Multi-Modal Learning For Adaptive Scene Understanding

Charika Sanjeewani De Alvis Weerasiriwardhane

A thesis submitted in fulfillment
of the requirements for the degree of
Doctor of Philosophy

Faculty of Engineering and Information Technologies
University of Sydney

2017

Declaration

I hereby declare that this submission is my own work and that, to the best of my knowledge and belief, it contains no material previously published or written by another person nor material which to a substantial extent has been accepted for the award of any other degree or diploma of the University or other institute of higher learning, except where due acknowledgement has been made in the text.

Charika Sanjeewani De Alvis Weerasiriwardhane

January, 2017

Abstract

Charika De Alvis
University of Sydney

Doctor of Philosophy
January 2017

Multi-Modal Learning For Adaptive Scene Understanding

Modern robotics systems typically possess sensors of different modalities such as colour cameras, inertial measurement units, and 3D laser scanners to perceive their environment. While there are undeniable benefits to combine sensors of different modalities the process tends to be complicated. Segmenting scenes observed by the robot into a discrete set of classes is a central requirement for autonomy as understanding the scene is the first step to reason about future situations. Equally, when a robot navigates through an unknown environment, it is often necessary to adjust the parameters of the scene segmentation model online to maintain the same level of accuracy in changing situations. This thesis explores efficient means of adaptive semantic scene segmentation in an online setting with the use of multiple sensor modalities

In computer vision many successful methods for scene segmentation are based on conditional random fields (CRF) where the maximum a posteriori (MAP) solution to the segmentation problem can be obtained by efficient inference. CRF encodes contextual information and longer-range relationships during the prediction process. Further, parameters learning of CRFs is a also widely studied area This thesis offers three main contributions.

First, we devise a novel CRF inference method for scene segmentation that incorporates global constraints, enforcing particular sets of nodes to be assigned the same class label. To do this efficiently, the CRF is formulated as a relaxed quadratic program whose MAP solution is found using a gradient-based optimisation approach. These global constraints are useful, since they can encode "a priori" information about the final labeling. This new formulation also reduces the dimensionality of the original image-labeling problem, which result in a decrease of the computational time. The proposed globally constrained CRF is employed in an urban street scene understanding task. Camera data is used for the CRF based semantic segmentation while global constraints are derived from 3D laser point clouds. Experimental results demonstrate the improvement achieved with global constraints. Comparisons with higher order potential CRF show the benefits of the proposed method.

Second, an approach to learn CRF parameters without the need for manually labelled training data is proposed. Parameter learning is of high importance when extending scene segmentation to an online setting since the nature of the input

data is unknown. The model parameters are estimated by optimising a novel loss function using self supervised reference labels. These reference labels are obtained purely based on the information from camera and laser, in a self-training manner with minimum amount of human supervision. Sensor data is pre-processed using methods such as convolutional nets, discriminant analysis, and Euclidean distance based clustering to extract reference labels

Third, an approach that can conduct the parameter optimisation while increasing the model robustness to non-stationary data distributions in the long trajectories of the robot is proposed. We adopted stochastic gradient descent to achieve this goal by using a learning rate that can appropriately grow or diminish to gain adaptability to changes in the data distribution. We demonstrate experimental results on KITTI dataset for long real world image sequences.

Acknowledgments

I would like to express special gratitude to my supervisor Fabio Ramos for the strong support and guidance you have provided to conduct quality research. The freedom to explore and timely feedback were incredibly helpful to make this journey a success. I would also thank NICTA for the financial support during the period my candidature. Furthermore I would also like to thank Lionel Ott for the support and collaborations throughout in the development of the thesis. My special thanks to all of my colleagues in Fabio's group for the interesting chats and the shared ideas. I would finally like to express my appreciation towards my loving husband, and to my mother for the confidence you built in me.

Nomenclature

General

$P(A)$	Probability of event A
$P(A\|B)$	Probability of event A given event B
K^{-1}	Matrix inversion
K^T	Matrix transpose
K_{ij}	Element of matrix K at row i and column j
L	Set of Labels
n	Number of labels
$\mathbf{x}$	Multidimensional observed variable
D	Dimension of $\mathbf{x}$
x_i	i^{th} element of vector $\mathbf{x}$
$\|\|\mathbf{x}\|\|$	L^2 norm of vector $\mathbf{x}$
$\mathbf{y}$	Multidimensional target variable
θ	Model parameters
θ^*	Optimal parameter values
$g(\mathbf{x})$	Function over $\mathbf{x}$

Classificication

a_i	Training data samples in class i
k_i	Number of data samples in a_i
$\bar{\mathbf{x}}_i$	Average of the data samples in a_i
Γ_1	Intra class covariance matrix
Γ_2	Inter class covariance matrix
Φ	Linear transformation matrix

Conditional Random Fields

G	Undirected graph
V	Set of vertices
$\mathbf{E}$	Set of edges
C	Set of cliques in a graph
m	Number of nodes in the graph
$\mathbb{N}_i$	Neighbours of node i
$\mathcal{Z}$	Normalising function

y_i	Trarget variable correspond to node i.
$\psi_i(y_i)$	Unary potental of node i
$\psi_{ij}(y_i, y_j)$	Pairwise potental between nodes i and j
$E(\mathbf{y}\|\theta)$	Energy of the model given parameters
Ω	Training set
N	Number of training samples
K	Number of parameters
$\mathrm{L}(\theta\|\Omega)$	Likelihood of θ given data
$l(\theta\|\Omega)$	Log likelihood function
$\mathbb{E}(f(x))$	Expected value of $f(x)$
$\mathbb{C}$	Global constraints

Convex Relaxation

$I(y_i), I(y_i, y_j)$	Indicator variables of label assignment
H	Edge potential matrix

Equality Constrained Quadratic Programming

Q	Negative edge potential matrix
A	Equality constraints matrix
e	Number of equality constraints
λ	Lagrange multiplier vector
$null(A)$	Null space of matrix A
Z	Basis for the null space
$\tilde{Q}$	Reduced hessian matrix

Belief Propagation

$m_{i\to j}$	Message from i to j
B_i	Belief in node i
$\mathbb{N}_{i/j}$	Neighbourhood of i except j

Optimisation

w	Image frame index
ω	Data sample
E	Expected risk

E_N	Empirical risk
η	Global learning rate
t	Iteration index
θ_t	Value of parameter at t^{th} iteration
γ	Global learning rate for SGD
$\bigtriangledown f(x)$	Gradient of f at x
α, λ	Loss function parameters
$\mathbf{z}$	Ground truth labels
B	Mini batch size

3D Point Cloud Processing

M	Horizontal plane model
h_0	Number of points required to learn parameters of M
P_n	Observed data distribution
T	Number of iterations
S_i	Set of points fit with the model
$\bar{S}_i$	Consensus set of S_i
ν	Probability of outlier occurrence
ϵ, γ	Threshold values
$\mathcal{R}$	kd tree formulation of point cloud
$Q_{\mathcal{R}}$	Queue of points
$\mathcal{C}_L$	List of clusters
d_n	Radius of point neighbourhood
d_u	Upper bound
$\mathcal{C}_i$	Cluster i

Visual Features and Metrics

l, a, b	LAB color metrics
x, y	Image pixel location coordinates
ω	Data sample
D_{xy}	Distance in $x - y$ coordinate frame
D_{lab}	Distance in LAB color space
D_r	Distance between cluster centers
D_B	Bhattacharyya Distance
D_E	Euclidean Distance
$\mathcal{M}$	Superpixel count
S	Superpixel size

m	Pixel count
ξ	Compactness indicator
N_B	Number of bins in a histogram

Abbreviations

CRF	Conditional Random Field
FCN	Fully Convolutional Net
GD	Gradient descent
GPS	Global positioning system
HOG	Histogram of oriented gradients
HOP	Higher order potentials
ICM	Iterative conditional modes
ILP	Integer linear programming
LBP	Loopy bilief propagation
LDA	Linear discriminant analysis
ML	Maximum Likelihood
MAP	Maximum a Posteriori
MCMC	Markov chain Monte Carlo
pLDA	Pseudo linear discriminant analysis
QP	Quadratic programming
RANSAC	Random sample consensus
SGD	Stochastic gradient descent
SIFT	Scale-Invariant Feature Transform
SLIC	Simple linear iterative clustering
UGM	Undirected graphical models
ADAGRAD	Adaptive gradient algorithm

Conference Papers

The contributions of the thesis are based on the following conference papers where I led the research, designed the experimental setup, conducted the experiments, drew conclusions and wrote the manuscript under the supervision of Dr. Lionel Ott and A/Prof. Fabio Ramos.

