Qualitative Analysis 76
3.5.3 Quantitative Analysis 78
3.6 Chapter Summary 84

Chapter 4: Along Scan Profile Conditional Random Field 85
4.1 Methodology Overview 86
4.2 Line Adjacent Graph 89
4.3 Short Range CRF (srCRF) 92
4.3.1 Graph Construction 92

4.3.2 Association Potential 93
4.3.3 Interaction Potential 94
4.4 Long Range CRF 94
4.4.1 Scene Layout 95
4.4.2 Long Range Vertical CRF (lrCRF(V)) 97
4.4.2.1 Graph Construction 97
4.4.2.2 Association and Interaction Potential 98
4.4.3 Long Range Horizontal CRF (lrCRF(H)) 102
4.4.3.1 Graph Construction 102
4.4.3.2 Association and Interaction Potential 103
4.5 Multi-Range CRF 105
4.5.1 Product Combination of Multiple CRF Classifiers 105
4.5.2 Single Integrated Model 107
4.6 Training and Inference of CRF 109
4.6.1 Training the Association and Interaction Potentials 111
4.6.2 Training the Weight of Potential Terms 112
4.6.3 Inference 114
4.7 Experiment Results 117
4.7.1 Qualitative Analysis 120
4.7.2 Quantitative Analysis 128
4.8 Chapter Summary 133

Chapter 5: Across Scan Profile Conditional Random Field..... 134

5.1 Context between Scan Profile..... 135

5.2 Across Scan Profile CRF Model..... 136

5.2.1 Graph Construction..... 137

5.2.2 Potential Design..... 140

5.2.3 Parameter Learning and Inference..... 140

5.3 Context Propagation through Adjacent Scan Profile..... 142

5.4 Experiments of Across Scan Profiles CRF models..... 146

5.5 Additional Experiments..... 149

5.5.1 SVM Based CRFs..... 149

5.5.2 York Blvd Datasets..... 152

5.5.3 Train Classifiers using York Village Dataset and Test on York Blvd Dataset

……..157

5.6 Chapter Summary..... 163

Chapter 6: Discussions..... 165

6.1 Conclusions..... 166

6.2 Future Work..... 170

Bibliography..... 173

List of Tables

Table 2.1: Technical specifications of RIEGL LMS Z-390i 19

Table 3.1: Object categorization of experimental dataset. 74

Table 3.2: Number of laser point, scan profile, line segment in York Village dataset. 75

Table 3.3: Confusion matrix of GMM classifier of data YV2. 79

Table 3.4: Confusion matrix of SVM classifier of data YV2. 79

Table 3.5: Precision of each class in ten classifiers. 82

Table 3.6: Recall of each class in ten classifiers. 82

Table 3.7: F1 score of each class in 10 classifiers. 83

Table 4.1: Total number of the spatial entities extracted from York Village datasets. .. 118

Table 4.2: Positive and negative transition from GMM to each CRF classifier. 125

Table 4.3: Test accuracy of GMM and the four CRF models. 129

Table 4.4: Confusion matrix of srCRF classifier on data YV2. 129

Table 4.5: Confusion matrix of lrCRF(V) classifier on data YV2. 130

Table 4.6: Confusion matrix of lrCRF(H) classifier on data YV2. 130

Table 4.7: Confusion matrix of maCRF classifier on data YV2. 130

Table 5.1: Total number of the spatial entities extracted from York Village datasets. .. 146

Table 5.2: Test Accuracy of sequential CRF Models. 149

Table 5.3: Test Accuracy of sequential CRF Models. 151

Table 5.4: Total number of the spatial entities extracted from York Blvd datasets. 154

Table 5.5: Test Accuracy of the proposed classifiers on York Blvd datasets..... 157

List of Figures

Figure 1.1: Examples of objects in terrestrial laser scanning data. 6

Figure 3.1: Examples of line extraction. 43

Figure 3.2: Post-processing using the Douglas–Peucker algorithm. 44

Figure 3.3: Neighborhood selection for context feature. 46

Figure 3.4: Averaged test accuracy over 5-fold cross validation. The value 59

Figure 3.5: Typical structure of ANN with three layers. 64

Figure 3.6: Averaged test accuracy over 5-fold cross validation as different minimum leaf size was selected. 68

Figure 3.7: Real scene of the York Village Data. 73

Figure 3.8: Classification result of GMM, SVM and ground truth. 77

Figure 3.9: Averaged accuracy of ten classifiers. 80

Figure 4.1: Height distributions of seven classes 87

Figure 4.2: Example of grid system and line-cell occupancy relations. 90

Figure 4.3: Multi-range neighborhood searching for each cell 91

Figure 4.4: Prior and likelihood estimation for vertical interaction term. 101

Figure 4.5: Prior and likelihood estimation for horizontal interaction term. 104

Figure 4.6: Parameter learning of maCRF model using SGD algorithm on data YV1. . 119

Figure 4.7: Classification result of the four CRFs of the data YV2. 121

Figure 4.8: Example of single scan profile analysis with different context 123

Figure 4.9: Label transition from GMM to srCRF. 126

Figure 4.10: Label transition from GMM to lrCRF(V). 126

Figure 4.11: Label transition from GMM to lrCRF(H). 127

Figure 4.12: Label transition from GMM to maCRF. 128

Figure 4.13: Precision of each class in five methods. 131

Figure 4.14: Recall of each class in five methods. 132

Figure 4.15: F1-Score of each class in five methods. 132

Figure 5.1: Across and along scan profile neighborhood. 138

Figure 5.2: Example of across/with scan profile multi-range graph. 139

Figure 5.3: Each amaCRF model is independent with each other. 142

Figure 5.4: Contextual information propagates through adjacent scan profiles. 144

Figure 5.5: Parameter learning of maCRF model on data YV1. 147

Figure 5.6: Classification result of the amaCRF model and sequential processing on the data YV2. 148

Figure 5.7: Parameter learning of the amaCRF model (SVM) on data YV1. ……150

Figure 5.8: Classification result of the SVM-based amaCRF model and sequential processing on the data YV2. 152

Figure 5.9: Surveying locations of York Blvd Dataset. 153

Figure 5.10: Classification results of the GMM and GMM-based amaCRF model with sequential processing on the data YB2. 155

Figure 5.11: Classification results of the SVM and SVM-based amaCRF model with sequential processing on the data YB2. 156

List of Abbreviations

AdaBoost	Adaptive Boosting
ALS	Airborne Laser Scanning
amaCRF	Across scan profile Multi-range Asymmetric CRF
amaCRF+	Across scan profile Multi-range Asymmetric CRF with sequential modeling
AMN	Associative Markov network
ANN	Artificial Neural Network
BP	Belief Propagation
CART	Classification And Regression Tree
CRF	Conditional Random Field
DLG	Digital Linear Graph
DOM	Digital Ortho-image Model
DRM	Digital Raster graph Model
DT	Decision Tree
DTM	Digital Terrain Model
EM	Expectation Maximization
GLCM	Gray-Level Co-occurrence Matrix
GMM	Gaussian Mixture Model
GPS	Global Position System
HMM	Hidden Markov Model

IMU	Inertial Measurement Unit
KNN	K-Nearest Neighbor
LBP	Loopy Belief Propagation
LiDAR	Light Detection And Ranging
LLF	Label Layout Filter
LMO	Low Man-made Object
LOD	Level of Detail
LR	Logistic Regression
LV	Low Vegetation
maCRF	Multi-range Asymmetric CRF
MG	Multivariate Gaussian
MLE	Maximum Likelihood Estimation
MRF	Markov Random Field
NB	Naïve Bayes
PR	Pedestrian Road
PS	Phase Shift
RANSAC	RANdom SAmple Consensus
RBF	Radial Basis Function
RF	Random Forest
SAMME	Stagewise Additive Modeling using a Multi-class Exponential loss function
SGD	Stochastic Gradient Descent

SVM	Support Vector Machine
TLS	Terrestrial Laser Scanning
ToF	Time-of-Flight
VR	Vehicle Road
YV	York Village
YB	York Blvd

Chapter 1
Introduction

1.1 Problem Domain

1.1.1 Research Context

Municipal infrastructure refers to the fundamental facilities and systems that serve for the public. Typical infrastructures include public buildings, transportation networks, bridges, train/bus stations, education facilities, and hospital service, etc. Urbanization is the global trend but the growing urban population brings challenges to municipal infrastructure management. The "State of World Population 2014", published by the United Nations Population Fund (UNPF) that infrastructure shortage is a significant problem in developing counties, especially those counties with fast population growth (UNPFA, 2014). Every day, new urban infrastructures are built while existing infrastructures deteriorate, which poses a great demand for a sustainable management of municipal infrastructure system, including construction, monitoring, and maintenance. A sustainable municipal infrastructure management system enables city governments and related civic service provides better services to the residences. Many governments have realized the significance of a sustainable municipal infrastructure system, and have already taken actions, such as Canada's National Guide to Sustainable Municipal Infrastructure (Boudreau and Brynildsen, 2003), and Singapore's Future Cities Laboratory (FCL) (Axhausen, 2011).

Risk assessment of infrastructures is one of the key elements of an infrastructure management system. A 3D municipal infrastructure system can significantly reduce the amount of cognitive effort, achieve a rapid response to plausible risks, and improve the efficiency of the decision-making process (Kolbe et al. 2005, Zlatanova 2008). As one of essential components of a municipal infrastructure system, 3D urban modeling is a crucial work. Recently, 3D photo-realistic urban modeling, especially the 3D building modeling has been attracting much attention from photogrammetric and computer vision communities as there is an increasing demand for urban modeling applications, such as urban planning, augmented reality and individual navigation. In 3D city visualization, the same city object needs to be represented with different geometric complexities according to users' request. The Level of Detail (LOD) is usually used to describe the geometric complexity of a 3D building, and allows the geometry of objects to be represented in varying accuracies and details (Emgard and Zlatanova, 2008). Lee and Nevatia (2003) proposed a hierarchical representation structure of 3D building models for 3D urban reconstruction, in which the visualization quality of the building model increases when LOD level upgrades. The coarsest LOD0 is essentially a 2.5D Digital Terrain Model (DTM), and building models in LOD0 do not contain volume. In the LOD1 level, building models are referred to as a block with flat roof structures. Both the outer facade and roof of the buildings at LOD2 level can be represented with multiple faces. Compared with lower-level models, LOD3 goes further by representing more detailed facade geometries, such as wall, roof, door, window, sidewall, window sill .etc. The LOD4 model completes a LOD3 model by adding interior structures.

For modeling realistic facilities, capturing digitized 3D geometric and textual information is the first step. Photogrammetry has been and is still used as the main method of collecting geo-spatial information of Earth surfaces over the past century. Photogrammetry is passive remote sensing technology, and recovers 3D geometric and photogrammetric information of real world by matching stereo pair images (Wolf et al., 2000). Typical products of photogrammetry-based methods include digital elevation model (DEM), digital ortho-image model (DOM), digital raster model (DRM), and digital linear graph (DLG), which have been widely used for urban planning and management. However, the main drawback of photogrammetric workflow is the low efficiency in generating dense 3D coordinator from stereoscopic pictures and somtimes-manual work (Alshawabkeh, Y., 2006). Recently, laser scanning compensates for this drawback of photogrammetry by providing direct 3D data and has become a standard tool for 3D data collection.

Airborne laser scanning (ALS) has been used for surveying and mapping since the 1980s, such as forest surveying (Rutzinger et al., 2008; Vehmas et al, 2009; Zhang and Sohn, 2010; Kantola et al, 2013), digital surface modeling (Kraus and Pfeifer 1998). Since ALS collects data from bird's-eye perspective, it can capture roofs of buildings efficiently but only get part of building facade that is essential for LOD3 model. Due to close range, high accuracy and cost-effectiveness, Terrestrial Laser Scanning (TLS) has been rapidly adopted for collecting massive urban street-view data. According to the platform carrying laser scanner, it can be categorized as tripod based (static TLS) or

vehicle based (Mobile TLS). Both of them could provide rich geometric information of building facades for producing realistic LOD3 city models (Pfeifer and Briese, 2007).

As the TLS is a relative young technology for infrastructure surveying, many problems on both hardware and software need to be solved. Popular research topics related with TLS data processing are calibration (Lichti et al., 2005; Schulz, 2007), multiple station registration (Al-Manasir and Fraser, 2006; Dold and Brenner, 2006; Barnea and Filin, 2007), geo-referencing (Lichti et al., 2005; Reshetyuk, 2009), integration of ALS and TLS (Böhm and Haala, 2005; Bremer and Sass, 2012), segmentation (Boulaassal et al., 2007; Moosmann et al., 2009; Wang and Shan, 2009; Aijazi et al, 2013), and classification (Belton and Lichti, 2006; Lim and Suter, 2008; Lim and Suter, 2009; Munoz et al., 2009; Pu and Vosselman, 2009; Brodu and Lague, 2013; Luo and Sohn, 2013; Luo and Sohn, 2014).

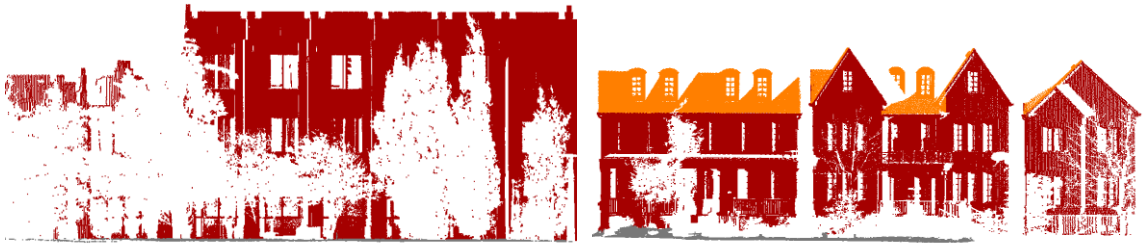
1.1.2 Problem Statement

According to spatial entity to label, classification algorithms for TLS data can be categorized into three types: point-based (Triebel, et al, 2006; Munoz et al, 2008), line-based (Manandhar and Shibasaki, 2001; Zhao et al., 2010) and surface-based (Belton and Lichti, 2006; Pu and Vosselman, 2009). The point-based method directly labels individual laser points. Though both line-based and surface-based methods partition the point cloud into homogeneous segments, such as line, plane, and cylinder firstly, and then label these segments. Since single laser point does not provide any semantic information about the scanned objects; therefore, point-based classification method has higher risk of misclassification than line-based classification. Although surface-based method reduces

computational cost by reducing the number of spatial entities to be labeled, it is still computational expensive in surface segmentation, which requires constructing adjacent relationship in 3D space. In contrast, line segmentation is implemented in 2D space. This advantage in computational efficiency of line-based method has been approved by (Jiang and Bunke, 1994). Indeed, extracting line in profiling data is more straightforward where the appearance of scanned objects can be well-characterized using lines. Moreover, as a higher level geometric primitive, lines carry more sematic information than single point about the scanned objects. Therefore we finally chose lines as geometric primitive for TLS data classification. The line-based classification method starts with extracting lines in each scan profile and subsequently labels these lines based on features vector.

Object recognition from massive TLS data still faces many challenges, such as complex urban scene, appearance variations, occlusions and various point density with range. For instances, the urban street scene is composed of various objects such as building facade that can include walls, windows, doors, columns, balconies, etc. Appearance variations means the same class could have great variation on appearance, for example, different tree species have different shapes (Figure 1.1(a)) and structures and building at different locations have different architectural styles (Figure 1.1(b)). In IQmulus & TerraMobilita mobile laser scanning data, pedestrian class can be further categorized into seven subdivision such as: still pedestrian, walking pedestrian, running pedestrian, stroller pedestrian, holding pedestrian, leaning pedestrian and other pedestrian (Vallet et al., 2015).

(a)



(b)

Figure 1.1: Examples of objects in terrestrial laser scanning data. Color setting: green-tree; brown-building; orange-roof.

Due to the limit of line-of-sight of static laser scanning, some objects are occluded by other objects that are closer to the laser scanner, which results in some holes in the occluded object. It is observed in Figure 1.1(b) that trees are in front of buildings and thus many tree-shape holes are founded in the building area. Occlusion reduces the information about the objects of interest and brings problem for further data processing.

The point density varies with the range between laser scanner and objects. The point density decreases when the distance between the object and the laser scanner increases. The various point density will make the same type of objects have different geometric appearance when the distance changes.

All of problems mentioned above will cause the problem of feature ambiguity, which is also called feature overlap. Feature distribution of different classes could overlap in the feature space, which results in a non-linear separable classification problem(Lalonde et al., 2005). Building classifiers only relying on these features with serious ambiguity poses risk of misclassification (Trappenberg and Back, 2000).

A popular solution to solve the problem of feature ambiguity is applying object context, or context for short, which can be defined as dependencies or correlation among spatial entities (such as points, lines, or surfaces) in a scene. With context, a spatial entity is perceived associated with its surrounding neighbors rather than independently. Classifiers that do not consider context are called local classifier and those considering context are called context based classifiers. Markov Random Field (MRF) was proposed by Clifford (1990), and is a commonly used context based model. The MRF model has been approved to be effective on laser scanning data classification (Anguelov et al., 2005; Triebel et al., 2006; Munoz et al., 2008; Zhang and Sohn; Häselich et al., 2011). However, MRF can only maximize the local label homogeneity between adjacent entities, but fails to capture those relations at global level. For example, MRF can model the relations as "the building is likely to be neighbor with the building", but is unable to express interactions between different objects, such as "the building is above the ground but below the roof". Therefore, a MRF based method is probably to produce an over-smoothness (minority objects are misclassified as the class that its surrounding majority objects are) classification result (Schindler, 2012).

To avoid the over-smoothness problem, the global object scene layout is usually considered. The scene layout corresponds to the relative locations of objects in a scene, and assumes that image (or a point cloud) is not a random collection of independent pixels (or points), but follows some rules on spatial arrangement. With the prior knowledge on scene layout, it is expected to estimate what types of objects could be above or below building, and so on. The scene layout can be modeled as a co-occurrence matrix, but it is more frequently modeled as data-dependant interaction potential function in a CRF model. Many achievements has been made on applying scene layout for object recognition from images (Winn and Shotton, 2006; Heesch and Petrou, 2010; Jahangiri et al., 2010; Ding et al., 2014). But just a few publications focus on applying scene layout on TLS data classification. Pu and Vosselman (2009) applied manually defined scene layout rules on classifying TLS data. Although such rule-based method is easily implemented and achieved satisfying classification result, but, it cannot cover all the rules that govern object layout, let alone conditions behind these rules.

All contexts mentioned above, local smoothness and scene layout provide contextual information on different scales. Each single context contains partial contextual information, so relying only on a single context could be risky as “part of the evidence is spent to specify the model” (Leamer, 1978). It is promising to combine all types of contextual information together in one CRF model.

1.2 Research Objectives

The main objective of this thesis is to develop an automatic, accurate and robust classification algorithm for TLS data processing. Accordingly, the specific objectives are as follows:

1. **Develop a line-based TLS data classification algorithm**. We will explore the potential of lines as the geometric primitive for TLS data classification. The lines extraction is based on the "line profiling" nature of laser scanning. Each scan profile is considered as a stream of sequentially observed laser points, and those neighboring points that have small range difference were merged into a line. The line is the highest level geometric primitive that can be extracted from profiling data, so that the line primitives are expected to be optimal for characterizing street objects and gaining computational benefits. As line extraction is implemented within each scan profile, it is also suitable for a real-time point cloud processing.
2. **Enhance classification accuracy using multi-range contexts**. As mentioned previously, complex urban scene, appearance variations, occlusions and various point density with range can result in the problem of features ambiguity. Relying only on these features with ambiguity, conventional local classifiers cannot properly identify the boundaries between classes. To improve the classification performance of local classifier, multi-range (short range and long range) contexts are introduced. The short range context imposes local smoothness constraint that neighboring lines are likely to have the same class label. While the long range context imposes scene layout regularity. The scene layout indicates spatial

arrangements of objects in the space, both in vertical ("above-below" relation) and horizontal ("front-behind") directions. Moreover, local smoothness constraint is also considered between lines at adjacent scan profiles, which makes lines gain additional contextual information.

3. **Enhance classification accuracy using context propagation**. The acquisition of laser scanning data can be regarded as the process that a set of vertical scan profiles are sequentially obtained along the azimuth direction. Thus, object can be viewed as "growing" along the direction that laser scanner sweeps, and so the class label also can be propagated in the spatial domain. To make the contextual information propagate from one scan profile to other scan profiles that far away, a sequential processing can be used. Each time, posterior of the previous multi-range based classifier is used as association term of the next multi-range based, so that posterior probability is dynamically updated and confidence gets stronger and stronger.

1.3 Methodology Overview

In this thesis, we are interested in classifying static terrestrial laser scanning data. The raw data we get from the laser scanner include 3D coordinates (X, Y, Z), range, azimuth angle and zenith angle. Line is used as the primitive entity of TLS data classification. The whole TLS data was firstly split into a set of vertical scan profiles according to azimuth angle. Points in each scan profile were further segmented into a set of lines based on range analysis (Manandhar and Shibasaki, 2001) and the Douglas-Peucker algorithm (Hershberger and Snoeyink, 1992). Then multi-scale features were extracted for each

line, including local appearance, circle- based and column-based features. Then the Principle Component Analysis (Krzanowski, 2000) was applied to reduce the feature dimension. To validate the effectiveness of line based TLS data classification, both generative and discriminative classifiers were tested, including Naïve Bayes (Bishop, 2006), Multivariate Gaussian (Bishop, 2006), Gaussian Mixture Model (Bishop, 2006), K-Nearest Neighbor (Bishop, 2006), Logistic Regression (Menard, S., 2002), Support Vector Machine (Burges, 1998), Artificial Neural Network (Bendiktsson et al., 1990), Decision Tree (Quinlan, 1986), and two Decision Tree based ensembles, Random Forest (Breiman, 2001) and Adaptive Boosting (Freund et al., 1995).

In order to overcome the problem of feature ambiguities in local classifiers, multi-range contexts along scan profile were used, including short range context that enforces local smoothness, as well as the long range vertical and horizontal context that provide priori information of scene-layout compatibility. The three types of adjacent relations of lines were defined with the assistant of a grid system. At first, the scan profile was projected into 2D space (XY-Z) and then the 2D space was quantized in a grid along the Z and XY directions, with cell size of 0.5m by 0.5m. Neighbor searching of a line is based on neighboring relations of cells. In particular, we adopted an asymmetric interaction potential to capture directional scene layout (e.g. ground is lower than building, not vice-versa). To integrate context into a classification problem, Conditional Random Filed (Lafferty et al., 2001) was used. Finally all the three different contexts are integrated together in the multi-range asymmetric CRF (maCRF) model. To compare the

effect of different types of contexts and validate the advantage of multi-range context, three single range CRF models were also constructed.

The maCRF was also extended to across scan profiles, which is called across scan profile multi-range asymmetric CRF (amaCRF). The amaCRF graph was built on three consecutive scan profiles; and four types edges are considered, short range, long range vertical and horizontal, as well as across scan profile edge. To make the contextual information propagate from one scan profile to other scan profiles that indirectly connect with it, a sequential processing was used (amaCRF+). Each time, posterior of the previous amaCRF classifier is used as association term of the next amaCRF, so that posterior probability is dynamically updated and confidence gets stronger and stronger.

There are two types of parameters in each of the five CRF models: parameters in each potential term, and parameters weighting the relative influence of potential terms. Learning all of the parameters simultaneously in each CRF models is still a challenge; thus, parameter learning was divided into two stages. At first, parameters in association and each interaction terms were learned individually, following which the weights of association and interaction terms were learned using Stochastic Gradient Descent (Vishwanathan et al., 2006). Given learned parameters, the loopy belief propagation (Frey et al., 1998), a variant of belief propagation (BP), was used for inference; and the final class label was selected by maxizing node belief.

Finally the proposed classifier was tested on several TLS data collected in York Village, Toronto. The performance of classification was evaluated both qualitatively and quantitatively. Quantitative measure includes confusion matrix, overall accuracy,

precision, recall and F1-score. To track how different types of context affect the classification result, one representative scan profile was selected for comparative analysis. In order to examine which classes are sensitive to which type of context, label transition analysis was analyzed, which is based on comparing label change from local classifier to CRF model.

To test the whether the function of multi-range context is dependent on association terms, both output of GMM and SVM were used as association term. To validate that the algorithm is not only work on a specific scene, another TLS data were tested, collected at York Blvd, Toronto.

1.4 Outline

Chapter 2: We present literature review on mechanism of terrestrial laser scanning technology and popular classification methods. Comparison of various classification methods are discussed, including rule-based methods verses machine learning methods, generative classifiers verses discriminative classifiers, local classifiers verses context based graphical models, MRFs verses CRFs. In particular, the information loss challenges in TLS data classification and potential of scene layout for enhancing classification performance is discussed.

Chapter 3: At first, data prepossessing for line-based classification will be introduced, including technique characteristics of the experimental laser scanner, data collection, data preprocessing, line segment extraction, and feature generation. Principle, learning and inference of three generative classifiers and seven discriminative classifiers

are presented. Finally, the ten classifiers are tested on TLS data collected at York Village, and performances of these classifiers are compared.

Chapter 4: We propose a multi-range asymmetric CRF model (maCRF) to enhance classification performance. Limitation of local classifier is discussed first using the experimental result of GMM for example. Then three types of object context within along scan profile are exploited: short range context that enforces local smoothness, as well as long range vertical and horizontal context that provide priori information of scene layout compatibility of objects. Three single range CRF models and the integrated multi-range asymmetric CRF model are presented. The output of GMM is used as association term of the four CRF models. Performances of the four CRF models are evaluated using the same experimental data, and compared with the results of GMM classifier.

Chapter 5: The maCRF model is extended from only along scan profile contexts to the across scan profile context (amaCRF). Furthermore, a sequential knowledge propagation method (amaCRF+) is proposed to make contextual information propagate through adjacent scan profiles. To validate that the multi-range context CRF model is not sensitive to the association term, output of GMM (generative) was replaced with SVM (discriminative). To validate that the multi-range context CRF model is not sensitive to dataset, TLS data collected at a different site, York Blvd, was also tested.

Chapter 6: conclusions of this study and directions of future works.

Chapter 2
Background

2.1 Terrestrial Laser Scanning Technology

2.1.1 Laser Scanning Mapping

Since the first laser instrument for distance measurement was invented in 1966, laser scanning has been the standard for a wide range of applications (Heritage and Large, 2009). LiDAR, which stands for Light Detection and Ranging, is an active remote sensing technology for detecting the surrounding environment. Laser scanning is an effective way of capturing surface information of targeted objects. Compared with traditional surveying and mapping technologies, laser scanning mapping provides advantages like high accuracy, fast collection and cost-efficiency. It has been widely used for civil surveying and mapping, such as high-resolution topographic mapping (Kraus and Pfeifer 1998), various infrastructure modeling (Kim and Sohn, 2010; Shapovalov et al., 2010) and forest studies (Rutzinger et al., 2008; Vehmas et al, 2009; Zhang and Sohn, 2010) , etc.

A typical laser scanning system consists of a laser scanner, and some additional onboard equipment for positioning and navigation, such as an onboard Global Position System (GPS) and Inertial Navigation System (INS) system (Wehr and Lohr, 1999). The GPS is used to translate laser system coordinates to the global geographic coordinates. The INS is used to estimate the attitudes of a moving rigid body by measuring the angular velocities. The laser scanner sends out laser pulses to a targeted region and then receives

signal reflected by the surface it encounters. By comparing the sending and reflected signal, the range between laser scanner and the object of interest can be calculated. To get the 3D coordinates of objects, the range value needs to be combined with position and orientation, from GPS and INS respectively. This set of points with coordinates is usually called “point cloud”.

The ranging technologies using a laser can be classified into two groups: phase comparison and time pulse method (Shan and Toth, 2009). In the phase comparison method, the scanning system transmits a continuous wave (CW) of laser radiation. The ranges between the laser scanner and objects are determined by comparing the transmitted and received wave patterns. The laser ranging system using a CW is usually used in terrestrial LiDAR systems aiming to measure relatively short distances. The drawback of the CW system is that the phase difference between reflected and emitted signals is measured by comparing them, but the integer number of wavelengths cannot be determined by the signal difference. It is known as the ambiguity resolution problem, which is similar to the GPS carrier-phase ambiguity problem. In modern systems, the problem is solved by making many changes to the wavelength (Shan and Toth, 2009). Second, ‘time pulse method’ transmits discrete pulses instead of the CW and records time difference between transmitted and reflected pulses to determine the distance for the round trip (Baltsavias, 1999; Wehr and Lohr, 1999). Usually, when the pulse is reflected from the specific targets such as grounds, buildings, and trees, the received pulses whose energy is higher than a predetermined threshold value can be detected. The detected pulse is recorded against the time between the signal emission and its reception in a graph,

which is known as the waveform. Since the speed of light is accurately known, the accuracy of the laser range is dominantly affected by the quality of the time measurement.

In the 1980s, NASA launched the first laser altimetry system, called Airborne Topographic Mapper, while the first commercial airborne LiDAR system was developed by 1995 at Optech Incorporation, Canada. In recent years, with the continuing improvement in accuracy and density of laser measurement, more accurate positioning and navigation system, as well as more advanced solutions for data processing, laser scanning has showed its potential in surveying and mapping (Vosselman and Maas, 2010).

2.1.2 Introduction of Terrestrial Laser Scanning

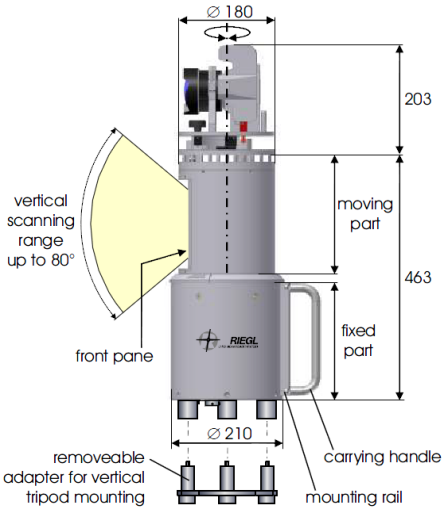
A laser scanner put on the platform of an airplane is called an airborne laser scanner (ALS). Due to rapid, accurate and dense data acquisition, ALS has been widely applied for DEM modeling (Kraus and Pfeifer 1998), forest inventory investigation (Rutzinger et al., 2008; Vehmas et al, 2009; Zhang and Sohn, 2010; Kantola et al, 2013), 3D power line modeling (Kim and Sohn, 2010), 3D city modeling (Shapovalov et al., 2010). There is an increasing demand for fine 3D urban object modelling, which aims to capture full geometric details of objects, such as roof, façade, even the interior structure. However, ALS collects data from the bird's eye view, and cannot completely cover details at the ground level, like building facades. Therefore, the ground-based terrestrial laser scanning (TLS) might be able to provide complementary measurements for ALS, by placing the laser scanner on the top of a tripod or a moving vehicle. Because of its high level of

surveying accuracy, terrestrial laser scanning is feasible for all kinds of detailed 3D documentation, such as digital factory, virtual reality, architecture, civic engineering and culture heritage, plant design and automation systems.

A laser scanner sends out signals toward a specific direction and receives the reflected signal, so only one point is detected at a time. To capture a broad view, the laser scanner changes beam emitting direction to sweep through the whole targeted area; laser beam direction change can be achieved by a system of rotating mirror or rotating the laser source itself (Vosselman and Maas, 2010). However, due to the limited view of static laser scanning, the background region is occluded by the foreground objects. This occlusion prevents the targeted area from being completely scanned, and poses a big challenge for object recognition.

The terrestrial laser scanner used in this research is RIEGL LMS Z-390i, which uses technology of TOF. RIEGL LMS Z-390i is a long range TLS scanner and its range varies between 1.5 m to 400 m. The system of rotating mirror is a two-axis system and allows measurement conducted simultaneously both along vertical and horizontal direction. The field of view covers 360 degrees horizontally and 80 degrees vertically. The minimum horizontal and vertical angular stepwidths are both 0.002 degrees. Table 2.1 presents some technical specifications of the device (Riegl, 2010).

Table 2.1: Technical specifications of RIEGL LMS Z-390i



RIEGL LMS Z-390i	
Measurement principle	**Time of flight**
Range	1.5m – 400m
Acquisition rate	11000 pts/sec
Horizontal FOV	360 degrees
Vertical FOV	80 degrees
Angular stepwidth	0.002 degrees

The scanner is controlled by RiSCAN PRO software, which provides complete data collection services, including sensor configuration, data acquisition, visualization, and manipulation. Direct measurements for each laser return include range, horizontal angles, and vertical angles. The RiSCAN PRO software is able to automatically calculate 3D coordinates from these direct measurements. Finally, data collected by the sensor is transferred to a computer via USB connection.

2.1.3 Terrestrial Laser Scanning Data Classification

With the development of laser scanning and related technology, TLS has been rapidly adopted for urban street data acquisition. Classification is a necessary step for further application, but classifying such complex urban street scenes in an automated manner still remains as a challenging vision task. According to the primitive spatial entity, TLS data classification can be categorized into point-based classification (Triebel, et al, 2006; Munoz et al, 2008), line-based classification (Manandhar and Shibasaki, 2001; Zhao et

al., 2010; Hu and Ye, 2013) and surface-based classification (Belton and Lichti, 2007; Pu and Vosselman, 2009).

As regard the features used for classification, commonly used features include spectral features and geometric features. Spectral features provide information on physical properties of objects. Intensity is typical spectral information; it is dependent on reflectivity and scattering characteristics of object surface (Pfeifer et al., 2007). Imagery from an attached camera also can provide additional spectral features; it usually needs to be registered with the point clouds (Forkuo and King, 2004). However, the laser scanner we used has a problem of outputting intensity, so this research relies purely on geometric information that is derived from 3D coordinates of point clouds.

Geometric feature can be extracted based only on a single spatial entity (e.g., point, line, and surface) without considering its neighborhood. Another type of feature is neighborhood-based feature, which provides contextual information. The neighborhood can be selected by searching neighbors in a pre-defined region (Niemeyer, et al., 2012; Kim and Sohn, 2010), or *k* nearest neighbors (Munoz et al., 2008; Niemeyer et al., 2011, Schmidt et al., 2012) are popular methods. Given the neighborhood, geometric features can be calculated, such as eigenvalue based features (Belton and Lichti, 2006), hough transformation based features (Kim and Sohn, 2010), point density based features (Rutzinger et al, 2008), and features based projected 2D space (Weinmann et al., 2013).

When features extraction is done, classifiers can be built based on these features. There are two primary classification strategies, rule-based classification and machine learning.

2.1.3.1 Rule Based Classification

Rule based methods usually implement classification by converting prior expert knowledge to simple "if this, then that" clause (Forlani et al., 2006; Goulette et al., 2006; Pu and Vosselman, 2009; Lehtomäki et al., 2010; Aijazi et al., 2013). Forlani et al. (2006) applied a set of hierarchically predefined rules to classify segmented laser scanning data into bare terrain, building, vegetation, courtyard, and water from ALS data; these rules were based on geometric and topological properties (e.g., regions exceeding a size of 200000 m^2 were classified as terrain). Goulette et al. (2006) detected ground from vehicle-based TLS data by assuming that ground points correspond to the peak of histogram of vertical coordinates. After removing ground, building and tree were then recognized by detecting peaks in the histogram of horizontal coordinates. Pu and Vosselman (2009) manually defined classification rules based on point segments' characteristics, such as size, position, orientation, and topological relations. In Lehtomäki et al. (2010), vertical pole-like objects were detected by fitting circle and arc models from horizontal slices of point clouds. Candidate circles can be classified as pole only if they fulfil all requirements on length, shape, orientation, etc. Authors claimed that thresholds they used need to be adjusted according to the real data. Aijazi et al. (2013) classified super-voxels into ground and other five non-ground objects using both geometrical models (e.g., roads represent a low flat plane, while the buildings are represented as large vertical blocks) and predefined rules (e.g., barycenters of tree and vegetation are greater than geometrical centers of them).

These rule based methods have many advantages: they are easily designed and implemented; the inference rules can be modified and updated according to real data; they do not require labeled training data. However, classification performance of rule based methods is highly dependent on the choice of features and thresholds; thus rule makers should have sufficient prior knowledge about the target classes. Unfortunately, rule makers often cannot discover all the rules that govern objects, let alone the various conditions behind these rules. In contrast, machine learning is able to learn classification rules automatically from labeled data; they also can be implemented and updated easily.

2.1.3.2 Machine Learning

Machine learning based laser scanning classification has attracted more and more attention over recent years. Supervised classification method is one of the most popular machine learning strategies and has been widely applied for object recognition. Supervised methods learn statistical rules automatically from labeled training data, and then generalize these rules on unseen data (Kotsiantis, 2007). Supervised methods can be categorized into "generative classifiers" and "discriminative classifiers". Generative classifiers model joint distributions of class label and features and provide rigorous framework to combine prior knowledge and observed data. Generative classifiers can freely generate new labeled instances according to these joint distributions. Many generative classifiers have been used for laser scanning data classification, such as Naïve Bayes (Premebida et al., 2009; Posner et al., 2009), Gaussian Mixture Model (Charaniya et al., 2004; Lalonde et al., 2006; Vandapel et al., 2004; Luo and Sohn, 2013), and Bayesian Network (Brunn and Weidner, 1997).

The Naïve Bayes classifier makes the assumption that each attribute of the feature vector is independent, and the likelihood is modeled as the product of class conditional probability of each attribute (Premebida et al., 2009), which is often modeled using Gaussian distribution. However, the class conditional probability is usually very complex, so single Gaussian distribution cannot fit it well. An alternative is Gaussian Mixture Model (GMM), which decomposes a distribution using linear combination of several Gaussian distributions (Charaniya et al., 2004; Lalonde et al., 2006). Parameters in the GMM are usually estimated using the classic Expectation Maximization (EM) algorithm (Dempster et al., 1977). If given sufficient expert knowledge on the classification problem domain, Bayesian Network is a proper choice; it models direct dependencies and local distributions between variables (Brunn and Weidner, 1997).

On the other hand, the "discriminative classifiers" are concerned with finding the boundaries between different classes, and directly model the posterior probability. Discriminative classifiers, such as *k*-Nearest Neighbour (Vehmas et al., 2009; Golovinskiy et al., 2009), Logistic Regression (Vehmas et al., 2009; Saxena et al., 2008), Support Vector Machine (Posner et al., 2007; Nüchter and Hertzberg, 2008; Golovinskiy et al., 2009; Himmelsbach et al., 2009; Brodu and Lague; 2012), Decision tree (Matikainen et al, 2007), Neural Network (Nguyen et al., 2005; Priestnall et al., 2000; Prokhorov, 2009) have been applied for laser scanning data classification.

K-nearest neighbour is a non-parametric method and assigns to a new instance with the majority class of its *k* nearest training samples (Cover and Hart, 1967). Nearest neighbour methods are easy to implement, but they are rather sensitive to the training

data (Vehmas et al., 2009), and choice of the number of neighbors (Golovinskiy et al., 2009). Logistic regression is a basic parametric method for binary classification and uses logistic transformation to make the relationship between the posterior probability and linear combination of features (Hosmer and Lemeshow, 2004; Saxena et al., 2008). More recently, Support Vector Machines (SVM) attracts more attention as an alternative for laser scanning data classification (Posner et al., 2007; Nüchter and Hertzberg, 2008; Golovinskiy et al., 2009; Himmelsbach et al., 2009; Brodu and Lague; 2012).The principle of SVM is maximizing the margin, which is defined as the shortest distance from the separating hyperplane to the closest positive (negative) example (Burges, 1998). However, the linear decision boundary found by the classic linear SVM has risk of misclassification if the dataset is not linearly separable, thus kernel function is often used to find a non-linear separating hyperplane by mapping original features into a new high-dimension space (Wang, 2005). An Artificial Neural Networks (ANN) is a computational model inspired by the mechanism of the human neurons. It is comprised of densely interconnected adaptive simple processing elements (called artificial neurons or nodes), which are capable of performing massively parallel computations for data processing and knowledge representation. Variants of ANN, such as Hopfield Neural Network (HNN) and Recurrent Neural Network (RNN) have shown its potential in classifying laser scanning data (Basheer and Hajmeer, 2000; Prokhorov, 2009).