Charika De Alvis*, Lionel Ott, and Fabio Ramos. Urban scene segmentation with laser-constrained crfs. In *International Conference On Intelligent Robots and Systems(IROS)*, 2016.

Charika De Alvis*, Lionel Ott, and Fabio Ramos. Online learning for scene segmentation with laser- constrained crfs. In *International Conference on Robotics and Automation(ICRA)*, 2017. To be published.

* - Lead Author and Corresponding Author

Students'Name: Charika Sanjeewani De Alvis Weerasiriwardhane

Signature:.....charM..... **Date:**...30/01/2017.....

As supervisor for the candidature upon which this thesis is based, I can confirm that the authorship attribution statements above are correct.

Supervisor's Name: Fabio Ramos

Signature:.....Fabio Ramos..... **Date:**...30/01/2017.....

Contents

List of Figures

List of Tables

List of Algorithms

Chapter 1

Introduction

1.1 Motivation

Intelligent autonomous systems are becoming increasingly popular in society. Driver assistance, robotic navigation and environmental exploration all include a level of autonomy. Autonomous driving is highly beneficial because it contributes to reducing road accidents, creating orderly traffic flow, optimising fuel consumption and providing mobility for the elderly and people with disabilities. Under autonomous driving, there are many areas of active research, such as road, vehicle, traffic sign and pedestrian detection and understanding. Therefore, it is necessary to establish a semantic and geometrical understanding of the changing environment surrounding the vehicle. For this purpose, identifying the precise class boundaries is critical. Object or class recognition is usually achieved through labelling every pixel of the image with a chosen class or object label. Figure 1.1 shows a semantically segmented street scene to 12 distinct classes. Google's self-driving car is arguably a successful attempt to fully automate the task of autonomous driving. However, complete autonomy is still infeasible due to the nonlinearities in real world applications. Figure 1.2 is an image of the Google self-driving car with a 360° Velodyne scanner.

Scene understanding is commonly studied in the context of autonomous driving and comes with major challenges. A successful scene understanding algorithm should have the capability to accommodate rich contextual information in the process of segmenting the image over accurate class boundaries and subsequently assigning class labels to each segment. For image labelling problems, Conditional Random Fields (CRF) are commonly used because they can integrate different levels of contextual information. CRF has unary potentials that can capture low-level cues derived from local texture, colour and location of the pixels and pairwise potentials that can assist in smoothing label predictions. Figure 1.3 depicts a simple CRF model built over image patches. Higher-level cues such as label consistency in regions, object co-occurrence statistics and shape information can be incorporated in the CRF model through higher order potentials or hierarchical connectivity.

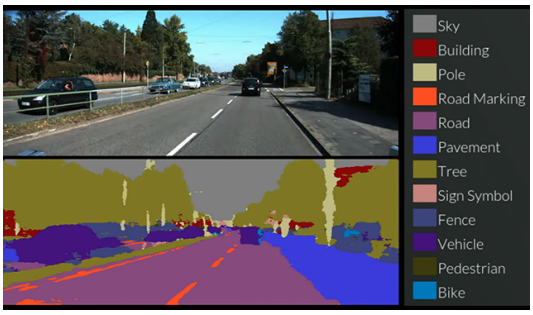


Figure 1.1: Illustration of semantic segmentation of an image. Pixels correspond to different object classes are individually. From: http://mi.eng.cam.ac.uk/projects/segnet/.

Figure 1.2: Google self driving car(courtesy NASA/JPL-Caltech).

Higher-level cues that contain longer-range information such as label consistency over image regions are typically derived using unsupervised image segmentation methods such as clustering. Clustering algorithms group similar data points within the given data. The similarity of the data in a group provides information about label consistency. The importance of clustering is that it requires minimum human supervision and can be modelled to adapt to changing environments. Furthermore, clustering algorithms do not require assumptions on the input data since they group data into separate clusters based on the dissimilarity metrics. Moreover, clustering is an attractive technique to reduce the dimensionality of the data and also especially suitable for processing sparse data. For sparse 3D laser point cloud processing, clustering techniques are efficiently implemented [90]. Figure 1.5 shows clusters generated from a 3D point cloud where each cluster correspond to an object or a part of an object. However, the accuracy of the final solution of CRF based semantic scene segmentation models is limited by the accuracy of the associated unsupervised methods.

For accurate scene labelling output it is necessary to learn the CRF parameters corresponding to the input data distribution, which requires inference over the CRF model.

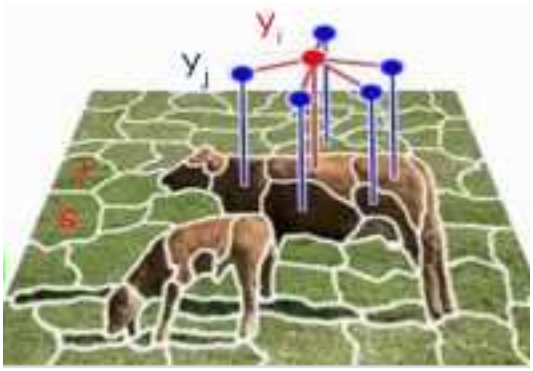


Figure 1.3: Autonomous driving platform Annieway with multiple sensor modalities (courtesy Annieway /KITTI).

The scene segmentation problem typically formulated by grid shaped CRF and it results in complicated dependencies in the model, further it also involves multiple classes/states, essentially rendering the inference problem intractable. Under these circumstances, accurate CRF parameter learning can be a challenging task.

Typically, CRFs are used to encode visual information, such as colour and texture for scene classification. However, it is evident that combination of multiple modalities can be beneficial, i.e. using depth information in addition to visual data can increase the robustness to changes in illumination and texture; and as a result, we see that contemporary robots are comprised of multiple sensors such as optical cameras, Velodynes, sonars, thermal cameras and flash lidars. Figure 1.4 shows autonomous driving platform of KITTI vision benchmark suite with multiple modalities mounted on it. Laser-based depth information is commonly used with visual information to CRF modelling. The tendency to use multiple modalities has drawn more attention towards efficient sensor fusion techniques.

Another consideration is that the location and orientation of each sensor may vary from each other. As a result, the visibility of an object might change from sensor to sensor due to occlusions. Laser sensors usually can perceive objects in a shorter range, while cameras can capture objects much further. This shows that, some sensors are capable of recognising particular object classes better than others. Therefore, efficient sensor fusion requires representing all different sensor inputs in one single domain. This is a complex task, and so substantial research is being conducted on fusing sensors to get the maximum use of input data. Apart from that the scene segmentation model should have the flexibility to fuse information from any new sensor input introduced to the system.

Semantic scene segmentation directly links with autonomous driving. Scene labelling models are learnt on training datasets, eventhough the model has to operate on newly encountered data. In scenarios where the autonomous vehicle

Figure 1.4: Part of a conditional random field built over an image patches. Demonstrates characterization of contextual information. From: http://spark-university.s3.amazonaws.com/stanford-pgm/slides/2.4.1-Repn-MNs-pairwise.pdf.

navigates in unknown changing environments, maintaining a high level of accuracy of image labelling is very challenging. One option is to use large datasets of labelled samples to train the model, since it should improve the generalisation properties. But this involves large amounts of human supervision, increases computational complexity and time consumption. In other words, autonomy becomes an unrealistic goal. Additionally, it is impractical to obtain labelled data when navigating in unknown environments. There can be an infinite number of different routes, changing weather, lighting conditions, traffic conditions and so on. Therefore it is essential to generate means of establishing adaptability in unexpected scenarios.

When developing adaptive scene segmentation models for autonomous driving, online learning plays a significant role. As new data instances are observed, CRF model parameters can be updated in an online setting. Batch optimisers that update parameter with the use of gradient and Hessian computations accumulated over the complete training set are a popular choice for parameter learning. However, in autonomous driving, the relevant data stream is continuous. As the vehicle moves, new data flows in which makes it hard to define a fixed-length batch of data. Furthermore, using batch optimizing on past data also can be infeasible due to the sheer amount of information. Consequently, stochastic gradient-based methods that update the model parameters based on the gradient over a single data instance are commonly utilised in place of batch optimisers. Stochastic methods scale appropriately with advanced computing resources and are also resilient to the inaccuracies that occur when approximating the gradients.

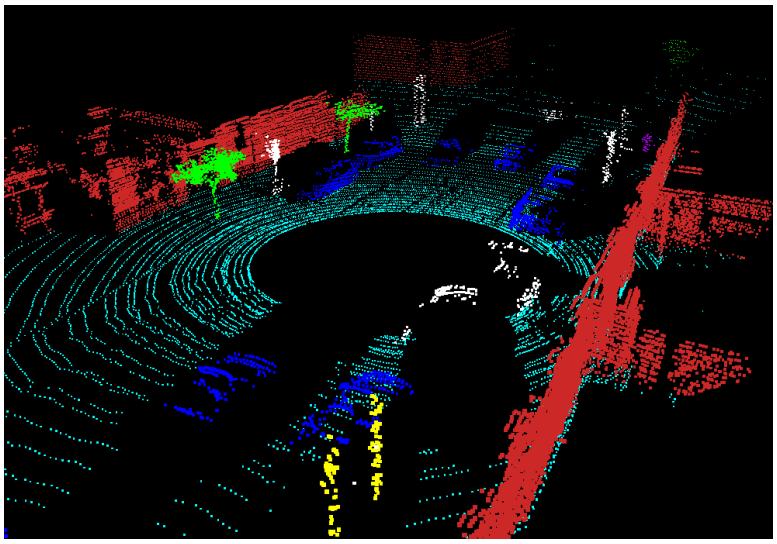


Figure 1.5: Point clusters generated from a 3D Velodyne point cloud conrrespond to individual objects. From: http://www.roboticsproceedings.org/rss05/p22.pdf.

Additionally, there is evidence showing that models trained using stochastic gradient descent(SGD) tend to have lower generalisation errors compared to batch learning methods.

However, using SGD in an online setting comes with challenges. Selecting an optimum learning rate for SGD methods can be complicated since larger learning rates result in divergence from the optimum parameter values and smaller learning rates can make the learning process extremely slow. Slow adaptation is unsuitable for real-time operation, especially when navigating in an urban environment, where it is essential to understand the environment in a real-time manner. Hence we need a way to optimise the parameters of the CRF efficiently and accurately. It has been shown in the literature that decreasing learning rates guarantee the convergence of SGD, and so diminishing learning rate is commonly used in practice. However, for non-stationary data distributions, the optimiser might become trapped in a local minimum as the the learning rate becomes infinitesimal, and so new information cannot be learned. Even fixed learning rates cannot address this issue, therefore it is important to have an adaptable learning rate that can increase or decrease according to changes in the data distribution.

1.2 Problem Statement

The thesis addresses the following critical issues in autonomous navigation. Initially, it focuses developing a convenient way of including “a priori ”knowledge about correct labelling to the scene segmentation model in the optimisation process, because this type of additional information is readily available and can be used to enhance the quality of scene understanding. This “a priori ”knowledge

is commonly obtained by combining information from different modalities. Secondly, we consider the problem in the context of typical autonomous platforms that consist of cameras and laser sensors as the primary sensors of interest. In this scenario, using laser-based information as additional knowledge to image based scene segmentation models has to be analysed further to increase the efficiency. Thirdly, this thesis addresses the issues involve with scene segmentation during long-term navigation where the main problem concerns modelling adaptability in changing environments. The core of the thesis thus demonstrates a method that can adapt to the variations in the perceived environment through an efficient parameter learning method. Furthermore, the computational cost of the semantic segmentation of an image frame is minimised, thus facilitating real-time operation. Finally, we explore means of eliminating the need for manually labelled data during the learning process.

1.3 Contributions

The major contributions of this thesis are as follows:

1. **A novel CRF formulation using global constraints capable of enforcing label consistency in a semantic scene segmentation model for autonomous driving. An application of the proposed method is demonstrated for urban street scene segmentation using camera and laser sensor data gathered by real robotic platforms**.
   CRF can be used to model the image labeling problem. Label prediction is formulated as the maximum a posteriori (MAP) estimation problem of CRF. Quadratic programming (QP) formulation is one of the most efficient means of solving the CRF inference problem. We propose an inference method to include "a priori" information about label consistency in the form of constraints. A side effect of the use of these constraints is a large reduction of the problem's dimensionality which facilitate real-time operation. Experiments shows how constrained CRF is used to efficiently fuse camera and laser sensors efficiently. The CRF model is formulated based on visual features obtained from camera images, and global constraints that enforce label consistency are extracted from the laser point cloud. This approach enhances the model's resilience to changes in lighting conditions and occlusions.

2. **Developing an approach for CRF parameters learning eliminating the need for manually labelled training data.**

CRF parameters estimation is essential to efficiently combine the different cues associated with the model since it enriches the adaptability of the model. Parameter learning is formulated on the optimisation of a loss function.

Our approach derives the reference labels necessary for the loss computation in a self supervised manner based on the outputs from a discriminant analysis classifier and a fully convolutional network combined with laser point based segments corresponding to objects in the image. This approach minimises the human supervision in learning by providing means for the model to automatically learn from unseen instances.

3. **A stochastic gradient based method to update CRF parameters while making the model robust to non-stationary data in long-term navigation.**

   The Semantic scene segmentation model is extended to an online learning algorithm, where the model updates its parameters to predict image labels more accurately over time. Since the input data stream is large and unknown, stochastic gradient descent is used to optimise the loss as new data is received. The learning rate is continuously adjusted, both decreasing over time to allow the model to reach an optimum point and increasing when necessary to leap out of a local minima. The proposed model has the capability to maintain or improve the accuracy of its initial estimates as the perceiving environment changes.

1.4 Thesis Outline

This section summarises the content of the thesis. The goal is to develop an efficient framework for scene understanding to assist with autonomous driving. Accurate scene understanding is expected to be achieved through the addition of "a priori" knowledge in the form of global constraints, while getting the maximum use of the input data from multiple modalities. The model is designed to adpat to the changes in the environment when navigating in urban environments.

Chapter 2: Theoretical Background

This chapter describes the fundamental theories and techniques necessary to develop the original contributions of the thesis. It starts with an introduction to supervised learning (section 2.1) and then details an specific classification algorithm, discriminant analysis. Afterwards, the theory of conditional random fields (CRFs) is explained (section 2.2). Under this section, the formulation of CRF,

general inference, inference using programming relaxations and parameter learning of CRF are described. Sections 2.3 and 2.4 detail the batch gradient descent and stochastic gradient descent methods, which are commonly used in parameter learning. Finally, section 2.5 presents information on sensor data processing, focusing specifically on camera images and laser point clouds.

Chapter 3: Urban Scene Segmentation with Laser-Constrained CRFs

This chapter discusses the first contribution of the thesis related to developing a reliable scene understanding framework. Section 3.1 consists of the introduction and related work of the proposed model. Section 3.2 introduces the CRF based semantic scene segmentation model, that incorporates camera-based visual information and laser based global constraints. In section 3.3, inference in the proposed CRF model is illustrated. This section expands on how quadratic programming formulation can be used to characterise the image labeling problem with global constraints. Finally, section 3.4 demonstrates the benefit of the proposed semantic segmentation model over other state of the art methods that only exploit visual information. Furthermore, it also showcases the advantage of having laser-based hard constraints over methods using soft constraints. Experimental results are presented for two real-world data sets.