Recently, more attention has been turned to ensemble learning (Drucker et al, 1994), which increase the accuracy of single classifier by combining results of some weak classifiers (Galar et al., 2012). Commonly used ensemble classifiers can be

categorized into bagging and boosting. In Breiman (1996), the concept of bootstrap aggregation was introduced, and the strategy using bootstrap to generate weak classifiers is called bagging. Random forest is a typical bagging method that constructs a set of decision trees using bootstrap. In addition to resampling, candidate features for splitting at each node are also randomly chosen, which increases independency of trees (Liaw and Wiener, 2002). Random forest has achieved good prediction result in urban scene classification (Chehata et al., 2009), power line corridor recognition (Kim and Sohn, 2010), forest type classification (Kantola et al., 2013) from laser scanning data. Instead of randomly sampling training data and combining classifiers with equal vote as the bagging method, the boosting method uses a weighted sample to focus learning on those samples that misclassified by previous weak classifiers, and finally combines results of weak classifiers using weighted vote (Freund et al., 1999). The adaptive boosting (Adaboost) is a typical boosting model, and has been applied to classify laser scanning data (Lodha et al., 2007).

2.2 Context Based Object Recognition

The machine learning based methods mentioned in section 2.1 are called local classifier because they only use appearance features, without considering relations between objects. Appearance variation, occlusion, various point density with range, all of which cause the problem of feature ambiguity. Relying only on these features with ambiguity, local classifiers have risk of misclassification.

Contextual information, or context for short, has been proved to be able to remove misclassification errors of local classifiers by considering relations of objects. Strat

(1993) defined the context as any and all information that may influence the way a scene and the objects within it are perceived. Therefore, data collected for the same object using different sensors, time of data collection, attributes of local region, and global scene layout of objects are all parts of context. The context can be defined at visual perception level and objective statistical level. Visual perception is the ability to interpret the surrounding environment by processing information that is contained in visible light; illusions (such as the Muller-Lyer illusion) and Stroop phenomenon are typical modalities of visual perceptual context (Toussaint, 1978). Meanwhile, the statistical context is defined under an elegant probabilistic framework (Song, 1999). In this research, we utilized the statistical method to model context.

2.2.1 Object Context

Object context in this research is indicates dependencies or correlations among entities (line) in a scene. With context, a line is perceived associated with its surrounding neighbors rather than independently. Galleguillos and Belongie (2010) categorized the statistical context used for object recognition into three types: semantic (probability), spatial (position) and scale (size).

Semantic context indicates the occurrence probability that an object can be found in a specific scene but not others. Early studies on semantic context mainly focused on manually-made rules, but current research prefers to extract context automatically from labeled training data. The symmetric, nonnegative co-occurrence matrix is a typical form of semantic context. Each entry of the co-occurrence matrix represents the number of times that a given class occurs in a particular relation to another another class.

Rabinovich et al. (2007) used this type of co-occurrence matrix among segment labels to enhance classification performance. Soh and Tsatsoulis (1999) defined the gray-level spatial dependence over pixels using a gray-level co-occurrence matrix (GLCM), where each entry *P(i,j)* of the GLCM corresponds with the number of co-occurrence of the pair of grey level *i* and *j* at a distance of *d*.

Spatial context specifies the likelihood of finding an object at some position. The spatial context can be defined based on absolute position (Shotton et al., 2006; Shotton et al., 2009; Bo et al., 2011; Liu et al., 2011; Zitnick et al., 2013) or relative position (Gould et al., 2013; Zitnick et al., 2013) in a scene. Shotton et al. (2006) encoded the probability of a class occurs at the specific location in the image as the form of a look-up table. Gould et al. (2013) used non-parametric relative location maps over super-pixels as a global feature, which not only allows modeling simple relative location relations (above, beside, or enclosed), but also complex relationships, such as both sky and car are found above road, but car tends to be much closer than sky. Zitnick et al. (2013) incorporated both absolute location prior and relative location prior in their probabilistic model.

Scale context refers to prior information about the most likely sizes at which objects might appear in the scene (Torralba, 2003). Meta-data (e.g. position, orientation, geometric horizon, and map) of cameras is able to generate hypothesis about the scene in which object's configurations are consistent with a global context (Strat and Fischler, 1991). Scale context is the hardest relation to access, since it requires more detailed information about the objects in the scene (Galleguillos and Belongie, 2010).

Actually, the boundaries between different types of context are not strictly defined. Most of publication we reviewed above perhaps used one or two explicit types of context. A critical contribution of this research is exploiting scene layout of object to improve classification; the scene layout can be sematic context or spatial context. While images have scaling problem because object size varies with the focal length, the TLS scanner captures direct 3D coordinate of target objects; thus, the scaling context is of no benefit and was not considered.

2.2.2 Scene Layout Prior

The scene layout corresponds to the relative locations of objects in a scene. An image (or a point cloud) is not a random collection of independent pixels (or points), but follows some rules on spatial arrangement. The spatial arrangement of objects in urban environment is rather clear and strict, e.g. roof is on the top of building facade, and building is behind of tree. With the prior knowledge on scene layout, it is expect to estimate what types of objects could be above or below building, and so on.

Many achievements have been made on applying scene layout for object recognition from images (Winn and Shotton, 2006; Heesch and Petrou, 2010; Jahangiri et al., 2010; Ding et al., 2014). Winn and Shotton (2006) modeled scene layout (above/bellow/left/right) over pixels using asymmetric pairwise potential. In Gould et al. (2008), layout of objects was modeled as relative location probability maps over pixels, which were based on the first-stage classification using appearance-based feature; and the final label prediction was made by combining appearance-based feature and contextual features extracted from relative location probability maps. Heesch and Petrou (2010) was

interested in modeling the probability distribution over labels for a segmented region given labels of its six local neighboring regions: above, bellow, left, right, as well as regions containing and being contained by the current region. Jahangiri et al. (2010) incorporated five different scene layout relations between segmented region pairs in one probabilistic model, including relative vertical and horizontal orientation, containment relation, and the ratio of width and height. In Desai et al. (2011), an image was represented as a collection of overlapping windows at multiple scales, and spatial relation between these windows was considered, such as above, below, overlapping, next-to, near, and far. Label layout filter (LLF) was proposed by Ding et al. (2014) to model the class distribution behavior and visual context appearance of labels over multi-scale segmented regions, such as location distribution of each class in the image, or the relative distance and orientation between two classes. The LLF combines label compatibility, spatial closeness (distance), and feature similarity on all pairs of pixels from the image scene in one potential term in forms of appearance kernel and smoothness kernel.

However, not too many was done on applying scene-layout to classify laser point cloud. Pu and Vosselman (2009) manually defined object's layout based on size, position, orientation, etc., from human knowledge and then apply these predefined rules on classifying TLS data. As it is mentioned above, although such rule based method is easily implemented and achieved satisfying classification result, it cannot cover all the rules that govern object layout, let alone conditions behind these rules. Instead, we used supervised training to learn scene layout rules automatically from labeled training data.

In this research, the scene layout specifies the relative location of lines in both the vertical and horizontal directions. The vertical scene layout was considered an "above-below" relation, such as building is bellow roof but above the pedestrian road. The horizontal scene layout was modeled as a "front-behind" relation, with respect to the distance between lines and laser scanner center, such as tree is in front of building, but behind of vehicle road.

2.3 Probabilistic Graphical Model

There exist two methods to utilize contextual information in a classification, contextual features and contextual classifiers. Contextual features are usually derived from a local or global neighborhood surrounding the interest region that is being analyzed (Haralick et al., 2013). Contextual features could be extracted directly from unlabeled data (Kim and Sohn, 2010; Niemeyer, et al., 2012), or based on an initial classification result that relies only on appearance features (Gould et al., 2008; Jahangiri et al., 2010). Contextual features are finally combined with appearance features to make a final decision using any classifier. Instead of modeling contextual information as features, contextual classifiers incorporate contextual information directly into a probabilistic graphical model.

2.3.1 Probabilistic Graphical Model

Probabilistic graphical model, or graphical model in short, gives a multivariate statistical modeling based on both the graph theory and probability theory (Koller and Friedman, 2009). By considering dependency of variables, the graphical model greatly simplifies the design of a complex probabilistic system, while the probability theory models

dependency of variables using potential functions. Thus, a graphical model refers to a family of probability distributions associated with the graph that can be parameterized by graph factorization. It has been widely applied to many fields, such as image processing, social network analysis, bioinformatics, marketing analysis, etc.

There are two elements in a graph, nodes and edges. The nodes in the graph are random variables, which can be discrete (take one of predefined finite number of values) or continuous (take one of infinite number of values). As classification is a problem to predict states of a discrete variable, all graphical models reviewed in this thesis are discrete graphical models. Edge represents the statistical dependencies between random variables. These dependencies could be directed or undirected, corresponding respectively with directed graph and undirected graph. Directed graph is consists of many subsets of nodes based on "parent-child" relations, which can be modeled using conditional probabilities. Typical directed graph models includes Bayesian Network, Hidden Markov Model (HMM), etc. Based on the idea of causality, directed graphical models have a simple causal interpretation (Pearl, 2000); however, if the some variables related with causality are not observed, an analysis of directed graphical models involving only the observed variables can be highly misleading (Andersson, et al., 1999). The undirected graphical model is targeted to model the problem given little causal structure. This thesis only focuses on discussion of undirected graphical model.

2.3.2 Markov Random Field

Undirected graphical model is also known as Markov Random Field (Clifford, 1990). Let *G(V, E)* be an undirected graph, where *V* is the set of nodes, which corresponds to

random variable Y, and E is the set of edges. The existence of an edge $e=(v_i, v_j)$ indicates a dependency relation between two random variable v_i, v_j, and the absence of an edge between two nodes implies that they are conditionally independent given all other random variables in the graph (Wallach, 2004). Clique is the basic subset of the undirected graph graph and nodes inside a clique are completely connected. Let C denote a collection of cliques of the graph, let ψ_c denote a nonnegative potential function for a given clique c. According to the Hammersley-Clifford theorem (Hammersley and Clifford, 1971), if a random field Y has the local Markov property, $p(Y)$ can be written as a Gibbs distribution

$$P(Y) = \frac{\prod_{c \in C} \psi_c(y_c)}{Z(Y)} \tag{2.1}$$

where Z is a normalization term, which is obtained by summing the product of the potential function over the collection of cliques C.

There exist many publication on apply MRF for laser scanning data classification (Anguelov et al., 2005; Triebel et al., 2006; Munoz et al., 2008; Zhang and Sohn, 2010; Häselich et al., 2011). Wellington et al. (2005) classified vehicle-based lasser scanning data into ground and obstacle using MRF with a prior on smooth ground and class continuity. Zhang and Sohn (2010) formulated detecting single tree from ALS data as a problem of energy minimization using MRF. In Häselich et al. (2011), 3D laser data was segmented into a 2D grid first, and then applied MRF to enforce a local smoothness

constraint. The Potts model is usually used as the interaction potential to encourage adjacent points to have the same class label (Häselich et al., 2011). However, the Potts model is set constantly and restricted only to the class labels. Therefore, a MRF based method is probably to produce an over-smoothness classification result. Schindler (2012) did detailed experimental comparison on classifiers with and without smoothness assumption and found that smoothness prior improved the classification accuracy up to 33% in presented data, but also confirmed that all smoothness based methods had over-smoothness effect.

A variant of MRF, Associative Markov network (AMN) has been recently studied for classifying laser scanning data. In AMN model, the pairwise potential is set as a contrast-sensitive Potts model, value of homogeneous relation is not 1 but can be a function of edge features, and pairwise potential between different classes is still set to 0 (Anguelov et al., 2005; Triebel et al., 2006; Munoz et al., 2008). Because interaction potential is associated with feature vector, it can reduce the risk of over-smoothness more or less. But the same as Potts model based MRF model, it still fails to capture the relations that neighboring nodes have different classes. For example, they can model the relations as "the building is likely to be neighbor with the roof", but is unable to express interactions between different objects, such as "the building is likely to be neighbor with the roof but lower than roof". By capturing dependencies between different labels and features simultaneously, the restriction of MRF is overcome by Conditional Random Field (CRF), which allows interaction potential terms conditioned on class label as well as global observations data (Kumar and Hebert, 2006).

2.3.3 Conditional Random Field

The Conditional Random Filed (CRF) is an undirected graphical model that was firstly proposed by Lafferty et al. (2001) to labelling sequence data. CRF model has shown its confidence in text mining (Lafferty et al., 2001; Pang and Lee, 2008), image processing (Kumar and Hebert, 2003; He, et al., 2004), and biomedical science (Settles, 2004).

As a discriminative model, instead of modeling the joint probability, CRF directly models the conditional distribution over class label Y give observation X. Lafferty et al. (2001) defined the conditional probability of a CRF model as a normalized product of potential functions,

$$P(Y \mid X) = \frac{\prod_{c \in C} \psi_c(y_c, X)}{Z(X)} \tag{2.2}$$

where ψ_c is potential function, which defines the compatibility among variables for a given clique c. The larger the potential value is, the more confidence the configuration gets. Compared with the potential function of MRF, CRF designs potential function as a data-dependent function (Kumar and Hebert, 2003; He, et al., 2004). If the maximal clique number is two, it is called pairwise potential; while if more than two, it is called high-order potential, examples of which can be found in (Munoz et al., 2009) and (Wegner et al, 2013). In this research, we only consider pairwise potential. There are two types of potential functions in a pairwise CRF model, node potentials and pairwise potential.

Parameters of potential functions of a CRF are usually unknown, and can be learned from a training data. Parameter learning of a CRF refers to the procedure of recovering model parameters that best fit the training data. Various methods have been used for training CRF, including maximum log-likelohood (Vishwanathan et al., 2006), maximum pseudo likelihood (Liao, 2006), and Logitboost Based Training (Vail et al., 2007). The inference of a CRF refers to computing the marginal distributions of each hidden variables or Maximum A-Posterior given parameters and observations. Common inference methods for CRF include Loopy Belief Propagation (Frey et al., 1998), and Markov Chain Monte Carlo (Liao, 2006).

Recently, many works on classifying laser scanning point using CRFs have been reported. Munoz et al. (2008) classified mobile laser data into five classes, wire, pole/trunk, scatter, ground and facade using feature-dependant pairwise potential, which is named Directional Associative Markov Networks (Directional AMN). Munoz et al. (2009) used a high-order CRF model, which further improved classification performance by the previous Directional AMN model. Shapovalov et al. (2010) classified airborne laser scanning data using non-associative Markov Network, in which the interaction term is modeled using Naïve Bayes classifier and so it is able to capture all types of relations between classes. Experiment results of (Niemeyer et al., 2011) validated the advantage of CRF over MRF on classifying Airborne LiDAR data; the flat-roofed part that MRF failed to detect was exactly extracted by the proposed CRF nearly without error.

However, these aforementioned CRF models only consider local context between closely neighboring points or segments, which could still mislead incorrect smooth label

configuration. Moreover, close-range neighbor searching perhaps fail when the data is partially observed. Thus, considering both regional and global context can be a solution to overcome this limitation. He et al. (2004) firstly proposed a multi-scale CRF model for image recognition, which used a multilayer perceptual fashion, modeling local, regional and global label compatibilities. Lim and Suter (2008) and Lim and Suter (2009) proposed multi-scale Conditional Random Fields to classify 3D outdoor terrestrial laser scanning, and the multi-scale contexts include connections between points within each super-voxel, and connections between super-voxels.

2.4 Chapter Summary

Object recognition from massive TLS data still faces many challenges, such as appearance variations, occlusions, various point density with range, which cause the problem of feature Ambiguity. Relying only on these features with ambiguity, local classifiers have risk of misclassification. Object context has shown its potential to improve the classification performance by considering the label interactions of neighboring objects, such as local context and global context. Scene layout is a type of global context and provides information on spatial arrangement of object in the space; however, automatically learning the scene layout of objects from terrestrial laser scanning data is still problem. Moreover, there is a need to combing local and global context in a probabilistic graphical model.

Chapter 3
Line-based TLS Data Classification

This chapter presents a line-based TLS data classification method. Existing methods on TLS data classification using different spatial entities and the advantage of line-based method will be discussed at the beginning of this chapter. Afterwards, the workflow of line-based TLS data classification is presented in section 3.1 and 3.2. The entire TLS was firstly split into a set of vertical scan profiles and then line segments were extracted in each scan profile. Two types of features were extracted for line-based classification: local features and contextual features. In order to validate the effectiveness of line-based TLS classification, both generative and discriminative classifiers were tested. Ten classifiers were designed, including Naïve Bayes (NB), Multivariate Gaussian (MG), Gaussian Mixture Model(GMM), K-Nearest Neighbor (KNN), Logistic Regression (LR), Support Vector Machine (SVM), Artificial Neural Network (ANN), Decision Tree (DT), and two Decision Tree based ensembles, Random Forest (RF) and Adaptive Boosting (AdaBoost). These classifier were then evaluated using TLS data collected in the residence regions of York University, Toronto. The performance of these classifiers was then evaluated both qualitatively and quantitatively. Quantitative measurements include confusion matrix, accuracy, precision, recall and F1-score. The experimental results show that all classifiers were efficient for line-based TLS data classification and achieved satisfying accuracy. Limitations of these classifiers are also discussed in the end.

3.1 Line-based Classification

According to spatial entity to label, classification methods for TLS data can be categorised into three types: point-based (Triebel, et al, 2006; Munoz et al, 2008), line-based (Manandhar and Shibasaki, 2001; Zhao et al., 2010) and surface-based (Belton and Lichti, 2006; Pu and Vosselman, 2009). The point-based method directly labels individual laser points; meanwhile both line-based and surface-based methods partition the point cloud into homogeneous segments firstly, such as line, plane, and cylinder, and then label these segments. Since a TLS scanner collects data by rapidly generating 2D profile scans of objects, the appearance of scanned objects can be characterized using lines in each scan profile. Therefore it is rather straightforward to extract lines and construct line adjacent graph in each scan profile. The line-based classification method starts with extracting line segments in each scan profile and subsequently labels these line segments based on linear features vector.

3.1.1 Motivation of Line-based Classification

Compared with single laser point, the line segment is higher-level geometric primitive and carries more semantic information; thus point-based method is not considered in this thesis. Compared with surface-based method, the line-based method is computational efficient for massive TLS data classification. Although surface-based method reduces computational cost by reducing the number of spatial entities to be labeled, segmenting large amounts of point clouds into surfaces still requires constructing adjacent relationship over points. Therefore, searching and storing neighborhood information needs a large amount of memory and produces a high computational load. These

techniques are not efficient for massive TLS data processing, let alone real time processing. In contrast, line-based method utilises the profiling nature of the laser scanning data. This advantage in computational efficiency has been approved by (Jiang and Bunke, 1994), in which line segments were extracted for range data processing. Moreover, most of objects can be well characterized using lines in each vertical scan profile. Lastly, because of the high point density, lines can be easily extracted from terrestrial laser scanning data.

Many efforts have been made on line segment extraction from laser scanning data classification. Axelsson (1999) divided ALS scan profile into line segments based on second derivatives analysis, and classified them using knowledge-based method. In Manandhar and Shibasaki (2001), objects with smooth surface, such as building and ground, were detected from TLS data by extracting horizontal and vertical line segments using range analysis. Hebel and Stilla (2008) detected building facade and roof from ALS data by extracting straight line segments using Random sample consensus (RANSAC) algorithm. Zhao et al. (2010) collected data using vehicle based single-row laser scanner and extracted line segments within each scan profile to characterize building and roads. Hu and Ye (2013) used the Douglas-Peucker algorithm to divide ALS scan profile into line segments. Instead of using hardware-generated scan profiles, Sithole and Vosselman (2003) manually defined two orthogonal scan profiles by slicing the ALS point cloud along x and y. Following this, points in each profile were split into line segments based on connectivity and continuity analyses.

All publications mentioned above extracted line segments from vertical TLS scan profiles (vertically slice the TLS data), but horizontal scan profiles (horizontally slice the TLS data) were also considered. Horizontal scan profiles are cross-section made by TLS data and pre-defined horizontal planes. In horizontal scan profiles, pole-like objects, such as truck and lamp post, is close to circular, and can be detected by fitting circle and arcs (Forsman, 2001; Aschoff and Spiecker, 2004; Bienert et al., 2007;). Building facade can be detected by fitting line segment (Lehtomäki et al., 2010).

In this study, we extracted line segments using range analysis referring to Manandhar and Shibasaki (2001). Firstly, the TLS data was split into a set of vertical scan profiles. Next, each scan profile was partitioned into line segments using range analysis. The Douglas-Peucker algorithm (Hershberger and Snoeyink, 1992) was then applied as a post-processing. The classification was finally implemented by labeling these line segments.

3.1.2 Scan Profile Generation

Prior to line segment extraction, the TLS data was firstly split into a set of vertical scan profiles. Each scan profile was considered a stream of points. The scanning TLS data is assumed to be sequentially observed in a discrete-time fashion, which is denoted by

$$\begin{bmatrix} SP_1: & P_{1,1}, P_{1,2}, P_{1,3,}... \\ SP_2: & P_{2,1}, P_{2,2}, P_{2,3,}... \\ & ..... \\ SP_n: & P_{1,1}, P_{1,2}, P_{1,3,}... \end{bmatrix} \tag{3.1}$$

The SP_n in the Equation 3.1 denotes the n-th scan profile, and $P_{i,j}$ denotes the j-th observation in the i-th scan profile. To generate vertical scan profile, the tripod where laser scanner is put above should be horizontally adjusted, and laser scanner body also need to be vertically adjusted; otherwise, this method does not guarantee vertical scan profiles. The width of each scan profile is set as the scanning angle precision, here 0.05 degree (refers to the horizontal alignment). Figure 3.1(a) shows an example of one vertical scan profile. For further processing, the points were then projected into XY-Z 2D space. The coordinate of XY dimension is the square root of X square and Y square.

3.1.3 Line Segment Extraction

Once vertical scan profiles are ready, line segments can be then extracted from each scan profile. Referring to the method suggested by Manandhar and Shibasaki (2001), line segments were extracted based on range analysis; range is the Euclidian distance between individual laser point and the laser scanner. Since this range analysis method is based on acquisition order of laser scanning, points within each scan profile were firstly sorted according to zenith angle.

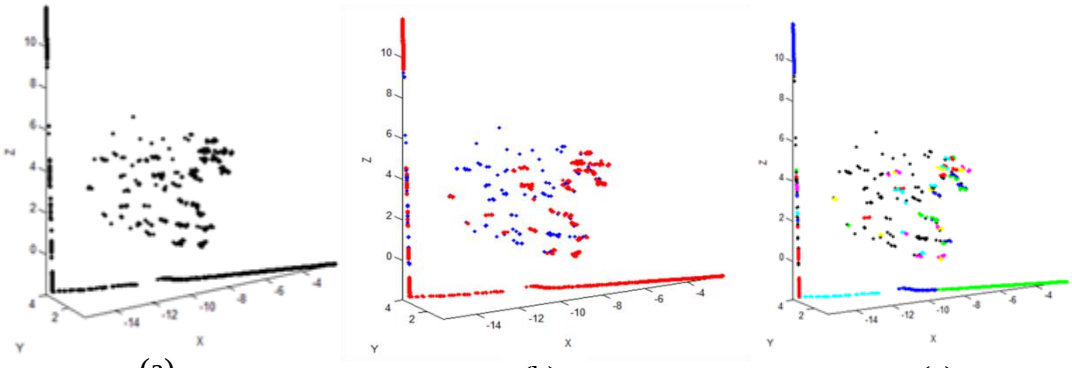
Most urban objects have very visible shapes that can be well characterized with line segments. It is observed that structured objects, like planar (building, facade, road) and cylinder (lamp post) objects, typically have continuous and smooth appearances. Therefore, neighboring points reflected from them have approximate range values. On the contrary, points located at the edge of an object have large range difference from previous and following observed points. In this research, points have large range differences from neighboring points were defined as “scattered points” and points having

small range differences with neighboring points were defined as "smooth points". Given point P_i, and its previous and following observation P_{i-1} and P_{i+1}, the range difference of point P_i is calculated as

$$RangeDifference(P_i) = \frac{|R_{i+1} - R_i| + |R_{i-1} - R_i|}{2} \quad (3.2)$$

where R_i, R_{i-1}, R_{i+1} are the range of P_i, P_{i-1}, P_{i+1} respectively. Points with range difference greater than 0.5 meters (empirical threshold) were considered scattered points, while other points were considered smooth points. Figure 3.1 (b) shows an example of scattered points (blue) and smooth points (red). Most points from structured objects, such as building, ground, etc., are smooth points, and only the edge points are scattered points. Conversely, scattered points appear more frequently in unstructured objects, such as tree, as laser pulse could penetrates them due to the "hole" inside of such objects.

Sequentially connected smooth points were then grouped as a single point cluster. Although points in each cluster do not make a straight line, they do show strong linear characteristic so that they are called *line segment*, or *line* in short in this thesis. Figure 3.1(c) shows the result of line extraction using range analysis, and line were rendered using different color.



(a) (b) (c)

Figure 3.1: Examples of line extraction. (a) Laser point in scan profile, (b) Range difference analysis result, Red: smooth points; Blue: scattered points, (c) Line segment extraction result.

However, under-segmentation problem was found where some lines contain points from multiple objects. For example, if there is no object in front of the building, the boundary between building and ground is rather smooth. As a result, the points observed from the building and points from ground were grouped into one line (see the blue line in Figure 3.2(a)). In order to fix this issue, the Douglas–Peucker algorithm was then applied as post-processing. The line that passes through the endpoints of each line segments was termed "baseline", and the distance between each member point and the baseline was calculated. If the maximum distance was greater than a certain selected threshold (0.1 meter here), the line was subdivided into two lines at the maximum distance point. The procedure was recursively implemented until no line met the subdivision requirement. It is observed that the long blue line in Figure 3.2(a) was divided into one blue line and one green line, which is presented in Figure 3.2(b).

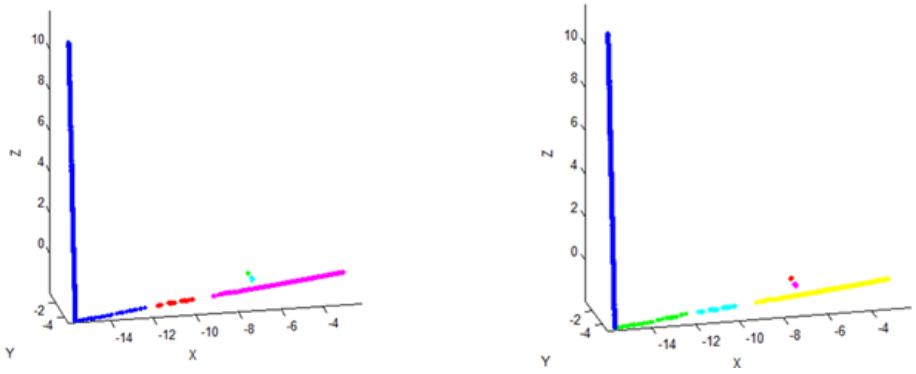


Figure 3.2: Post-processing using the Douglas–Peucker algorithm. (a) The blue line captures points both from building and ground, (b) After line segment subdivision, building points and ground points are separated.

3.2 Linear Feature Extraction

Feature provides discriminative information between classes. To classify the lines, two types of features were extracted: local and contextual. Both circle-based and vertical column-based neighborhoods were used to compute contextual features.

3.2.1 Local Features

Local features characterize the local appearance of a line segment. They were extracted based only on a single line. The elevation (z) is expected to efficiently separate ground, building, and other low-rise objects on the ground; thus three local features were extracted based elevation: 1) maximum height (*z*); 2) minimum height (*z*); 3) averaged height (*z*). To extract linear characteristics, all member points of a line were fit into one straight line using the least square line fitting method. The following additional four

features were extracted based on the fitted line: 4) length (maximum extension in the major direction); 5) mean absolute residual; 6) standard deviation of residual; and 7) orientation (angle between the fitted line and z axis). Length can be used to separate large objects (ground, building, etc.) and small objects (tree leafs, etc.); the residual measures the roughness of each line; the orientation is expected to separate horizontal objects and vertical objects.

3.2.2 Contextual Features

Contextual features can provide the grouping characteristics of a line and its surrounding neighbours. In this study, two types of neighboring systems were used to extract context features: circle-based (Figure 3.3(a)) and vertical column-based (Figure 3.3 (b)). Figure 3.3 (a) illustrates an example of circle-based neighborhood system, which is a circle with 1m radius at the centre of a line centroid (black dot). Lines whose centroids fall inside the circle (both red and pink dots in Figure 3.3 (a)) are considered as neighbours of the current line of interest. Figure 3.3 (b) presents the vertical column-based neighboring system. For this method, a scan profile was quantized into a set of non-overlapping vertical columns (rectangle area between dotted blue lines) with 0.5m in width. Subsequently, neighboring lines were searched within the vertical column (blue filled area) into which the current line falls.

Once two neighboring systems were generated for a line segment, seven contextual features were computed. These include: 1) maximum *z*; 2) sum of line length. Points belong to the line and its surrounding neighbours were fitted into one straight line and the following three features are extracted from the fitted line: 3) mean residual; 4)

standard deviation; 5) orientation (angle between the fitted line and z axis). The other two feature are: 6) point density (point number in the line group) and 7) line density (line number in the line group). Therefore, two types of neighbourhood systems produced a total of fourteen contextual features.

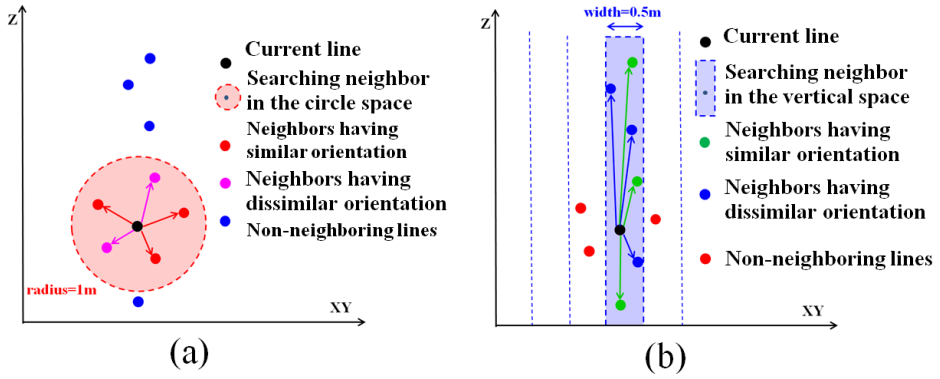


Figure 3.3: Neighborhood selection for context feature. (a) circle-based neighborhood, (b) vertical column-based neighborhood.

3.2.3 Feature Selection

A total of twenty-one features (seven local features and fourteen contextual features) were extracted as described in the previous section. More features bring more discriminative information for classification. However, the high-dimensional features are usually highly correlated and bring computational complexity problem. To avoid such as issue, principle component analysis (PCA) was applied to reduce the feature dimensionality. Adopting the cumulative energy (90%) criteria proposed by Krzanowski (2000), we finally chose the eight most significant principle components for classification. It is noted that in order to equally balance the impact of features, original

features were normalized using z-score before applying PCA algorithm, which causes distribution to have a mean of zero and standard deviation of one.

3.3 Generative Classifiers

The line-based object recognition from TLS data was modeled as supervised classification problem. Given a set of labeled training data and observed feature vectors, the supervised classification method is able to induce a statistical model that can label unseen data. Supervised classifiers can be categorized into generative classifiers and discriminative classifiers according to the way they model posterior probability. Generative classifiers estimate the underlying generalized joint probability distribution over the class label y and feature vector x; the posterior probability $p(y|x)$ is then calculated using Bayes rule, and the class label is finally determined by maximizing the posterior. On the other hand, discriminative classifier directly models the posterior probability. In other words, generative classifiers aim to model which area of the feature space is covered by each class, and discriminative classifiers aim to find a good decision boundary between classes. Detailed comparison of the two methods can be found in Jordan (2002), which used Naïve Bayes classifier and linear logistic regression as examples. In this study, we used both generative (section 3.3) and discriminative (section 3.4) algorithms to test the effectiveness of line-based classification.

Joint distribution modeling is the most significant aspect of generative classifiers. Generative classifiers factorize the joint distribution in the form of product of likelihood $p(x|y)$ and prior $p(y)$. Likelihood $p(x|y)$ models the distribution of feature vector given the class label, which is also called class conditional probability. The prior provides

information about how likely a specified class is expected to be seen before it is actually observed. Bayes' theorem provides an elegant probabilistic framework to model the posterior probability with the concepts of likelihood and prior. To classify a new instance x_i, generative classifier estimates probability of the new instance belonging to each class as follows:

$$P(y_i \mid x_i) = \frac{P(x_i \mid y_i)P(y_i)}{P(x_i)} \tag{3.3}$$

where $P(x_i)$ is the probability of observation data, regardless of its class label. $P(x_i)$ is estimated by marginalizing the joint distribution over all classes.

$$P(x_i) = \sum_{y_i} P(x_i \mid y_i)P(y_i) \tag{3.4}$$

Since the denominator is a scale-factor to normalize the density, it is always dropped in practice and hence the posterior is proportional with the joint probability as follows:

$$P(y_i \mid x_i) \propto P(x_i \mid y_i)P(y_i) \tag{3.5}$$

Prior can be regarded as an uncertainty variable, so it can be modeled using a distribution, such as Gaussian distribution (Bishop, 2006; Lawrence, 1998). However, the distribution of prior was more often selected on the basis of mathematical convenience rather than as a reflection of any prior beliefs (Bishop, 2006). Since reliable prior probabilities are not easily available in practice, Bayesian classifier usually makes an assumption that prior probabilities are equal (Kumar et al., 2011). Thus, in this thesis, priors were assumed equal and posterior expression was identical to the following expression:

$$P(y_i \mid x_i) \propto P(x_i \mid y_i) \tag{3.6}$$

This indicates that final decision was made solely based on maximizing likelihood. Actually, the three generative classifiers used in this thesis differ mainly in likelihood modeling. Generative classifiers individually model likelihood for each class, and the accuracy of likelihood estimation increases with the amount of training data available. Since the prior of each class is equally assumed and the likelihood estimation of each class is not affected by imbalanced training, we did not consider balanced training in learning generative classifiers.

3.3.1 Naïve Bayes（NB）

The Naïve Bayes (NB) algorithm is one of the most popular Bayesian classifiers. NB makes the conditional independence assumption that attributes of feature vectors are

independent of each other given the class label. Thus, likelihood $P(x_i|y_i)$ equals to the production of conditional probabilities of each attribute x_i^k given the class y_i according to

$$P(x_i \mid y_i) = \prod_{k=1}^{K} P(x_i^k \mid y_i) \tag{3.7}$$

Due to this independence assumption, correlations of attributes are ignored and so the computational complexity of NB is considered very low when compared with other supervised classification algorithm. Despite the simplifying assumption, NB classifier still gives high classification accuracy in practice. Moreover, NB is not sensitive to irrelevant feature.

Gaussian distribution is often used to model likelihood. A single-variable Gaussian distribution has two parameters, mean and standard deviation. The Gaussian distribution of single real-valued variable x is defined as a quadratic function of the variable x, mean μ and standard deviation σ as follows:

$$N(x \mid \mu, \sigma) = \frac{1}{\sigma\sqrt{2\pi}} \exp[-\frac{1}{2\sigma^2}(x - \mu)^2] \tag{3.8}$$

Here, the maximum likelihood estimation (MLE) was used to learn the parameters of Gaussian distribution. Given a set of data $\{x_1, \ldots, x_n\}$, the likelihood function is defined as the joint density of all samples as

$$L(\mu,\sigma \mid x_1,...,x_n) = f(x_1,...,x_n;\mu,\sigma) = \prod_{i=1}^{n} \frac{1}{\sigma\sqrt{2\pi}} \exp(-\frac{1}{2\sigma^2}(x_i - \mu)^2) \qquad (3.9)$$

It is quite difficult to directly maximize this likelihood function; as an easier alternative, the log-likelihood function is often used. As logarithm is a monotonic function, maximizing log-likelihood will also maximize likelihood. The log-likelihood is written as

$$\ln L(\mu,\sigma \mid x_1,...,x_n) = -\frac{n}{2}\ln(2\pi) - \frac{n}{2}\ln(\sigma^2) - \frac{1}{2\sigma^2}\sum_{i=1}^{n}(x_i - \mu) \qquad (3.10)$$

MLE estimates parameters by differentiating log-likelihood with respect to each parameter. Partial derivative of μ and σ are such that

$$\frac{\partial \ln L(\mu,\sigma \mid x_1,...,x_n)}{\partial \mu} = -\frac{n\sum_{i=1}^{n}(x_i - \mu)}{2\sigma^2} = 0 \qquad (3.11)$$

$$\frac{\partial \ln L(\mu,\sigma \mid x_1,...,x_n)}{\partial \sigma^2} = -\frac{n}{2\sigma^2} + \frac{1}{(\sigma^2)^2}\sum_{i=1}^{n}(x_i - \mu)^2 = 0 \qquad (3.12)$$

The final estimation of μ and σ are the following formulation:

$$\mu = \frac{\sum_{i=1}^{n} x_i}{n} \tag{3.13}$$

$$\sigma^2 = \frac{\sum_{i=1}^{n} (x_i - \mu)}{n} \tag{3.14}$$

3.3.2 Multivariate Gaussian (MG)

The independence assumption of NB has advantage in computation reduction, but it difficult to judge attributes of feature are dependent or not in practice. As such, the performance of NB usually is not satisfying in the domains with correlated features. In this case, multivariate Gaussian distribution is able to capture the correlation of features, and the corresponding classifier is termed as multivariate Gaussian classifier (MG). A *K-variate* Gaussian distribution is parameterized by mean vector μ and a variance Σ as follows:

$$P(x_i \mid y_i) = N(x_i \mid \mu, \Sigma) = \frac{1}{\sqrt{(2\pi)^k \mid \Sigma \mid}} \exp[-\frac{1}{2\sigma^2}(x_i - \mu)^T \mid \Sigma \mid^{-1} (x_i - \mu)] \tag{3.15}$$

MLE was also used to learn parameters in multivariate Gaussian distribution. Given a set of data $\{x_1, \ldots, x_n\}$, log-likelihood of the data is written as

$$\ln L(\mu, \sum \mid x_1, \ldots, x_n) = -\frac{n}{2} \ln \mid \sum \mid - \frac{nk}{2} \ln(2\pi) - \frac{1}{2} \sum_{i=1}^{n} (x_i - \mu)^{\mathrm{T}} \sum^{-1} (x_i - \mu) \tag{3.16}$$

Taking partial derivative with respect to μ and Σ, and then setting these parameters to zero, solution of μ is the same as Equation 3.13, and Σ is given as

$$\Sigma = \frac{1}{n}\sum_{i=1}^{n}(x_i - \mu)(x_i - \mu)^T \tag{3.17}$$

3.3.3 Gaussian Mixture Model (GMM)

The two methods discussed earlier do not fit well in distributions that do not follow normal distribution or have highly complex distributions. In said situations, the Gaussian Mixture Model (GMM) is an alternative. GMM is a linear combination of several Gaussian distributions. It provides an attractive semi-parametric framework to approximate unknown distributions based on available data (McLachlan and Peel, 2000). GMM has been widely applied in many areas, such as speaker identification (Reynolds, 1995), image segmentation (Zhang et al., 2001), and image texture detection (Permuter, et al., 2003). The mixture Gaussian approximation is a farily appropriate method for modeling complex likelihood. The GMM density function is given by

$$P(x_i \mid y_i) = \sum_{k=1}^{K}\alpha_k N(x_i; \mu_k, \Sigma_k) \tag{3.18}$$

$$\alpha_k = p(x_i \in N_k), \quad 0 < \alpha_k < 1, \quad \sum_{i=1}^{K}\alpha_k = 1 \tag{3.19}$$

where $N(x\ ;\ \mu_k,\ \Sigma_k)$ is the *k-th* Gaussian mixture component, α_k is the prior that x_i is produced by the component N_k, and K indicates the total number of mixture components. The value of α_k ranges from 0 to 1, and sum of $\{\alpha_{1,\ldots}\ \alpha_K\}$ equals 1. The parameters $\theta=\{\alpha_1,\ldots,\ \alpha_K,\ \mu_1,\ldots,\ \mu_K,\ \Sigma_1,\ldots,\ \Sigma_K\}$ define the Gaussian mixture probability density function.