Chapter 4: Online Learning for Scene Segmentation With Laser Constrained CRFs

This chapter focuses on the second and third contributions of this thesis, which are developed by extending the proposed scene segmentation model to an online adaptive model. Section 4.1 provides the introduction to the framework and discusses related work. Section 4.2 describes the process of extending the constrained QP problem in to an adaptive model by parameter learning using self supervised reference labels. It details that how stochastic gradient descent methods can be used to learn the parameters efficiently. Lastly, section 4.4 demonstrates the enhancement in quality achieved for individual object classification with adaptive learning. It also showcases the robustness of the scene segmentation model over long image sequences to simulate real-world driving. Finally, it illustrates the adaptability of the model to abrupt changes of input data distribution

Chapter 5: Conclusion and Future Work

Chapter 5 summarises the contributions of the thesis and draw conclusions based on the proposed methods. This chapter concludes by suggesting directions for future research on semantic scene segmentation based on the proposed framework.

Chapter 2

Background

This chapter presents the theoretical background necessary to understand the thesis. In Section 2.1 we introduce supervised learning techniques. We discuss classification algorithm, Discriminant Analysis, which we have utilised in our work to obtain the basic predictions on image labels.

Section 2.2 provides a description of Conditional Random Fields, which is also a sophisticated method for enhancing the quality of image classification considering the longer-range contextual information. Here we discuss about the inference in CRF and possible approaches to solve the inference problem. We majorly focus on the Quadratic Programming that can be efficiently applied in image classification. Further this section provides information on parameter estimation of CRF models.

The approaches used for optimisation in machine learning are described in the Section 2.3 and 2.4. Section 2.3 introduces gradient descent algorithm in general. Our research involves with developing an adaptive model requiring parameter optimisation. Stochastic gradient decent, which is detailed in section 2.4, is implemented for optimising the parameters for our CRF model to conduct image segmentation. Since our problem operates in an online setting the problem is large scale and stochastic gradient methods are more attractive.

The Section 2.5 presents information on sensor data processing. This section contains two portions. The first part explains image data processing. The second part is focused on 3-D point cloud processing. We provide details on super pixels generation and feature extraction. Laser point cloud based 3D information plays a major role in our image classification framework. We describe 3D point cloud processing methods such as Euclidean cluster extraction and ground plane removal methods.

2.1 Supervised Learning

Supervised learning is a major branch in machine learning. Consider target variables $\mathbf{y}$ and observed variables $\mathbf{x}$ where $\mathbf{y} = g(\mathbf{x})$. In supervised learning, the main task is learning the function g parameterised by a set of parameters θ given

training set. Learning is conducted using a training set of input and output samples $\Omega = \{(\mathbf{x_i}, y_i)\}_{i=1}^{N}$ assumed to be independent and identically distributed. N refers to the number of training samples. $\mathbf{x}_i$ has D number of dimensions, where each dimension links to a feature or a attribute. Features are extracted from an image, sentence, electronic signal or voice recording. In the case where y_i is a real valued scalar the prediction problem is referred as **regression**. Normally the output y_i is considered to be a categorical or nominal variable and can be assigned with a value from the set $L = \{1, .., n\}$ where n is the number of classes. This type of a problem is known as **classification**. Murphy *et al.* [75] offers a more extensive theoretical description of the properties and the applications of supervised learning.

2.1.1 Classification

In classification problems, when $n = 2$ then the problem reduces in to a binary classification when $n > 2$ it is a multiclass classification. Through machine learning the classifier function g is obtained. Subsequently, the learnt function can be used to predict the label of newly observed data. However, it is important that function g generalises well to unseen data. There are several algorithms used for classification such as Support vector machines [20], Decision Trees [84] and Neural Networks [3] . In the next section we explain the theory behind discriminant analysis utilised in extracting labels for the image pixels in later chapters.

Linear Discriminant Analysis

Linear discriminant analysis (LDA) linearly combines features to distinguish between object classes. It can be directly used as a linear classifier or for dimensionality reduction. This is a popular technique for pattern classification mainly due to the ease of computation since it has closed form solutions. It also provides decent class separability and can be used for multiclass problems intrinsically. This classifier has demonstrated good performance in practice. Additionally LDA does not require learning of hyper parameters. In our work, we use LDA to classify image patches. Li et al. [64] proposed the first LDA model to map multivariate input variables to univariate output variables. Here the model ensures that the outputs generated from each of the classes are far from each other as much as possible. Consider a training set Ω. The dimensionality D of input vector $\mathbf{x}$ has to be sufficiently large to contain adequate information to conduct the classification accurately. LDA is developed based on the analysis of scatter matrix, which is a metric utilised to evaluate the covariance matrix of the multivariate normal

distribution. The corresponding scatter matrix for class i is denoted by:

$$\chi_i = \sum_{\mathbf{x}_j \in a_i} (\mathbf{x}_j - \bar{\mathbf{x}}_i)(\mathbf{x}_j - \bar{\mathbf{x}}_i)^T, \tag{2.1}$$

where the number of classes is denoted by n. a_i referred to input data samples of i^{th} class. $\bar{\mathbf{x}}_i$ referred to the mean of the example input instances that has the label of the i^{th} class. Fisher et al. [114] introduce a discriminative feature transform using intra-class and inter class co-variance matrices denoted by Γ_1 (Eq. (2.2)) and Γ_2 (Eq. (2.3)) respectively,

$$\Gamma_1 = \sum_{i=1}^{n} \chi_i, \tag{2.2}$$

$$\Gamma_2 = \sum_{i=1}^{n} k_i(\bar{\mathbf{x}}_i - \bar{\mathbf{x}})(\bar{\mathbf{x}}_i - \bar{\mathbf{x}})^T, \tag{2.3a}$$

$$\bar{\mathbf{x}} = \frac{1}{N} \sum_{i=1}^{n} k_i \bar{\mathbf{x}}_i. \tag{2.3b}$$

Here k_i refers to the number of sample inputs belong to the class i. N denotes the total number of samples. There are important characteristics of the matrix $T = \Gamma_1^{-1}\Gamma_2$ so it can convey information on class compactness and class separation. T provides a discriminative feature transform through the eigenvectors related to the largest eigenvalues. According to Fishers criterion [114] a linear transformation Φ can be defined by maximising the Rayleigh coefficient indicated below,

$$\text{Rayleigh coefficient} = \frac{|\Phi^T \Gamma_2 \Phi|}{|\Phi^T \Gamma_1 \Phi|}. \tag{2.4}$$

This linear transformation matrix can be utilised as a distance measure (similar to Euclidean distance) to do the classification in the transformed space. The class label for some input $\mathbf{x}_j$ is given by:

$$y_j = i^* \text{ where } i^* = \min_{i \in L} \mathbf{x}_j \Phi - \bar{\mathbf{x}}_i \Phi. \tag{2.5}$$

2.2 Conditional Random Fields

In artificial intelligence, problems such as scene understanding, natural language processing and voice recognition require computing the assignments to a sequence $\mathbf{y}$ given a known set of inputs $\mathbf{x}$. The prediction of $\mathbf{y}$ can be difficult due to the complex dependencies between output variables. For instance, in image labelling problems neighbouring image patches are likely to have similar labels. Graphi-

cal models [19, 81] unite the techniques in probability theory with the efficient strategies in graph theory to overcome the complexity. They can be used to represent such problems, since they can efficiently characterise the dependencies between output variables. There are several families of graphical models such as neural networks, Markov random fields [51], ising models [113], Bayesian networks [47]and factor graphs [57] for structured prediction. Commonly, graphical models use a generative approach, which focus on modeling a joint probability distribution over input and output variables. However, CRF based approaches can result in intractable models when the dimensionality of the input is massive and there are complex dependencies between input variables. For more information on graphical models please refer to [54].

Conditional random fields (CRFs) are a variation of Markov random fields and use a discriminative approach to overcome the tedious joint probability computations. CRFs characterise distributions of structured output variables that are conditioned on some observed variables. These conditional distributions can be utilised for solving sequential classification problems. Typically, discriminative models do not require modelling the input distribution and also they permit to use pre-processed inputs, which can be useful in image classification. Another advantage of CRFs in the context of image classification is that it has flexibility to incorporate global features.

A discrete random field can be defined over the graph $G = (V, \mathbf{E})$ where V and $\mathbf{E}$ are the set of vertices and set of edges in the graph respectively. In this context $\mathbf{y} = \{y_1, y_2, .., y_m\}$ is a set of random variables where each vertex is associated with each node i. Each random variable can have a label from the label set $L = \{1, 2, .., n\}$. $\mathbf{x}$ represent the observed variables. Neighbours of each node i are indicated by $\mathbb{N}_i = \{j \in V | (i, j) \in \mathbf{E}\}$. Through the conditional distribution $P(\mathbf{y}|\mathbf{x})$ the mapping from $\mathbf{x}$ to $\mathbf{y}$ is modeled. When $\mathbf{y}$ is conditioned on $\mathbf{x}$ it assumes the Markov property. That implies the conditional distribution of y_i, given its neighbours in G, is independent from the other variables which are not in the neighbourhood.

The conditional distribution for the random variable set can be indicated by:

$$P(\mathbf{y}|\mathbf{x}, \theta) = \frac{1}{\mathcal{Z}(\mathbf{x})} \prod_{c \in C} \Psi_c(\mathbf{y}_c|\mathbf{x}, \theta). \tag{2.6}$$

Here a set of conditionally dependent random variables is defined as a clique (fully-connected sub-graphs), which is denoted by c. Set of random variables correspond to clique c is denoted by $\mathbf{y_c}$, where C denotes the set of all cliques and Ψ_c is a non negative clique potential. Set of model parameters are indicated by θ. In this scenario the partition function (normalising function) $\mathcal{Z}(x)$ is a

function of the input $\mathbf{x}$,

$$\mathcal{Z}(\mathbf{x}) = \sum_{\mathbf{y}} \prod_{c \in C} \Psi_c(\mathbf{y}_c|\mathbf{x}, \theta). \tag{2.7}$$

Generally in the cases where $\mathbf{y}$ is discrete, log potential is described by a linear combination of parameters,

$$\log\Psi_c(\mathbf{y}_c|\mathbf{x}, \theta) = \theta_c^T \psi_c(\mathbf{x}, \mathbf{y}_c), \tag{2.8}$$

where $\theta_c \in \mathbb{R}$ is a parameter vector. ψ_c indicate sufficient statistics or feature functions learnt from the observed data. Now the log of the conditional probability can be denoted by:

$$log(P(\mathbf{y}|\mathbf{x}, \theta)) = \sum_{c \in C} \theta_c^T \psi_c(\mathbf{x}, \mathbf{y}_c) - log(\mathcal{Z}(\mathbf{x})). \tag{2.9}$$

This formulation is known as the log-linear model. According to the theories in statistical physics, a probability distribution can be defined using the energy of the variables. This formulation is known as the gibs distribution [63],

$$P(\mathbf{y}|\mathbf{x}, \theta) = \frac{1}{\mathcal{Z}(\mathbf{x})} exp(-\sum_{c \in C} E(\mathbf{y_c}|\theta_c)), \tag{2.10}$$

where $E(\mathbf{y_c}|\theta_c) \geq 0$ is the energy correspond to the clique c. The conditional distribution of the CRF can be represented from a Gibbs distribution by defining the potential as follows:

$$\Psi_c(\mathbf{y}_c|\mathbf{x}, \theta) = exp(-E(\mathbf{y_c}|\theta_c)). \tag{2.11}$$

Now the energy of the CRF model can be denoted by:

$$E(\mathbf{y}|\theta) = -log(P(\mathbf{y}|\mathbf{x}, \theta)) - log(\mathcal{Z}(\mathbf{x})). \tag{2.12}$$

CRF for Image Labelling

Image classification can be characterised as a process of assigning labels to image pixels or patches (small groups of pixels). These labels depend on the application, i.e. foreground, background or object class label. Relationships among the labels of image pixels or patches are very important. CRF models are commonly used to solve the image classification problem. Usually for image classification problems, size of maximal clique is considered as 2 by confining the model into unary and pairwise cliques.

Unary cliques correspond to nodes, and each node (a pixel or an image patch)

is associated with an unary potential $\psi_i(y_i)$ that is defined as the log likelihood of node i is assigned with label y_i. This potential is computed based on features extracted locally to a node, i.e. colour, texture and location. Similarly, pairwise cliques correspond to edges. Edge potentials are denoted by $\psi_{ij}(y_i, y_j)$. Typically these edge potentials encourage connected nodes to take the same label. In practice contrast sensitive Potts models [17] are used to formulate pairwise potentials can be expressed by,

$$\psi_{ij}(y_i, y_j) = \begin{cases} 0 & \text{if } (y_i = y_j) \\ \gamma(i,j) & \text{otherwise} \end{cases}, \tag{2.13}$$

here $\gamma(i, j)$ is a feature function derived on the contrast of colour, texture and location of the connecting nodes in the graph. In the general case neighbourhood $\mathbb{N}_i$ is stated as to connect 4 to 8 neighbouring pixels. These connections are important since they decide the amount of contextual information is used in classification. Energy of the image classification problem can be indicated by:

$$E(\mathbf{y}) = \sum_{i \in V} \underbrace{\psi_i(y_i)}_{\text{Unary Potentials}} + \sum_{i \in V, j \in \mathcal{N}_i} \underbrace{\psi_{i,j}(y_i, y_j)}_{\text{Pairwise potentials}} . \tag{2.14}$$

Higher Order Potentials

The pairwise potential encourages smooth boundaries between different object classes. However these potentials can be highly useful but still have some drawbacks. For example, these smoothing terms are less efficient in identifying intricate boundaries of different object classes. In addition, the boundaries of the objects in the segmentation based on pairwise potentials can deviate from the actual boundaries due to the over smoothing. In order to address these problems higher order potentials (HOPs) are introduced to the image classification problem. Higher order potentials impose soft constraints about label consistency. Thus it encourages the consistency of labelling with in image regions or segments. In this manner HOP assist to capture longer range relationships. Energy function of CRF can be modified by introducing HOPs as in Eq. (2.15), where S denotes the set of image regions/segments, on which ψ_a, the higher order potentials are defined on,

$$E(\mathbf{y}) = \sum_{i \in V} \psi_i(y_i) + \sum_{i \in V, j \in \mathcal{N}_i} \psi_{i,j}(y_i, y_j) + \sum_{a \in S} \psi_a(y_a). \tag{2.15}$$

2.2.1 CRF Inference

In learning the CRF model parameters and predicting the query variables, accurate and fast inference is a main concern. The goal of this section is to describe the main inference problems and commonly used approaches to solve them. The section mainly focus on the inference methods which would be efficiently used for image labelling problem. A more comprehensive explanation can be found in [75]. Generally two inference problems can be described for CRF.

- Marginal Distribution Computation:
  The marginal distribution for a set of variables in a CRF is obtained by marginalising all the other variables to obtain $P(\mathbf{y}_c|\mathbf{x},\theta)$ where c is a subset of $\mathbf{y}$. The computation involves summing out all the random variables in the conditional distribution of $\mathbf{y}$ that do not belong to clique c. When there are large number of variables associated with $\mathbf{y}$ or when variables having a higher number of states, computational complexity can increase and problem can become intractable. Selecting a simpler graph structure might reduce the computational cost.