In this thesis, the well-known Expectation Maximization (EM) algorithm was used for parameter estimation of GMM. Given n independent samples generated from a GMM distribution, the log-likelihood function was written as

$$l(\theta) = \sum_{i=1}^{n} \log\left(\sum_{i=1}^{k} \alpha_k N(x_i \mid \mu_k, \Sigma_k)\right) \tag{3.20}$$

Firstly, the partial derivative of the log-likelihood function with respect to the mean μ_k was taken and set to zero.

$$\frac{\partial l(\theta)}{\partial \mu_k} = \sum_{s=1}^{n} \frac{\alpha_k N(x_i; \mu_k, \Sigma_k)}{\sum_{i=1}^{k} \alpha_k N(x_i; \mu_k, \Sigma_k)} \Sigma_k^{-1}(x_i - \mu_k) = 0 \tag{3.21}$$

However, the log-likelihood function is not a linear function with respect to parameters, so this partial derivative expression is difficult to optimize and cannot achieve a closed form solution. This problem results from incomplete data that we do not know which Gaussian component is responsible for the generation of each training

sample. EM algorithm was introduced by Dempster (1977) and yields a closed form solution to the estimation issue with incomplete data (McLachlan and Peel, 2000) by artificially completing the data with additional pseudo data. EM is an iterative algorithm with two steps in each iteration, the Expectation-step (or E-step) and the Maximization-step (or M-step). Starting with an initial model by *K-means* clustering or any other initialization method, the EM algorithm alternates between the E-step and M-step.

The E-step computes the expected value of the complete log-likelihood, conditioned on the training data and the current parameter estimate θ_t. The partial derivative with respect to mean μ_k can be written as

$$\frac{\partial l(\theta)}{\partial \mu_k} = \sum_{s=1}^{n} p(k \mid x_i, \theta^t) \Sigma_k^{-1} (x_i - \mu_k) = 0 \tag{3.22}$$

$$p(k \mid x_i, \theta) = \frac{\alpha_k N(x_i; \mu_k, \Sigma_k)}{\sum_{i=1}^{k} \alpha_k N(x_i; \mu_k, \Sigma_k)} \tag{3.23}$$

where $p(k|x_i,\Theta^t)$ is the posterior probability that data x_i belongs to the *k-th* Gaussian component given the current estimate, which is also called membership probability. The membership probability provides knowledge on which sample are generated from which Gaussian mixture component.

The "M-step" improves the current model by maximizing expected log-likelihood found on the E-step. Maximization is implemented and parameters are updated as follows:

$$\mu_k^{t+1} = \frac{\sum_{s=1}^{n} p(k \mid x_i, \theta^t) x_i}{\sum_{s=1}^{n} p(k \mid x_i, \theta^t)} \tag{3.24}$$

$$\Sigma_k^{t+1} = \frac{\sum_{i=1}^{n} p(k \mid x_i, \theta^t)(x_i - \mu_k^t)(x_i - \mu_k^t)^T}{\sum_{i=1}^{n} p(k \mid x_i, \theta^t)} \tag{3.25}$$

$$\alpha_k^{t+1} = \sum_{i=1}^{n} p(k \mid x_i, \theta^t) \tag{3.26}$$

The EM algorithm has been approved to be stable, and converge to an ML estimate (Zhang et al., 2001). At each iteration, the parameter update made an increase in the likelihood function until a local maximum is found. Another issue of using GMM is that mixture component number *K* is unknown in most conditions. (Figueiredo and Jain, 2002) summarized existing methods on finding the optimal mixture component number. In this research, five-fold cross validation was used to choose the optimal *K*.

3.4 Discriminative Classifiers

Instead of modeling joint distribution, discriminative classifiers directly optimize the posterior probability *p(y|x)*. Discriminative classifiers lack the elegant probabilistic concepts of priors, structure, and uncertainty of generative classifiers; instead, penalty functions, regularization, kernels etc., are often used (Jebara, T., 2012). This section

introduces adopted discriminative classifiers used for line-based classification, including k-nearest neighbour (KNN), logistic regression (LR), support vector machine (SVM), decision tree (DT), artificial neural network (ANN), random forest (RF) and adaptive boosting (AdaBoost).

Training data is often imbalanced that sample size of each class is not equal. Many recent publications have pointed out that the decision boundary of a discriminative classifier skews towards the minority class for imbalanced training data, which results in high misclassification error of minority classes (Chawla et al, 2004; Imam et al, 2006). To avoid this risk, we used balanced training data to train these discriminative classifiers; however we did not compare the performance of balanced training and imbalanced training.

3.4.1 K-Nearest Neighbors (KNN)

KNN assumes that an instance tends to have a similar label with training samples that are similar to it. It is a typical non-parametric classification model. Let $T = \{t_1, t_2, \ldots, t_n\}$ denote the set of labelled training samples, and $S = \{s_1, s_2, \ldots, s_k\}$ be the set of k nearest training samples to a test instance t according to some similarity measurements. In such case, the KNN assigns the sample t to the class that occurs most frequently among the k nearest training samples.

Commonly used similarity measurement methods include Euclidean distance, Mahalanobis distance and Minkowski distance. Euclidean distance is suitable for continuous variables, while the other methods are better for categorical variables. Since all features extracted for line classification are continuous, Euclidean distance was used

for distance measurement. Assuming that each feature vectore x is an M dimensional vector, Euclidean distance between test instance t and *i-th* training samples was calculated as

$$d = \sqrt{\sum_{m=1}^{M} (x_t^m - x_i^m)^2} \tag{3.27}$$

The only parameter that can adjust the complexity of KNN is k, the number of nearest neighbors. KNN is sensitive to value of k (Golovinskiy et al., 2009). The larger k is, the smoother the classification boundary and the less the misclassification risk; however, a large k often brings problem in computational efficiency. Moreover, the value of k is dependent on the training sample, and changing the position of a few training samples could significantly change the decision boundary. Here the five-fold cross validation method was used to select the optimal k. The Figure 3.4 shows the averaged test accuracy when the number of neighbors iterates from one to twenty. As the value of k increases, averaged test accuracy also increases, but the amount of improvement becomes less and less after ten. Therefore, we chose the 10NN model.

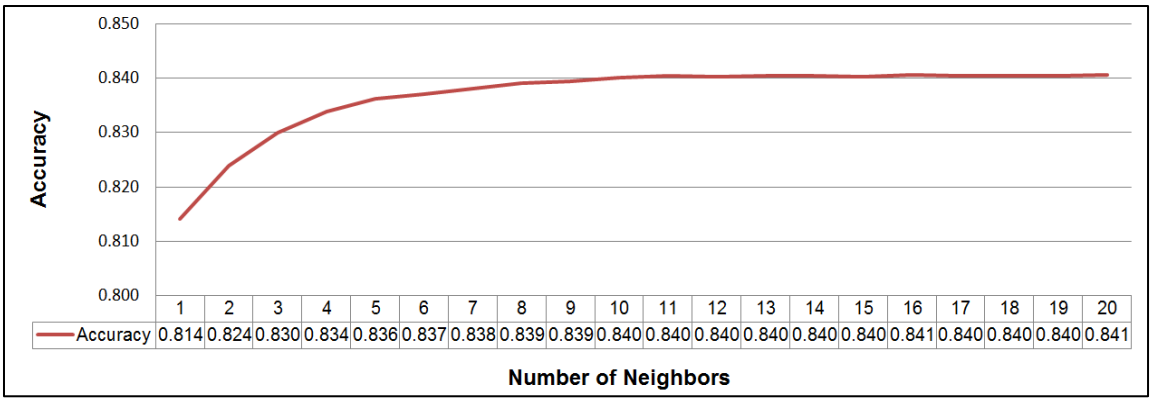


Figure 3.4: Averaged test accuracy over 5-fold cross validation. The value Of K (Gaussian mixture component number) ranges from 1 to 20.

3.4.2 Logistic Regression (LR)

Logistic Regression is a parametric method for binary classification that uses logistic transformation between the posterior probability and the linear combination of observation data (Menard, S., 2002). If x is the feature vector, and $C=\{C_1,C_2\}$ as the class label, then the posterior probability of class C_1 can be written as a logistic function of the linear combination of x as follows:

$$p(y = C_1 \mid x) = \frac{1}{1+\exp(w^T x)} \tag{3.28}$$

where w is the model parameter, and $p(y=C_2|x) = 1- p(y=C_1|x)$. Logistic function is a monotonic, s-shaped, continuous function between 0 and 1.

Maximum likelihood was used to estimate parameters of the LR model. Given a set of training data $\{(x_1, y_1), \ldots, (x_n, y_n)\}$, let $y_i=1$ when the sample takes class label C_1,

and let y_i=0 when the sample takes class label C_2.The likelihood function is defined as follows:

$$L(w; x_1,...x_n) = f(x_1,...x_n; w) = \prod_{i=1}^{n} p^{y_i}(1-p)^{(1-y_i)} \tag{3.29}$$

After taking a logarithm, the log-likelihood function was written as

$$\begin{aligned} \ln L(w; x_1,...x_n) &= \sum_{i=1}^{n} y_i \ln p + (1-y_i)\ln(1-p) \\ &= \sum_{i=1}^{n} y_i \ln(\frac{1}{1+\exp(w^T x)}) + (1-y_i)\ln(1-\frac{1}{1+\exp(w^T x)}) \end{aligned} \tag{3.30}$$

The log likelihood function is convex (Rennie, 2005) and traditional method of estimating parameter is to set the first-order derivative with respect to each parameter equal to zero. Unfortunately, there is no known closed-form way to estimate the parameters in LR. Thus an iterative algorithm, such as gradient descent, needs to be used. Gradient descent requires estimation of the partial derivative. Partial derivative of the *j-th* parameter w_j is written as follows:

$$\begin{aligned} \frac{\partial \ln L(w; x_1,...x_n)}{\partial w_j} &= \sum_{i=1}^{n} y_i \frac{\partial}{\partial w_j}\ln(\frac{1}{1+\exp(w^T x)}) + (1-y_i)\frac{\partial}{\partial w_j}\ln(1-\frac{1}{1+\exp(w^T x)}) \\ &= \sum_{i=1}^{n} (y_i(1+\exp(w^T x)) + (1-y_i)(1+\frac{1}{\exp(w^T x)}))x_j \exp(1+w^T x) \end{aligned} \tag{3.31}$$

Once an initial setting of w_0 is chosen, it can be updated at each iteration as follows:

$$w_j^{t+1} = w_j^t + \alpha \frac{\partial \ln L(w; x_1, ...x_n)}{\partial w_j} \tag{3.32}$$

where α is the stepsize. Standard LR is designed for binary classification, and cannot be directly used for multiclass problems. There are many solution to apply logistic regression to multi-class classification, such as softmax regression, "one-against-all" and "one-against-one" (Bishop, 2006). The "one against all" strategy was chosen in this thesis project. The "one against all" strategy builds one LR for each class, which is trained to distinguish one class from all remaining classes, and the label of a new instance is determined by the maximizing posterior.

3.4.3 Support Vector Machine (SVM)

The LR algorithm focuses on maximizing the likelihood function that considering the all training samples. On the contrary, the SVM classifier attempts to find the separating hyperplane that maximizes the margin (the support vectors), which is defined as the shortest distance from the separating hyperplane to the closest positive (negative) example (Burges, 1998). SVM solely considers points near the margin, instead of the entire training data.

Given a set of training data $\{(x_1, y_1), \ldots, (x_n, y_n)\}$, where x_i is the feature vector and $y_i \in Y=\{-1, +1\}$is class label, let the separating hyperplane be defined by a vector w

with a bias w_0. The vector w makes $wx+w_0 \geq +1$ when class label is +1 and $wx+w_0 \leq -1$ when class label is -1. There are many possible hyperplanes that can separate the two classes, but there is only one optimal hyperplane that represents the largest separation. Finding optimal hyperplane can be modeled as a convex quadratic programming problem as follows: *min* $||w^2||/2$, subject to $y_i(wx+w_0) \geq +1$. Because the training set is often not linearly separable in real applications, a set of variables called slack variables ξ_i were introduced into the marginal maximization function as follows:

$$J(w, w_0, \xi) = \frac{1}{2} \| w^2 \| \sum_{i=1}^{n} \xi_i, \xi_i \geq 0, \quad i = 1,...,n \tag{3.33}$$

$$y_i(wx_i + w_0) \geq 1 - \xi_i \qquad i = 1,...,n \tag{3.34}$$

where C is a regularisation parameter. Using a Lagrangian formulation, the above quadratic programming problem can be translated to the following dual problem.

$$\max(\sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j < x_i, x_j >) \tag{3.35}$$

$$\sum_{i=1}^{n} \alpha_i y_i = 0, 0 \leq \alpha_i \leq C, i = 1,...,n \tag{3.36}$$

where α_i represents the *i-th* Lagrange multiplier. Under this formulation, the equation of the optimal hyperplane discriminant function was then written as following:

$$\sum_{i \in S} \alpha_i y_i < x_i, x > + w_0 \tag{3.37}$$

where *S* is the set of marginal points. According to Mercer's theorem, the inner product of the vectors in the mapping space, can be expressed as a function of the inner products of the corresponding vectors in the original space (Mercer, 1909), which is also called the *"kernel trick"*. The equation can be expressed using kernel function as follows:

$$\sum_{i \in S} \alpha_i y_i K(x_i, x) + w_0 \tag{3.38}$$

The kernel function plays an important role in SVM because it maps original features into higher dimension space, which could alter a non-linear separable problem into a linear separable problem. Common kernel functions used in SVM include polynomial function, Gaussian radial basis function (RBF), and sigmoid function, etc. In this thesis project, we used the LibSVM package to implement SVM and chose BRF kernel. More detail on parameter estimations of LibSVM can be found in Chang and Lin (2011). The SVM was primarily designed to solve binary classification problems. To solve the multiclass problem, the "one against all" strategy was adopted.

However, the decision function in a Traditional SVM classifier produces a categorical value, not a continuous posterior probability that is suitable for association term. To convert the output of the decision function to a posterior probability, we used a modified version of the method in Wu et al. (2004).

3.4.4 Artificial Neural Networks (ANN)

The human brain is a powerful decision making system with millions of neuron connected in a complex way. It is composed of multiple parallel layers of neurons, such that each neuron in any given layer receives input signals from all neurons in the previous layer and sends out different output signals to all neurons in the next layer. By training and memorizing the interaction of neurons across layers, the human brain is able to process information from outside and makes appropriate response. An artificial neural network (ANN) is a computational model whose design is inspired by the mechanism of the human neurons.

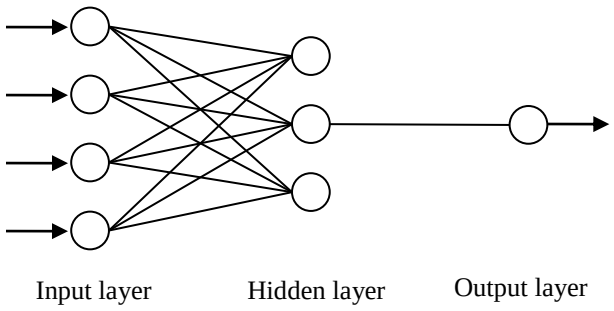


Figure 3.5: Typical structure of ANN with three layers.

A typical ANN consists of three main parts: the input layer, the hidden layer(s), and the output layer. The graphical representation of ANNs are abstractly illustrated in Figure 3.5. Although the figure depicts only a single hidden layer, there can be multiple hidden layers in an ANN. The activation functions and the weights are not shown. The input layer consists of the ANN's inputs (x), which is feature vector. The hidden layer

consists of many hidden neurons, which take in a set of weighted inputs and apply an activation function to their sum. The most commonly used activation function is logistic function, which was accordingly adopted in this thesis. The output from the *k-th* hidden node h_k is given by

$$h_k = f(\sum_i w_{ij} x_i + b) \tag{3.39}$$

where w_{ik} is the weight between the *i-th* input node x_i and the *k-th* hidden node, and b is the bias. The output layer receives the weighted inputs from the hidden layer neurons, and then provides the final prediction result, which can be written as

$$O = f(\sum_k \alpha_k h_k + c) = f(\sum_k \alpha_k f(\sum_i w_{ij} x_i + b) + c) \tag{3.40}$$

where α_k is the weight of the *k-th* hidden node for the final output, and c is the bias term. There are two significant variables need to be considered when working with ANNs: the number of hidden layers and the number of neurons for each hidden layer. As (Bendiktsson et al., 1990) demonstrated that a single hidden layer ANN has good potential, we also decide to use single hidden layer ANN. Referring to Kolmogorov's theorem ANN (Kůrková, V., 1992), we chose the number of neurons for each hidden layer as *2n+1*, where *n* is the number of neurons in input layer.

The backpropagation algorithm is a commonly used iterative method for training ANN. Given an initial weight and training sample, errors, the difference between actual and predicted results, that occurred in the output units are calculated; then they pass backwards, first to the hidden layer and then to the input layer. At each iteration, a gradient descent search is performed to adjust the weights that minimize the error. Further details about the backpropagation algorithm can be found in (Rumelhart et al., 1995).

3.4.5 Decision Tree (DT)

The decision tree is one of the most widely used inductive inference algorithm. It is a non-parametric classifier, and is based on a "divide and conquer" strategy, which is learned by recursively dividing the feature space from a training data (Quinlan, 1986). Decision tree has many advantages over other traditional supervised classification algorithm. Firstly, it is a non-parametric method, thus it does not require any assumptions about the distributions of the input data. As well, it is not a *"black box"* like a neural network, so we can convert decision tree into classification rules that easy to understand for non-experts.

A typical learned decision tree consists of three types of nodes: one root node, a set of interior nodes, and terminal nodes, which are also called "leaves". A new instance starts from the root node, and travels down to its consecutive branch node by testing the feature specified by the current node. This process is repeated until it meets a terminal node. According to which terminal node that it falls into, the final label of the new instance is determined.

Popular decision tree algorithms include ID3, C4.5 and classification and regression tree (CART). These trees mainly differ in the splitting criteria; the ID3 and C4.5 use information theory to split the training samples, while CART uses the Gini index. A critical issue of decision tree is how to select the splitting threshold. For categorical features, the test values are simply the different possible categories; for continuous features, the data need to be sorted at each node and the threshold is obtained by choosing the split between two consecutive values that maximize the criteria. In this research, the Gini index was used and we were concerned with only continues features.

Another critical issue is tree pruning, which is necessary to avoid over-fitting (1987). A fully grown tree is able to classify all training data correctly, but it has potential risk of over-fitting when the training data is noisy, bias or too small. In this research, we terminated tree growth if the number of instances fall below a specified threshold, which was selected using 5-fold cross validation. Figure 3.6 shows the averaged test accuracy of the five folds when the minimum leaf size varies from one to twenty. When the minimum leaf size exceeds twelve, averaged test accuracy becomes stable; thus, minimum leaf size of DT was chosen as 12.

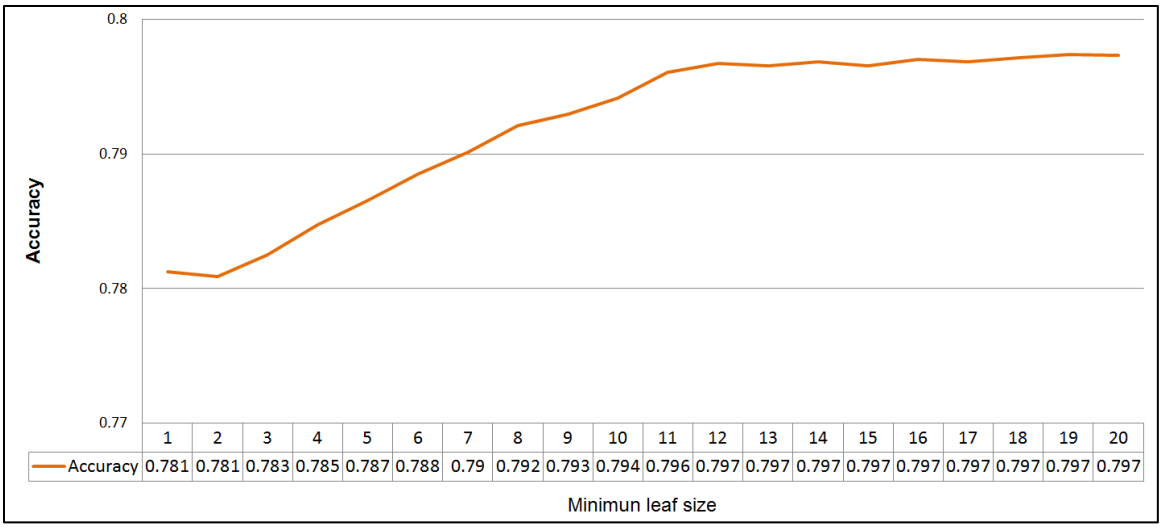


Figure 3.6: Averaged test accuracy over 5-fold cross validation as different minimum leaf size was selected.

3.4.6 Random Forest (RF)

Recently, ensemble classification has gathered increasing amount of attention from the machine learning community. The fundamental idea is that a combination of multiple classifiers can perform better than each the individual classifier alone. The final combined classifier is called the ensemble, and the member classifiers are called as base models, each of which could be any traditional machine learning model, such as decision tree or Naïve Bayes classifier. The ensemble can be regarded as a weighted combination of these base models as shown in Equation 3.41.

$$g(x) = \sum_{k=1}^{K} w_k h_k(x) \tag{3.41}$$

where w_k is the weight of the *k-th* base learner. The variance error of ensemble can be decreased by reducing the correlations of base models, increasing the number of base models, or improving the performance of a single base model (Hsieh, 2009). Two most popular ensemble techniques are bagging and boosting.

The bagging method generates base models by making multiple bootstrap training sets from the original training set. These bootstrap training sets are randomly drawn with replacement, and each of them is used to train a different base classifier. Finally the outputs of these individual base classifiers are combined to make a majority voting. A representative bagging ensemble method is Random Forest (RF) algorithm, which was proposed by Breiman (2001). The word "random" has two types of meaning, random sampling (bootstrap aggregation) and random feature selection, both of which reduce the correlations of base models. It is named "forest" because this ensemble classifier consists of many binary decision trees.

There are two important parameters in RF: the number of trees, and the number of features selected at each node. In theory, the larger the number of trees, the better performance RF has, but this also increases computational complexity. As a result, in this study the number of trees was set 100 as Breiman (2001) suggested. If the number of features is too small, performance of individual tree will decrease, but if the number used is too large, the correlation between trees increases. Therefore, Breiman (2001) suggested a middle value, the square root of the total number of features. The critical tree pruning in DT is no longer considered here because over-fitting risk is supposed to be prevented by

the random aspects of RF. Given the training data and determined parameters, the learning of each tree in RF is the same as decision tree learning in section 3.4.6.

Once the forest is learned, a new instance runs across all the trees in the forest. Each tree votes a prediction label, and votes from all trees are then combined to make a majority voting.

3.4.7 Adaptive Boosting (AdaBoost)

Instead of random sampling of training data and combining classifiers with equal vote as in the bagging method, the boosting method uses a weighted sample to focus learning on misclassified samples by the previous weak classifier and finally combines all classifiers using a weighted vote (Freund et al., 1995). Base models in the bagging method are independent, but in the boosting algorithm, base model is highly dependent on the previous one, and focuses on the previous model's errors. Adaboost was proposed by Freund and Schapire (1995), and was reported in (Freund and Schapire, 1995; Freund and Schapire, 1996) as the most successful boosting algorithm. Boosting algorithm has two main concerns: how to update sample weights at each boosting round, and how to combine these weak base models into a single prediction rule. AdaBoost addressed these two questions by selecting a special parameter α on each round for both updating the sample weight and assigning voting weight for each base model.

Given a training set $\{(x_1, y_1), \ldots, (x_n, y_n)\}$, where x_i is the feature vector and $y_i \in Y=\{-1, +1\}$ is class label. The first step is initializing the weights of all training samples, for which equally weighting is commonly used. The weight of the *i-th* training sample on round t is denoted $D_t(i)$. On each round $t = 1, \ldots T$, the weights of incorrectly classified

samples under hypothesis model h_t are increased so that the model is forced to focus more on these hard samples in next round. The error of a hypothesis model h_t is measured by summarizing the weights of the misclassified training examples as follows:

$$\varepsilon_t = \sum_{i=1}^{n} D_t(i)[y_t \neq h_t(x_i)] \tag{3.42}$$

AdaBoost chooses a parameter α_t, which is specifically selected by minimizing the training error of the combinational classifier. α_t is based on the error ϵ_t as follows:

$$\alpha_t = \frac{1}{2}\ln(\frac{1-\varepsilon_t}{\varepsilon_t}) \tag{3.43}$$

The sample weights are then updated using the rule in Eqaution (3.44). The rule increases the weight of misclassified samples by h_t, and decreases the weight of correctly classified samples.

$$\begin{aligned} D_{t+1}(i) &= \frac{D_t(i)}{Z_t}\begin{cases}\exp(-\alpha_t) & if \quad y_t = h_t(x_i) \\ \exp(\alpha_t) & if \quad y_t \neq h_t(x_i)\end{cases} \\ &= \frac{D_t(i)}{Z_t}\exp(\alpha_t y_t h_t(x_i)) \end{aligned} \tag{3.44}$$

After *T* updates, the final hypothesis *H(x)* is a weighted voting of the T weak hypotheses as

$$H(x) = sign(\sum_{t=1}^{T} \alpha_t h_t(x)) \tag{3.45}$$

It is noted that AdaBoost was originally designed for binary classification problems. There are several methods of extending AdaBoost to the multiclass case, such as AdaBoost-M1 (Freund and Schapire, 1995), Stagewise Additive Modeling using a Multi-class Exponential loss function (SAMME) suggested by Zhu et al. (2009), and AdaBoost-Cost (Mukherjee and Schapire, 2011). In this thesis, SAMME was used to solve the multiclass Adaboost classification; further details about SAMME can be found in Zhu et al. (2009).

3.5 Experiment Results

The effectiveness of line-based TLS classification was validated using ten classifiers mentioned in section 3.3 and section 3.4, with two TLS data sets. For each classifier, the two-fold cross validation was used to test its generalization ability. Since the number of mixture number of GMM and number of neighbor of KNN need to be fixed before model learning, they were selected using five-fold cross validation only on one piece of data, YV1, which is introduced in section 3.5.1.

The three generative classifiers were implemented using Matlab software; while the discriminative classifiers were implemented using open source packages. Implementation of the artificial neural network refers the R package “nnet” (Ripley et al., 2015), and implementation of other discriminative classifiers refer to scikit-learn package

(Pedregosa et al., 2011). This package provides existing function, but parameters of each classifier were still tuned based on the experimental data.

3.5.1 Experimental Data

The data set was collected at two different sites, on Kidd Terrace (Figure 3.7), York village community, Toronto. The two datasets are noted as YV1 and YV2 respectively. Both of them show typical North American residential street views, where two or three story houses are built densely along the street. Architectural styles of buildings, tree species of the two sites are different. And both of them have a problem of occlusion.

Figure 3.7: Real scene of the York Village Data.

The experimental dataset is categorized into seven classes: building, roof, pedestrian road (PR), tree, low man-made object (LMO), vehicle road (VR), and low vegetation (LV). The Table 3.1 presents the object categorization and description of each class.

Table 3.1: Object categorization of experimental dataset.

Class	Objects belong to the class
Building	Building façade
Roof	Roof
Pedestrian Road(PR)	Grass land + Pedestrian roads
Tree	Tree
Low Man-made object (LMO)	Car, Pedestrian, Garbage bin, Steps, Fence, Railing
Vehicle Road(VR)	Vehicle road
Low Vegetation(LV)	Bush, flower

To collect TLS data, the RIEGL LMS Z390i laser scanner was put on a Leica tripod, which was adjusted at a horizontal plane using levelling adaptor in advance. Due to safety concerns, it is not possible to put laser scanner in the center of street to collect panoramic point cloud. Thus, the laser scanner was put on the pedestrian walk and scanned objects on the other side; the distance between the laser scanner and building facade ranged from 20 to 70 meters. Both vertical and horizontal scanning angular precision were set to 0.05 degree. The horizontal view of field determines the data acquisition time and the number of points. Because only one side of street needed to be scanned at a time, the horizontal view of field was smaller than 180 degrees. A 50 meter street requires about 30 minutes for surveying, and generates around 2 million points.

To transfer coordination of a point cloud from the scanner coordinate system to the geodetic coordinate system, geo-reference is required. As this research mainly focuses on TLS data classification, geo-reference was not considered. Instead, before mounting the laser scanner, the tripod was adjusted to be at a horizontal plane using the levelling adaptor as previously mentioned.

Both of the two datasets have about three million points, and they were split into 2810 and 2580 scan profiles respectively. Finally, about 105620 lines were extracted from the data YV1 and 100648 lines from the data YV2. Table 3.2 summarizes the total number of spatial entities extracted from the two datasets.

Table 3.2: Number of laser point, scan profile, line segment in York Village dataset.

Spatial entities	**YV1**	**YV2**
Laser scanning point	3,294,337	3,087,301
Scan profiles	2,810	2,580
Line segments	105,620	100,648

Local features and contextual features were then extracted for each line, and PCA was used to reduce feature dimension into eight. When feature is ready, classifiers can be learned. The two-fold cross validation method was used to evaluate the performance of 10 classifiers. Each classifier was learned from one dataset and then tested on the remaining dataset, which was repeated two times. Classification performance was measured individually, and was also averaged. At first, all points were manually labeled and then ground truth of each line was assigned to be the majority of its member points' labels. Because of varying point density and occlusion, it was difficult to visually identify the nature of some points. Therefore, these ambiguous points were labeled 'unknown' and were not used for performance evaluation. The percentage of "unknown" point is lower than 1%. Both qualitative (section 3.5.2) and quantitative (section 3.5.3) analyses were done.

3.5.2 Qualitative Analysis

GMM and SVM were selected as representatives of generative and discriminative classifiers. The classification results of both GMM and SVM over the data YV2 are presented respectively in Figure 3.8(a) and 3.8(b), while Figure 3.8(c) shows the ground truth. Most lines from building, roof, pedestrian road and vehicle road were correctly classified. However some misclassification errors were apparent in the result. We categorized the misclassification errors into two types, local inconsistency and incorrect scene layout. Red bounded regions in Figure 3.8(a) and Figure 3.8(b) show examples of local inconsistency, in which a few of building lines were misclassified as tree or roof, which is also called *pepper and salt* noise. Blue bounded regions in Figure 3.8(a) and Figure 3.8(b) shows examples of incorrect scene layout, in which the roof was found below building and the tree was surrounded by building. These misclassification errors result from ambiguities in appearance feature among classes in varying vision conditions, which could affect the distinguish ability. As local classifier only relying on local appearance, it has big risk of misclassification.

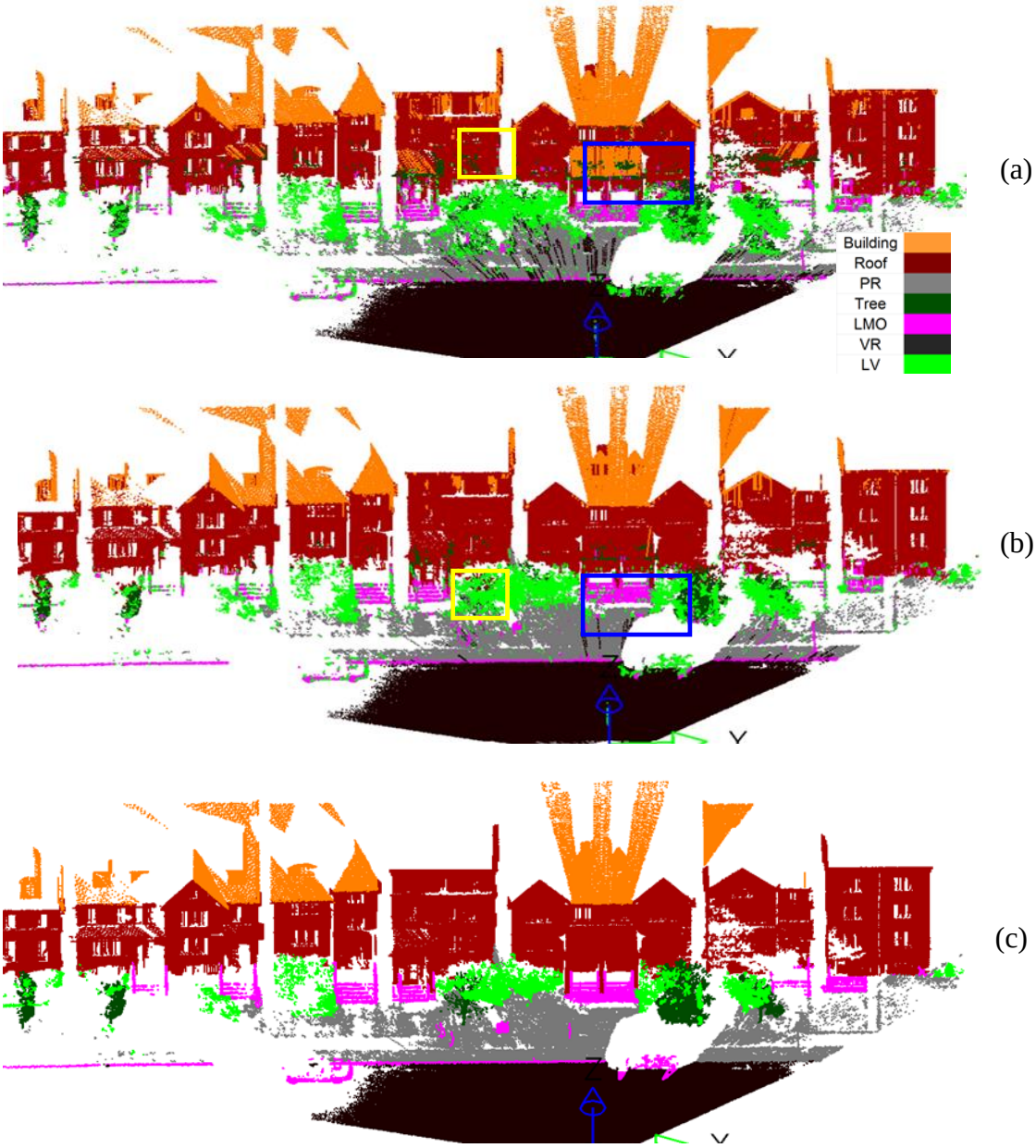


Figure 3.8: Classification result of GMM, SVM and ground truth. (a) classification result of GMM; (b) classification result of SVM; (c) ground truth.

3.5.3 Quantitative Analysis

Five classical evaluation metrics were used to quantitatively measure the classification performance, namely confusion matrix, accuracy, precision, recall and F1 score. Of these, the last four were derived from the confusion matrix. All quantitative measurements in this research are based on lines not on points.

The confusion matrix is an effective way to quantitatively visualize the classification performance. A confusion matrix shows the number of correct (diagonal elements) and incorrect (off-diagonal elements) predictions made by the classifier compared with the data's ground truth. The matrix is *k* by *k*, where *k* is the number of class types. Each row represents the instances in a predicted class and each column represents the instances in an actual class. Non-diagonal elements at row *i* column *j* indicates the number of true class *i* misclassified as class *j*. GMM and SVM over data YV2 were selected as representatives of generative and discriminative classifiers, and confusion matrices of them are presented respectively in Table3.3 and Table3.4. It is observed that misclassification errors mainly occurred in distinguishing building and roof, building and tree, building and LMO, tree and LV, LV and LMO, etc.

Table 3.3: Confusion matrix of GMM classifier of data YV2.

		Prediction						
		Building	Roof	PR	Tree	LMO	VR	LV
Ground Truth	Building	35537	1654	20	1674	686	0	284
	Roof	892	2860	0	0	0	0	0
	PR	138	0	12678	732	860	426	2113
	Tree	1186	8	6	8341	226	0	1356
	LMO	742	0	282	290	6241	68	1771
	VR	6	0	967	1	58	6727	59
	LV	197	0	185	2555	673	3	8003

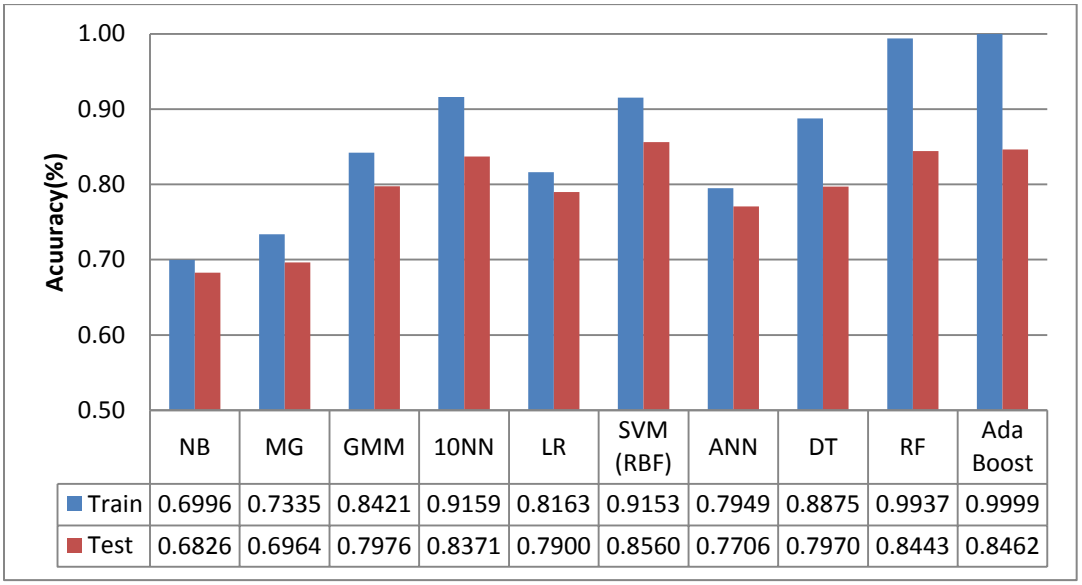
Table 3.4: Confusion matrix of SVM classifier of data YV2.

		Prediction						
		Building	Roof	PR	Tree	LMO	VR	LV
Ground Truth	Building	38288	684	3	411	314	0	155
	Roof	1009	2743	0	0	0	0	0
	PR	94	0	14276	49	701	218	1609
	Tree	2574	0	7	7532	123	0	887
	LMO	579	0	327	91	6693	28	1676
	VR	2	0	1049	0	49	6712	6
	LV	334	0	180	1261	333	0	9508

Accuracy measures the average performance of all classes. It is the proportion of the sum of correct predictions (diagonal elements) to the total amount of data, and is defined as follows:

$$accuracy = \frac{\sum_{i=j} C_{i,j}}{\sum_{i=1,\ldots,7} \left(\sum_{j=1,\ldots,7} C_{i,j} \right)} \tag{3.46}$$

Figure 3.9 presents the averaged train / test accuracy of the 10 classifiers. An averaged accuracy was calculated based on overall data that combines YV1 and YV2 data. Comparing train and test accuracy, it is observed that all classifiers did not have high over-fitting risk, except for RF and AdaBoost. Based on comparison of test accuracy, GMM (79.76%) showed the best performance of the generative classifiers, followed by MG (69.64%) and NB (68.26%). Among discriminative classifiers, SVM with RBF kernel (85.60%) had the best performance, followed by AdaBoost (84.62%), RF (84.43%), 10NN (83.71%), DT (79.70%), LR (79.00%), and ANN (77.06%). The averaged accuracy over all ten classifier was 79.19%. As well, on the whole, discriminative classifiers performed better than generative classifiers. As expected, the two decision tree based ensemble methods were better than single decision tree.



	NB	MG	GMM	10NN	LR	SVM (RBF)	ANN	DT	RF	Ada Boost
Train	0.6996	0.7335	0.8421	0.9159	0.8163	0.9153	0.7949	0.8875	0.9937	0.9999
Test	0.6826	0.6964	0.7976	0.8371	0.7900	0.8560	0.7706	0.7970	0.8443	0.8462

Figure 3.9: Averaged accuracy of ten classifiers.

The accuracy measures the overall correctness of all classes, and consequently it fails to measure the performance of any single class. So we also used recall and precision to evaluate the performance of each single class. Precision, which is also called producer accuracy, measures the percentage of objects that are correctly classified as "building" from all the objects that truly are "building". It is defined by

$$precision = \frac{C_{i,i}}{\sum_{j=1,...,7} C_{i,j}} \tag{3.47}$$

Table 3.5 presents the precision of each class in 10 classifiers. All classifiers achieved high precision on building. The averaged precisions of PR and VR are higher than 80%, but precision greatly varies with classifiers. For example, the precision of VR is 94.31% in AdaBoost, but falls down to 60.12% in MG. LV and Tree also showed satisfying precision. Precision of roof is rather sensitive to different classifier, with maximum precision in 86.46% (RF) and minimum precision of 39.38% (MG).

The recall, which is also called user accuracy, is the proportion of objects that are correctly classified as "building" from all the objects that are predicted as "building". Recall is defined by

$$recall = \frac{C_{i,i}}{\sum_{i=1,...,7} C_{i,j}} \tag{3.48}$$

Table 3.6 presents the recall of each class in 10 classifiers. Building still lead the precision rank, but were also sensitive to different classifiers, with maximum recall of 95.54% (RF) and minimum recall of 68.46%. The averaged recalls of VR and PR were over 80%. The averaged recalls of other classes were not as high as those classes mentioned above, but still over 60%.