- Maximum A Posteriori Estimation:
  Considering a CRF with parameter set θ, we obtain the labelling of $\mathbf{y}$ correspond to input $\mathbf{x}$ by picking the most likely output $\mathbf{y}^{MAP}$ that is stated as the maximum a posteriori denoted by $\text{argmax}_{\mathbf{y}}\, P(\mathbf{y}|\mathbf{x},\theta)$,

  $$\mathbf{y}^{MAP} = \underset{\mathbf{y}}{\text{argmax}}\, P(\mathbf{y}|\mathbf{x},\theta) = \underset{\mathbf{y}}{\text{argmax}}\, \frac{1}{\mathcal{Z}(\mathbf{x})} \prod_{c\in C} \Psi_c(\mathbf{y}_c|\mathbf{x},\theta). \tag{2.16}$$

  MAP estimation is widely used since it represent an optimisation problem, that can be solved by efficient algorithms. MAP inference for general graphs tend to be NP-hard in most of the cases. For those instances approximate inference is conducted. It is clear that since the normalisation function is not a function of $\mathbf{y}$, it can be ignored during the optimisation process. Therefore, according to the Eq. (2.12) MAP solution also can be attained by minimising the energy function,

  $$\mathbf{y}^{MAP} = \arg\min_{\mathbf{y}} E(\mathbf{y}). \tag{2.17}$$

  MAP estimation is similar to a point estimate. Hence it does not comes with any uncertainty measures.

Exact Inference

This section focuses on the exact inference in conditional random fields. For more detailed description on the algorithms mentioned here refer to [54],[50] and [80]. Usually the computational cost for the inference operations exponentially grows with the number of random variables. This causes the intractability of the exact inference in general CRF. However there are some special cases where the inference problem in CRFs can be solve in polynomial time. Variable elimination (VE) [54] is the simplest approach to obtain the marginal distribution or the MAP estimation of CRFs. However, this method has an exponentially growing complexity with the number of random variables. Usually VE can only be used to conduct the inference in graphs with low treewidth. Typically, when the graph has a chain or a tree structure, message-passing algorithms can provide exact solutions for the inference problems. In addition, if a CRF consists only pairwise terms and the random variables associated with the nodes can only take binary values (binary graph) then it can be solved exactly using max flow−min cut algorithm [32]. Here MAP solution is obtained as the equivalent maxflow solution. Vision based problems such as foreground/background segmentation (binary problems) can be exactly solved by this method. Graphs with lower tree weight can also be solved exactly using the junction tree algorithm [50]. In this case marginal distribution is obtained. Yet the computational complexity exponentially increases with the treewidth. In our work we are mainly interested in solving scene segmentation problems, which involves multiple classes, graphs with loops and higher tree width. Generally these problems are intractable and cannot be solved using the exact inference methods.

Approximate Inference

CRF models, commonly used to solve computer vision problems, have posterior distributions, which are infeasible to compute in polynomial time due to the high dimensionality. Instead several approximation techniques are proposed to solve inference problems.

One of the common approaches is stochastic approximation [88]. In this process sufficient number of samples are drawn from the true distribution and an approximated sample distribution is generated. Sampling distributions can asymptotically converge to the original distribution. This characteristic is used in solving the problems. Markov chain Monte Carlo [78] methods are popular sampling methods to solve the CRF inference problem.

Variational inference [113] is also a widely used deterministic technique. The main idea behind this approach is to use an analytical distribution that can

approximate the posteriori distribution. Different types of Gaussian distributions are commonly used to approximate true posterior distributions. Further, variational method deal with minimising the distance between the true and the analytical distribution.

Message passing techniques such as loopy belief propagation [76] also provide good approximation to the CRF inference problem. The next sections elaborate the theory behind some approximate inference methods.

Loopy Belief Propagation

Loopy belief propagation (LBP) ([75] Chapter 22) is a technique for approximate inference on discrete graphical models. LBP is an extension to standard belief propagation algorithms to conduct inference on the graphs with loops. As we know CRF based image classification models commonly exploit grid shaped graphs that connects all the neighborng nodes because it requires modeling contextual relationships. This type graphical structures have loops. LBP methods are used in practice to perform inference in CRF models that are used in image classification. Marginal distribution $P(\mathbf{y}|\mathbf{x})$ of the random variables associated with the nodes can be obtained using LBP. The fundamental concept behind LBP is letting all nodes to receive messages from the neighbouring nodes. Given an all edges in the graph, messages flow through every edge in both directions. Standard form of a message sent from a certain node i to its neighbour j can be denoted by:

$$m_{i\to j}^{new}(y_j) = \sum_{y_i}[\psi_i(y_i)\psi_{ij}(y_i, y_j) \prod_{k\in\mathcal{N}_i\setminus j} m_{k\to i}^{old}(y_i)]. \tag{2.18}$$

Common practice is to normalise messages as follows:

$$\sum_{y_j} m_{i\to j}(y_j) = 1. \tag{2.19}$$

The belief of node i is proportional to the following terms:

$$\beta_i(y_i) \propto \psi_i(y_i) \prod_{j\in\mathcal{N}_i} m_{j\to i}(y_i). \tag{2.20}$$

The algorithm updates node beliefs and send the updated messages to their corresponding neighbours. These steps are repeated until the marginal beliefs are stabilised.

Algorithm 1: Loopy Belief Propagation for Pairwise CRF

Input: unary/node potentals $\psi_i(y_i)$, pairwise/edge potentials $\psi_{ij}(y_i, y_j)$

Output: $\beta_i(y_i)$

Initialise:

Messages $m_{i\to j} = 1 \quad \forall$ edges $i - j \in \mathbf{E}$

Beliefs $\beta_i(y_i) = \psi_i(y_i) \quad \forall i \in V$

do

  Transmit message from each node to its corresponding neighbours

  $m_{i\to j}(y_j) = \sum_{y_i}[\psi_i(y_i)\psi_{ij}(y_i, y_j) \prod_{k\in\mathcal{N}_i\setminus j} m_{k\to i}(y_i)]$

  Update marginal belief correspond to each node

  $\beta_i(y_i) \propto \psi_i(y_i) \prod_{j\in\mathcal{N}_i} m_{j\to i}(y_i)$;

while *change of* $\beta_i(y_i)$ *is significant*;

return $\beta_i(y_i)$

Consider the algorithm 1. All the messages $m_{i\to j}$ are initialised to 1. Marginal belief of each node is initialised to its local node potential. Subsequently messages are sent from each node to its neigbours $\mathcal{N}_i$ parallely. Then new messages can be computed (for repeating the process) by multiplying all the incoming messages except the one from the receiver. In this manner marginal beliefs can be updated until convergence. Theoretically, node beliefs is expected to converge to true marginals. However, this sum product belief updating process does not guarantee to converge, even if it converges the solution might not be accurate. It has been proven that the approximation error of the marginal is linked with the convergence rate. In the other words, if the LBP is converging fast, that implies the answer is more accurate.