Table 3.5: Precision of each class in ten classifiers.

Classifier	**Building**	**Roof**	**PR**	**Tree**	**LMO**	**VR**	**LV**
NB	0.8792	0.4657	0.7765	0.4303	0.5805	0.7657	0.5598
MG	0.9417	0.3938	0.8262	0.5076	0.6822	0.6012	0.5844
GMM	0.9189	0.6588	0.8936	0.6059	0.6589	0.8752	0.6479
10NN	0.9049	0.8346	0.8384	0.7255	0.7611	0.9152	0.7083
LR	0.9251	0.6043	0.7887	0.6254	0.6971	0.8818	0.6372
SVM	0.9099	0.8440	0.8429	0.8088	0.7807	0.9397	0.7396
ANN	0.8653	0.6528	0.7278	0.6333	0.7038	0.8908	0.6268
DT	0.8757	0.7174	0.8019	0.6943	0.6653	0.8840	0.6655
RF	0.8816	0.8646	0.8599	0.7772	0.7666	0.9366	0.7342
AdaBoost	0.8949	0.8515	0.8672	0.7787	0.7492	0.9431	0.7172
Average	0.8997	0.6887	0.8223	0.6587	0.7046	0.8633	0.6621

Table 3.6: Recall of each class in ten classifiers.

Classifier	**Building**	**Roof**	**PR**	**Tree**	**LMO**	**VR**	**LV**
NB	0.6846	0.7250	0.6804	0.6674	0.4656	0.7963	0.7760
MG	0.6919	0.8810	0.5609	0.7825	0.5672	0.9366	0.7346
GMM	0.8637	0.8018	0.7699	0.7527	0.6941	0.8904	0.6978
10NN	0.9309	0.6958	0.8784	0.6986	0.6843	0.8891	0.7221
LR	0.8637	0.7478	0.8038	0.6752	0.7200	0.8446	0.6616
SVM	0.9506	0.7091	0.8944	0.7227	0.7433	0.8773	0.7279
ANN	0.9075	0.4807	0.8160	0.5776	0.7061	0.7887	0.5590
DT	0.9190	0.6135	0.8222	0.6472	0.6252	0.8653	0.6441
RF	0.9554	0.6033	0.8753	0.6832	0.6976	0.9015	0.7339
AdaBoost	0.9474	0.6295	0.8752	0.6946	0.7177	0.8996	0.7429
Average	0.8997	0.6887	0.8223	0.6587	0.7046	0.8633	0.6621

Individual precision and recall cannot describe the entire performance of a classifier, thus F1 score that combines both precision and recall, was introduced. Thus F1 score is also called the harmonic mean of precision and recall, and is calculated as follows:

$$F1 \quad score = \frac{precision \times recall}{precision + recall} \tag{3.49}$$

Averaged F1 score of each class in 10 classifiers was presented in Table 3.7. Building (0.8819) had the highest F1-score, followed by two types of roads, VR (0.8619) and PR (0.8063); all the three classes were effectively detected by all classifiers. Other four classes had satisfying F1 scores, between 0.65 and 0.70.

Table 3.7: F1 score of each class in 10 classifiers.

Classifier	**Building**	**Roof**	**PR**	**Tree**	**LMO**	**VR**	**LV**
NB	0.7698	0.5671	0.7253	0.5233	0.5167	0.7807	0.6504
MG	0.7977	0.5443	0.6682	0.6158	0.6194	0.7323	0.6509
GMM	0.8905	0.7233	0.8271	0.6714	0.6761	0.8827	0.6719
10NN	0.9177	0.7589	0.8579	0.7118	0.7207	0.9020	0.7152
LR	0.8934	0.6685	0.7962	0.6493	0.7084	0.8628	0.6492
SVM	0.9298	0.7706	0.8679	0.7633	0.7616	0.9074	0.7337
ANN	0.8859	0.5537	0.7694	0.6042	0.7049	0.8367	0.5910
DT	0.8968	0.6614	0.8119	0.6699	0.6446	0.8745	0.6546
RF	0.9171	0.7107	0.8676	0.7272	0.7305	0.9187	0.7341
AdaBoost	0.9204	0.7239	0.8712	0.7342	0.7331	0.9208	0.7298
Average	0.8819	0.6682	0.8063	0.6670	0.6816	0.8619	0.6781

3.6 Chapter Summary

To summarize, in this chapter, a line-based TLS data classification method was proposed. Firstly the lines were extracted from each vertical scan profile. Ten popular generative and discriminative classifiers were then used to validate the effectiveness of line-based method. For each classifier, two-fold cross validation was used to test its generalization ability; each classifier was learned from one dataset and then tested on the remaining dataset, which was repeated two times. The experiment results showed that all ten classifier achieved satisfying accuracy, with averaged accuracy of the ten classifiers of 79.19%.

However, the limitations of the classifier were also observed in classification errors that result from similar local appearance, which is a typical drawback of using local classifier. Misclassification errors are mainly found between building and roof, building and tree, building and LMO, tree and LV, LV and LMO, among other classes. When feature distribution of one class is not clearly discriminated from another, which is presented in figure, misclassification tends to occur. Even if the overlapping area is limited on the training data level, this does not necessarily hold with the test data.

To improve the classification performance of local classifier, object context will be considered in next Chapter.

Chapter 4
Along Scan Profile Conditional Random Field

This chapter demonstrates that the performance of local appearance based classifier can be improved by considering multi-range contexts in conditional random field (CRF) model. Since the context is only considered in each scan profile, all CRF models proposed in this chapter are along scan profile CRFs. Firstly, the limitations of local classifier were discussed using classification results of GMM as an example of. Then three types of object context were exploited: short range context that enforces local smoothness, as well as the long range vertical and horizontal context that provide priori information of scene-layout compatibility. To examine the effect of different contexts, three single range CRF models were separately constructed. The final goal is to integrate multi-range asymmetric contexts in one CRF model, which is called maCRF. The posterior probability of GMM was used as association term for each of the four CRF models. To evaluate the advantage of multi-range context, the four CRF models were tested on the York Village data using cross-validation, and their classification performance was evaluated and compared.

4.1 Methodology Overview

Classification is the problem of identifying corresponding class label that belongs to an "entity" (e.g., point, line and plane in laser point space) with given observations ("features"). A typical approach uses information at a local level without considering the object context, only relying on apparent features to differentiate the object from the others; this approach is called local classifier. Local classifiers are classic supervised classification methods and have already been proved to be efficient for classifying TLS data, details of which were presented in Chapter 3.

However, due to ambiguities in appearances of objects and varying vision conditions, overlap between the territories of multiple classes in feature space can be found. The overlapping of feature distribution causes one class to be not clearly discriminated from others, and thus classification errors are anticipated by local classifiers. Figure 4.1 presents height distributions of seven classes. It is noticed that overlapping problem is very serious, which results in a non-linear separable classification problem. If only these apparent features are used to building supervised classifier, there will be a risk of misclassification, which was validated by experimental results in the Chapter 3. From the experimental results of the Chapter 3, it is observed that misclassification errors were mainly from building and roof, building and tree, low vegetation and tree, low vegetation and pedestrian road.

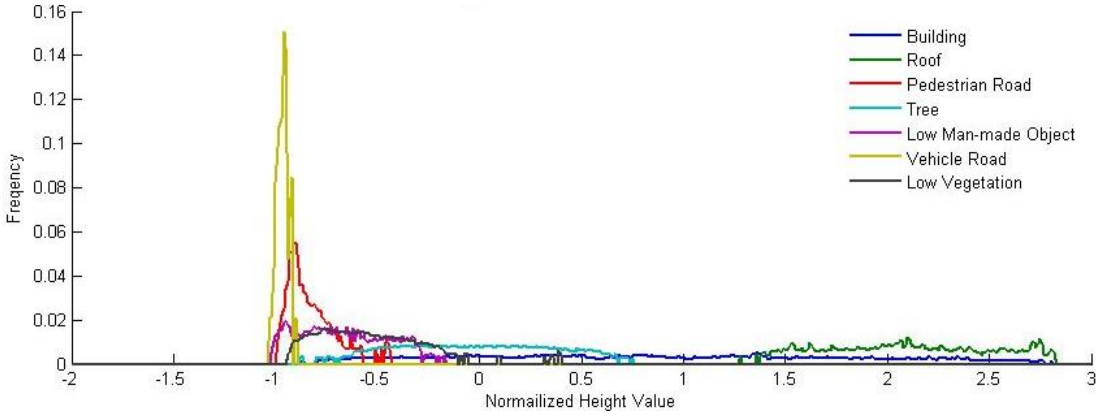


Figure 4.1: Height distributions of seven classes.

Recent work on object context has shown its power in improving classification performance when it collaborates with local appearance features (Gamba and Dell'Acqua, 2003; Oliva and Torralba, 2007). Object context makes assumptions on spatial consistency or compatibility of objects, which compensates for insufficient information of local appearance features (Oliva and Torralba, 2007). A natural and simple method to apply contextual assumption is taking post-processing on existing classification output, such as $k \times k$ filtering window, to make a smoother result (Gamba and Dell'Acqua, 2003). This filtering window based method is easy to implement in practice, but always brings the risk of over-smoothness.

Another way applying contextual assumption is to directly impose spatial dependence between adjacent entities in a classifier. Conditional Random Field (CRF) is a well-known classifier that enables the modelling of object dependence and local appearance in a single model (Lafferty et al., 2001; Kumar and Hebert, 2003; He, et al., 2004). A commonly used spatial dependency is local smoothness, which maximizes the

local label homogeneity between adjacent entities. The first CRF model we developed is short range CRF (srCRF) that enforce local smoothness and emphasizes on local label consistency. However, the srCRF fails to captures the long range global dependency of objects. Moreover, because of occlusion, some lines do not have even short range neighbor.

Therefore, another type of spatial dependency was also exploited, the regularity of spatial arrangement between long range adjacent objects, which is a global prior on scene-layout. For instance, the pedestrian road is usually below its adjacent objects, like building and tree, and trees or lamp post is generally closer to vehicle road than building. Such scene-layout spatial dependency was modeled as pairwise interaction potential in vertical and horizontal direction respectively, corresponding CRF models of which were called long range vertical CRF (lrCRF(V)) and long range horizontal CRF (lrCRF(H)). In particular, we adopted an asymmetric interaction potential to capture directional scene layout (e.g. it allows ground is lower than building, not vice-versa).

Then the power of all three different context sources were integrated together (short range, long range vertical and long range horizontal) with local appearance in one single CRF model, which is called multi-range asymmetric CRF (maCRF). Following the work of Chapter 3, we selected the line primitive as the entity for constructing CRF models and each CRF model was built within each scan profile; the adjacent relations between lines were constructed with the assistance of a grid. Finally, the four classifiers were tested on TLS data, and their performances were both qualitative and quantitatively analyzed.

4.2 Line Adjacent Graph

In a CRF model, dependent relations of nodes are defined by an adjacent graph. To construct a graph, the first thing needs to be considered is how to defined neighboring relation between classification primitives. Defining adjacent relation in image space can be based on the grid pattern. In pixel based image classification, neighbours of a pixel can be searched for using standard 4-connected neighborhood (Kumar and Hebert, 2006; Shotton et al., 2006) or 8-connected neighborhood (He et al., 2004). In super-pixel based image classification, an adjacent relation is defined when two super-pixels share part of boundary (Gould, et al., 2008). However, graph construction methods for image do not work for laser scanning data, because laser scanning data does not conform to a regular grid pattern and point distribution is very sparse. Delaunay triangulation (DT) and *k* nearest neighbours are commonly used methods to build adjacent connections between laser points. Delaunay triangulation (implemented in 2D space in cited publications) is a popular method for finding nearest neighbors, and has already been used for laser scanning data processing, such as planar faces detection (Vosselman, 1999), surface reconstruction (Gopi et al., 2000), and segmentation (Hyyppa et al., 2001). In recent work by Douillard et al. (2008), an adjacent graph of the CRF model was determined via Delaunay triangulation over laser points. Another popular method for building adjacent graphs over laser points is *k* nearest neighbours, which adds all the edges that connecting with *k* nearest neighbours in spherical space (Munoz et al., 2008; Niemeyer et al., 2011, Schmidt et al., 2012), vertical cylindrical space (Niemeyer, et al., 2012), or some projected 2D space (Shapovalov et al., 2010).

The above two methods are effective in finding short ranges nearest neighbors, but fail to capture long range neighbors. Li and Huttenlocher (2008) proposed a sparse long-range random field (SLRF) model, which represents interactions between distant pixels using sparse edges with a clique size of three. In Lim and Suter (2009), points from neighboring super-pixels were defined as long range neighbors. In the recent work of Najafi, et al. (2014), point segments were projected on the ground plane and segments with more than 50% overlapping on this ground plane were considered as high-order neighbors.

In this research, the adjacent relations of lines were defined with the assistant of a grid system. Firstly, the points were then projected into XY-Z 2D space. The coordinate of XY dimension is the square root of X square and Y square. Then the 2D space was quantized along the Z and XY directions in a grid, with cell size of 0.5m by 0.5m. All the cells that a line passes through were regarded as cells occupied by the line. Figure 4.2 presents the quantized grid and gives a few example of the line-cell occupancy relations.

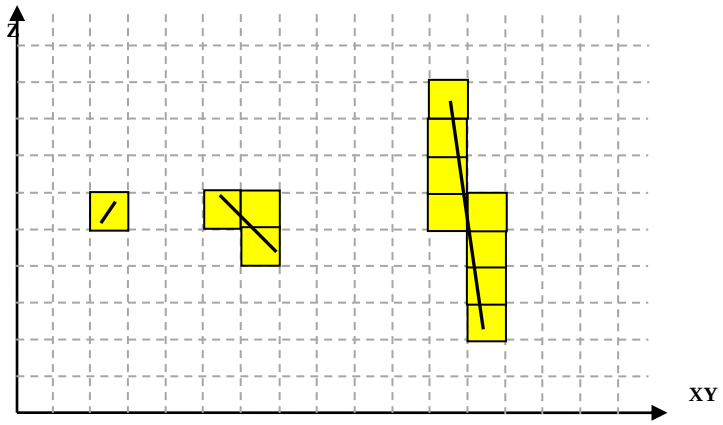


Figure 4.2: Example of grid system and line-cell occupancy relations. The occupied cells of each lines are marked in yellow.

In this research, we considered short range, long range vertical and long range horizontal neighbours. Neighbor searching of a line in this research is based on the grid, and can be divided into three steps. Firstly, cells that a line occupies were queried. Secondly, neighboring cells for all occupied cells were searched according to a predefined rule, such as 8-connected neighborhood. Lines pass those neighboring cells were neighbors the current line. The neighbor finding is visualized in Figure 4.3. Given an occupied cell (yellow), the 8-connected cells and the current occupied are considered as short range neighbors. Outside the 8-connected neighborhood, cells right above and right below (blue) are potential long range vertical neighbors, while cells at the left and right (purple) are considered as potential long range horizontal neighbors.

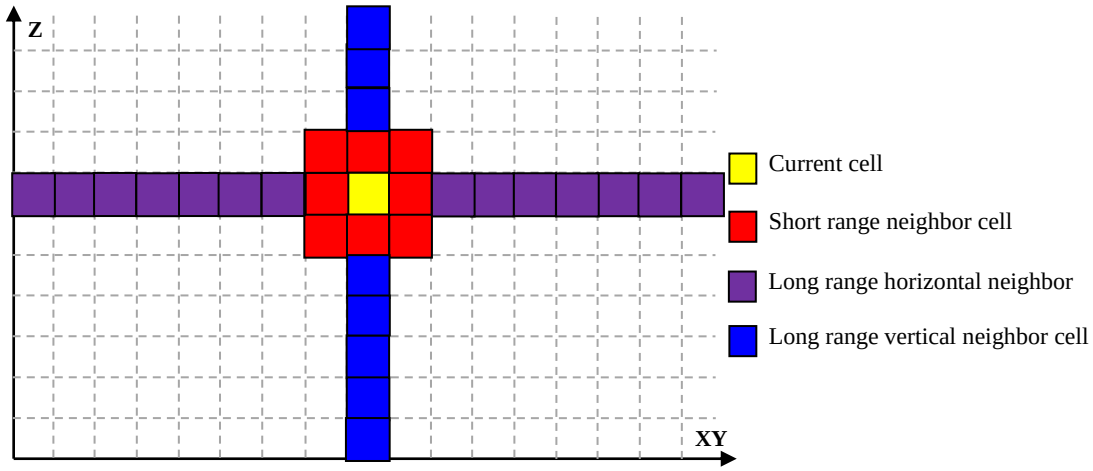


Figure 4.3: Multi-range neighborhood searching for each cell. Each types of neighbor are marked using different color.

4.3 Short Range CRF (srCRF)

The short range CRF makes a smooth assumption that objects in a given local neighbourhood tend to have the same class label. At first, graph construction of srCRF will be introduced. To conduct a comparative research, outputs of local classifiers were used as association potential. From the experimental results of chapter 3, the output of GMM classifier was used as input of association term. The interaction potential of srCRF was designed as Potts model.

4.3.1 Graph Construction

Let $G_S = (V, E_S)$ be a short range graph, each of which node, $v \in V$ represents a line segment (centroid of a line) extracted from one scan profile. Its node adjacency relation, $e_S \in E_S$ was constructed if a line passes the occupied cells of the other line or the 8-connected neighboring cells of these occupied cells, which is illustrated in Figure 4.3. Lines pass red cells are considered as short-range neighbor of the line occupies the yellow cell. It is noted that, in contrast with a graph model represented in image space, our line-based graph does not follow a regular grid pattern.

Given the fundamental theorem of random fields, the conditional distribution over the labels Y given observed data X in the graph G_s is defined in Equation 4.1.

$$P_S(Y \mid X) = \frac{1}{Z_S(X)} \exp\left(\sum_{i \in V} \lambda A_i(X, y_i) + \alpha \sum_{(i,j) \in NS_i} S_{ij}(X, y_i, y_j)\right) \qquad (4.1)$$

where $A_i(X, y_i)$ is the association potential, which measures the probability that class label y_i is assigned to a single node i given global observations X, without considering a relational regularity (interaction) with other nodes; $S_{ij}(X, y_i, y_j)$ is the short range potential and measures how the labels at neighboring nodes (y_i, y_j) interact given the observation X; λ and α are the corresponding weights of potential terms; and Z_S is the normalization term (partition function), which is always computed using a forward-backward algorithm.

4.3.2 Association Potential

The association potential term in Equation 4.1 encodes the cost of assigning label y_i to node i given observation x_i, and it corresponds to a log posterior probability. Generalized linear models, which is a quadratic expansion of all node features in order to find a more accurate quadratic decision surface instead of a linear one, is often used to model associate potential (Kumar and Hebert, 2006). Theoretically, the posterior probability of any classifier can be used, generative or discriminative classifier, such as multilayer perceptron (He et al., 2004), support vector machines (Najafi, et al., 2014). Recently random forests attract more and more attention for modeling association potential, and posterior probability of each class is assigned proportionally to the number of trees voting for the class label (Shapovalov, et al., 2010; Fröhlich, et al., 2013).

To make a comparative research, the log posterior probability of GMM was used as associate potentials of the srCRF model.

$$A_i(X, y_i) = \log(P(y_i \mid x_i)) \tag{4.2}$$

where *i* indicates a line segment.

4.3.3 Interaction Potential

The interaction potential measures how compatible the labels of neighboring objects are, given the observation. In general, arbitrary non-negative functions can be designed as CRF interaction potential. The Potts model (Ising model for binary classification) is a widely used interaction potential, which is extensively used for modeling random fields (Winkler, 2003). The Potts model enforces an assumption of constant smoothing of labels, and penalizes when neighboring objects have different class labels. The Potts model is easy to design and implement; thus it was used to model interaction term in this research. For each short range edge connecting two nodes *i* and *j*, the energy of short range interaction potential S_{ij} is expressed as below:

$$S_{ij}(X, y_i = l, y_j = k) = \begin{cases} 1 & l = k \\ 0 & l \neq k \end{cases} \tag{4.3}$$

4.4 Long Range CRF

The Potts model in srCRF is based on the assumption that smooth distribution of objects in space, and neighboring lines tend to have the same class label. This assumption achieved excellent classification performance in (Anguelov et al., 2005; Kumar and Hebert, 2006; Munoz et al., 2008). However, the distribution of urban objects in space tends to follow some underlying organization rules rather than being randomly placed or

following only a simple homogeneous rule. Moreover, because of occlusion, some lines do not even have short range neighbor. Compared with spatial relation at short range, long range neighbor searching can connect an isolated line with other lines far apart. More importantly, long range level relation can provided global context in spatial arrangement. Scene layout is a commonly used long range global context; it provides strong spatial contextual cues as for where and how objects are expected to be found in the space (Bao, et al., 2011). In this research, scene layouts of urban objects were considered in both vertical ("above-below" relation) and horizontal direction ("front-behind" relation). To model the directional scene layout of objects, asymmetric interaction potentials were designed.

4.4.1 Scene Layout

The scene layout corresponds to the relative locations of objects in a scene. It answers questions such as: *which objects are expected to be above and below another object in urban environment?* Pu and Vosselman (2009) manually defined scene layout rules of objects based on size (e.g., wall has the largest size), position (e.g., roof always on the top of walls), orientation (e.g., walls are vertical and roofs are never vertical), etc. These predefined rules were then applied in classifying segmented TLS data. Such unsupervised rules learning does not require labeled training data, and rules inference can be modified and updated. So rule based method is easily made and implemented. However, the major issue of this method is that it cannot cover all the rules that govern object layout, let alone the conditions behind these rules. In contrast, supervised training is able to learn scene-layout rules automatically from labeled data and can also be updated easily. The learning

of the underlying object scene layouts can be modeled to be a problem of optimizing objective functions that incorporate the layout structure, the layout parameters, and the appearance. Once the learned model is ready, posterior probabilities for all possible label configurations are estimated and the final label can be decided by maximum a posterior.

Winn and Shotton (2006) modeled four types of relative location relations (above/below/left/right) over pixels using asymmetric pairwise potential, whilst also propagating long-range spatial constraints using only local pairwise interactions. The parameters of the asymmetric pairwise potentials were learned using cross-validation, using a search over a sensible range of positive values. In Gould et al. (2008), the layouts of objects were modeled as non-parametric relative location probability maps over pixels, from the statistics of first-stage classification results that only based an appearance features. The final classifier was trained using both appearance-based features and contextual features from relative location probability maps. Heesch and Petrou (2010) modeled scene layout as the conditional distribution of a segmented region, given the objects in its six local neighboring regions, above, below, left, right, as well as regions containing and being contained by the current region. Jahangiri et al. (2010) defined three types of scene layout between segmented region pairs, relative vertical or horizontal orientation, and containment relation. Potentials of relative vertical and horizontal orientation were modeled as sine and cosine functions respectively with respect to the angle between two regions. The potential of other relations were formulated as a Potts model. Ding et al. (2014) modeled scene layout using the label layout filter (LLF), which provides local context clues like 1) which classes exist around certain position, 2) the

proportion of each class, and 3) difference distances and orientations of the context connections implied by different forms of region.

In summary, the principle of scene layout is that the relative location of objects in urban environments is not arbitrary but follows some rules. In this research, we modeled scene layouts of objects in both vertical and horizontal direction. Vertical scene layout is rather strong in urban environments because of the function of objects. For example, roof is designed to protect people and their possessions inside of a building from climatic elements, and so should be above the building. Because almost all of daily human activity happens above the ground, as the main activity region, buildings are above ground. Therefore, vertical scene layout is modeled as a "above-below" relation, such as building is below roof but above the ground. Along the scanning direction, objects are also placed in order, and horizontal scene layouts are modeled as a "front-behind" relation. For example, tree is in front of building, but behind vehicle road. Remaining part of this section will introduce the detail how vertical and horizontal scene layouts were modeled in CRF.

4.4.2 Long Range Vertical CRF (lrCRF(V))

4.4.2.1 Graph Construction

Let $G_{LV} = (V, E_{LV})$ be a long-range vertical graph over lines. Each line is regarded as one node in G_{LV}, $v \in V$ represents lines extracted from one scan profile, and its node adjacency relation, $e_{LV} \in E_{LV}$ was constructed if a long range vertically neighboring relation is found. The "above-below" relation in the vertical direction between adjacent objects was

considered here. A line finds its long range vertical neighbors upward and downward as shown in Figure 4.3. Rather than using a completely connected graph, a sparse long range graph was constructed, similar with Li and Huttenlocher (2008). After excluding those short range neighboring cells, lines with the two nearest (both upward and downward) were selected as its long range vertical neighbors. Thus, the maximum number of long range neighbors corresponds to four (2 upward and 2 downward). Please note that some lines may not have any long range neighbors. The conditional distribution over labels Y given observed data X in G_{LV} can be defined as follows:

$$P_{LV}(Y \mid X) = \frac{1}{Z_{LV}(X)} \exp\left(\sum_{i \in V} \lambda A_i(X, y_i) + \beta \sum_{(i,j) \in NLV_i} LV_{ij}(X, y_i, y_j)\right) \tag{4.4}$$

where $A_i(X, y_i)$ is the association potential; $LV_{ij}(X, y_i, y_j)$ is the long range vertical potential that penalizes incorrect spatial arrangement between labels of neighboring nodes; and Z_{LV} is the normalization term.

4.4.2.2 Association and Interaction Potential

To compare the performance of local classifier versus CRF model, and effect of different context, the association term of each CRF model used prediction result from the same local classifier. The log posterior probabilities of GMM classifier was used as the associate potential of each CRF model respectively. Thus, association potential modeling of this lrCRF(V) model and other following CRF models will be explained, details of which can be found in section 4.3.2.

As regards the long range interaction, it encodes the scene layout between objects. Seven classes make forty-nine class pairs, and so forty-nine interaction potentials are needed. It is not effective and reliable to define so many interaction potentials manually based on human knowledge. Therefore, we modeled interaction potentials as posterior of a forty-nine-class classifier, which allows scene layout to be learned statistically from training data. As the frequency of each class pair varies a lot, it generates an unbalanced training data. Discriminative classifier is rather sensitive to training data; thus, we chose a generative classifier to model the long range vertical interaction.

The vertical long-range interaction term was formulated as the log posterior of a multivariate Gaussian classifier, which is described in Equation 4.5.

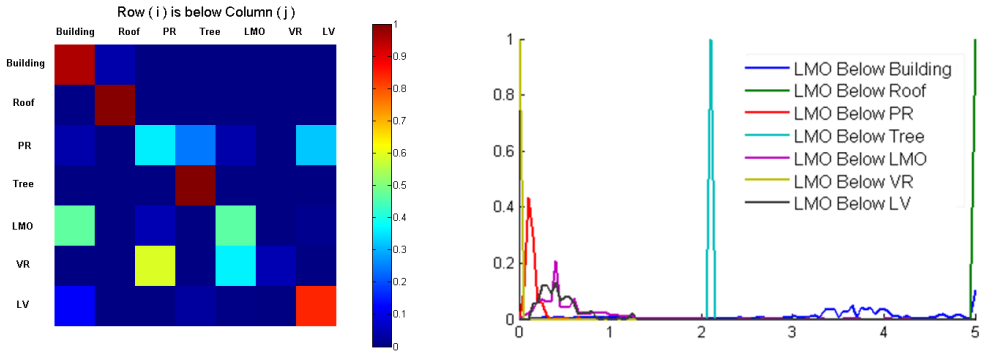
$$LV_{ij}(u_{ij}, y_i, y_j) = \log(P(y_{above} = l, y_{below} = k \mid u_{ij}) = \frac{P(u_{ij} \mid y_{above} = l, y_{below} = k)P(y_{above} = l, y_{below} = k)}{\sum_{y_{above} \in L, y_{bellow} \in L} P(u_{ij} \mid y_{above} = l, y_{below} = k)P(y_{above} = l, y_{below} = k)} \quad (4.5)$$

where(y_i, y_j) is a pair of lines forming an edge in G_{LV}; y_{above} is defined if one of (y_i, y_j) is placed higher than the other, otherwise as y_{below}.

Equation 4.5 estimates the probability of y_{above} labelled as l, given edge feature u_{ij} and y_{below} labelled as k. In Equation 4.5, the prior probability measures what can be found above the given object. $P\ (y_{above}= l, y_{below}=k)$ is the co-occurrence rate of class l that is placed above class k. This prior was calculated over all label pairs, which represents *a priori* knowledge of spatial arrangements between object pairs. This statistically-derived

knowledge was formed in a look-up table shown in Figure 4.4(a). The likelihood in Equation 4.5 is the probability distribution of edge feature u_{ij} given a configuration of that class l is above class k, which quantitatively measures how class l can be found above class k. The edge feature u_{ij} is a six dimension vector, $\{|h_i+h_j|, |o_i+o_j|, |l_i+l_j|, |h_i-h_j|, |o_i-o_j|, |l_i-l_j|\}$, h, mean height; o, orientation; l, length. We assumed that the likelihood follows a multivariate Gaussian distribution (mean vector: $\mu_{l,k}$; covariance matrix $\Sigma_{l,k}$), which described in Equation 4.6. The normalization term is a marginal probability over y_{above}. Figure 4.4(b) gives an example of probability distribution of height difference when one low man-made object (LMO) is below other objects. The x-axis represents the value of the height difference.

$$P(u_{\mathrm{ij}} \mid y_{above} = l, y_{below} = k) = \frac{\exp(-\frac{1}{2}(u_{\mathrm{ij}} - \mu_{l,k})^T \sum_{l,k}{}^{-1}(u_{\mathrm{ij}} - \mu_{l,k}))}{2\pi\sqrt{|\sum_{l,k}|}} \tag{4.6}$$



(a) (b)

Figure 4.4: Prior and likelihood estimation for vertical interaction term. (a) Look-up table: row *i* is below column *j*; (b) probability distribution of height difference when LMO is placed below the other objects.

We used asymmetric pairwise interactive potential to reflect the directional scene layout between adjacent long range objects. Firstly, there is no evidence show that the look-up table is symmetric. Moreover, there is no direct symmetric mathematic relation between likelihoods of symmetric class-pairs. The asymmetric prior and likelihood finally generates asymmetric long range potential $L_{ij}(x, y_i, y_j) \neq L_{ji}(x, y_j, y_i)$. With the asymmetric interaction potential design, when node *i* is above node *j*, the lrCRF(V) model encourages the configuration of $\{y_i = building, y_j = LMO\}$, but penalizes the configuration $\{y_j = building, y_i = LMO\}$.

4.4.3 Long Range Horizontal CRF (lrCRF(H))

4.4.3.1 Graph Construction

Let G_{LH} = (V, E_{LH}) be a long-range horizontal graph over line segments. Each line is regarded as one node in G_{LV}, $v \in V$ represents lines extracted from one scan profile, and its node adjacency relation, $e_{LH} \in E_{LH}$ was constructed if a long range horizontally neighboring relation is found.The scene layout in horizontal direction between adjacent object is considered as "front-and-behind" relation respect to the distance to laser scanner center. Long range horizontal neighbors were searched both forward and backward, as it is shown in Figure 4.3. To make a sparse connection graph, excluding those short range neighboring cells, only the two nearest (both forward and backward) lines were selected. Similar with long range vertical neighbors, the maximum number of long range horizontal neighbors is four. The conditional distribution over labels Y given observed data X in G_L can be now defined as below:

$$P_{LH}(Y \mid X) = \frac{1}{Z_{LH}(X)} \exp(\lambda \sum_{i \in V} A_i(X, y_i) + \gamma \sum_{(i,j) \in NLH_i} LH_{ij}(X, y_i, y_j)) \qquad (4.7)$$

where $A_i(X, y_i)$ is the association potential; $LH_{ij}(X, y_i, y_j)$ is the long range horizontal potential that penalizes incorrect scene layout compatibility in horizontal direction, such as building is closer to laser scanner than tree. Z_{LH} is the normalization term.

4.4.3.2 Association and Interaction Potential

The lrCRF(H) model shares the same association term with srCRF and lrCRF(V), details of which can be found in section 4.3.2. As regard the horizontal long-range interaction, it was also designed as the log posterior probability of a forty-nine-class classifier. The interaction potential design is showed in Equation 4.8.

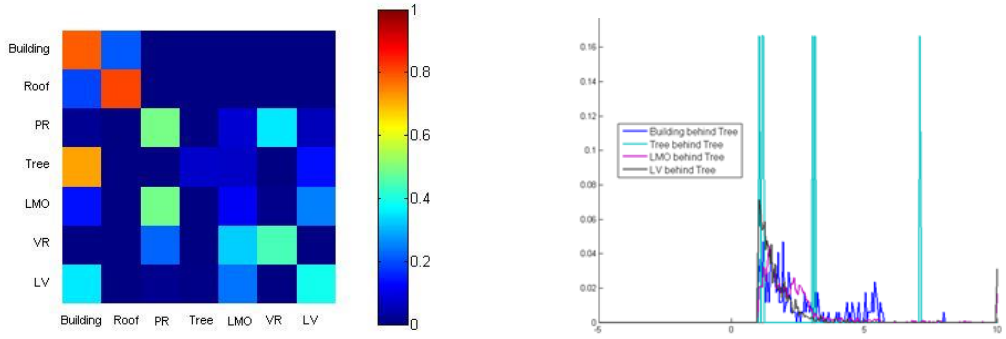
$$LV_{ij}(v_{ij}, y_i, y_j) = \log(P(y_{behind} = l, y_{front} = k \mid v_{ij}) = \frac{P(v_{ij} \mid y_{behind} = l, y_{front} = k)P(y_{behind} = l, y_{front} = k)}{\sum_{y_{behind} \in L, y_{front} \in L,} P(v_{ij} \mid y_{behind} = l, y_{front} = k)P(y_{behind} = l, y_{front} = k)} \quad (4.8)$$

where(y_i, y_j) is a pair of lines forming an edge in G_{LH}; y_{front} is defined if one of (y_i, y_j) is placed closer than the other, otherwise as y_{behind}. The equation 4.8 estimates the probability of y_{front} labelled as l, given edge feature u_{ij} and y_{behind} labelled as k. In Equation 4.8, $P\ (y_{\text{behind}} = l, y_{front} = k)$ models a co-occurrence rate of class l that is placed behind of class k. The same as vertical context, prior was calculated over all label pairs and its look-up table is shown in Figure 4.5 (a). The likelihood in Equation 4.8 is the probability distribution of edge feature u_{ij} given a configuration of that class l is behind class k. The design of horizontal edge feature is the same as vertical edge features, $v_{ij} = \{|r_i+r_j|, |o_i+o_j|, |l_i+l_j|, |r_i-r_j|, |o_i-o_j|, |l_i-l_j|\}$, r, range; o, orientation; l, length. The likelihood is estimated using multivariate Gaussian distribution (mean vector: $\mu_{l,k}$; covariance matrix $\Sigma_{l,k}$) described in Equation 4.9. The normalization term is a marginal probability over y_{above}. Figure 4.5 (b) shows the distribution of range difference when one object is behind the

tree; the x axis represents value of range difference. If a class is not found behind tree, such as building, its distribution can be seen.

$$P(u_{ij} \mid y_{behind} = l, y_{front} = k) = \frac{\exp(-\frac{1}{2}(u_{ij} - \mu_{l,k})^T \Sigma_{l,k}^{-1}(u_{ij} - \mu_{l,k}))}{2\pi\sqrt{|\Sigma_{l,k}|}} \quad (4.9)$$

The long range horizontal interaction term is also asymmetric. For example, given the condition that node *i* is closer than node *j*, the asymmetric potential encourages the configuration of $\{y_i = tree, y_j = building\}$ but penalizes the configuration of $\{y_i = building, y_j = tree\}$.



(a) (b)

Figure 4.5: Prior and likelihood estimation for horizontal interaction term. (a) Look-up table: row *i* is in front of column *j*; (b) probability distribution of range difference when other objects is placed behind tree.

4.5 Multi-Range CRF

The short range and long range context, in vertical and horizontal direction provide different contextual information on different scales. Each single context contains partial contextual information, so relying only on a single context could be risky as "part of the evidence is spent to specify the model" (Leamer, 1978). It is promising to combine all three types of contextual information together. The combination can 1) maximize the smoothness between short-range nodes; 2) maximize the regularity of spatial dependency between objects in long range nodes; and 3) consider asymmetric properties of scene layout regularity. In this research, two multiple range context combination strategies were adopted, the product combination of multiple CRF classifiers, a single CRF model with multiple range.

4.5.1 Product Combination of Multiple CRF Classifiers

Combining the predictions of different classifiers could significantly improve classification performance. There exist two combination rules, average combination and product combination (Kittler, et al., 1998). The posterior of average combination is computed by averaging the estimated posterior probabilities of multiple classifiers. Average combination is simple to implement and has already been proven effective (Taniguchi and Tresp, 1997). Unlike the averaging combination, the product combination is based on a Bayesian foundation, so it is more robust. Product combination makes independent assumptions between classifiers and estimates posterior by multiplying the posteriors of these classifiers (Tax, et al., 2000).

Supposing that there exist K classifiers and L possible class labels, the product combination rule is defined as follows:

$$p(y = l \mid x) = \frac{\prod_{k=1}^{K} p(y = l \mid x, M_k)\, p(M_k \mid x)}{\sum_{l=1}^{L} \{\prod_{k=1}^{K} p(y = l \mid x, M_k)\, p(M_k \mid x)\}} \tag{4.10}$$

where M_k is the *k-th* classifier, x is the feature vector of new instance and y is corresponding class label. The *k-th* classifier M_k produces posterior $P(y=l|x, M_k)$ and $p(M_k|x)$ is the prior of M_k. Here, the influence of each classifier was set equally and the product rule can be re-written as

$$p(y = l \mid x) = \frac{\prod_{k=1}^{K} p(y = l \mid x, M_k)}{\sum_{l=1}^{L} \prod_{k=1}^{K} p(y = l \mid x, M_k)} \tag{4.11}$$

Thus, to combine the effect of multi-range contexts, label predictions from three CRF models were firstly individually estimated, and the final posterior probability was accomplished by combining their prediction results as follows:

$$P_M(Y \mid X) = \frac{P_S(Y \mid X) P_{LV}(Y \mid X) P_{LH}(Y \mid X)}{\sum_{Y \in L} P_S(Y \mid X) P_{LV}(Y \mid X) P_{LH}(Y \mid X)} \tag{4.12}$$

where $P_S(Y|X)$, $P_{LV}(Y|X)$, $P_{LH}(Y|X)$ are the posterior of srCRF, lrCRF(V) and lrCRF(H) model respectively, and $P_M(Y|X)$is the final posterior.

4.5.2 Single Integrated Model

The assumption of the product combination is that the three classifiers are independent. However, because the three CRF models were learned from the same training data, they couldn't be absolutely independent. Moreover, since the three CRF classifiers share the same association term, the correlation of them is not weak. Thus, it is not appropriate to enforce an independent assumption on these three CRF classifiers.

Another issue is that product combination decreases the influence of interaction terms. Posterior probability of the product combination is showed in Equation 4.13. The expansion form of the Equation 4.13 can be written as Equation 4.14. From the Equation 4.14, it is clear that the combination classifier relies heavily on the association term, while influence of interaction terms decreases relatively.

$$P_M(Y \mid X) = P_S(Y \mid X)P_{LV}(Y \mid X)P_{LH}(Y \mid X) \tag{4.13}$$

$$\begin{aligned} &P_M(Y \mid X) \\ &= \exp((\lambda_1 + \lambda_2 + \lambda_3) \sum_{i \in V} A_i(X, y_i) + \alpha \sum_{(i,j) \in NS} S_{i,j}(X, y_i, y_j) + \beta \sum_{e(i,j) \in N_{LV}} LV_{i,j}(X, y_i, y_j) \\ &+ \gamma \sum_{e(i,j) \in N_{LH}} LH_{i,j}(X, y_i, y_j)) / Z_S(X) Z_{LV}(X) Z_{LH}(X) \end{aligned} \tag{4.14}$$

To overcome this limitation of product combination, a multi-range asymmetric CRF model (maCRF) was developed, which integrated multi range object context in a single CRF model, including short range smoothness constraint, and long range scene layout both in vertical and horizontal direction. The maCRF model incorporates appearance, local smoothness and global scene layout in a single unified model, which is defined as follows:

$$P(Y \mid X) = \frac{1}{Z(X)} \exp[\lambda \sum_{i \in V} A_i(X, y_i) + \alpha \sum_{se=(i,j) \in E_S} I^S_{se}(X, y_i, y_j) + \beta \sum_{lve=(i,j) \in E_{LV}} I^{LV}_{lve}(X, y_i, y_j) + \gamma \sum_{lhe=(i,j) \in E_{LH}} I^{LH}_{lhe}(X, y_i, y_j)] \quad (4.15)$$

where X is the entire observation and Y is the global label configuration. V is the set of nodes. E_s, E_{LV}, E_{LH} are the sets of short range edges, vertical long range edges and horizontal long range edges respectively. A, I^S, $I^{LV,}$ $I^{LH,}$ are the short range potential, vertical long range potential, and horizontal long range potential respectively, while λ, α, β, γ are corresponding weights. A_i measures the likelihood of node i belong to certain class. I^S is the pairwise potential and makes a local smoothness constraint. I^{LV} is the pairwise compatible potential and makes scene layout constraint in vertical direction and I^{LH} makes constraints in horizontal direction.

In this jointly integrated model, multi-range neighborhood searching is the same as in each single range CRF model. Association term A_i is still the log posterior of local

classifier. Design of the three interaction terms I^S, I^{LV}, I^{LH} are the same as those in individual CRF model.

4.6 Training and Inference of CRF

Parameters in CRF models can be learned by maximizing the conditional likelihood of true class labels given the training data. However, because partial derivative is a non-linear function with respect to each parameter, direct maximization of likelihood cannot provide a closed form for CRF model learning. Instead, iterative numerical optimization techniques, like gradient descent, are popularly used to find the local maximum conditional likelihood. At each iteration, parameters are updated on the basis of the gradient. In practice, the gradient requires some manual adjustment and this adjustment is called learning rate (or step size). Choosing a proper learning rate and schedule is rather difficult. In practice learning rate is either set to a small enough constant value that gives stable convergence, or adaptively updated as learning progresses that makes cost function converge faster (Bowling and Veloso, 2002). In this research, we prefer to get a stable convergence result and so finally chose a small constant learning rate, 0.0001.

Traditional gradient descent computes the gradient using the whole dataset, which is also called batch gradient descent. Since batch method uses the "true" gradient direction for parameter update, it moves directly towards an optimum solution, either local or global. As batch method has to scan through the entire training set before taking a single step, it is not computationally efficient to train a large and redundant dataset. As an alternative to reduce computation complexity, Besag (1986) proposed a pseudo-likelihood estimation method, which used parameter learning of markov random field as

an example. The pseudo-likelihood normalizes over the possible labels at each node, rather than directly maximizing the conditional likelihood over entire image. Entire training dataset can be divided into small pieces and each piece is trained independently. Instead of updating parameters until they have scanned the entire training set, it takes a small step in the direction given by the gradient of one piece only, thus it converges faster. When the amount of training data tends to infinity, the pseudo-likelihood coincides with that of the "true" likelihood (Winkler, 1995).

As a variant of pseudo-likelihood estimation, the stochastic gradient descent (SGD) method is an alternative for parameter estimation of CRFs (Vishwanathan et al., 2006). SGD updates parameter after looking at a randomly selected subset of the training set, thus the stochastic gradient descent (SGD) algorithm is a drastic simplification. The SGD approximation speeds up the convergence and make training more efficient even on large and redundant data sets. It makes a balance between convergence quality and speed. In order to accelerate the parameter learning training, Vishwanathan et al. (2006) also used the gain vector adaptation, and experimental results validated its advantages. Thus, parameter estimation of all CRF models in this research adopted the SGD algorithm following (Vishwanathan et al., 2006). As all the four CRF models were built in single scan profile, parameters were updated when a randomly selected scan profile was scanned.

There are two types of parameters in each of the four CRF models: parameters in each potential term, and parameters weighting the relative influence of potential terms. As both association term and long range interaction terms are complex quadratic

functions, learning all of the parameters simultaneously in each of four proposed CRF models using SGD is still a challenge. Some previous research simplified the parameter learning by assuming each potential term has equal influence, and assigning them with equal weight (Shotton et al., 2006; Rabinovich et al., 2007; Gould et al., 2008). Instead of making such an ad-hoc assumption, the two-stage training is a more commonly used that parameters of each potential term and weights of potentials are is separately learned. This method does not guarantee an optimal estimation, but it usually archives satisfying estimation (Lafferty et al., 2001; He, et al., 2004; Yang, et al., 2010). In this research, parameter learning in CRF model was divided into two stages. At first, parameters in association and each interaction terms were learned individually, following which the weights of association and interaction terms were learned.

4.6.1 Training the Association and Interaction Potentials

As regards association term, parameters of GMM were estimated using EM algorithm, detail of which can be found in section 3.3.2. Because short range interaction terms in srCRF were designed as exponents of the Potts model that is an identical matrix, no parameter needs to be learned. As long range vertical and horizontal interaction terms were designed as log posterior of a forty-nine-class multivariate Gaussian classifier, prior and likelihood need to be estimated. The prior was obtained from a look-up table, which is a frequency table of co-occurrence rate over forty-nine class pairs. The likelihood was designed as a multivariate Gaussian distribution, parameters of which were estimated using classic Maximum Likelihood (ML) algorithm.

4.6.2 Training the Weight of Potential Terms

Once all parameters in association and interaction terms are known, the weight of each term $\{\lambda, \alpha, \beta, \gamma\}$ can be learned using stochastic gradient descent. The joint conditional probability (likelihood function) over the whole training sample is given in Equation 4.16. Optimal parameters can be found by maximizing the likelihood function. Due to the monotonic property of logarithm, the log-likelihood function has the same maximizing argument with original likelihood function. Because maCRF model integrates interaction terms from other three single range CRF models, weight estimation of maCRF was used as an example. Log-likelihood function of the maCRF is written as:

$$\begin{aligned} L(\Theta) &= \log P(Y \mid X) \\ &= \lambda \sum_{i \in V} A_i(X, y_i) + \alpha \sum_{se=(i,j) \in E_S} I_{se}^{S}(X, y_i, y_j) + \beta \sum_{lve=(i,j) \in E_{LV}} I_{lve}^{LV}(X, y_i, y_j) \\ &+ \gamma \sum_{lhe=(i,j) \in E_{LH}} I_{lhe}^{LH}(X, y_i, y_j) - \log[Z(X)] \end{aligned} \tag{4.16}$$

Maximum likelihood estimates parameters by differentiating the likelihood function with respect to parameter. Equation 4.17 gives the partial derivative of short range interaction weight.

$$\begin{aligned} g_{\alpha}^{t} &= \frac{\partial L(\Theta^t)}{\partial \alpha} = \sum_{se=(i,j) \in E_S} I_{se}^{S}(X, y_i, y_j) - \frac{1}{Z(X)} * \frac{\partial Z(X)}{\partial \alpha} \\ &= \sum_{se=(i,j) \in E_S} I_{se}^{S}(X, y_i, y_j) - \sum_{y \in L} \sum_{se=(i,j) \in E_S} I_{se}^{S}(X, y_i, y_j) * P(Y \mid X, y_i, y_j, \Theta^t) \end{aligned} \tag{4.17}$$

where g_{α}^{t}, Θ^{t} are respectively the partial derivative of α and parameter set after t updates. The integral of posterior probability $P(Y|X, y_i, y_j, \Theta^t)$ and short range interaction can be regarded the expectation of short range interaction of edge e_{ij} given the parameter Θ^{t} over all possible labeling, which is noted as $E(P(Y|X, y_i, y_j, \Theta^t))$. Computation of this expectation is actually a graph inference is given the current parameters. Thus, the partial derivative of short range interaction weight can be rewritten as follows:

$$g_{\alpha}^{t} = \sum_{se=(i,j)\in E_S} I_{se}^{S}(X, y_i, y_j) \quad - \sum_{se=(i,j)\in E_S} I_{se}^{S}(X, y_i, y_j) E(P(Y \mid X, y_i, y_j, \Theta^{t})) \tag{4.18}$$

It is observed from Equation 4.18 that the computation complexity of gradient is mainly from the computation of expectation of edge interaction. As the SGD calculates gradient from a randomly selected scan profile rather than the whole training data, it greatly reduces the computational complexity. The parameter can be updated when gradient is given as follows:

$$\alpha^{t+1} = \alpha^{t} - g_{\alpha}^{t} * \eta \tag{4.19}$$

To make the log-likelihood function converges stably, the learning rate η should be set small enough. The learning rate was set as 0.0001 for all weights in each CRF model. In a similar way, the gradient of long range vertical and horizontal interaction term can be estimates as follows:

$$g_{\beta}^{t} = \sum_{lve=(i,j)\in E_{LV}} I_{lve}^{LV}(X, y_i, y_j) - \sum_{lve=(i,j)\in E_{LV}} I_{lve}^{LV}(X, y_i, y_j) E(P(Y \mid X, y_i, y_j, \Theta^t)) \quad (4.20)$$

$$g_{\gamma}^{t} = \sum_{lhe=(i,j)\in E_{LH}} I_{lhe}^{LH}(X, y_i, y_j) - \sum_{lhe=(i,j)\in E_{LH}} I_{lhe}^{LH}(X, y_i, y_j) E(P(Y \mid X, y_i, y_j, \Theta^t)) \quad (4.21)$$

The weights $\{\lambda, \alpha, \beta, \gamma\}$ can be scaled up or down, and the scaling does not affect CRF inference. However, to make the weights converge faster, weight of association term was fixed to 1 in all CRF models.

4.6.3 Inference

Once parameter estimation is done, the next step is to find the most likely label configuration Y for given entire observations X and parameter Θ, which is also called inference. Belief propagation (BP) is a message passing algorithm proposed by Pearl (1988) for inference of a graphical model, such as Bayesian Networks (Huang and Darwiche, 1996) and Markov Random fields (Smyth, 1997). The BP algorithm finds marginal distributions over nodes in the graph. It guarantees exact inference when the graph structure is a tree, but possibly does not converge when the graph has loops. The algorithm is then sometimes called "loopy" belief propagation (LBP), because graphs typically contain cycles (or loops), and the LBP has been reported effective in solving graphs with cycles (Murphy et al., 1999). Since all four CRF models in this research have cyclic structure, the LBP was used for inference. Implementation of the LBP algorithm referred to the open source code provided by Schmidt (2007). And the final decision was made on each single node by maximizing the marginal node belief.

LBP works by sending messages along the edges of the graph. Message is the confidence that a node believes one of its neighboring nodes takes certain label. As CRF is an undirected graph, message passes in both directions of an edge. LBP is an iterative algorithm, so messages are updated iteratively until convergence. Any vector of real-valued can be set to initial messages, and a typical method is to assign equal value over all the possible class labels. The messages node *i* sends to node *j* about the confidence that node *i* believe node *j* take a label *l* can be initialize as follows:

$$msg(y_j = l)_{i \to j} = 1/L \tag{4.22}$$

where *L* is The dimension of the message, which is the same as the number of possible class labels. To update message, there exist two strategies, max-product (Pearl, 1988) and sum-product (Mooij, 2007). Weiss (1997) compared the performance of sum-product and max-product on a "toy" turbo code problem, and found that sum-product is signicantly better than max-product when implemented on the nonconvergent cases, because max-product method usually tends to produce a discontinuous gradient estimate. Thus, we used the sum-product update algorithm in this research. The message sent from a node *i* to another node *j* by an edge e_{ij} is updated using sum-product update algorithm as follows:

$$msg(y_j = l)_{i \to j} = \sum_{k \in L} [\varphi_i(y_i = k) \psi_{i,j}(y_i = k, y_j = l) \prod_{p \in N(i) \setminus j} msg(y_i = k)_{p \to i}] \tag{4.23}$$

where Φ_i is the unary factor (association term); $\Psi_{i,j}$ is the pairwise factor (interaction term). When the edge is short range edge, the pairwise factor can be calculated using Equation 4.25.

$$\varphi_i(y_i = k) = \exp[\lambda A_i(X, y_i = k)] \tag{4.24}$$

$$\psi_{i,j}(y_i = k, y_j = l) = \exp[\alpha F^S I_{se}^S(X, y_i = k, y_j = l)] \tag{4.25}$$

To make numerical stability, and to avoid overflow or underflow, the message was normalized to sum to 1.

$$msg(y_j = l)_{i \to j} = \frac{msg(y_j = l)_{i \to j}}{\sum_{l \in L} msg(y_j = l)_{i \to j}} \tag{4.26}$$

Actually, the LBP algorithm does not guarantee that the message converge to a fixed point after any number of iterations. However, under relatively mild conditions, it may guarantee the existence of fixed points. Even if the fixed points not be unique, the LBP still gives a reasonable set of approximations to the correct marginal distributions in practice. In this research, the convergence condition was set as that the sum of absolute difference of old and newly updated messages over the entire graph is small than 0.0001. Moreover, to avoid infinite iterative loops, the maximum iteration number was set as 100. Experiment results showed that only 5 scan profiles was found not converged.

$$\sum_{(i,j)\in G} msg(y_j = l)_{i\to j} < 0.0001 \tag{4.27}$$

Once the message update terminates, the marginal probability (node belief) of each node can be computed by multiplying its own potential with all the messages it receives from its neighbors as follows:

$$nodeBelief(y_i = l) = \frac{\exp[A_i(X, y_i = l)] * \prod_{k\in N(i)} msg(y_i = l)_{k\to i}}{\sum_{l\in L} \exp[A_i(X, y_i = l)] * \prod_{k\in N(i)} msg(y_i = l)_{k\to i}} \tag{4.28}$$

And the final label decision is made by maximizing the node belief.

$$y_i = \arg\max_l nodeBelief(y_i = l) \tag{4.29}$$

4.7 Experiment Results

To evaluate the importance of multi-range contexts, we conducted a comparative analysis of classification results obtained from five different classifiers: 1) local classifier without label interactions, and four CRF models; 2) with short-range interaction (srCRF); 3) with long-range vertical interaction (lrCRF(V)); 4) with long-range horizontal interaction (lrCRF(H)); and 5) with integrated multi-range model (maCRF). The five classifiers were tested on the same experimental data that used in Chapter 3, YV1 and YV2. More detail

about the experimental data can be found in section 3.5.1. The two-fold cross validation was used. For each classifier, model parameters were learned using one of the datasets, while the other site was used for testing the learned classification model. Each CRF model was implemented using Matlab and C++. Implementation of parameter learning and inference referred the UGM code (Schmidt, 2007). Classification performance was measured on each site and then averaged.

Short range, long range vertical and horizontal neighbors were searched for each line within scan profile. Edge number for each type of context is presented in Table 4.1. It shows that short range edge has the largest edge density, followed by long range vertical edge and long range horizontal edge. There were 260579 short range edges extracted between 1056020 lines in the data YV1, and one node has about 2.5 short ranges in average; while one node has about 1.6 long range vertical edges and only 0.2 edges in average. The data YV2 has similar result.

Table 4.1: Total number of the spatial entities extracted from York Village datasets.

Nodes and Edges	**YV1**	**YV2**
Line segment	105,620	100,648
Short range edge	260,579	277,584
Long range vertical edge	158,276	156,023
Long range horizontal edge	18,787	16,463

Short range energy was Potts model, and so there is no parameter need to be trained in the energy term. For long range vertical and horizontal potential, two forty-nine-class classifiers were trained respectively. When parameters in each potential were

known, weight of each term can be learned using SGD algorithm. As it is mentioned in section 4.6.2, the weight of association term was fixed to 1. Figure 4.6 shows weight of each term versus iteration number using SGD algorithm to train maCRF model on data YV1. The vertical axis indicates the weight value and the horizontal axis indicates the iteration number. In this figure, weight of long range vertical interaction (LongRange(V)) converges rather faster, maybe because the number of long range edge is stable across scan profiles. Weight of short range interaction (LongRange) fluctuates in a small shrinking range, which perhaps results from varying short range edge number. Due to small amount of edge, weight of long range horizontal interaction (LongRange(H)) has a large value. Although the weight learning of maCRF did not converged at fixed points, but they were still considered converge since because the fluctuation ranges were rather small.

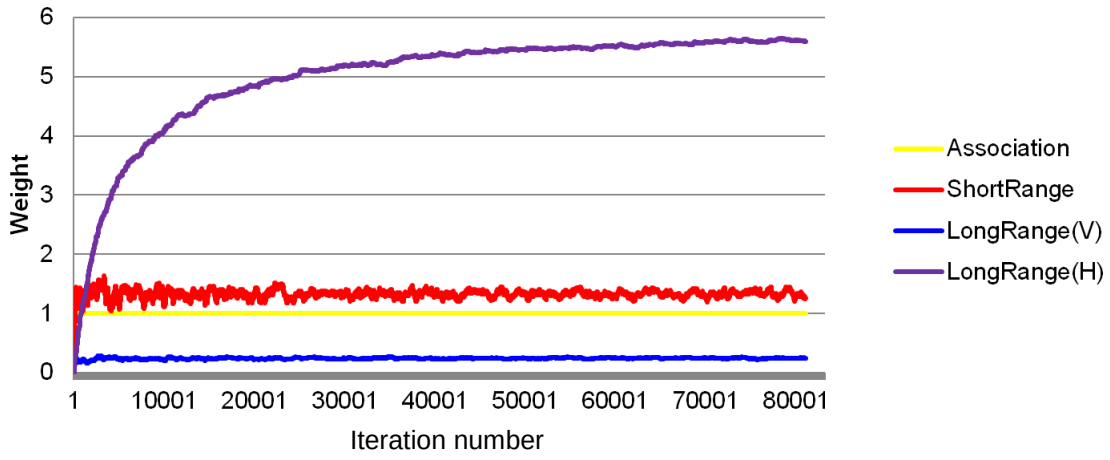


Figure 4.6: Parameter learning of maCRF model using SGD algorithm on data YV1.

4.7.1 Qualitative Analysis

Figure 4.7 presents the classification results of the four CRF models (output of GMM was used as association term) of the data YV2. Generally speaking, the three context-based classifiers achieved better classification quality than GMM classifier (can be found in Chapter 3). Figure 4.7(a) shows the srCRF is able to makes smoothness effect in the local region (e.g., less *salt-pepper* noise in facade region). Figure 4.7(b) shows that the lrCRF(V) rectifies some spatial arrangement errors in vertical direction (e.g., roof is below building as well as tree inside of building). Horizontal context is not that strong as short range context and vertical context, but we still can find some rectification (e.g., most of errors that vehicle road behind pedestrian road were removed) in Figure 4.7(c). Combining contexts of multiple ranges in a single graphical model, the maCRF had the best classification quality.

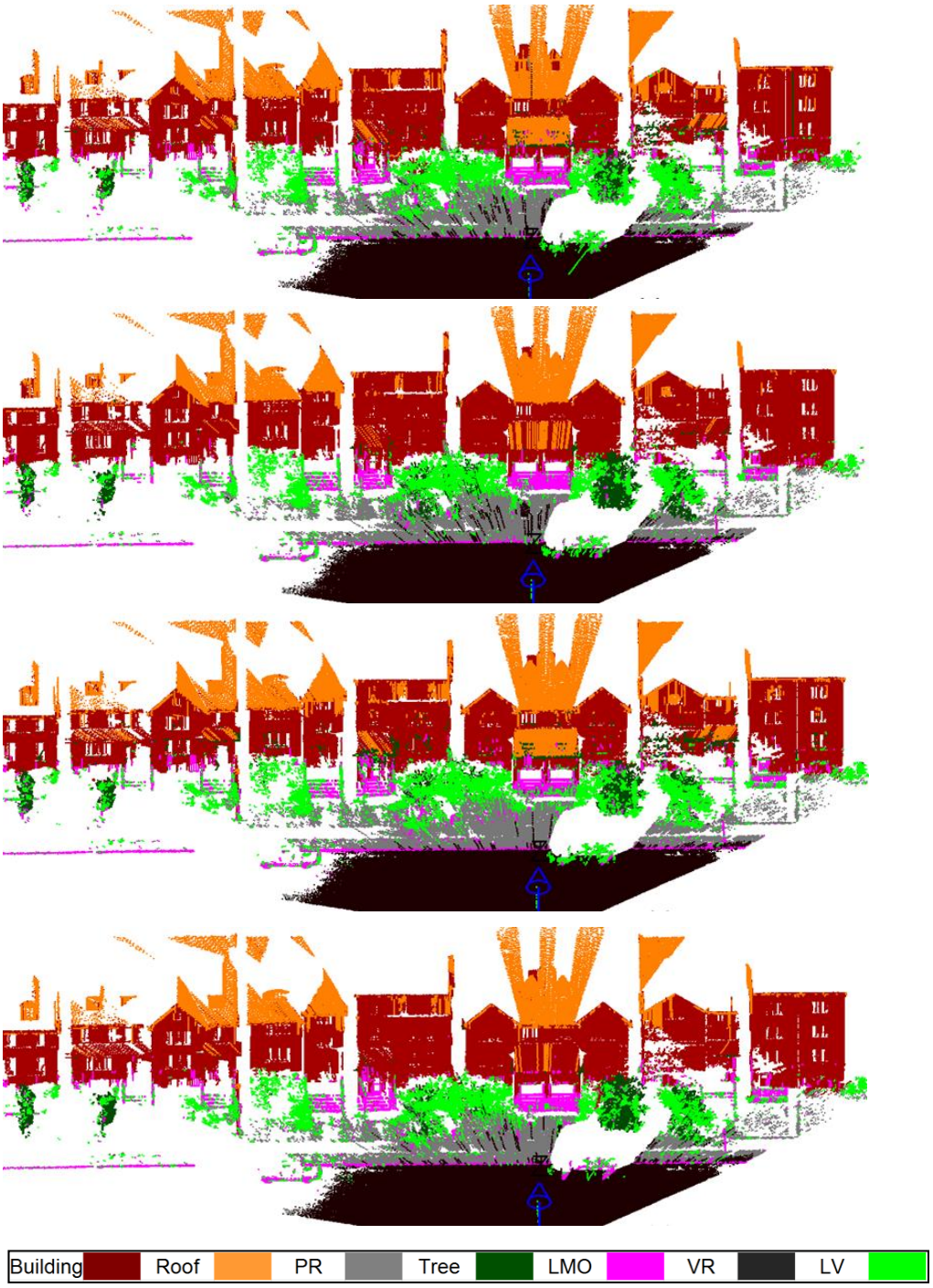


Figure 4.7: Classification result of the four CRFs of the data YV2. (a) srCRF; (b) lrCRF(V); (c) lrCRF(H); (d) maCRF.

To track how the local smoothness constraint and long range scene-layout effect the classification, one representative scan profile selected for comparative analysis, which is showed in Figure 4.8. Compared to the other classifiers, Figure 4.8(e) indicates that maCRF model yields significant improvement in line-based classification compared to the other classifiers. It can be observed in Figure 4.8(a) that GMM-EM produced the largest commission errors between building and tree, building and LMO and tree and LMO. It is clear to see that tree appears inside building and building locates inside a building, which is always called "salt and pepper" noise in image processing. Figure 4.8(b) shows some portion of those commission errors were rectified by srCRF through enforcing local regularities. Benefited from label interaction with many neighbors, noise lines surrounded by dominant neighborhood of tree and building are likely to be effected by local smoothness constraint. However, the smoothness constraint could fails if a misclassified line has only a few short range neighbor or even worse that some lines are isolated because of the occlusion problem. Moreover, the short range interaction makes a local smoothness but it did not work effectively to guarantee global spatial arrangement. For instance, srCRF does greatly rectify the "salt and pepper" noise but still produced spatial arrangement errors such "trees are placed on building façade" and "building are placed at the treetops" (see Figure 4.8(b)). It is noted that the objects in different scan profile could have different appearance and effect of multi-range context on different scan profile are different as well.

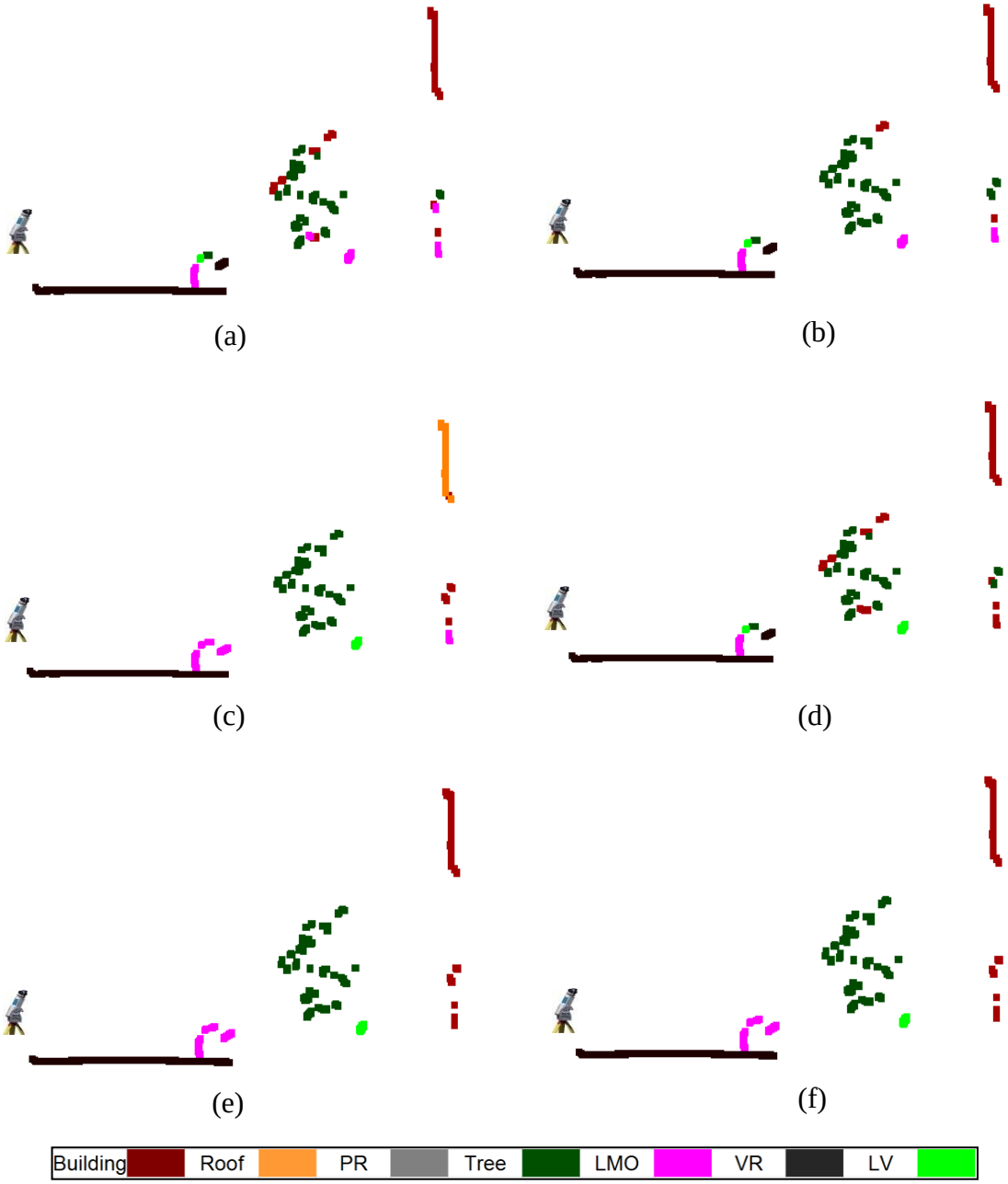


Figure 4.8: Example of single scan profile analysis with different context. (a)-(e) respectively present the classification result of GMM-EM classifier, srCRF, lrCRF(V), lrCRF(H) and maCRF; (f) represents the ground truth.

The long range CRF models assume scene-layout constraints on vertical by considering long range interactions of line segments. lrCRF(V) model makes vertical spatial arrangement constraint and lrCRF(H) model makes horizontal spatial arrangement constraint. As shown in Figure 4.8(c), lrCRF(V) is able to rectify the spatial arrangement error between tree and building by introducing and "above-below" relation prior and feature likelihood of each relation. Horizontal spatial arrangement does not allow tree behind building, so some building lines that misclassified as tree by GMM were rectified by lrCRF(H), which is showed in Figure 4.8(d). However, we also found that long range context was able to rectify some scene layout errors, but failed to correct inconsistence in local region.

So far, single range CRF models have showed their respective benefits and limitations. By considering local smoothness and global scene layout together, the combined maCRF model was expected to makes objects interact simultaneously with their neighbors of multiple ranges. As showed in Figure 4.8(e), maCRF produced the most accurate classification results the result, which is in accordance with the expectation.

In order to examine which classes are sensitive to which type of context, label transition analysis was done. The label transition analysis is based on comparing label change from local classifier to CRF model. There are three types label transitions, *False to False* (local classifier gives false label and CRF models gives another false label), *True to False* (local classifier gives true label and CRF models gives false label) and *False to True* (local classifier gives false label and CRF models gives true label); and they are

marked in blue, red and green respectively in each label transition figure. *False to True* is positive transition and the other are negative transition. The numbers of negative and positive transition from GMM classifier to each CRF classifier over the data YV2 are presented in Table 4.2. Details of label transitions from GMM to each CRF model are showed in Figure 4.9, Figure 4.10, Figure 4.11, and Figure 4.12 respectively.

Table 4.2: Positive and negative transition from GMM to each CRF classifier.

Classifier	Total	Negtive	Positve	Negtive rate	Positve rate
srCRF	11959	5488	6471	45.89	54.11
lrCRF(V)	13438	4501	8937	33.49	66.51
lrCRF(H)	4985	2228	2757	44.69	55.31
maCRF	12247	3061	9186	24.99	75.01

Figure 4.9 presents label transition from GMM to srCRF. The local smoothness constraint works rather well on rectify true low vegetation that misclassified as tree, but not very significantly on other misclassification errors.

Figure 4.10 presents label transition from GMM to lrCRF(V). It is observed that building and tree, low vegetation and tree, roof and building, building and low man-made object, were positively affected by the long range vertical scene layout constraint.

Figure 4.11 presents label transition from GMM to lrCRF(H). Commission errors between tree and low vegetation, pedestrian road and vehicle road, pedestrian road and low vegetation, low man-made object and low vegetation were more sensitive to horizontal scene layout constraint.

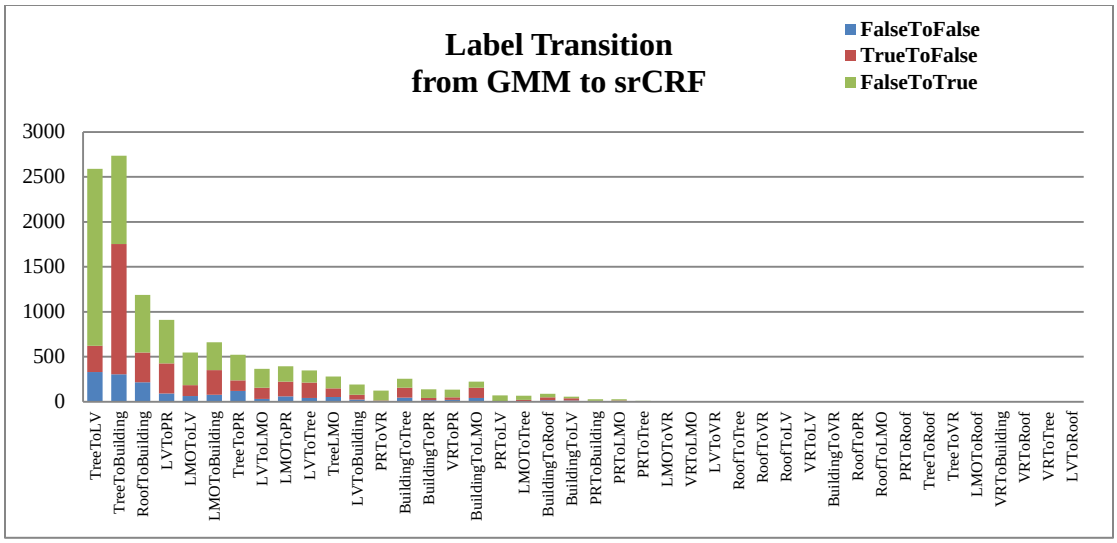


Figure 4.9: Label transition from GMM to srCRF.

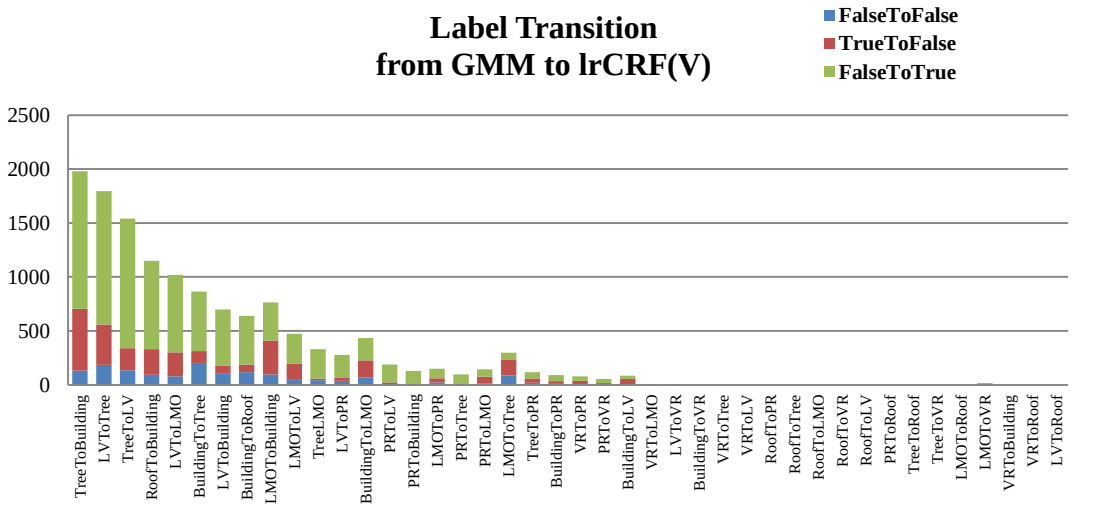


Figure 4.10: Label transition from GMM to lrCRF(V).

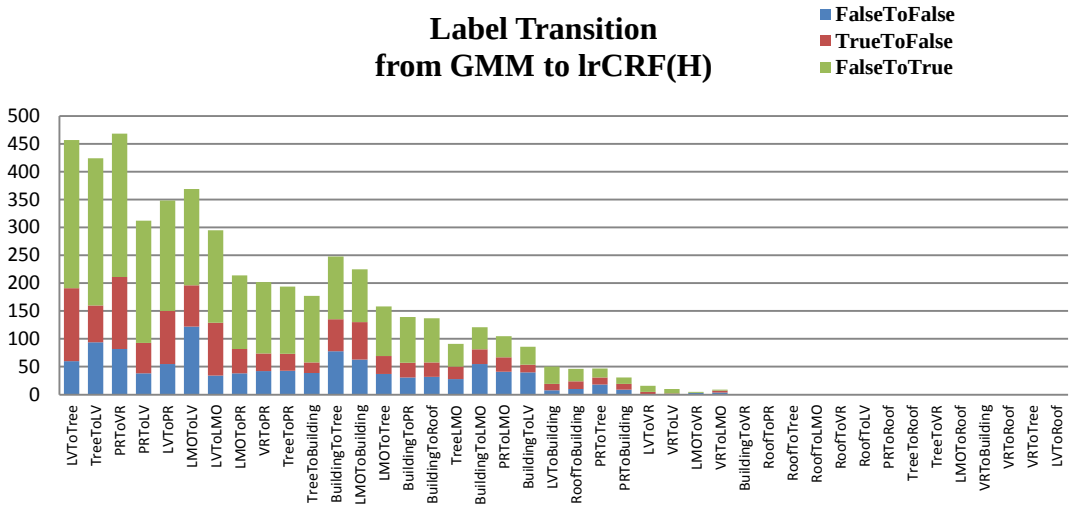


Figure 4.11: Label transition from GMM to lrCRF(H).

Figure 4.12 presents label transition from GMM to maCRF. It is clear to see that positive transition was the dominant label change (75%). Table4.1 shows the total number of label transition of maCRF is less than that of lrCRF(V), but the number of positive transition is more than that of lrCRF(V); and this result validates that by combining multi-range interaction, maCRF is able to integrate advantages of each single range context.

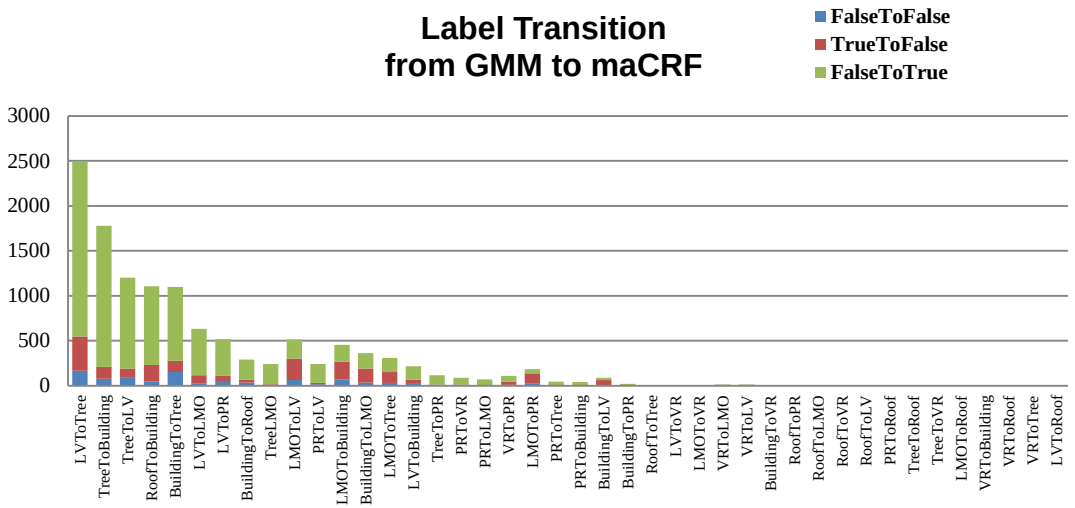


Figure 4.12: Label transition from GMM to maCRF.

4.7.2 Quantitative Analysis

By comparing prediction result and ground truth, confusion matrix was created for each context based classifier. Based on confusion matrix, overall accuracy, per class precision, recall and F1-score were computed. Following the experiment setup of Chapter3, two fold cross validation was used. Test accuracy of GMM and the four CRF models on each data and the averaged accuracy were presented in Table4.3. Confusion matrix of the four CRF models on data YV2 were presented in Table 4.4, Table 4.5, Table 4.6 and Table 4.7 respectively.

From Table 4.3, it is obvious to see the advantage of contextual information; all five contextual classifiers showed higher accuracy than the GMM classifier. By combing multiple range interaction, maCRF improved its classification accuracy by 6.25% compared with GMM. The long range vertical constraints worked better than long range

horizontal constraints. One possible reason is that the placement of objects in horizontal direction is not stable. For example, both curb and garbage bin are low man-made object, but curb is in front of pedestrian road and garbage bin is behind of pedestrian road. Another reason is from the nature of single view laser scanning that if there is an object already reflect laser signal back, the laser cannot capture objects behind it, which makes the objects have less connection in the horizontal direction.

Table 4.3: Test accuracy of GMM and the four CRF models.

Classifier	YV1	YV2	Averaged
GMM-EM	79.53	79.98	79.76
srCRF	82.05	81.73	81.89
lrCRF(V)	86.13	85.04	85.59
lrCRF(H)	80.25	80.41	80.33
maCRF	86.51	85.79	86.01

Table 4.4: Confusion matrix of srCRF classifier on data YV2.

		Prediction						
		Building	Roof	PR	Tree	LMO	VR	LV
Ground Truth	Building	36590	1428	65	674	848	0	384
	Roof	1158	2592	1	0	0	1	0
	PR	193	0	13106	99	900	449	2108
	Tree	2404	2	152	7128	198	0	1232
	LMO	751	0	570	113	6181	58	1722
	VR	18	0	784	6	88	6814	69
	LV	276	0	349	805	449	1	9739

Table 4.5: Confusion matrix of lrCRF(V) classifier on data YV2.

		Prediction						
		Building	Roof	PR	Tree	LMO	VR	LV
Ground Truth	Building	37450	1414	4	93	884	0	144
	Roof	590	3162	0	0	0	0	0
	PR	273	0	12272	596	946	404	2364
	Tree	515	1	33	10122	22	0	423
	LMO	580	0	234	30	7410	53	1088
	VR	9	0	730	22	91	6852	75
	LV	596	0	98	2396	328	0	8201

Table 4.6: Confusion matrix of lrCRF(H) classifier on data YV2.