2.2.2 Approximate MAP Inference Using Convex Relaxation

This section describes the formulation of the MAP estimation as a mathematical optimisation problem. For discrete CRFs, MAP estimation problem is generally NP-hard. Therefore, solutions are obtained through convex relaxations which is a approximation of the original problem with a much simpler problem. The previously mentioned CRF model for image classification is a discrete model. For MAP estimation of discrete CRFs it has been proposed tight relaxations [120] that is described in this section. Consider a standard integer linear (ILP) [34]

program given bellow:

$$\arg\min_{\mathbf{Y}} \mathbf{e}^T\mathbf{Y} \tag{2.21a}$$

$$\text{s.t } A\mathbf{Y} + \mathbf{s} = \mathbf{b} \tag{2.21b}$$

$$\mathbf{Y} \geq 0 \tag{2.21c}$$

$$\mathbf{Y} \in \mathbb{Z}^n, \tag{2.21d}$$

where $\mathbf{Y}$ is the vector containing the random variables, $\mathbf{e}$ and $\mathbf{b}$ are real valued vectors, A is integer valued matrix, $\mathbf{s}$ is a slack variable. MAP problem can be reformulated as an integer linear program to apply relaxations. ILP indicated in Eq. (2.21) consist of a linear objective built over variables that are constrained to attain integer values. This ILP formulation of MAP problem is also an NP-hard problem.Yet it allows convex relaxation by expanding the feasibility region from integer space to real valued space. There are several methods based on convex relaxations [58] that can be used to solve the ILP problem. These relaxations provide an approximation to the ILP problem. Some of the relaxation techniques are commonly used in practice are listed below

1. Linear Programming Relaxation
2. Quadratic Programming Relaxation
3. Semi definite Programming Relaxation
4. Second-Order Cone Programming Relaxation

Notations

Consider the pairwise CRF model and assume that each output variable of y_i is assigned with values from the discrete label set L. Then the potential functions can be defined as a linear combination of indicator functions as in Eq. (2.22) and Eq. (2.23),

$$\psi_i(y_i) = \sum_r \psi_i(r) I_r(y_i), \tag{2.22}$$

$$\psi_{ij}(y_i, y_j) = \sum_{r,s} \psi_{ij}(r,s) I_{rs}(y_i, y_j), \tag{2.23}$$

here $r, s \in L$ and $i, j \in V$. Now the indicator variables can be denoted by:

$$I_r(y_i) = \begin{cases} 1 & y_i = r \\ 0 & \text{otherwise} \end{cases}, \tag{2.24}$$

$$I_{rs}(y_i, y_j) = \begin{cases} 1 & y_i = r \text{ and } y_j = s \\ 0 & \text{otherwise} \end{cases}. \tag{2.25}$$

Since the normalising function is only a function of the observed variables we can rewrite the equation for conditional likelihood of the $\mathbf{y}$ as:

$$P(\mathbf{y}|\mathbf{x}, \theta) \propto exp(\sum_{i,r} \delta_{i;r} I_r(y_i) + \sum_{i,r;j,s} \delta_{i,r;j,s} I_{rs}(y_i, y_j)). \tag{2.26}$$

For convenience we have defined new notations $\delta_{i;r} = \theta_{ir}\psi_i(r)$ and $\delta_{i,r;j,s} = \theta_{irjs}\psi_{ij}(r, s)$. Based on all above derivations, the MAP estimation can be formulated using indicator functions,

$$\mathbf{y}^* = \underset{\mathbf{y}}{\text{argmax}} \sum_{i,r} \delta_{i;r} I_r(y_i) + \sum_{i,r;j,s} \delta_{i,r;j,s} I_{rs}(y_i, y_j). \tag{2.27a}$$

Linear Programming Relaxation

This section formulates the MAP problem as a standard ILP problem and as a the linear programming relaxation, as originally introduced in [99]. The indicator variables $I_r(y_i)$ and $I_{rs}(y_i, y_j)$ can be replaced by binary variables $\mu(i; r)$ and $\mu(i, r; j, s)$. The new form of the MAP estimation is,

$$\max \sum_{i,r} \delta_{i;r} \mu(i; r) + \sum_{i,r;j,s} \delta_{i,r;j,s} \mu(i, r; j, s) \tag{2.28a}$$

$$\text{subject to } \sum_{s} \mu(i, r; j, s) = \mu(i; r) \tag{2.28b}$$

$$\sum_{r} \mu(i; r) = 1 \tag{2.28c}$$

$$\mu(i; r) \in \{0, 1\} \tag{2.28d}$$

$$\mu(i, r; j, s) \in \{0, 1\}. \tag{2.28e}$$

Due to the structure of the indicator variables in Eq. (2.24) and Eq. (2.25), the binary variable also satisfy the marginalisation constraint Eq. (2.29b). Further, the constraints in Eq. (2.28c) are enforced to confine each variable to have only one label. Here all the constraints are assumed to be linearly independent. This formulation of the MAP problem satisfies the requirement for a ILP. The linear relaxation for this problem is given by Eq. (2.29). The random variable μ is relaxed such that it can lie in the range of [0,1]. This expands the feasibility

region for the solution,

$$\max \sum_{i,r} \delta_{i;r}\mu(i;r) + \sum_{i,r;j,s} \delta_{i,r;j,s}\mu(i,r;j,s) \tag{2.29a}$$

$$\text{subject to } \sum_{s} \mu(i,r;j,s) = \mu(i;r) \tag{2.29b}$$

$$\sum_{r} \mu(i;r) = 1 \tag{2.29c}$$

$$0 \leq \mu(i;r) \leq 1 \tag{2.29d}$$

$$0 \leq \mu(i,r;j,s) \leq 1. \tag{2.29e}$$

Quadratic Programming Relaxation

Graphical model energy can be precisely represented by a quadratic objective function. Thus quadratic programming relaxation of the MAP problem can yield exact solutions to the original problem. Quadratic programming (QP) relaxation is originally proposed in [87] and has applied to image classification with a MAP estimation [120]. According to the definition of the indicator function in Eq. (2.25), it automatically satisfies the independence constraint,

$$I_{rs}(y_i, y_j) = I_r(y_i)I_s(y_j). \tag{2.30}$$

Consider the relaxation variables of the indicator functions $I_{rs}(y_i, y_j)$ and $I_r(y_i)$ that are indicated by variable $\mu(i,r;j,s)$ and $\mu(i;r)$ in Eq. (2.29). We constrain these relaxation variables in a similar fashion to the indicator functions as in shown in Eq. (2.31), by letting the relaxation to be tighter,

$$\mu(i,r;j,s) = \mu(i;r)\mu(j;s). \tag{2.31}$$

Now we can rewrite Eq. (2.29) as a quadratic programming problem by substituting Eq. (2.31) for $\mu(i,r;j,s)$ as follows:

$$\max \sum_{i,r} \delta_{i;r}\mu(i;r) + \sum_{i,r;j,s} \delta_{i,r;j,s}\mu(i;r)\mu(j;s) \tag{2.32a}$$

$$\text{subject to } \sum_{r} \mu(i;r) = 1 \tag{2.32b}$$

$$0 \leq \mu(i;r) \leq 1. \tag{2.32c}$$

This quadratic programming formulation leads to a tighter relaxation of original MAP problem, according to Theorem 2.2.1 and Theorem 2.2.2 given below.

Theorem 2.2.1. *The optimal value of relaxed quadratic problem Eq.* (2.32) *is equal to the optimal value of the MAP problem in Eq.* (2.27)*. (Proof in [87])*

Theorem 2.2.2. *Any solution to the MAP problem Eq.* (2.27) *efficiently yields a solution of the relaxation and Eq.* (2.32) *and vice versa. Thus the relaxation Eq.* (2.32) *is equivalent to the MAP problem Eq.* (2.27)*. (Proof in [87])*

Note that relaxed QP problem consists of nm number of random variables which is much less compared to the $n^2|\mathbf{E}|$ number of variables associated with the LP problem.

Convex Approximation

In the cases where the MAP problem is convex it can be solved in polynomial time. Ravikumar et al. [87] state that if the pairwise coefficient matrix $H = [\delta_{i,r;j,s}]_{mn\times mn}$ of the proposed QP relaxation is negative definite, then the MAP problem becomes a convex program. They also propose a convex approximation to quadratic programming problems which enables to solve the problem in polynomial time. Consider a situation where H is non-negative definite. To convert H to a negative definite matrix, pairwise potentials are modified as,

$$H_{i,r;j,s} = \begin{cases} \sum_{k,p} \delta_{i,r;k,p} & \text{if } i = j, r = s \\ \delta_{i,r;j,s} & \text{otherwise} \end{cases}, \tag{2.33}$$

$$H_{i,r} = \delta_{i,r} - \sum_{k,p} \delta_{i,r;k,p}. \tag{2.34}$$

This means, positive potential value is added to each of the pairwise potentials in the diagonal of H. Subsequently, the value of the added potential is subtracted from the corresponding unary potential as indicated in Eq. (2.34) in order to cancel the effect of the addition with in the objective function. This modification of the pairwise potentials guarantees that the H is negative semi-definite. The updated QP program is denoted as follows:

$$\underset{\mu}{\operatorname{argmax}} \sum_{i,r} H_{i;r}\mu(i;r) + \sum_{i,r;j,s} H_{i,r;j,s}\mu(i;r)\mu(j;s) \tag{2.35a}$$

$$\text{subject to } \sum_{r} \mu(i;r) = 1 \tag{2.35b}$$

$$0 \leq \mu(i;r) \leq 1. \tag{2.35c}$$

The convexity of the QP problem in Eq. (2.35) makes it feasible to solve it in polynomial time. According to [87], in following two scenarios the QP relaxation in Eq. (2.35) obtains a solution which is close to MAP estimation of the original problem:

- The original edge potential matrix H is close to negative definite in case diagonal terms for the convex approximation is close to zero.
- Final solution of $\mu(i, r)$ yield values close to one or zero.