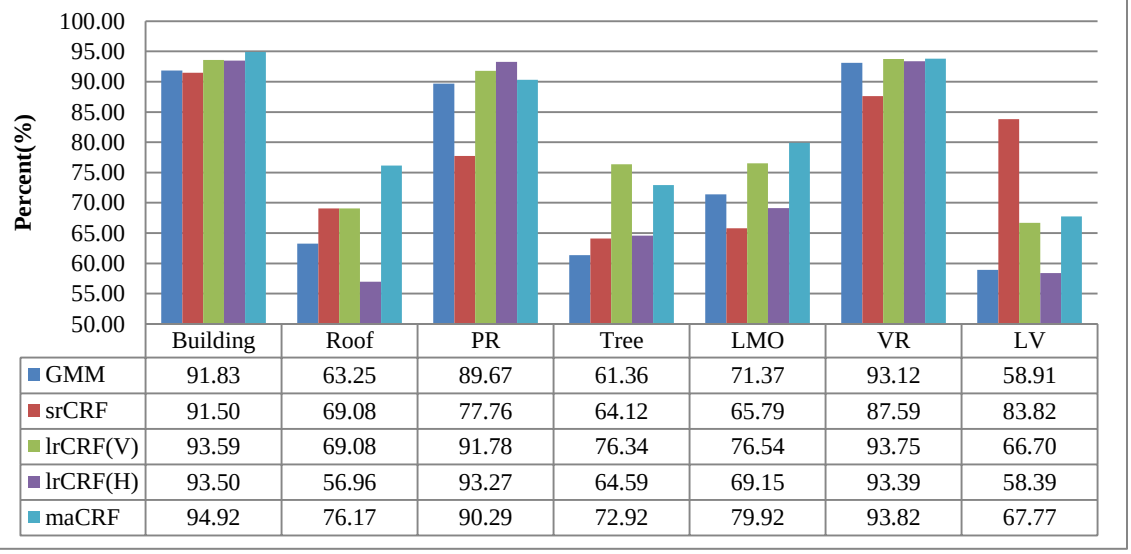
		Prediction						
		Building	Roof	PR	Tree	LMO	VR	LV
Ground Truth	Building	34735	2285	15	1622	935	0	397
	Roof	724	3028	0	0	0	0	0
	PR	100	0	12429	291	998	432	2605
	Tree	912	3	29	9167	172	0	833
	LMO	519	0	245	365	6286	70	1910
	VR	1	0	503	1	169	7092	13
	LV	157	0	105	2746	531	0	8080

Table 4.7: Confusion matrix of maCRF classifier on data YV2.

		Prediction						
		Building	Roof	PR	Tree	LMO	VR	LV
Ground Truth	Building	38047	917	5	159	543	0	184
	Roof	815	2937	0	0	0	0	0
	PR	158	0	13069	813	700	376	1831
	Tree	240	2	7	10498	56	0	320
	LMO	630	0	266	62	6929	63	1444
	VR	6	0	1008	0	65	6682	57
	LV	189	0	119	2865	377	1	8065

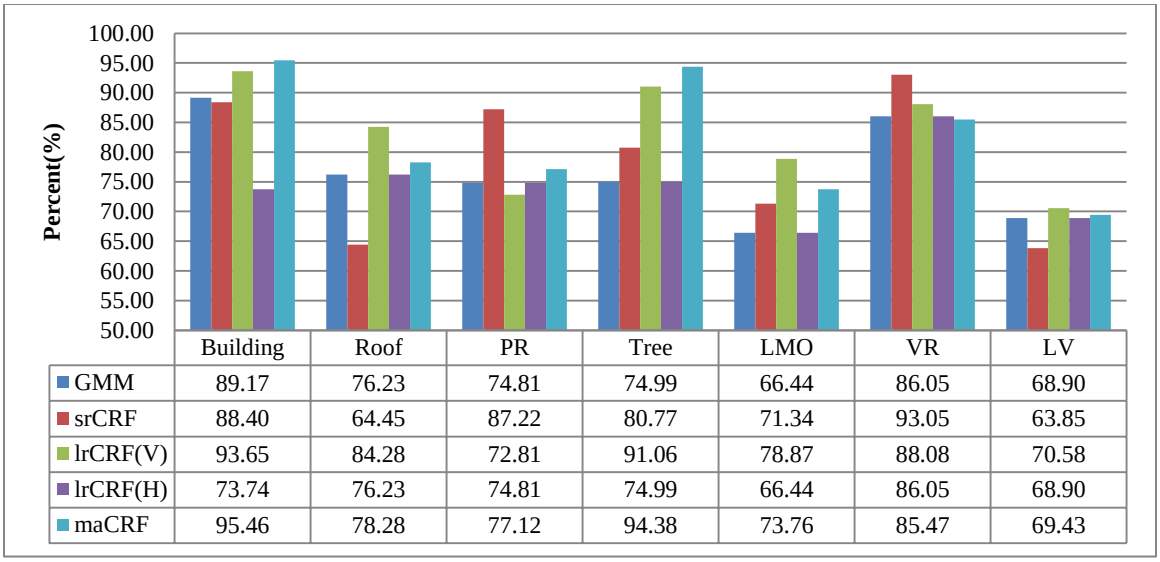
Figure 4.13, Figure 4.14 and Figure 4.15 present precision, recall, and F1-score respectively on the data YV2. Precisions of roof and tree benefited most from multi-range

context, improved more than 10% compared with GMM; improvement of other five classes were not significant but still can be observed. Recalls of most objects were improved by multi-range context; but recalls of VR decreased a little. As regards F1-score, all objects had higher value in maCRF than in GMM. It is also observed that maCRF does not guarantee that every class has better performance than that in each single range CRF model. For example, lrCRF(V) produced the best F1-score for tree, rather than maCRF.



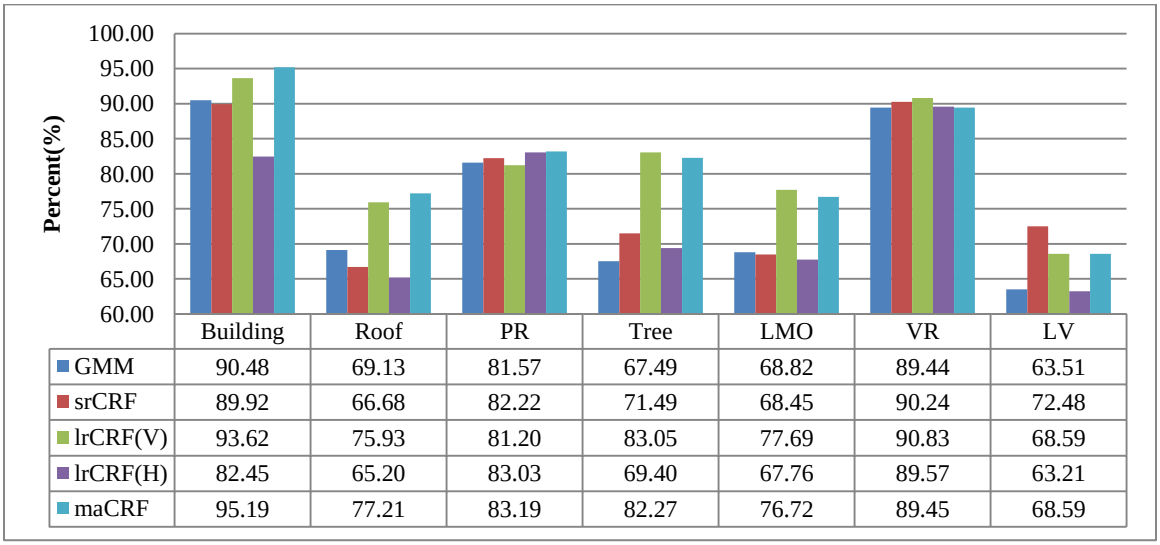
	Building	Roof	PR	Tree	LMO	VR	LV
GMM	91.83	63.25	89.67	61.36	71.37	93.12	58.91
srCRF	91.50	69.08	77.76	64.12	65.79	87.59	83.82
lrCRF(V)	93.59	69.08	91.78	76.34	76.54	93.75	66.70
lrCRF(H)	93.50	56.96	93.27	64.59	69.15	93.39	58.39
maCRF	94.92	76.17	90.29	72.92	79.92	93.82	67.77

Figure 4.13: Precision of each class in five methods.



	Building	Roof	PR	Tree	LMO	VR	LV
GMM	89.17	76.23	74.81	74.99	66.44	86.05	68.90
srCRF	88.40	64.45	87.22	80.77	71.34	93.05	63.85
lrCRF(V)	93.65	84.28	72.81	91.06	78.87	88.08	70.58
lrCRF(H)	73.74	76.23	74.81	74.99	66.44	86.05	68.90
maCRF	95.46	78.28	77.12	94.38	73.76	85.47	69.43

Figure 4.14: Recall of each class in five methods.



	Building	Roof	PR	Tree	LMO	VR	LV
GMM	90.48	69.13	81.57	67.49	68.82	89.44	63.51
srCRF	89.92	66.68	82.22	71.49	68.45	90.24	72.48
lrCRF(V)	93.62	75.93	81.20	83.05	77.69	90.83	68.59
lrCRF(H)	82.45	65.20	83.03	69.40	67.76	89.57	63.21
maCRF	95.19	77.21	83.19	82.27	76.72	89.45	68.59

Figure 4.15: F1-Score of each class in five methods.

4.8 Chapter Summary

In this chapter, limitation of local classifier was first discussed. Then the detail of the proposed multi-range asymmetric CRF model model was given. The proposed maCRF model introduces short-range and long-range (both vertical and horizontal) interactions among labels as well as observed features. By maximizing object label agreement according to the contextual coherence, maCRF model compensates for ambiguity in local appearance of objects. Classification performance of GMM-EM, single range context based CRF models (srCRF, lrCRF(V) and lrCRF(H)), and the multi-range integrated maCRF were evaluated. Our experimental results showed that maCRF performed the best, which validates the advantages of multi-range context constraints.

The proposed maCRF model considers multi-range contexts only in each scan profile, but neglect the contextual information between adjacent scan profiles. In the next chapter, the contextual information across scan profiles will be discussed.

Chapter 5
Across Scan Profile Conditional Random Filed

In the chapter 4, the maCRF model only considers object contexts along scan profile, but neglects the dependency between objects at neighboring scan profiles. It assumes that lines at adjacent scan profiles are independently, which is not coincident with the actual facts. Because of the sweeping nature of laser scanning, the sequentially acquired TLS data has strong spatial dependency, which can provide additional contextual information. Thus, we propose the across scan profile multi-range asymmetric CRF model (amaCRF), which is built over every three consecutive scan profiles. The amaCRF model is an extension of the previous maCRF model by introducing an additional across scan profile context enforces local homogeneity constraints on lines at adjacent scan profiles. Finally we proposed a sequential classification strategy that allows contextual information propagate through adjacent scan profiles, which is called amaCRF+. Along the sweeping direction, amaCRF models are sequentially constructed, and the posteriors of the previous amaCRF are used as association term of the next amaCRF model; thus posteriors of those lines at overlapping scan profiles of the two amaCRF can be sequentially updated.

Three additional experiments were implemented. In order to validate that the multi-range context is independent with association terms, output of SVM is tested as association term. To validate that the algorithm does not only work on a specific scene,

data collected at York Blvd was also tested. And finally, classifiers trained from York Village dataset were tested on the York Blvd dataset.

5.1 Context between Scan Profile

Label propagation has attracted much attention to object recognition from video sequence. Because of the strong correlation between consecutive frames, priors of object context are possible to be propagated from some early observed frame to other late observed frames. Semi-supervised method is often used for label propagation in video sequences, and the propagation engine can be invoked by a few manfully labelled frames. Zhu and Ghahramani (2002) first formulated the label propagation problem as a problem of assigning soft labels to nodes of a fully connected graph with few labelled nodes; labels were propagated with a combination of random walk and clamping. In contrast, as a sequential data, the label propagation problem is more naturally modelled using directed graphs, such as Hidden Markove Random Field (Badrinarayanan, et al.， 2010; Vijayanarasimhan and Grauman, 2012).

As semi-supervised label propagation methods usually require an amount of hand labeled as input, the propagation result is highly dependent on the input labels and there is no guarantee to an optimal result (Vijayanarasimhan and Grauman, 2012). Therefore, many researchers turn to find automatic inference solutions without human intervention. In Yang and Rosenhahn (2014), trajectory of foreground object (human being, animal, etc.) in video image was defined as a sequence of space-time points. A spatial-temporal graph was formulated over pixels in the same frame and trajectories across frames.

Trajectory clustering potentials in the spatial-temporal CRF model was designed as Laplacian matrix to encourage coherent labeling of trajectories across neighboring frames.

As the sequential acquisition nature of laser scanning data, label also can be propagated both in the spatial and temporal domain. In Vale and Mota (2004), acquisition of airborne LiDAR data was treated as a set of sequentially collected vertical sweep; it detects the power line anomaly based on the assumption that power line points "grows" along the direction that airplane moves and potential anomaly is found when power line points tracking across vertical sweeps fails. The detection by tracking method belongs to template matching, so that the final results needs additional manual intervention. In Stamos et al. (2012), each vertical scan profile was considered as a stream of observation and points were sequentially connected by from top to down. A three state (vertical object, horizontal object and vegetation) HMM model was built based on the assumption that label transits from one state to another can be characterised a shift pattern of surface normal. However, the label propagation was only implemented along scan profile.

Label propagation in this research is closer to Vale and Mota (2004) because we consider the label consistence across scan profiles. It is assumed that object "grows" along the direction that laser scanner sweeps and forces neighboring lines at neighboring scan profiles to have the same class label. By considering TLS data as a sequence, contextual information propagates through adjacent scan profiles.

5.2 Across Scan Profile CRF Model

In Chapter 3 and 4, entire laser scanning data was split into a set of sequentially observed scan profiles. The space width of each scan profile can be defined as follows:

$$d = \pi r\theta \tag{5.1}$$

where r is the distance between laser scanner and objects; θ is the angular width of one scan profile. The space width is not a fixed value but proportional with the distance r. Take distance in 50m away for example, the space width can be calculated as: $d_{20m} = 3.14 * 50m * (0.05/180) = 0.044m = 4.4cm$ .The ranges of most objects in the experimental data are within 50m, thus maximal space width of each scan profile is less than 4.4cm, which is rather small compared with urban objects size. The space width of each scan profile is so small that objects can "grows" along the direction that laser scanner sweeps.

5.2.1 Graph Construction

The proposed across scan profile multi-range asymmetric CRF (amaCRF) is an extension of the previous maCRF model, details of which can be found in Chapter 4. The amaCRF model is built over every three consecutive scan profiles. Following the line adjacent graph constructing method in Chapter4, adjacent relations of lines were created with assistance of grid system, which is depicted in Figure 5.1. Let $G_A = (V, E)$ be an undirected graph, each of which node $v \in V$, which represents line sets from the three consecutive scan profiles. There are four types edges, short range edge (e_S), long range vertical edge (e_{LV}), long range horizontal edge (e_{LH}) and across scan profile edge (e_A). In Figure 5.1, the four types of edges are marked using red, blue, purple and black respectively. The first three types of edges are line relation along scan profile, and they

were constructed using the same method as described in Chapter 4. Construction of across scan profile edge will be introduced in the following paragraph.

To find out across scan profile edge, the grid system was used. Because the space width of scan profile is rather small, grid system of neighboring scan profiles can be regarded as the same. Suppose there is line *l* at *s-th* scan profile (middle) and it passes the cell [*i, j*] (yellow), cells at corresponding position and their 4-connected neighborhood in the previous (left) and the following (right) scan profile are considered as across scan profile cell neighbors (black). Lines pass these cells are across scan profile neighbors of the line *l*. Figure5.2 presents an example of the line adjacent graph of the amaCRF model over three consecutive scan profiles.

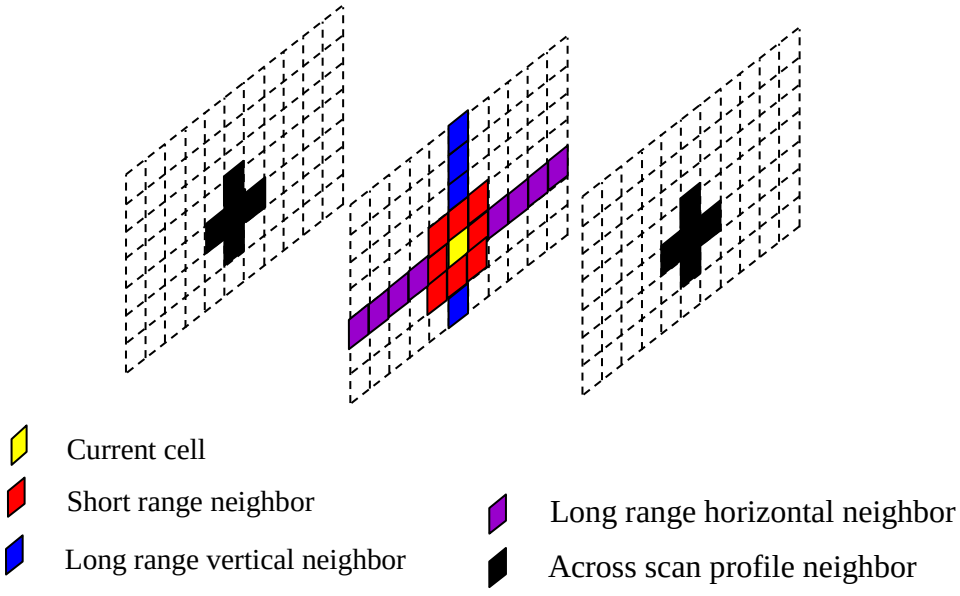


Figure 5.1: Across and along scan profile neighborhood.

Given the across scan profile graph, the conditional distribution over the labels Y given observed data X in the graph G_A can be defined as follows:

$$P(Y \mid X) = \frac{1}{Z(X)} \exp[\lambda \sum_{i \in V} A_i(X, y_i) + \alpha \sum_{se=(i,j) \in E_S} I^S_{se}(X, y_i, y_j) + \beta \sum_{lve=(i,j) \in E_{LV}} I^{LV}_{lve}(X, y_i, y_j) + \gamma \sum_{lhe=(i,j) \in E_{LH}} I^{LH}_{lhe}(X, y_i, y_j) + \delta \sum_{ae=(p,q) \in E_A} I^A_{ae}(X, y_p, y_q)] \quad (5.2)$$

where X is the entire observation and Y is the entire label configuration. E_s, E_{LV}, E_{LH} are sets of short range edges, vertical long range edges and horizontal long range edges respectively within scan profile, while E_A is the set across scan profile edges. A, I^S, I^{LV}, I^{LH}, I^A are the short range potential, vertical long range potential, horizontal long range potential and across scan profile potential respectively. λ, α, β, γ, δ are corresponding weighting coefficients of potential terms.

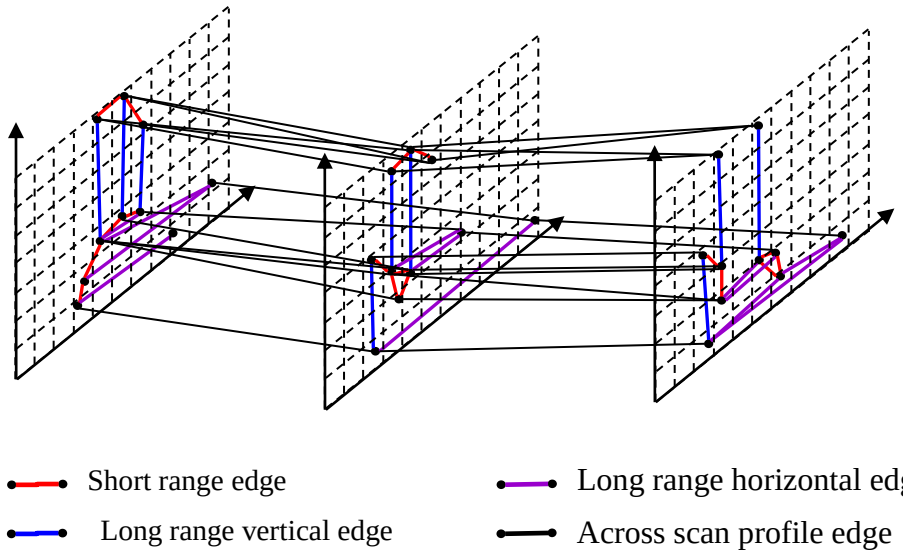


Figure 5.2: Example of across/with scan profile multi-range graph.

5.2.2 Potential Design

There are five potential terms in the Equation 5.1. To make a comparative research, association term, short and each long rang interaction potential keep the same format as they are expressed in the maCRF model. To model the compatibility of lines across scan profiles, the classic Potts model was used. The energy value is set to zero if two neighboring lines are given different labels and set to 1 when they are assigned the same label. For each across scan profile edge connecting two nodes *i* and *j*, the interaction potential is expressed as below:

$$A_{ij}(X, y_i = l, y_j = k) = \begin{cases} 1 & l = k \\ 0 & l \neq k \end{cases} \tag{5.3}$$

5.2.3 Parameter Learning and Inference

There are two types of parameters in the across scan profile maCRF model: parameters in each potential term, and parameters weighting the relative influence of potential terms. We used the same parameter learning strategy as maCRF model used. At first, parameters in association and each interaction potential term were learned individually, following which the weights of these terms were learned. Short range and across scan profiles interaction potentials are non-parametric model. Parameters of long range vertical and horizontal interaction terms were respectively learned using the Maximum Likelihood method. When all parameters in all potential terms are known, the weights of potential

terms $\{\lambda, \alpha, \beta, \gamma, \delta\}$ were jointly learned using the SGD algorithm. The log-likelihood function of the across scan profile maCRF model can be written as

$$\begin{aligned} L(\Theta) &= \log P(Y \mid X) \\ &= \lambda \sum_{i \in V} A_i(X, y_i) + \alpha \sum_{se=(i,j) \in E_S} I_{se}^{S}(X, y_i, y_j) + \beta \sum_{lve=(i,j) \in E_{LV}} I_{lve}^{LV}(X, y_i, y_j) \\ &+ \gamma \sum_{lhe=(i,j) \in E_{LH}} I_{lhe}^{LH}(X, y_i, y_j) + \delta \sum_{ae=(i,j) \in E_A} I_{ae}^{A}(X, y_i, y_j) - \log[Z(X)] \end{aligned} \tag{5.4}$$

In order to make the parameters fast converge to an optimal point, the weight of association term λ is set as 1. SGD is an iterative optimization algorithm, and parameters are updates based on gradients that are computed given current parameters. Equation 5.5 – 5.8 gives partial derivative of each interaction weight. Detail of parameter learning can be found in Chapter 4.

$$g_{\alpha}^{t} = \sum_{se=(i,j) \in E_S} I_{se}^{S}(X, y_i, y_j) \; - E_{P(Y|X, y_i, y_j, \Theta^t)} \sum_{se=(i,j) \in E_S} I_{se}^{S}(X, y_i, y_j) \tag{5.5}$$

$$g_{\beta}^{t} = \sum_{lve=(i,j) \in E_{LV}} I_{lve}^{LV}(X, y_i, y_j) \; - E_{P(Y|X, y_i, y_j, \Theta^t)} \sum_{lve=(i,j) \in E_{LV}} I_{lve}^{LV}(X, y_i, y_j) \tag{5.6}$$

$$g_{\gamma}^{t} = \sum_{lhe=(i,j) \in E_{LH}} I_{lhe}^{LH}(X, y_i, y_j) \; - E_{P(Y|X, y_i, y_j, \Theta^t)} \sum_{lhe=(i,j) \in E_{LH}} I_{lhe}^{LH}(X, y_i, y_j) \tag{5.7}$$

$$g_{\delta}^{t} = \sum_{ae=(i,j) \in E_A} I_{ae}^{A}(X, y_i, y_j) \; - E_{P(Y|X, y_i, y_j, \Theta^t)} \sum_{ae=(i,j) \in E_A} I_{ae}^{A}(X, y_i, y_j) \tag{5.8}$$

Given the parameters, inference was implemented using the LBP algorithm, which is a standard iterative message passing algorithm for graphs with cycles and has been validated effective in Chapter 4.

5.3 Context Propagation through Adjacent Scan Profile

The graph of across scan profile maCRF (amaCRF) model is built over every three consecutive scan profiles, which is depicted in Figure 5.3. In Figure 5.3, each plane represents one scan profile. These amaCRF graphs are independent each other; for example, scan profiles in the amaCRF(1) model do not have any connection with scan profiles in the amaCRF(2) model. Thus, contextual information only can be propagated in scan profiles within the same amaCRF model. Although the amaCRF model considers object context across scan profiles, but each scan profiles only has chance to be connected with its closet neighboring scan profiles.

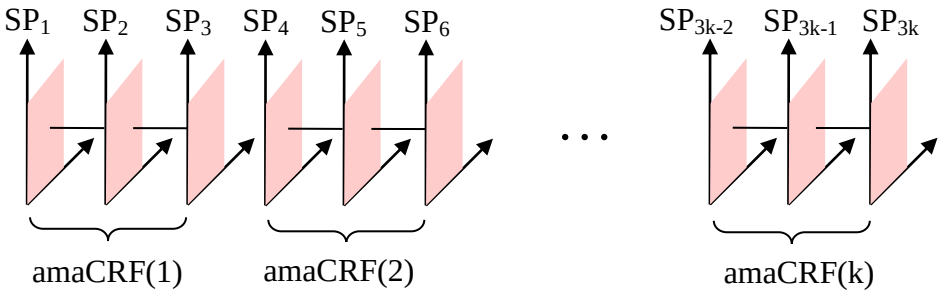


Figure 5.3: Each amaCRF model is independent with each other.

To let contextual information propagated from one scan profile to its neighbors far away, an intuitive way is putting all scan profiles between in one amaCRF. This

method does construct the connection between them, but inference of a large graph will be rather computational complex when the two scan profiles are too far away. Therefore, we proposed a sequential processing method, amaCRF+ to make contextual information flows between adjacent scan profiles. The "sequential" here does not mean the proposed CRF model is a directed graphical model, but means that CRF models are sequentially constructed and output of the previous model is used as the input of the next CRF model. This sequential processing method makes contextual information propagates through adjacent scan profiles by dynamically updating posterior probability. Figure 5.4 depicts the sequential processing method taking the first five scan profiles as an example. We will focus more on how posteriors of lines at the third scan profile are updated.

In Figure 5.4, the color indicates how many times posterior provability has been updated; the darker the color, more times of update is implemented. Before applying CRF model, posterior probabilities of lines at the third scan profile are from a local classifier (e.g., GMM and SVM). At time T1, the first, second and third scan profiles are selected to construct the first amaCRF model, which is noted amaCRF(1,2,3). After implementing the amaCRF(1,2,3) model, posteriors of lines at the third scan profile are updated the first time, and posterior is noted as $CRF^{(1)}$. Then log posterior of $CRF^{(1)}$ is used as association term of the amaCRF(2,3,4) model, and posteriors of lines at the third scan profile are updated again, which is noted as $CRF^{(2)}$. Finally, log posterior of $CRF^{(2)}$ is used as association term of the amaCRF(3,4,5) model and posteriors of lines at the third scan profile are updated the third time, which is noted as as $CRF^{(3)}$. In this manner, outputs of the previous amaCRF model are used as association potential of the next amaCRF model

so that posterior of those common scan profiles of the two amaCRF model can be dynamically updated. It is observed that except for the first two and the last two scan profiles, lines in all other scan profiles have three chances to update posterior probability.

Since the output of amaCRF(1,2,3) contributes to amaCRF(2,3,4), it is conclude that contextual information of the first scan profile is propagated to the fourth scan profile through the second and third scan profiles. The rest can be done in the same manner so that contextual information of the first scan profile can be propagated to the last scan profile.

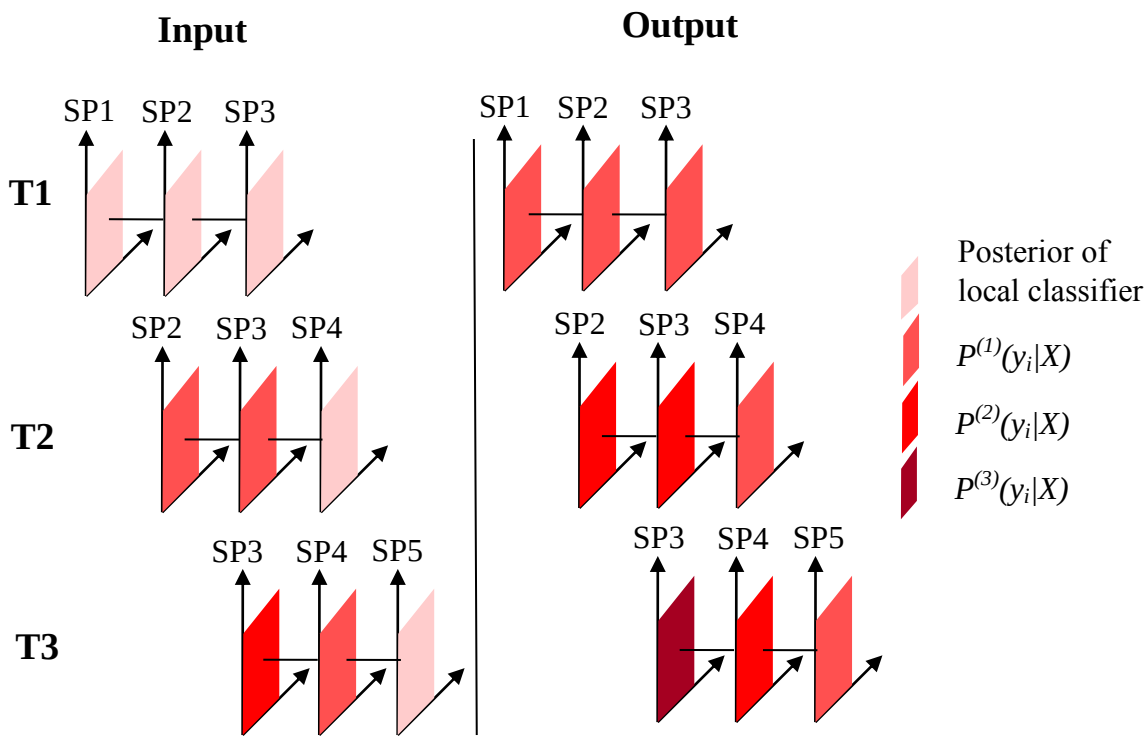


Figure 5.4: Contextual information propagates through adjacent scan profiles.

In most case, there should be one scan profile that already updated two times, one scan profile that already updated one times and one scan profile that never been updated. We can write the posterior of the sequential amaCRF model as follows:

$$
\begin{aligned}
P(Y \mid X) = \exp\{ & \lambda^{(0)} \sum_{i \in V^{(0)}} \log(P^{(0)}(y_i \mid X)) + \lambda^{(1)} \sum_{i \in V^{(1)}} \log(P^{(1)}(y_i \mid X)) + \lambda^{(2)} \sum_{i \in V^{(1)}} \log(P^{(1)}(y_i \mid X)) \\
& + \alpha \sum_{se=(i,j) \in E_S} I^{S}_{se}(X, y_i, y_j) + \beta \sum_{lve=(i,j) \in E_{LV}} I^{LV}_{lve}(X, y_i, y_j) \\
& + \gamma \sum_{lhe=(i,j) \in E_{LH}} I^{LH}_{lhe}(X, y_i, y_j) + \delta \sum_{ae=(i,j) \in E_A} I^{A}_{ae}(X, y_i, y_j) - \log[Z(X)]\}
\end{aligned}
\tag{5.9}
$$

where $P(Y|X)$ is the posterior of sequential amaCRF model. $P^{(0)}(y_i|X)$, $P^{(1)}(y_i|X)$ and $P^{(2)}(y_i|X)$ are posteriors updated zero, one and two times update; and $\lambda^{(0)}$, $\lambda^{(1)}$, and $\lambda^{(2)}$ are corresponding weights. $V^{(0)}$, $V^{(1)}$, and $V^{(2)}$ are node set of three scan profile, and the upper index notes how many time posterior has been updated. E_S, E_{LV}, E_{LH}, E_A are respectively the edge set of short range, long range vertical / horizontal along scan profile and across scan profile edge set, and α, β, γ and δ are corresponding weights. Compared with non-sequential amaCRF model, the sequential method updates association terms gradually, but does not change the graph structure and interaction potential design of amaCRF model. To simplify the model, we assume that three types of posteriors have equal weight and so it has the same equation as amaCRF model. Therefore, under this assumption, parameters of amaCRF model can be shared with the sequential amaCRF model. For inference, the sequential amaCRF model also used the LBP algorithm.

5.4 Experiments of Across Scan Profiles CRF models

In this chapter, two models were proposed, amaCRF and an improved model using sequential label propagation, sequential amaCRF model. To validate the advantages of context of across scan profile and sequential knowledge propagation, the two models were tested on the same datasets that we already used in previous chapters. Each CRF model was implemented using Matlab and C++. Implementation of parameter learning and inference referred the UGM code (Schmidt, 2007). Scan line number, line segment number and number of short range, long range vertical and horizontal edge, and across scan profile edges of each dataset are presented in Table 5.1.

Table 5.1: Total number of the spatial entities extracted from York Village datasets.

Nodes and edges	**YV1**	**YV2**
Scan line	2810	2580
Line segment	105,620	100,648
Short range edge	260,579	277,584
Long range vertical edge	158,276	156,023
Long range horizontal edge	18,787	16, 463
Across scan profile edges	167,098	169,832

Parameters in amaCRF model were learned using SGD, and the parameter learning on the data YV1 is showed in Figure 5.5. The horizontal axis indicates the iteration number and vertical axis indicates weight value. There are 2810 scan profiles in the training data, and it is observed that weights converge very fast and get stable after scanning the entire dataset 10 times.

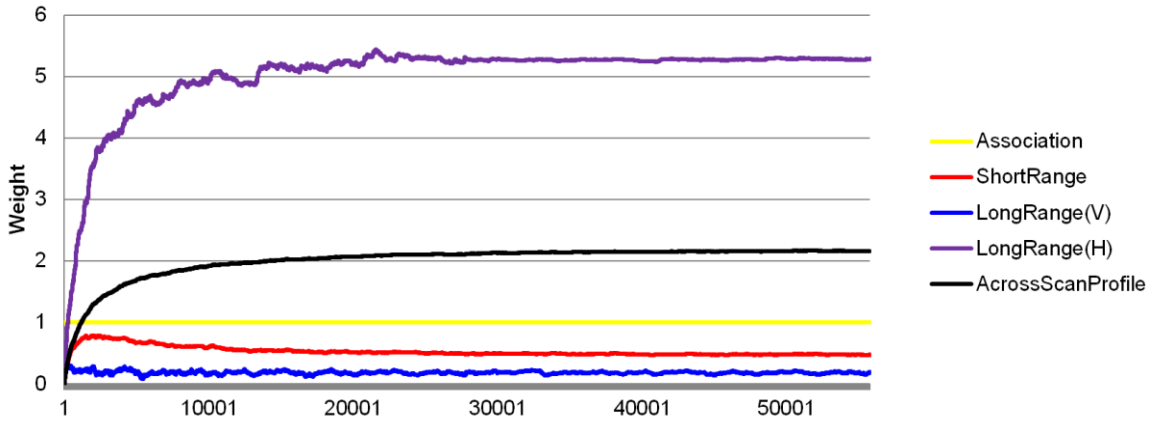


Figure 5.5: Parameter learning of maCRF model on data YV1. X-axis is the interation number and Y-axis is the weight value.

Classification results of the two amaCRF models on the data YV2 are presented in Figure 5.6. Compared with classification result of maCRF, which is presented in the Figure 4.6(d), the amaCRF model removes most of "pepper and salt" noises on facade area (Figure 5.6(a)). It is also observed that those area with serious occlusion is more likely to be affected by the across scan profile. Using sequential processing, classification quality is further improved (Figure 5.6(b)).

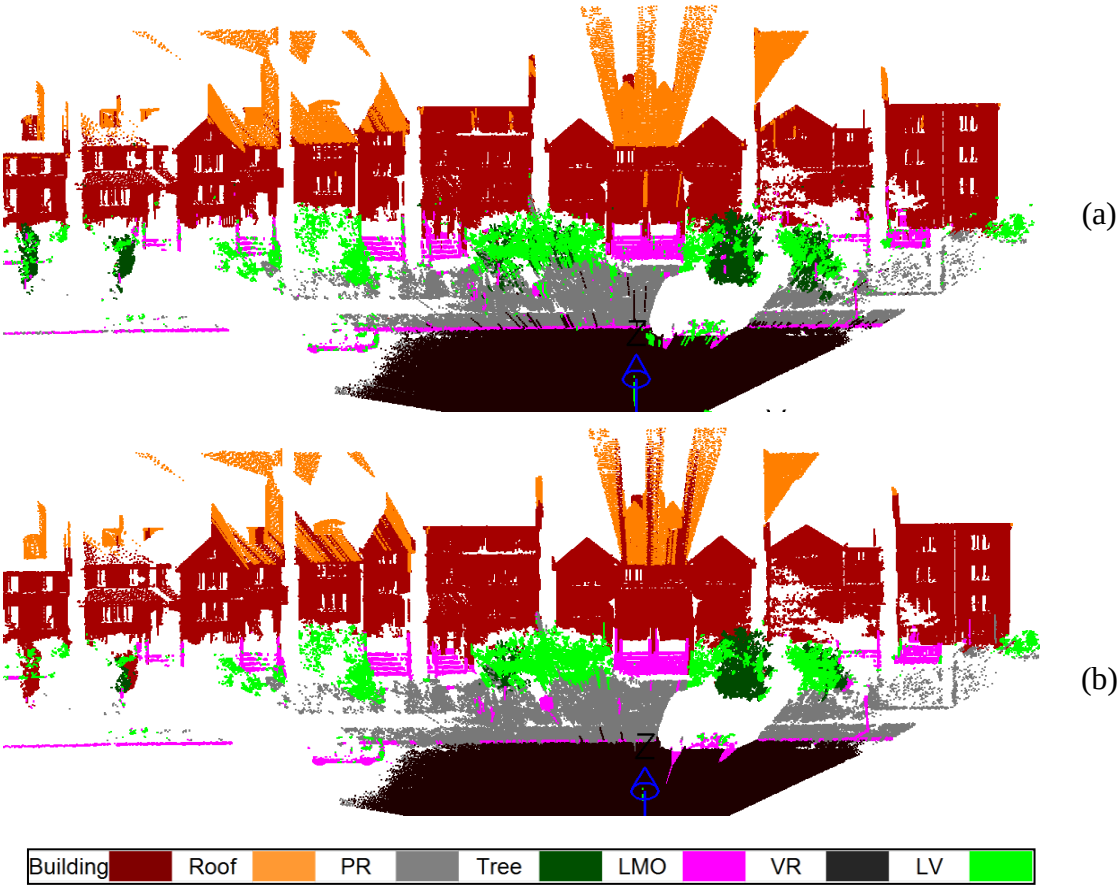


Figure 5.6: Classification result of the amaCRF model and sequential processing on the data YV2. (a) amaCRF model; (b) amaCRF+ model.

Test accuracies of GMM, maCRF and two amaCRF models on data YV1 and YV2 are shown in Table 5.2. Compared with maCRF, the amaCRF improved the classification accuracy by 1.29%, and the sequential processing improved further by 2.05%. Compared with GMM classifier, the amaCRF+ improved the accuracy up to around 10%.

Table 5.2: Test Accuracy of sequential CRF Models.

Classifier	YV1	YV2	Averaged	Improvement
GMM	79.53	79.98	79.76	
maCRF	86.51	85.79	86.01	+6.25
amaCRF	87.01	87.79	87.40	+7.64
amaCRF+	89.10	89.79	89.45	+9.69

5.5 Additional Experiments

In this section, we will study potential generalization ability of the proposed multi-range asymmetric CRF models and the sequential processing. The objective of the additional experiments is to investigate whether the proposed classifier is dependent on association term, and whether the proposed classifier is dependent on the scene type.

5.5.1 SVM Based CRFs

GMM and SVM are champions of generative classifiers and discriminative classifiers respectively. But in previous CRF models, we only used output of GMM as input of the association term. Although experimental results validated the advantage of using multi-range contexts, a question comes up: *is the context only compatible with the output of generative classifiers?* To address this problem, the first experiment is replacing the output of GMM with output of SVM. To convert the output of the decision function to a posterior probability, we used a modified version of the method in Wu et al. (2004). Given the posterior probabilities of SVM classifier, six SVM-based CRF models, srCRF, lrCRF(V), lrCRF(H), maCRF, amaCRF and amaCRF+ was modeled. The conditional

distribution over the labels Y given observed data X in the across scan profile amaCRF graph G_A can be defined as follows:

$$P(Y \mid X) = \frac{1}{Z(X)} \exp[\lambda \sum_{i \in V} \log(P_{SVM}(x_i \mid y_i)) + \alpha \sum_{se=(i,j) \in E_S} I_{se}^{S}(X, y_i, y_j) + \beta \sum_{lve=(i,j) \in E_{LV}} I_{lve}^{LV}(X, y_i, y_j) + \gamma \sum_{lhe=(i,j) \in E_{LH}} I_{lhe}^{LH}(X, y_i, y_j) + \delta \sum_{ae=(p,q) \in E_A} I_{ae}^{A}(X, y_p, y_q)] \quad (5.10)$$

Parameter learning of amaCRF model on the data YV1 is showed in Figure 5.5. The horizontal axis indicates the iteration number and vertical axis indicates weight value. It is observed that weights start to converge after scanning through the entire training data three times, which is even faster than the GMM-based amaCRF model.

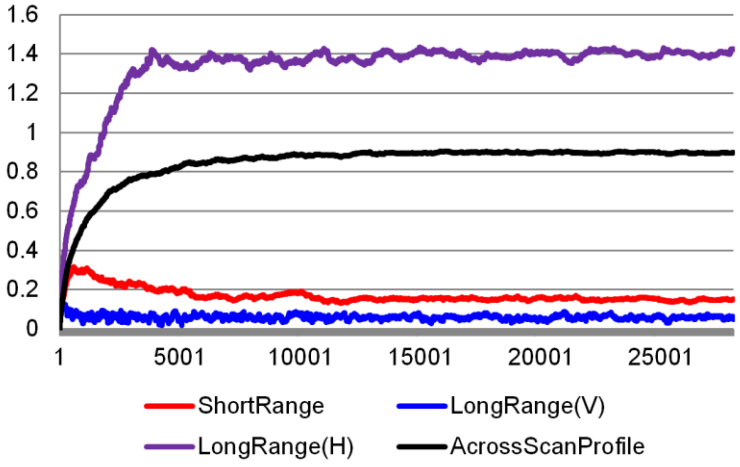


Figure 5.7: Parameter learning of the amaCRF model (SVM) on data YV1. X-axis is the interation number and Y-axis is the weight value.

Test accuracies of SVM and the six SVM-based CRF models on data YV1 and YV2 are presented in Table 5.3. It is observed that effect of multi-range contexts and sequential processing of SVM-based CRF models are similar with those GMM-based CRF models.

Table 5.3: Test Accuracy of sequential CRF Models.

Classifier	**YB1**	**YL2**	**Averaged**
SVM	85.19	85.32	85.26
srCRF	86.82	86.77	86.80
lrCRF(V)	88.62	88.51	88.57
lrCRF(H)	86.32	85.81	86.07
maCRF	89.72	89.85	89.78
amaCRF	90.03	90.14	90.09
amaCRF+	90.18	90.36	90.27

Classification results of SVM and the sequential amaCRF model on the data YV2 are presented in Figure 5.8. As a local classifier, SVM produced a result with visible misclassification errors (Figure 5.8(a)). Considering multi-range contexts and sequential processing, most of local inconsistence errors were removed (Figure 5.8(b)). It is concluded that the proposed multi-range based CRF models are not sensitive to association term. It is noticed that since the SVM already gives high classification accuracy, the improvement space of multi-range and sequential processing is very limited.

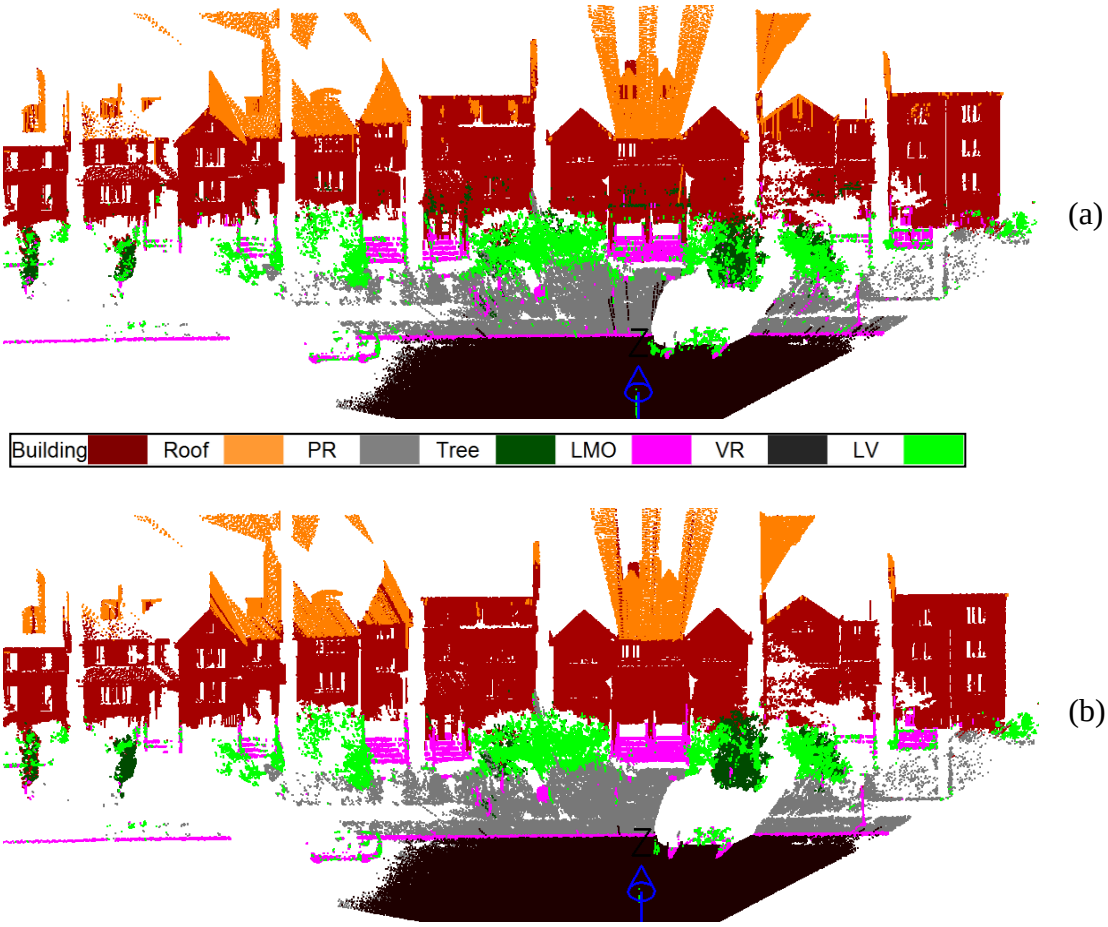


Figure 5.8: Classification result of the SVM-based amaCRF model and sequential processing on the data YV2. (a) amaCRF model; (b) sequential processing.

5.5.2 York Blvd Datasets

The second generation is applying the proposed classification algorithms on different dataset. The new dataset was collected at two different sites, on York Blvd, York University campus, Toronto. The two datasets are noted as YB1 and YB2 respectively. YB1 locates at the north of the York Blvd, and mainly covers the south facade of York

Lanes Mall. YB1 locates at the south of the York Blvd, and mainly covers the north facade of the Center for film and Theatre. Different from York Village dataset, building roofs in York Blvd datasets are flat so that roof is not visible in the TLS data and we have only six classes: building, pedestrian road (PR), tree, low man-made object (LMO), vehicle road (VR), and low vegetation (LV). Scene type of York Village and York Blvd data are different, such as Architectural style of building, tree species.

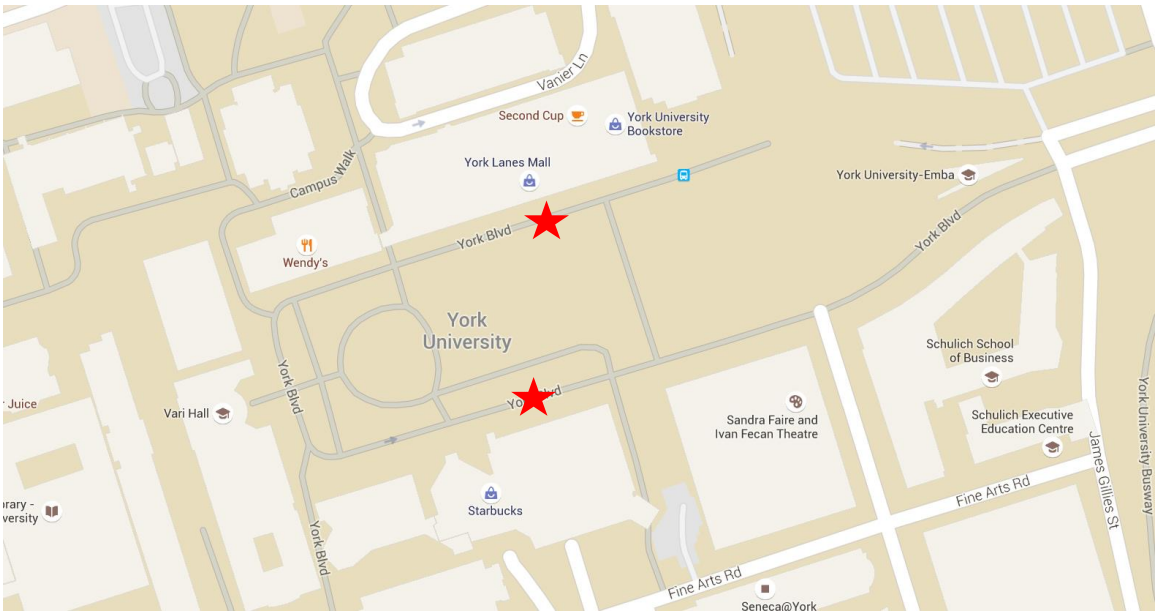


Figure 5.9: Surveying locations of York Blvd Dataset. Dataset was collected at two different locations (red pentagram) of York Blvd.

The two York Blvd datasets were collected using RIEGL LMS Z390i laser scanner and all dataset acquisition setup is the same as York Village dataset. All thresholds of data processing, including scan profile generation, line extraction, feature extraction and line adjacent neighboring searching are also the same as York Village

dataset. Table 3.1 summarizes the number of spatial entities extracted from the two datasets.

Table 5.4: Total number of the spatial entities extracted from York Blvd datasets.

Spatial entities	YB1	YB2
Laser points	3,673,257	3,484,462
Scan profiles	2,800	2,600
Line segments	152,978	162,053
Short range edge	579,872	582,341
Long range vertical edge	400,299	408,247
Long range horizontal edge	9,687	15,274
Across scan profile edge	410,244	413,192

We still used the two-fold cross validation to evaluate the performance of proposed classifiers on York Blvd dataset. We only tested the amaCRF and sequential processing methods. Both GMM and SVM were used as association terms of the two CRF models. Classification results of the GMM and GMM-based amaCRF model with sequential processing on the data YB2 is presented in Figure 5.10. Classification errors of GMM were mainly found between building and tree, low vegetation and low man-made object. Most of these errors were removed after applying the multi-range contexts and sequential processing. We still found the many lines of pedestrian road were misclassified as vehicle road using GMM, and unfortunately they cannot be effectively rectified by using multi-range contexts and sequential processing, which need to be further examined. SVM and SVM-based CRFs had similar results as GMM and GMM-based CRFs had, and classification results on the data YB2 are presented in Figure 5.11.

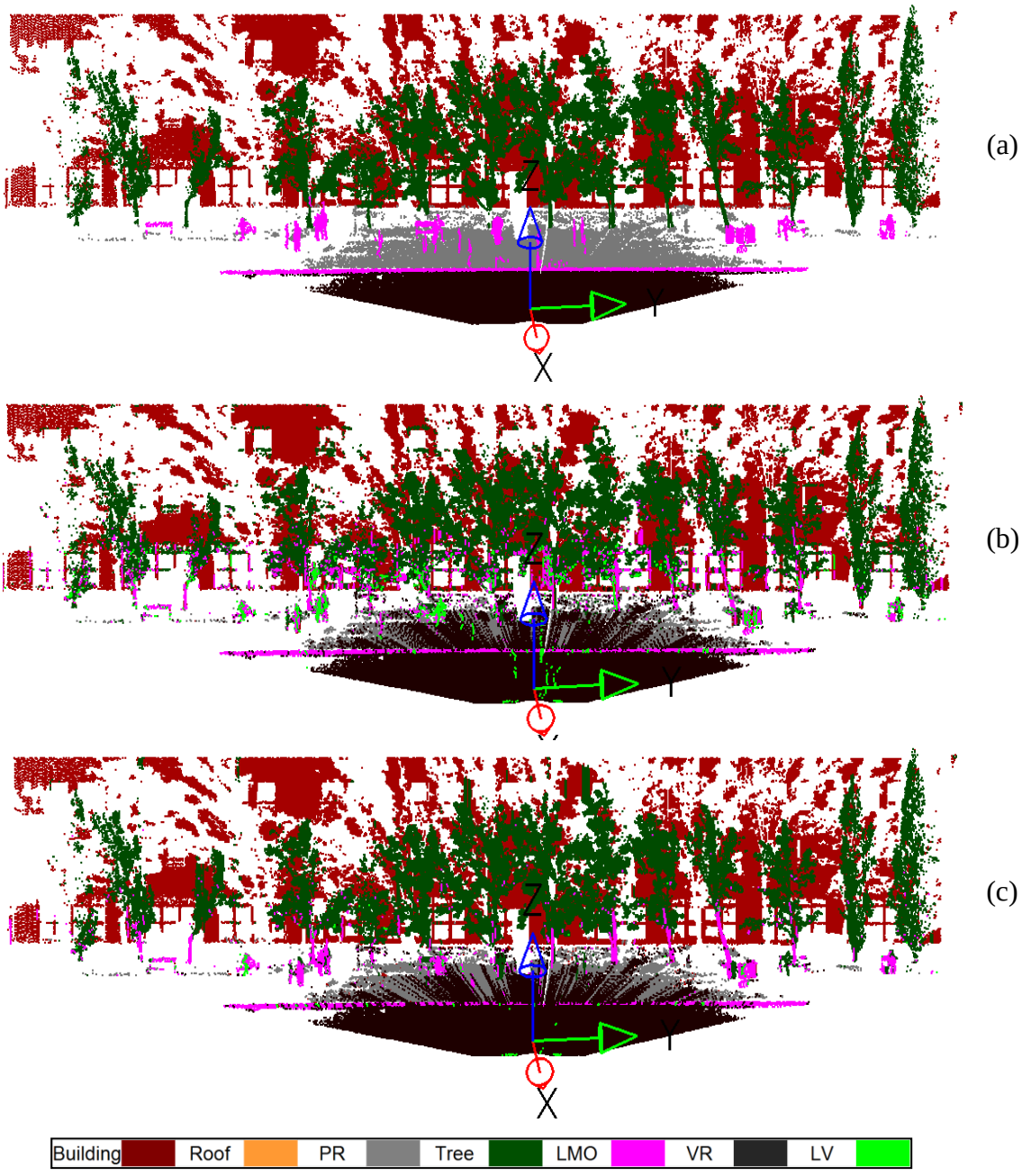


Figure 5.10: Classification results of the GMM and GMM-based amaCRF model with sequential processing on the data YB2. (a) Ground truth; (b) GMM; (c) GMM-based amaCRF model with sequential processing.

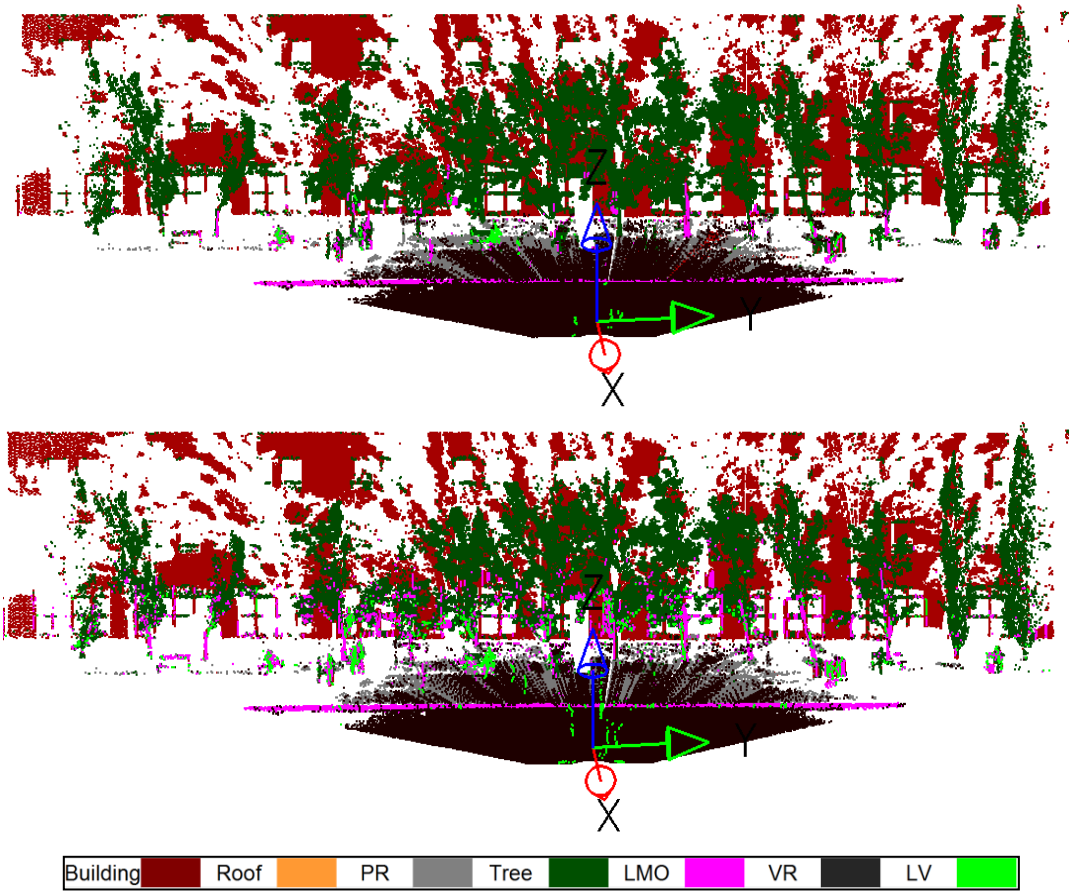


Figure 5.11: Classification results of the SVM and SVM-based amaCRF model with sequential processing on the data YB2. (a) SVM; (b) SVM-based amaCRF model with sequential processing.

Test accuracy of the local classifiers and CRF models on the two York Blvd datasets are presented in Table 5.4. Although the two local classifiers, GMM and SVM, already make high classification performance, improvement still can be achieved by amaCRF and sequential processing. Compared with GMM, the GMM-based amaCRF with sequential processing improved averaged test accuracy by 4.36%. Compared with

SVM, the SVMM-based amaCRF with sequential processing improved averaged test accuracy by 3.38%.

Table 5.5: Test Accuracy of the proposed classifiers on York Blvd datasets.

Classifier	YB1	YB2	Averaged
GMM	88.27	89.19	88.73
GMM based amaCRF	92.25	92.71	92.48
GMM based amaCRF+	92.93	93.24	93.09
SVM	89.73	90.16	89.95
SVM based amaCRF	92.92	93.23	93.08
SVM based amaCRF+	93.18	93.47	93.33

5.5.3 Train Classifiers using York Village Dataset and Test on York Blvd Dataset

Training and testing datasets in the previous experiments were collected at the same street. In the third additional experiment, training and testing datasets were collected at different streets. Classifiers were trained from the YV1 data and then tested on the York Blvd datasets, both YB1 and YB2. There are seven types of objects in the York Village data (building, roof, PR, tree, LMO, VR, LV), and only six types of objects in the York Blvd data (building, PR, tree, LMO, VR, LV). Because the York Village data has the class "roof" that does not appear in York Blvd data, testing classification performance of York Village data using the classifiers trained from York Blvd dataset was not implemented.

Four classifiers, GMM, SVM, GMM based amaCRF+ and CRF based amaCRF+ were trained from the YV1 data, and then were tested on YB1 and YB2. Classification

accuracy of their performance is shown in the Table 5.6. Performance of the local classifiers on the YB1 data is not very high, but higher than 60%. But performance of the local classifiers on the YB1 data is rather bad, lower than 50%. The amaCRF+ improved the classification performance of local classifiers on the YB1 data, but achieved even worse performance than local classifiers on the data YB2. Thus, this additional experimental does not validate that the multi-range context and the sequential modeling method can improve classification performance of local classifier.

Table 5.6: Test Accuracy of York Blvd dataset using Classifiers Trained from YV1 dataset.

Classifier	**YB1**	**YB2**
GMM	64.42%	48.91%
GMM based amaCRF+	65.16%	35.30%
SVM	67.61%	47.63%
SVM based amaCRF+	69.58%	34.17%

The classification results of GMM and GMM based amaCRF+ on the data YB1 and YB2 are presented in the Figure 5.12. It is observed that amaCRF+ always can make a more coherent classification result on building facade area both in data YB1 and YB2. Some building lines were misclassified as roof in GMM, and then these misclassification errors were removed by amaCRF+. However, many tree lines that correctly classified in GMM were misclassified as building by amaCRF+.

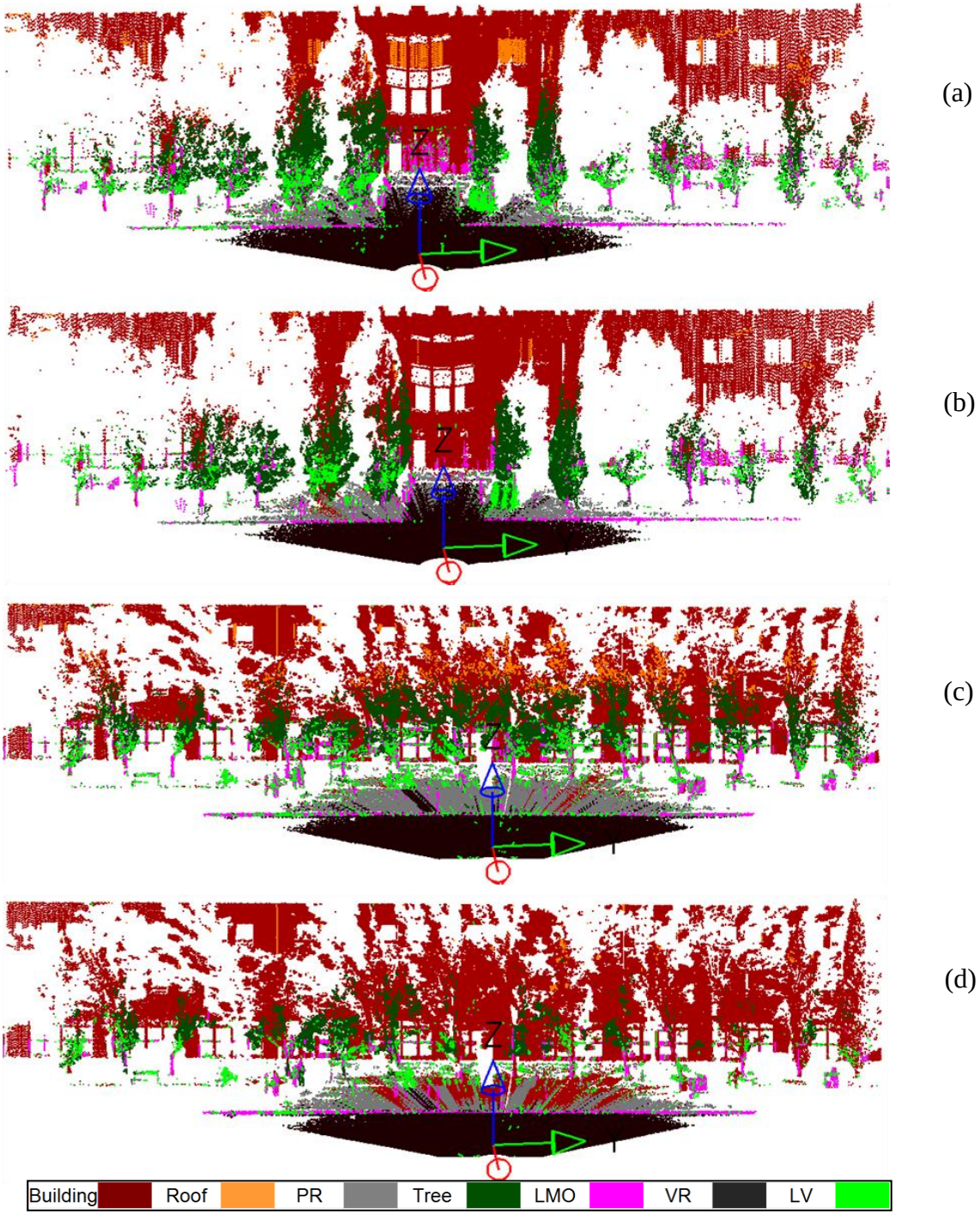


Figure 5.12: Classification results of York Blvd datasets using classifiers trained from data YV1. (a) GMM of YB1; (b) amaCRF+ of YB1; (c) GMM of YB2; (b) amaCRF+ of YB2.

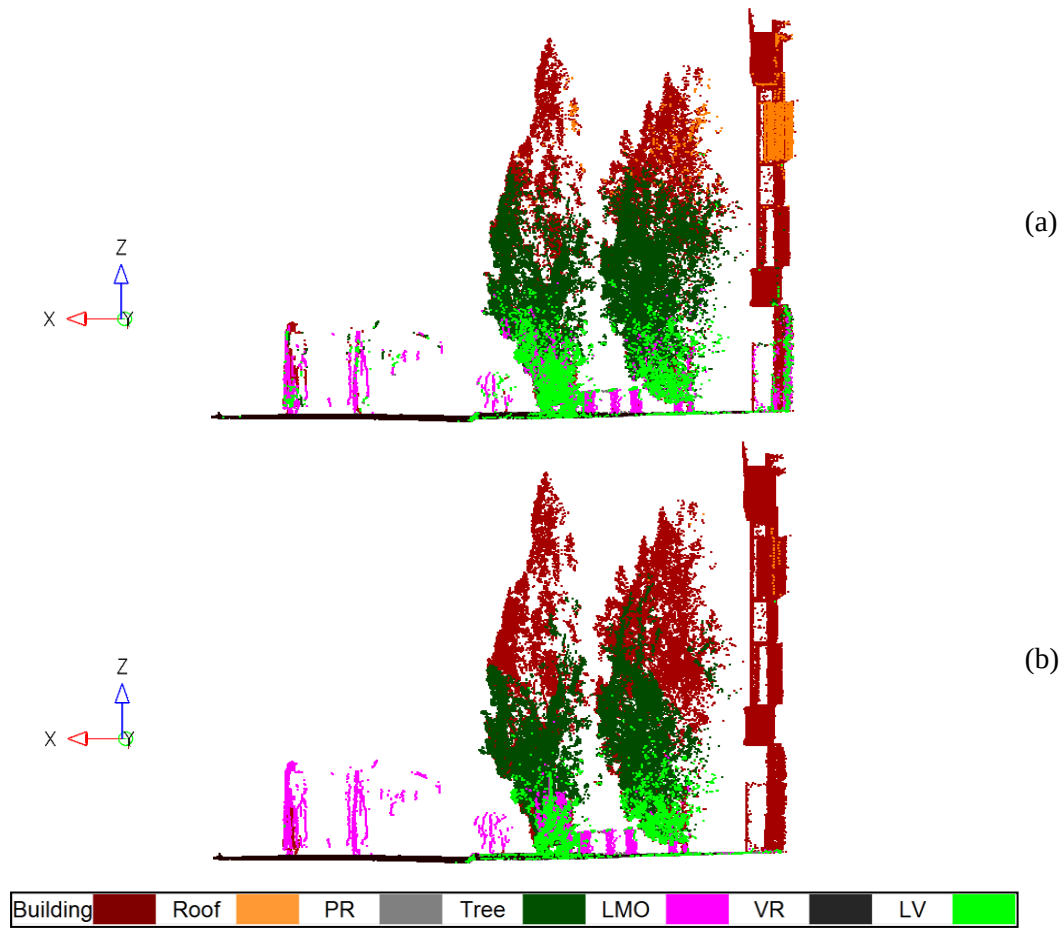


Figure5.13: Sideview of classification results of YB1 data. (a) GMM; (b) amaCRF+.

Sideview of the classification results of YB1 data are presented in the Figure 5.13. In the Figure 5.13(a), tree has serious misclassification problem using GMM. Those high tree lines were misclassified as building or roof, and those low tree lines were misclassified as low vegetation. This problem could result from that trees in the test data are different from the trees in the training data, such as species, structure, especially the height. Tree height in testing data is double of that in the training data, so that it is similar

with building in the training data; therefore the high tree lines have risk to be classified as building. And low tree lines in testing data have more similar height distribution with low vegetation, and this perhaps the reason that why many of them were misclassified as low vegetation. In the Figure 5.13(b), it is observed that amaCRF+ rectified most of misclassification errors in building, low man-made objects (bus); however, misclassification problem of tree is even worse.

Two representative scan profiles were selected from YB1 data for further analysis, which are noted as SP-A and SP-B are presented in the Figure 5.13 and the Figure 5.14 respectively.

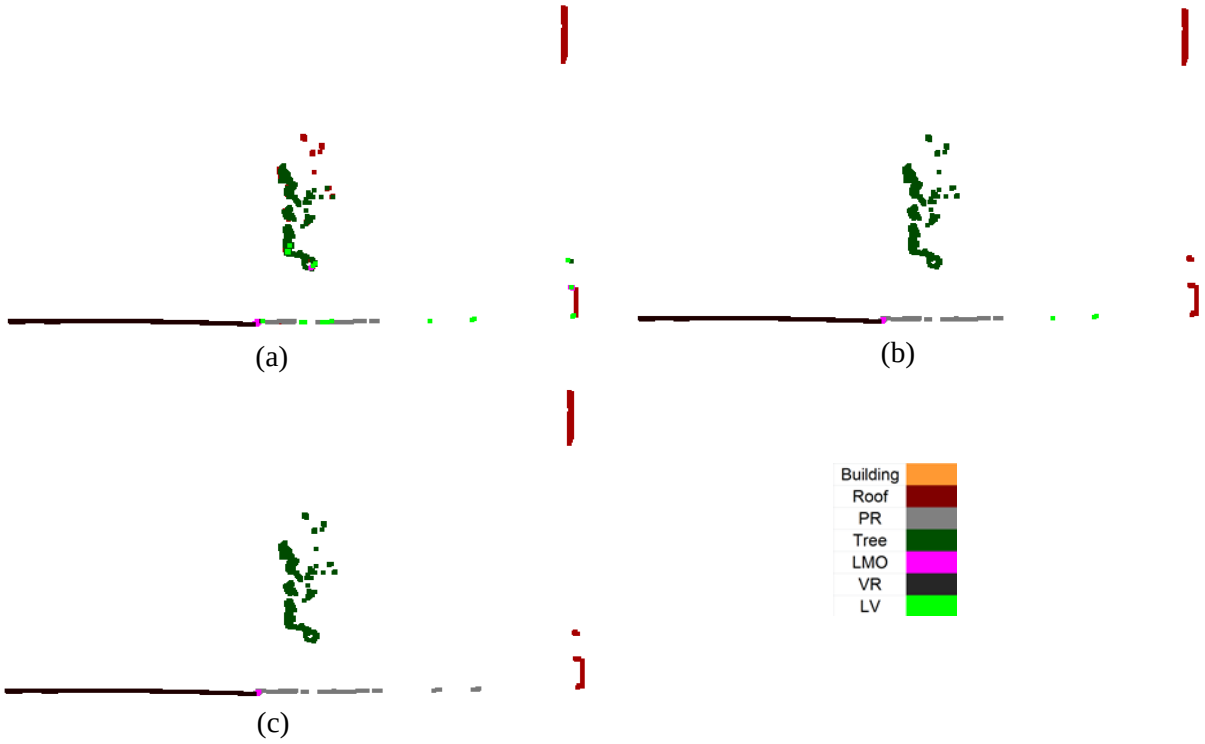


Figure 5.14. classification results of the scan profile SP-A. (a) GMM; (b) amaCRF+; (c) ground truth.

In the scan profile SP-A, only a few lines were incorrectly classified by GMM (Figure 5.14(a)), and then they were rectified by considering neighbors in along and across scan profiles (Figure 5.14(a)). In the scan profile SP-B, because most of tree lines were misclassified as building using GMM (Figure 5.15(a)), those correctly classified tree lines were affected by the misclassified majority and then changed the label from true to flase after applying amaCRF+ (Figure 5.15(b)).

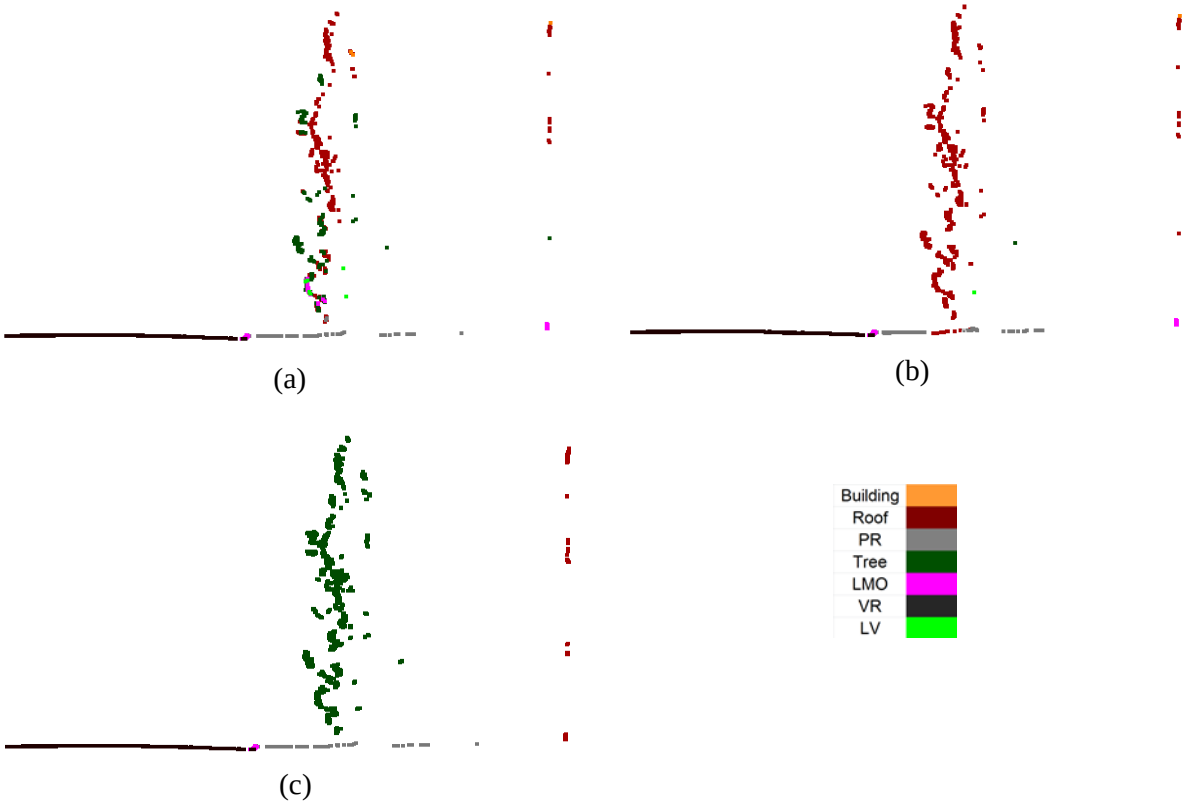


Figure 5.15: Sideview of classification results of YB1 data. (a) GMM; (b) amaCRF+.

From analysis of the two scan profiles, it is concluded that the multi-range contexts do enforce constraints on local smoothness and global scene layout; however the effect of

the multi-range contexts is related with the association term (local classifier) as well. When local classifier achieves a satisfying classification performance, applying multi-range contexts can further improve the classification accuracy; otherwise, when local classifier is weak, applying multi-range contexts has the risk of decreasing the classification accuracy.

5.6 Chapter Summary

In this chapter, the maCRF model was extended to across scan profile, which is called across scan profile multi-range CRF (amaCRF) model. The amaCRF model incorporates contexts along scan profile (short range, long range vertical and horizontal) and across scan profile context into a unified probabilistic graphical model. The amaCRF model is built over every three consecutive scan profiles and contextual information one scan profiles can be propagated to adjacent scan profile through across scan profile edges; however, scan profiles at different amaCRF models are absolutely independent. Therefore, we proposed a sequential processing method (amaCRF+), which allows contextual information propagate through adjacent scan profiles. Along the sweeping direction, amaCRF models are sequentially constructed. The posteriors of the previous amaCRF are used as association term of the next amaCRF model so that posteriors of those lines at overlapping scan profiles can be updated. In this way, contextual information of the first scan profile can be propagated to the last scan profile. The experiment results showed that the amaCRF and amaCRF improved success rate of GMM by 2% and 4% respectively.

We also examined the generalization ability of the proposed methods. To validate that the multi-range context CRF model is not dependent on the association term, output of GMM was replaced by the output of SVM. Experiment results showed that SVM-based CRF models can achieve similar classification improvement as GMM did. The algorithm was also tested using another TLS data, which was collected at York Blvd, and the experimental results verify that the proposed algorithm has good generalization ability, not only work on specific scene. Finally, classifiers trained from York Village were tested on York Blvd data. Although the experimental results do not validate that multi-range contexts and sequential modeling is able to improve the classification performance, the effect of local smoothness and global scene layout enforced by multi-range contexts can be observed.

Chapter 6
Discussions

This thesis aims to achieve two primary objectives for addressing the research problems to label urban street scenes from massive laser point clouds acquired by TLS. On one hand, the study focused on the design and implementation of an automatic, accurate and robust classifier, which can be employed for a real-time laser point cloud processing. In this study frame, a concept of "per-scan profile" classification, following the scanning nature of range profiler such as TLS was proposed. On the other hand, the thesis discussed significant roles of spatial context and regularities for improving the performance achieved by conventional local classifiers. This spatial regularity has been studied in the framework of CRF. These objectives were achieved by developing three major methods presented in this thesis: (1) implemented a new "per-scan profile" classifier, which characterize key street objects with apparent and context linear features and validate the effectiveness of "per-scan profile" classifier using ten different generative and discriminative classifiers; (2) proposed a multi-range asymmetric CRF model (maCRF), which augments spatial layout compatibility by integrating multi-range smoothness (short, long range vertical and long range horizontal) in CRF; and finally (3) extended maCRF by labeling point clouds, not only along scan profile, but also across scan profiles; and proposed two classifiers, called amaCRF and amaCRF+ by updating the posterior probability of label decision through non-overlapping (amaCRF) or overlapping (amaCRF+) sequential processing scheme. This chapter will give an

overview of this research and discuss our conclusions on this subject, and then the future directions could follow.

6.1 Conclusions

In this research, we proposed a line based multi-range asymmetric CRF (maCRF) model, which is aimed at real-time TLS data classification. This work can be decomposed into three parts as follows:

1. **Line-based object representation**

   We explored the potential of lines as the geometric primitive for classification purpose. In our "per-san profile" classification scheme, we believe that the line primitives are optimal for characterizing street objects and gaining computational benefits. In this study, the lines were extracted from each vertical scan profile. Each scan profile was considered as a stream of observed points, and those points that have similar range were merged into a line. To avoid the "under-segmentation", the Douglas–Peucker algorithm was then applied as a post-processing for splitting the under-segmented line into separate lines. The line extraction result shows that as high as 99% points can be represented by the lines, and all types of object we are interested in are well characterized by line primitives.

2. **Line-based TLS data classification (Local Classifier)**

   To classify the extracted line primitives, we implemented local classifiers by proposing two types of line-based features (i.e., apparent features and contextual features). Two neighboring systems (circle-based and vertical column-based)

were used for extracting context features. The total thirty-five features were reduced into eight dimensions using PCA algorithm. Based on these features, we designed and implemented 10 different local line-based classifiers covering both generative and discriminative ones, which include NB, MG, GMM, KNN, LR, SVM, ANN, DT, and two ensembles based on decision tree, RF and AdaBoost. The performance of these classifiers was then quantitatively evaluated using confusion matrix, accuracy, precision, recall and F1-score. The strongest classifiers achieved accuracy up to 85.60% (SVM with RBF kernel), while the weakest classifier achieved accuracy in 68.82% (NB). We observed that the averaged classification accuracy over all the ten classifier is as high as 79.19%. The overall experimental results suggested that the line-based local classifiers are efficient to produce reasonable classification outcomes. However, the labeling errors produced by the local classifiers are locally irregulars, which do not follow compatible spatial relations amongst objects. For instance, tree objects labeled by the local classifiers are often found in the middle of building facades. We concluded this local labeling irregularity was caused by the locality of neighboring smoothness implemented in the local classifiers and resolved by introducing another type of regularity, such as layout compatibility amongst spatial objects. These problem observations lead to the development of multi-range and layout compatible context (regularity) within the framework of CRF for improving classification results in this thesis.

3. **Along scan profile CRF model(maCRF)**

As local classifiers are trained only relying on apparent features, they are likely to produce misclassification errors when two classes overlap in the feature space. To overcome the limitations of local classifiers, we proposed multi-range and asymmetric CRF (maCRF), which augments the semantic context between adjacent labels, not only considering the local homogeneity, but also in sparse neighboring system (long range). This context augmentation considers both local labeling homogeneity and implicit regularity of spatial layout relations amongst objects. Two types of contexts were used, short range and long range context. The short range context imposes local smoothness constraint that neighboring lines are likely to have the same class label. While the long range context forces regularity on scene layout that objects follow some specific spatial arrangements along each scan profile, both in vertical and horizontal directions. Rather than using predefined rules, the scene-layout compatibility functions are automatically learned from training data. The experiment results validated three multi-range and asymmetric context regularity terms contributed to the improvement of the performance of local classifier (GMM-EM). We observed that all context terms provided positive effects to the classification results. However, we found that each type of context terms affect the classification differently. Especially, the vertical layout compatibility term provided the most benefits to improve the classification results, higher than 5% success rate compared to the horizontal term. We believe that more scene complexity (more numbers of objects, relations

and occlusion) is present in the horizontal direction, which is likely to cause more ambiguity to impose spatial layout regularity compared to in the vertical direction. The experimental result also suggests the integrated multi-range CRF model combine benefits of all three single contexts and makes the best classification performance by improving 8% classification compared to the local classifier (GMM-EM).

4. **Across scan profile CRF model (amaCRF and amaCRF+)**

   TLS typically scans the scenes, not only in vertical direction, but also horizontal direction as well. We extended the capacity of maCRF to classify laser point clouds by propagating label probability estimated within each vertical scan profile into across scan profiles. For achieving this goal, we proposed two multi-range asymmetric CRF models, called amaCRF and amaCRF+. These two classifiers were developed based on the same frame of maCRF, but are different each other with respect to the ways of label propagation. The amaCRF model was built over every three consecutive scan profiles. Compared to maCRF, the amaCRF model enforces its label decision with additional context regularity from neighbours along scan profiles. The experimental results demonstrated that the classification quality produced by amaCRF was greatly improved, especially over occluded regions compared to maCRF.

   However, the amaCRF limits its labeling decision only within three scan profiles involved in the local graphical model construction. To address this limitation, we proposed amaCRF+, which allows sequential propagation of semantic knowledge

across CRF models. In amaCRF+ scheme, amaCRF models were sequentially constructed and adjacent models share identical scan profiles. The posteriors of the previous amaCRF were used as the association term of its next amaCRF model so that posteriors of those lines at overlapping scan profiles can be updated. In this way, contextual information of the first scan profile can be propagated to the last scan profile. The experiment results suggested that by dynamically updating posteriors, classification confidence of each line get stronger, which leads to additional gains of classification performance compared to maCRF and amaCRF. We observed this performance improvement is more obvious when GMM (representative of generative classifier) was used as a local classifier compared to SVM (representative of discriminative classifier).

6.2 Future Work

Upon summarizing and highlighting the contributions of this doctoral research project, it is essential to identify the limitation of current methodology design and address them appropriately in future considerations.

1. **Application to real-time classification**

   In recent years, many engineering applications using TLS requires real-time scene understanding for supporting on-site decision making, such as for autonomous car, unmanned vehicle and robot navigation, sense-and-avoid decision, facility risk monitoring and emergency response. In this thesis, our along and across scan profile CRFs were designed for providing computational benefits by limiting labeling spaces to per-scan profiles and thus suit for a real-

time point cloud processing. Also, our experimental results demonstrated the effectiveness and satisfactory classification performance of the proposed classifiers. However, in this thesis, the implementation of our classification methods wasn't realized in a truly real-time mode (on-board processing integrated with laser scanners). In our future research, we will implement our proposed classifiers tightly coupled with laser scanning hardware and evaluate its effectiveness for supporting emerging on-the-go decision applications in a truly real-time environment.

2. Generalization of classification methods

In this thesis, we designed, implemented and validated several new classifiers, but for targeting a limited numbers of street objects within certain limited environments. In a short-term, we plan to further investigate the sensitivity of our proposed classifiers to: 1) different scene types and complexity; 2) various point density; and 3) different laser scanning mechanism. Thus we will investigate how these variations from our current experimental setting might produce different quality and density of line-based object representation and thus lead to non-optimal classification results. In this regard, our future research efforts will focus on the adaptive design of line-based object representation, which performance will be more robust to the variations of point density and scanning mechanism. In addition, we will investigate an intelligent fusion of object representation to combine the line primitives with others such as points and surfaces; and also incorporate various attributes including colors and intensity within current

classification models. In a long-term, we will extend our classification methods to mobile and airborne applications, enabling the real-time scene classification.

3. **Optimization of parameters**

   Many parameters were manually set in this research, such as the threshold (0.5m) to separated scattered point and smoothness points, the threshold (0.1m) of Douglas-Peucker algorithm, radius (1m) of circle-based neighboring system, width (0.5m) of column-based neighboring system, and size of cell (0.5m by 0.5m) in the grid system. Although this ad-hoc parameter setting achieved satisfying results, it does not guarantee to optimal results. Therefore, these parameters will be chosen using optimization methods.

4. **High-order scene layout regularity**

   In this research, we modeled the scene layout regularity using pairwise potential functions (first-order dependency). The first-order dependency can only allow to model relations between two nodes, like “building is on top of ground” or “roof is on top of building”. In reality, one spatial object have much complex layout relations with multiple objects, which is difficult to be interpreted through the pairwise context. In our future work, we will consider a strategy to increase the power of object layout context, which depends on a large number of entities by implementing a high-order potential function in our current graphical models. Defined over multiple entities, the higher-order potential function will be able to model complex interaction between objects, such as “building is on top of ground but also below of roof”.

Bibliography

Aijazi, A. K., Checchin, P., and Trassoudaine, L., 2013. Segmentation based classification of 3D urban point clouds: A super-voxel based approach with evaluation. *Remote Sensing*, 5(4), pp. 1624-1650.

Al-Manasir, K., and Fraser, C. S., 2006. Registration of terrestrial laser scanner data using imagery. *The Photogrammetric Record*, 21(115), pp. 255-268.

Alshawabkeh, Y., 2006. Integration of laser scanning and photogrammetry for heritage documentation.

Andersson, S. A., multivariate Madigan, D., Perlman, M. D., and Richardson, T. S. (1999). Graphical Markov models in analysis. Statistic Textbooks And Monographs, 159, pp. 187-230.

Anguelov, D., Taskarf, B., Chatalbashev, V., Koller, D., Gupta, D., Heitz, G., and Ng, A., 2005. Discriminative learning of markov random fields for segmentation of 3d scan data. In *Computer Vision and Pattern Recognition (CVPR 2005)*, Vol. 2, pp. 169-176.

Aschoff, T., and Spiecker, H., 2004. Algorithms for the automatic detection of trees in laser scanner data. *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, 36(Part 8), W2.

Axelsson, P., 1999. Processing of laser scanner data — algorithms and applications. *ISPRS Journal of Photogrammetry and Remote Sensing*, 54(2), pp. 138-147.

Axhausen, K. W., 2011. Future Cities Laboratory.

Badrinarayanan, V., et al., 2010. Label propagation in video sequences. *Computer Vision and Pattern Recognition (CVPR 2010)*, pp. 3265-3272.

Bao, S. Y., Sun, M., and Savarese, S., 2011. Toward coherent object detection and scene layout understanding. *Image and Vision Computing*, *29*(9), pp. 569-579.

Barnea, S., and Filin, S., 2007. Registration of terrestrial laser scans via image based features. *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, 36, pp. 32-37.

Basheer, I. A., and Hajmeer, M., 2000. Artificial neural networks: fundamentals, computing, design, and application. *Journal of microbiological methods*, 43(1), pp. 3-31.

Belton, D., and Lichti, D. D., 2006. Classification and segmentation of terrestrial laser scanner point clouds using local variance information. *IAPRS, Xxxvi*, *5*.

Baltsavias, E. P., 1999. A comparison between photogrammetry and laser scanning. *ISPRS Journal of photogrammetry and Remote Sensing*, 54(2), pp. 83-94.

Benediktsson, J., Swain, P. H., and Ersoy, O. K., 1990. Neural network approaches versus statistical methods in classification of multisource remote sensing data. *Geoscience and Remote Sensing, IEEE Transactions on*, *28*(4), pp. 540-552.

Besag, J., 1986. On the statistical analysis of dirty pictures. *Journal of the Royal Statistical Society B*, 48 (3), pp. 259–302.

Bienert, A., Scheller, S., Keane, E., Mohan, F., and Nugent, C., 2007. Tree detection and diameter estimations by analysis of forest terrestrial laserscanner point clouds. In *ISPRS workshop on laser scanning*, Vol. 2007, pp. 50-55.

Bishop, C. M. (2006). *Pattern recognition and machine learning*. springer.

Bo, Y., and Fowlkes, C. C., 2011. Shape-based pedestrian parsing. In *Computer Vision and Pattern Recognition (CVPR 2011)*, pp. 2265-2272.

Böhm, J., and Haala, N., 2005. Efficient integration of aerial and terrestrial laser data for virtual city modeling uusing lasermaps.

Boudreau, S., and Brynildsen, D., 2003. National guide to sustainable municipal infrastructure. Ottawa, Ontario, Canada: National Research Council of Canada.

Boulaassal, H., Landes, T., Grussenmeyer, P., and Tarsha-Kurdi, F., 2007. Automatic segmentation of building facades using terrestrial laser data. *In ISPRS Workshop on Laser Scanning 2007 and SilviLaser 2007*, pp. 65-70.

Bowling, M., and Veloso, M., 2002. Multiagent learning using a variable learning rate. *Artificial Intelligence*, 136(2), pp. 215-250.

Burges, C. J. (1998). A tutorial on support vector machines for pattern recognition. *Data mining and knowledge discovery*, 2(2), pp. 121-167.

Breiman, L., 1996. Bagging predictors. *Machine learning*, 24(2), pp. 123-140.

Breiman, L., 2001. Random forests. *Machine learning*, 45(1), pp. 5-32.

Bremer, M., and Sass, O., 2012. Combining airborne and terrestrial laser scanning for quantifying erosion and deposition by a debris flow event.*Geomorphology*, 138(1), pp. 49-60.

Brodu, N., and Lague, D., 2012. 3D terrestrial lidar data classification of complex natural scenes using a multi-scale dimensionality criterion: Applications in geomorphology. *ISPRS Journal of Photogrammetry and Remote Sensing*, 68, pp. 121-134.

Brunn, A., and Weidner, U., 1997. Extracting buildings from digital surface models. *International Archives of Photogrammetry and Remote Sensing*, 32(3 SECT 4W2), pp. 27-34.

Chang, C. C., and Lin, C. J., 2011. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology (TIST)*,2(3), 27.

Charaniya, A. P., Manduchi, R., and Lodha, S. K., 2004. Supervised parametric classification of aerial lidar data. *In Computer Vision and Pattern Recognition Workshop(CVPRW2004)*. pp. 30-30.

Chawla, N. V., Japkowicz, N., and Kotcz, A., 2004. Editorial: special issue on learning from imbalanced data sets. *ACM Sigkdd Explorations Newsletter*, 6(1), pp. 1-6.

Chehata, N., Guo, L., and Mallet, C., 2009. Airborne lidar feature selection for urban classification using random forests. *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 39(Part 3/W8), pp. 207-212.

Clifford, P., 1990. Markov random fields in statistics. *Disorder in physical systems: A volume in honour of John M. Hammersley*, pp. 19-32.

Cover, T., and Hart, P., 1967. Nearest neighbor pattern classification. *Information Theory, IEEE Transactions on*, 13(1), pp. 21-27.

Dempster, A. P., Laird, N. M., and Rubin, D. B., 1977. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the royal statistical society*, Series B (methodological), 1-38.

Desai, C., Ramanan, D., and Fowlkes, C. C., 2011. Discriminative models for multi-class object layout. *International journal of computer vision*, 95(1), pp. 1-12.

Ding, Y., Li, Y., and Yu, W., 2014. SAR image classification based on CRFs with integration of local label context and pairwise label compatibility. *Selected Topics in Applied Earth Observations and Remote Sensing, IEEE Journal of*,7(1), pp. 300-306.

Dold, C., and Brenner, C., 2006. Registration of terrestrial laser scanning data using planar patches and image data. *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, 36(5), pp. 78-83.

Douillard, B., Fox, D., and Ramos, F., 2008. Laser and Vision Based Outdoor Object Mapping. In *Robotics: Science and Systems*.

Drucker, H., Cortes, C., Jackel, L. D., LeCun, Y., & Vapnik, V. (1994). Boosting and other ensemble methods. *Neural Computation*, 6(6), 1289-1301.

Emgård, L., and Zlatanova, S., 2008. Implementation alternatives for an integrated 3D Information Model. In *Advances in 3D Geoinformation Systems* (pp. 313-329). Springer Berlin Heidelberg.

Figueiredo, M. A., and Jain, A. K., 2002. Unsupervised learning of finite mixture models. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 24(3), pp. 381-396.

Forkuo, E. K., and King, B., 2004. Automatic fusion of photogrammetric imagery and laser scanner point clouds. *International Archives of Photogrammetry and Remote Sensing*, 35, pp. 921-926.

Forlani, G., Nardinocchi, C., Scaioni, M., and Zingaretti, P., 2006. Complete classification of raw LIDAR data and 3D reconstruction of buildings. *Pattern Analysis and Applications*, 8(4), pp. 357-374.

Forsman, P., 2001. Three-Dimensional Localization and Mapping of Static Environments by Means of Mobile Perception. Ph.D Thesis, Helsinki University of Technology, Espoo, Finland.

Freund, Y., and Schapire, R. E., 1995. A desicion-theoretic generalization of on-line learning and an application to boosting. In *Computational learning theory*, pp. 23-37, Springer Berlin Heidelberg.

Freund, Y., and Schapire, R. E., 1996. Experiments with a new boosting algorithm. In ICML (Vol. 96, pp. 148-156).

Freund, Y., Schapire, R., and Abe, N., 1999. A short introduction to boosting. *Journal-Japanese Society For Artificial Intelligence*, 14(771-780), 1612.

Frey, B. J., and MacKay, D. J., 1998. A revolution: Belief propagation in graphs with cycles. *Advances in neural information processing systems*, pp. 479-485.

Fröhlich, B., Rodner, E., and Denzler, J., 2013. Semantic segmentation with millions of features: Integrating multiple cues in a combined random forest approach. In *Computer Vision–ACCV 2012* (pp. 218-231). Springer Berlin Heidelberg.

Galar, M., Fernandez, A., Barrenechea, E., Bustince, H., and Herrera, F., 2012. A review on ensembles for the class imbalance problem: bagging-, boosting-, and hybrid-based approaches. *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, 42(4), pp. 463-484.

Galleguillos, C., and Belongie, S., 2010. Context based object categorization: A critical survey. *Computer Vision and Image Understanding*, 114(6), pp. 712-722.

George Vosselman, Hans-Gerd Maas, 2010. Airborne and terrestrial laser scanning. Whittles Publishing.

Golovinskiy, A., Kim, V. G., and Funkhouser, T., 2009. Shape-based recognition of 3D point clouds in urban environments. In *12th IEEE International Conference on Computer Vision* (pp. 2154-2161).

Gopi, M., Krishnan, S., and Silva, C. T., 2000, September. Surface reconstruction based on lower dimensional localized Delaunay triangulation. In *Computer Graphics Forum* (Vol. 19, No. 3, pp. 467-478). Blackwell Publishers Ltd.

Gould, S., Rodgers, J., Cohen, D., Elidan, G., and Koller, D., 2008. Multi-class segmentation with relative location prior. *International Journal of Computer Vision*, 80(3), pp. 300-316.

Goulette, F., Nashashibi, F., Abuhadrous, I., Ammoun, S., and Laurgeau, C., 2006. An integrated on-board laser range sensing system for on-the-way city and road modelling. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 34(Part A).

Gueuning, F., Varlan, M., Eugene, C., and Dupuis, P., 1996. Accurate distance measurement by an autonomous ultrasonic system combining time-of-flight and phase-shift methods. In *Instrumentation and Measurement Technology Conference, 1996. IMTC-96. Conference Proceedings. Quality Measurements: The Indispensable Bridge between Theory and Reality., IEEE* (Vol. 1, pp. 399-404).

Hammersley, J. M., and Clifford, P., 1971. Markov fields on finite graphs and lattices.

Häselich, M., Arends, M., Lang, D., and Paulus, D., 2011. Terrain Classification with Markov Random Fields on fused Camera and 3D Laser Range Data. In *ECMR* (pp. 153-158).

Haralick, R. M., Shanmugam, K., and Dinstein, I. H., 1973. Textural features for image classification. *IEEE Transactions on Systems, Man and Cybernetics*, (6), pp. 610-621.

He, Xuming, Richard S. Zemel, and M. A. Carreira-Perpindn. Multiscale conditional random fields for image labeling. In *Proceedings of the IEEE computer society conference on Computer vision and pattern recognition (CVPR 2004)*,Vol. 2.

Hebel, M., and Stilla, U., 2008. Pre-classification of points and segmentation of urban objects by scan line analysis of airborne LiDAR data. *International Archives of*

Photogrammetry, Remote Sensing and Spatial Information Sciences, 37(B3a), pp. 105-110.

Heesch, D., and Petrou, M., 2010. Markov random fields with asymmetric interactions for modelling spatial context in structured scene labelling. *Journal of Signal Processing Systems*, 61(1), pp. 95-103.

Hershberger, J.E. and Snoeyink, J., 1992. *Speeding up the Douglas-Peucker line-simplification algorithm* (pp. 134-143). University of British Columbia, Department of Computer Science.

Heritage, G., and Large, A. (Eds.)., 2009. Laser scanning for the environmental sciences. John Wiley and Sons.

Himmelsbach, M., Luettel, T., and Wuensche, H., 2009. Real-time object classification in 3D point clouds using point feature histograms. In *IEEE/RSJ International Conference on Intelligent Robots and Systems,(IROS 2009)*, pp. 994-1000.

Hsieh, W. W., 2009. Machine learning methods in the environmental sciences: neural networks and kernels. Cambridge university press. pp. 146-p149.

Hosmer Jr, D. W., and Lemeshow, S., 2004. Applied logistic regression. John Wiley and Sons.

Hu, X., and Ye, L., 2013. A fast and simple method of building detection from Lidar data based on scan line analysis. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 3, W1.

Huang., J. Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference. Morgan Kaufmann, 1988.

Hyyppa, J., Kelle, O., Lehikoinen, M., and Inkinen, M., 2001. A segmentation-based method to retrieve stem volume estimates from 3-D tree height models produced by laser scanners. *IEEE Transactions on Geoscience and Remote Sensing*, 39(5), 969-975.

Imam, T., Ting, K. M., and Kamruzzaman, J., 2006. z-SVM: an SVM for improved classification of imbalanced data. In: *Advances in Artificial Intelligence* (pp. 264-273). Springer Berlin Heidelberg.

Jahangiri, M., Heesch, D., and Petrou, M., 2010. CRF Based Region Classification Using Spatial Prototypes. In *IEEE International Conference on Digital Image Computing: Techniques and Applications (DICTA,2010),* pp. 510-515.

Jebara, T., 2012. Machine learning: discriminative and generative (Vol. 755). Springer Science and Business Media.

Jiang, X., and Bunke, H., 1994. Fast segmentation of range images into planar regions by scan line grouping. *Machine vision and applications*, 7(2), pp. 115-122.

Kantola, T., Vastaranta, M., Lyytikäinen-Saarenmaa, P., Holopainen, M., Kankare, V., Talvitie, M., and Hyyppä, J., 2013. Classification of needle loss of individual Scots pine trees by means of airborne laser scanning. *Forests*, 4(2), pp. 386-403.

Kim, H. B., and Sohn, G., 2010. 3D classification of power-line scene from airborne laser scanning data using random forests. *Int. Arch. Photogramm. Remote Sens*, 38, pp. 126-132.

Kittler, J., Hatef, M., Duin, R. P., and Matas, J., 1998. On combining classifiers. *IEEE Transactions on Pattern Analysis and Machine Intelligence* , 20(3), 226-239.

Kolbe, T.H., Gröger, G. and Plümer, L., 2005. CityGML: Interoperable access to 3D city models. In *Geo-information for disaster management* (pp. 883-899). Springer Berlin Heidelberg.

Koller, D., and Friedman, N., 2009. Probabilistic graphical models: principles and techniques. MIT press.

Kotsiantis, S. B., Zaharakis, I., and Pintelas, P., 2007. Supervised machine learning: A review of classification techniques.

Kumar, J., Rottensteiner, F., and Soergel, U., 2012. Conditional random fields for lidar point cloud classification in complex urban areas. *ISPRS annals of the photogrammetry, remote sensing and spatial information sciences*, 1(3), pp. 263-268.

Kumar, S., and Hebert, M., 2006. Discriminative random fields. *International Journal of Computer Vision*, 68(2), pp. 179-201.

Kumar, U., Raja, S.K., Mukhopadhyay, C. and Ramachandra, T.V., 2011. Hybrid Bayesian classifier for improved classification accuracy. *Geoscience and Remote Sensing Letters, IEEE*, 8(3), pp.474-477.

Kůrková, V., 1992. Kolmogorov's theorem and multilayer neural networks. *Neural networks*, 5(3), pp. 501-506.

Lafferty, John, Andrew McCallum, and Fernando CN Pereira, 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data.

Lalonde, J. F., Unnikrishnan, R., Vandapel, N., and Hebert, M., 2005. Scale selection for classification of point-sampled 3D surfaces. In *Fifth International Conference on 3-D Digital Imaging and Modeling (3DIM 2005)*, pp. 285-292.

Lalonde, J. F., Vandapel, N., and Hebert, M., 2006. Automatic three-dimensional point cloud processing for forest inventory. *Robotics Institute*, 334.

Leamer, E. E., 1978. *Specification searches: Ad hoc inference with nonexperimental data* (Vol. 53). John Wiley and Sons Incorporated.p.91.

Lee, S. C., and Nevatia, R., 2003. Interactive 3D building modeling using a hierarchical representation. In *First IEEE International Workshop on Higher-Level Knowledge in 3D Modeling and Motion Analysis (HLK 2003)*, pp. 58-65.

Lehtomäki, M., Jaakkola, A., Hyyppä, J., Kukko, A., and Kaartinen, H., 2010. Detection of vertical pole-like objects in a road environment using vehicle-based laser scanning data. *Remote Sensing*, 2(3), pp. 641-664.

Li, Y., and Huttenlocher, D. P., 2008. Sparse long-range random field and its application to image denoising. In *Computer Vision–ECCV 2008* (pp. 344-357). Springer Berlin Heidelberg.

Liao, L., 2006. Location-based activity recognition. Ph.D Thesis, University of Washington, USA.

Liaw, A., and Wiener, M., 2002. Classification and regression by randomForest. *R news*, 2(3), pp. 18-22.

Lichti, D. D., Gordon, S. J., and Tipdecho, T., 2005. Error models and propagation in directly georeferenced terrestrial laser scanner networks. *Journal of surveying engineering*.

Lim, E. H., and Suter, D., 2008. Multi-scale conditional random fields for over-segmented irregular 3D point clouds classification. *In IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops(CVPRW2008)*, pp.1-7.

Lim, E. H., and Suter, D., 2009. 3D terrestrial LIDAR classifications with super-voxels and multi-scale Conditional Random Fields. *Computer-Aided Design*,41(10), pp. 701-710.

Liu, C., Yuen, J., and Torralba, A. 2011. Nonparametric scene parsing via label transfer. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(12), pp. 2368-2382.

Lodha, S. K., Fitzpatrick, D. M., and Helmbold, D. P., 2007. Aerial lidar data classification using adaboost. In *Sixth IEEE International Conference on 3-D Digital Imaging and Modeling (DIM 2007)*, pp. 435-442.

Luo C., Sohn G., 2013. Line-based Classification of Terrestrial Laser Scanning Data using Conditional Random Field. *ISPRS International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 1(2), pp. 155-160.

Luo, C., and Sohn, G., 2014. Scene-layout compatible conditional random field for classifying terrestrial laser point clouds. *ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences*, 1, pp. 79-86.

Manandhar, D., and Shibasaki, R., 2001. Feature extraction from range data. In *22nd Asian Conference on Remote Sensing*, Vol. 5, p. 9.

Matikainen, L., Kaartinen, H., and Hyyppä, J., 2007. Classification tree based building detection from laser scanner and aerial image data. *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, 36(Part 3), W52.

McLachlan, G., and Peel, D., 2000. Mixtures of factor analyzers. Finite Mixture Models, pp. 238-256.

Menard, S., 2002. Applied logistic regression analysis (Vol. 106). Sage.

Mercer, J., 1909. Functions of positive and negative type, and their connection with the theory of integral equations. *Philosophical transactions of the royal society of London. Series A, containing papers of a mathematical or physical character*, pp. 415-446.

Mooij, J. M., and Kappen, H. J., 2007. Sufficient conditions for convergence of the sum–product algorithm. *IEEE Transactions on Information Theory*, 53(12), pp.4422-4437.

Moosmann, F., Pink, O., and Stiller, C., 2009. Segmentation of 3D lidar data in non-flat urban environments using a local convexity criterion. In *IEEE on Intelligent Vehicles Symposium*, pp. 215-220.

Mukherjee, I., and Schapire, R. E., 2011. A theory of multiclass boosting. *arXiv preprint arXiv*:*1108.2989.*

Gamba, P., and Dell'Acqua, F.,2003. Increased accuracy multiband urban classification using a neuro-fuzzy classifier. *International Journal of Remote Sensing*, 24(4), pp. 827-834.

Munoz, D., Vandapel, N., and Hebert, M., 2008. Directional associative markov network for 3-d point cloud classification.

Munoz, D., Bagnell, J. A., Vandapel, N., and Hebert, M., 2009. Contextual classification with functional max-margin markov networks. In *IEEE Conference on Computer Vision and Pattern Recognition(CVPR 2009)*, pp. 975-982.

Murphy, K. P., Weiss, Y., and Jordan, M. I., 1999. Loopy belief propagation for approximate inference: An empirical study. In *Proceedings of the Fifteenth conference on Uncertainty in artificial intelligence* (pp. 467-475). Morgan Kaufmann Publishers Inc..

Najafi, M., Namin, S. T., Salzmann, M., and Petersson, L., 2014. Non-Associative Higher-Order Markov Networks for Point Cloud Classification. In *Computer Vision–ECCV 2014 (pp. 500-515).* Springer International Publishing.

Niemeyer, J., Wegner, J. D., Mallet, C., Rottensteiner, F., and Soergel, U., 2011. Conditional random fields for urban scene classification with full waveform LiDAR data. In *Photogrammetric Image Analysis (pp. 233-244)*. Springer Berlin Heidelberg.

Nguyen, M. Q., Atkinson, P. M., and Lewis, H. G., 2005. Super resolution mapping using a Hopfield neural network with LIDAR data. *IEEE on Geoscience and Remote Sensing Letters*, 2(3), pp. 366-370.

Nüchter, A., and Hertzberg, J., 2008. Towards semantic maps for mobile robots. *Robotics and Autonomous Systems*, 56(11), pp. 915-926.

Oliva, A., and Torralba, A., 2007. The role of context in object recognition. *Trends in cognitive sciences*, 11(12), pp. 520-527.

Pang, B., and Lee, L., 2008. Opinion mining and sentiment analysis. *Foundations and trends in information retrieval*, 2(1-2), pp. 1-135.

Pearl., J., 1998. Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference. Morgan Kaufmann.

Pearl, J., 2000. Causality: models, reasoning and inference (Vol. 29). Cambridge: MIT press.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V. and Vanderplas, J., 2011. Scikit-learn: Machine learning in Python. *The Journal of Machine Learning Research*, 12, pp.2825-2830.

Permuter, H., and Francos, J., 2003. Gaussian mixture models of texture and colour for image database retrieval. In *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP'03)*, Vol. 3, pp. III-569.

Pfeifer, N. and Briese, C., 2007. *Laser scanning-Principles and applications*. na.

Pfeifer, N., Dorninger, P., Haring, A., and Fan, H., 2007. Investigating terrestrial laser scanning intensity data: quality and functional relations (pp. 328-337). na.

Posner, I., Schroeter, D., and Newman, P., 2007. Describing composite urban workspaces. In *IEEE International Conference on Robotics and Automation*, pp. 4962-4968.

Posner, I., Cummins, M., and Newman, P., 2009. A generative framework for fast urban labeling using spatial and temporal context. *Autonomous Robots*, 26(2-3), pp. 153-170.

Premebida, C., Ludwig, O., and Nunes, U., 2009. Exploiting lidar-based features on pedestrian detection in urban scenarios. *In 12th International IEEE Conference on Intelligent Transportation Systems(ITSC2009)*, pp. 1-6.

Priestnall, G., Jaafar, J., and Duncan, A., 2000. Extracting urban features from LiDAR digital surface models. *Computers, Environment and Urban Systems*,24(2), pp. 65-78.

Prokhorov, D. V., 2009. Object recognition in 3d lidar data with recurrent neural network. In *IEEE Conference on Computer Society*, pp. 9-15.

Pu, S. and G. Vosselman, 2009. Knowledge based reconstruction of building models from terrestrial laser scanning data. *ISPRS journal of photogrammetry and remote sensing* 64(6): pp. 575-584.

Quinlan, J. R., 1986. Induction of decision trees. *Machine learning*, 1(1), pp. 81-106.

Quinlan, J. R., 1987. Simplifying decision trees. *International journal of man-machine studies*, 27(3), pp. 221-234.

Rabinovich, A., Vedaldi, A., Galleguillos, C., Wiewiora, E., and Belongie, S., 2007. Objects in context. In *IEEE 11th international conference on Computer vision (ICCV 2007)*, pp. 1-8.

Rennie, J. D. (2005). Regularized logistic regression is strictly convex. Unpublished manuscript. URL people. csail. mit. edu/jrennie/writing/convexLR. pdf.

Reshetyuk, Y., 2009. Self-calibration and direct georeferencing in terrestrial laser scanning.

Reynolds, D. A., 1995. Speaker identification and verification using Gaussian mixture speaker models. *Speech communication*, 17(1), pp. 91-108.

Riegl, 2010. System Configuration 3D Terrestrial Laser Scanner. www.riegl.com.

Ripley, B., Venables, W., & Ripley, M. B. (2015). Package 'nnet'.

Rumelhart, D. E., Durbin, R., Golden, R., and Chauvin, Y., 1995. Backpropagation: The basic theory. *Backpropagation: Theory, Architectures and Applications*, pp. 1-34.

Rutzinger, M., Höfle, B., Hollaus, M. and Pfeifer, N. 2008. Object-Based Point Cloud Analysis of Full-Waveform Airborne Laser Scanning Data for Urban Vegetation Classification. *Sensors*, Vol. 8(8), pp. 4505-4528.

Saxena, A., Driemeyer, J., and Ng, A. Y., 2008. Robotic grasping of novel objects using vision. *The International Journal of Robotics Research*, 27(2), pp. 157-173.

Schindler, K., 2012. An overview and comparison of smooth labeling methods for land-cover classification. *IEEE Transactions on Geoscience and Remote Sensing*, 50(11), pp. 4534-4545.

Schmidt, M., 2007. UGM: Matlab code for undirected graphical models. https://www.cs.ubc.ca/~schmidtm/Software/UGM.html

Schmidt, A., Rottensteiner, F., and Soergel, U., 2012. Classification of airborne laser scanning data in Wadden sea areas using conditional random fields. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, XXXIX-B3,* pp. 161-166.

Schulz, T., 2007. Calibration of a terrestrial laser scanner for engineering geodesy. Ph.D. Thesis, Technical University of Berlin.

Settles, B., 2004. Biomedical named entity recognition using conditional random fields and rich feature sets. In *Proceedings of the International Joint Workshop on Natural Language Processing in Biomedicine and its Applications* (pp. 104-107). Association for Computational Linguistics.

Shapovalov, R., et al., 2010. Non-associative markov networks for 3D point cloud classification. *Networks* 38(Part 3A): 103-108.

Shotton, J., Winn, J., Rother, C., and Criminisi, A., 2006. Textonboost: Joint appearance, shape and context modeling for multi-class object recognition and segmentation. In *Computer Vision–ECCV 2006* (pp. 1-15). Springer Berlin Heidelberg.

Shotton, J., Winn, J., Rother, C., and Criminisi, A., 2009. Textonboost for image understanding: Multi-class object recognition and segmentation by jointly

modeling texture, layout, and context. *International Journal of Computer Vision*,81(1), pp. 2-23.

Sithole, G., and Vosselman, G., 2003. Automatic structure detection in a point-cloud of an urban landscape. In *2nd GRSS/ISPRS Joint IEEE Workshop on Remote Sensing and Data Fusion over Urban Areas*, pp. 67-71.

Smyth, P., 1997. Belief networks, hidden Markov models, and Markov random fields: a unifying view. *Pattern recognition letters*, 18(11), pp. 1261-1268.

Soh, L. K., and Tsatsoulis, C., 1999. Texture analysis of SAR sea ice imagery using gray level co-occurrence matrices. *IEEE Transactions on Geoscience and Remote Sensing*, 37(2), pp. 780-795.

Song, X., 1999. Contextual pattern recognition with applications to to biomedical image identification. Ph.D. thesis, California Institute of Technology.

Stamos, I., Hadjiliadis, O., Zhang, H., and Flynn, T., 2012. Online algorithms for classification of urban objects in 3d point clouds. *In Second IEEE International Conference on 3D Imaging, Modeling, Processing, Visualization and Transmission (3DIMPVT)*, pp. 332-339.

Strat, T. M., and Fischler, M. A., 1991. Context-based vision: recognizing objects using information from both 2D and 3D imagery. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(10), pp. 1050-1065.

Strat, T. M., 1993. Employing contextual information in computer vision. *DARPA93*, pp. 217-229.

Toussaint, G. T., 1978. The use of context in pattern recognition. *Pattern Recognition*, 10(3), pp. 189-204.

Taniguchi, M., and Tresp, V., 1997. Averaging regularized estimators. *Neural Computation*, 9(5), pp. 1163-1178.

Tax, D. M., Van Breukelen, M., Duin, R. P., and Kittler, J., 2000. Combining multiple classifiers by averaging or by multiplying?. *Pattern recognition*, 33(9), pp. 1475-1485.

Torralba, A., 2003. Contextual priming for object detection. *International journal of computer vision*, 53(2), pp. 169-191.

Trappenberg, T. P., and Back, A. D., 2000. A classification scheme for applications with ambiguous data. In *Proceedings of the IEEE-INNS-ENNS International Joint Conference on Neural Networks(IJCNN 2000)*, Vol. 6, pp. 296-301.

Triebel, R., Kersting, K., and Burgard, W., 2006. Robust 3D scan point classification using associative Markov networks. In *Proceedings of IEEE International Conference on Robotics and Automation (ICRA 2006)*, pp. 2603-2608.

UNPA, 2014. State of World Population 2014. http://www.unfpa.org/swop.

Vail, D. L., Lafferty, J. D., and Veloso, M. M., 2007. Feature selection in conditional random fields for activity recognition. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2007)*, pp. 3379-3384.

Vale, A., and Mota, J. G., 2004. Detection and Classification of Clearance Anomalies on Over-Head Power Lines. Albatroz Engenharia, Controlo.

Vallet, B., Brédif, M., Serna, A., Marcotegui, B., and Paparoditis, N., 2015. TerraMobilita/iQmulus urban point cloud analysis benchmark. *Computers and Graphics*.

Vandapel, N., Huber, D. F., Kapuria, A., and Hebert, M., 2004. Natural terrain classification using 3-d ladar data. In *Proceedings of IEEE International Conference on Robotics and Automation (ICRA2004)*, Vol. 5, pp. 5117-5122.

Vehmas, M., Eerikäinen, K., Peuhkurinen, J., Packalén, P., and Maltamo, M., 2009. Identification of boreal forest stands with high herbaceous plant diversity using airborne laser scanning. *Forest Ecology and Management*, 257(1), pp. 46-53.

Vijayanarasimhan, S., and Grauman, K., 2012. Active frame selection for label propagation in videos. In *Computer Vision–ECCV 2012* (pp. 496-509). Springer Berlin Heidelberg.

Vishwanathan, S. V. N., Schraudolph, N. N., Schmidt, M. W., and Murphy, K. P., 2006. Accelerated training of conditional random fields with stochastic gradient methods. In *Proceedings of the 23rd international conference on Machine learning* (pp. 969-976). ACM.

Vosselman, G., 1999. Building reconstruction using planar faces in very high density height data. *International Archives of Photogrammetry and Remote Sensing*, 32(3; SECT 2W5), pp. 87-94.

Wallach, H. M., 2004. Conditional random fields: An introduction. Technical Reports (CIS), 22.

Wang, L. (Ed.), 2005. Support Vector Machines: theory and applications (Vol. 177). Springer Science and Business Media.

Wang, J., and Shan, J., 2009. Segmentation of LiDAR point clouds for building extraction. In *American Society for Photogramm. Remote Sens. Annual Conference*, Baltimore, MD, pp. 9-13.

Wegner, J. D., Montoya-Zegarra, J. A., and Schindler, K., 2013. A higher-order CRF model for road network extraction. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1698-1705.

Wehr, A., & Lohr, U. (1999). Airborne laser scanning—an introduction and overview. ISPRS Journal of Photogrammetry and Remote Sensing, 54(2), 68-82.

Weinmann, M., Jutzi, B., and Mallet, C., 2013. Feature relevance assessment for the semantic interpretation of 3D point cloud data. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 5, W2.

Weiss., Y. Belief propagation and revision in networks with loops, 1997. Technical Report 1616, MIT AI lab.

Wellington, C., Courville, A. C., and Stentz, A., 2005. Interacting Markov Random Fields for Simultaneous Terrain Modeling and Obstacle Detection. In *Robotics: Science and Systems*, pp. 1-8.

Winkler, G., 1995. Image Analysis, Random Fields and Dynamic Monte Carlo Methods. Springer Verlag.

Winkler, G., 2003. Image analysis, random fields and Markov chain Monte Carlo methods: a mathematical introduction (Vol. 27). Springer Science and Business Media.

Winn, J., and Shotton, J., 2006. The layout consistent random field for recognizing and segmenting partially occluded objects. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Vol. 1, pp. 37-44.

Wolf, P. R., and Dewitt, B. A., 2000. Elements of Photogrammetry: with applications in GIS (Vol. 3). New York: McGraw-Hill.

Wu, T. F., Lin, C. J., & Weng, R. C., 2004. Probability estimates for multi-class classification by pairwise coupling. *The Journal of Machine Learning Research*,5, pp. 975-1005.

Yang, M. Y., Förstner, W., andDrauschke, M., 2010. Hierarchical Conditional Random Field for Multi-class Image Classification. In *VISAPP (2)*, pp. 464-469.

Yang, M. Y., and Rosenhahn, B., 2014. Video segmentation with joint object and trajectory labeling. In *IEEE Winter Conference on Applications of Computer Vision (WACV 2014)*, pp. 831-838.

Zhang, J., and Sohn, G., 2010. A Markov Random Field Model for individual tree detection from airborne laser scanning data. In *Proceedings of Photogrammetric Computer Vision (PCV)* , pp. 01-03.

Zhang, R., and Ding, X., 2001. Offline handwritten numeral recognition using orthogonal Gaussian mixture model. In *Proceedings of International Conference on Image Processing*, Vol. 1, pp. 1126-1129.

Zhao, H., Liu, Y., Zhu, X., Zhao, Y., and Zha, H., 2010. Scene understanding in a large dynamic environment through a laser-based sensing. In *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 127-133.

Zhu, J., Zou, H., Rosset, S., and Hastie, T., 2009. Multi-class adaboost. *Statistics and its Interface*, 2(3), pp. 349-360.

Zhu, X., and Ghahramani, Z., 2002. Learning from labeled and unlabeled data with label propagation. Technical Report CMU-CALD-02-107, Carnegie Mellon University.

Zlatanova, S., 2008. SII for emergency response: the 3D challenges. *J. Chen, J. Jiang and S. Nayak* (Eds.), pp.1631-1637.

Zitnick, C. L., Parikh, D., and Vanderwende, L., 2013. Learning the visual interpretation of sentences. In *IEEE International Conference on Computer Vision (ICCV 2013)*, pp. 1681-1688.