Linearly Constrained Quadratic Programming

Both scenarios of MAP problem, denoted in Eq. (2.32a) and convex approximated version in Eq. (2.35a) maximise a quadratic function which is defined over linearly constrained (equality and/or inequality) random variables. The algorithms such as interior point [69], active set [40] and augmented Lagrangian [21] are used to directly solve the problem. However when the number of nodes is higher and large numbers of states are associated with variables, the above-mentioned methods tend to fail due to the computational complexity. However, if a quadratic function is derived only based on equally constrained variables, then there are mathematical approaches to reduce the dimensionality of the original problem and derive an equivalent unconstrained optimisation problem. This allows us to solve the problem using unconstrained optimisation approaches such as conjugate gradient method [38].

Equality Constrained Quadratic Programming Problems

Equality Constrained Quadratic Programming Problem refers to a type of optimisation problems where the objective is a quadratic function of some variables, which are only subjected to equality constraints. Consider a quadratic function similar to Eq. (2.32a), and assume the corresponding random variables are only equality constrained. General form of this problem can be written in matrix notation as a minimisation problem as follows:

$$\text{minimize } \frac{1}{2} Y^T Q Y + \mathbf{e}^T Y \tag{2.36a}$$

$$\text{subject to } AY = \mathbf{b} \tag{2.36b}$$

$$Y \in \mathbb{R}^{mn}. \tag{2.36c}$$

Here $Q \in \mathbb{R}^{mn \times mn}$ is symmetric pairwise potential matrix and $Q = -H$, $\mathbf{e} = [-\delta_{i,r}]_{mn \times 1}$ and $Y = [\mu(i, r)]_{mn \times 1}$. $A \in \mathbb{R}^{u \times mn}$ with $u < mn$, A has full row rank allowing $AY = \mathbf{b}$ to have u number of linearly independent equations (equality constrains) and $\mathbf{b} \in \mathbb{R}^u$. $Y^* \in \mathbb{R}^{mn}$ denotes the optimum solution to the problem. According to first order necessary conditions [77] for Y^* to be a solution of Eq. (2.36), it is true that there is a vector λ^* such that the following linear system satisfied,

$$\begin{bmatrix} Q & -A^T \\ A & 0 \end{bmatrix} \begin{bmatrix} Y^* \\ \lambda^* \end{bmatrix} = \begin{bmatrix} -\mathbf{e} \\ \mathbf{b} \end{bmatrix}, \tag{2.37}$$

here $\lambda^* \in \mathbb{R}^m$ is the vector of Lagrange multipliers. By introducing a new variable p such that $p = Y^* - Y$, where Y be a feasible point satisfying the equality constraints, the linear system can be reformulated as follows,

$$\underbrace{\begin{bmatrix} Q & A^T \\ A & 0 \end{bmatrix}}_{K} \begin{bmatrix} -p \\ \lambda^* \end{bmatrix} = \begin{bmatrix} \mathbf{e} + QY \\ AY - \mathbf{b} \end{bmatrix}. \tag{2.38}$$

The matrix K is stated as Karush−Kuhn−Tucker (KKT) matrix [77]. When the KKT matrix is non-singular, it results in important conditions as indicted in Lemma 2.2.1.

First, consider a matrix Z, that is a basis for null space of A where $Z \in \mathbb{R}^{mn\times(mn-u)}$ and $AZ = 0$. The matrix Z^TQZ is stated as the reduced Hessian matrix.

Lemma 2.2.1. *Assume that A has full row rank and reduced-Hessian matrix Z^TQZ is positive definite. Then the Karush−Kuhn−Tucker matrix $\begin{bmatrix} Q & A^T \\ A & 0 \end{bmatrix}$ is non-singular. Hence the linear system (Eq. (2.37)) has a unique solution (Y^*, λ^*) [77] (Proof in [77])*

According to Lemma 2.2.1, the KKT conditions (first order necessary conditions [77](see Chapter 12)) are satisfied, therefore the quadratic problem in Eq. (2.36a) has a unique optimal solution. Further it also implies that the system satisfies the second order sufficient conditions [77](see Chapter 12) such that there exist a local minimiser for the problem. Using these facts the following argument has been derived to prove that the unique solution is a global minimum given the KKT conditions.

Theorem 2.2.3. *Given the assumptions in Lemma 2.2.1, linear system yield a unique solution (Y^*, λ^*). Then Y^* is the unique global solution of equality constrained QP problem Eq. (2.36) (Proof in [77] (Chapter 16)).*

The solution (Y^*, λ^*) can be obtained by range space or null space methods [77].

Range Space Methods

This method is applicable only when Q is strictly positive definite and invertible. In addition, it also require the number of constraints u to be small. Under these assumptions, considering the linear equations Eq. (2.38), Y^* can be eliminated

to derive a linear system for λ^* as follows,

$$(AQ^{-1}A^T)\lambda^* = (AQ^{-1}e + b). \tag{2.39}$$

Then p can be obtained by the first equation in the linear system Eq. (2.38) by substituting for λ^* as follows,

$$Qp = A^T\lambda^* - (e + QY). \tag{2.40}$$

The optimum solution is obtained by $Y^* = Y + p$. Where Y is some feasible point satisfying the equality constrains in the original minimisation problem.

Null-space Methods

This method exploit the matrix Z that is in the null space of A. Consider given feasible solution Y_0 that satisfy the equality constraints . We can denote $Y = Y_0 + Zv$ for some new vector $v \in \mathbb{R}^{mn-u}$. Substituting for Y it can be shown that the equality-constrained minimisation problem(Eq. (2.36)) can be represented from an equivalent unconstrained minimisation problem.

Firstly, it require to always satisfy the equality constraints during the optimisation. To this end, $AY - b = 0$ is substituted by $Y = Y_0 + Zv$, then $\underbrace{AY_0 - b}_{=0} + \underbrace{AZ}_{=0}v = 0$. As we can see the constraint equation is always satisfied. The quadratic objective in Eq. (2.36a) is a function of Y hence we denote it as $q(Y)$. Now we substitute $Y = Y_0 + Zv$ to $q(Y)$. Here Y_0 and Z are constants hence the substitution makes the quadratic objective $q(Y)$, a function of v. The new unconstrained problem given by Eq. (2.41),

$$\begin{aligned} q(v) =& \frac{1}{2}(Y_0 + Zv)^T Q(Y_0 + Zv) + e^T(Y_0 + Zv) \\ =& \frac{1}{2}v^T\tilde{Q}v + \tilde{e}^T v + \text{const} \text{ , where } \tilde{Q} = Z^TQZ, \tilde{e} = Z^T(QY_0 + e). \end{aligned} \tag{2.41}$$

The Quadratic function $q(v)$ implicitly satisfy the constraint equations as shown earlier and it can be considered as an unconstrained optimisation equivalent to the equality constrained minimisation in Eq. (2.36). Now by eliminating the constant term, the new minimisation problem can be denoted as,

$$\min_v \frac{1}{2}v^T\tilde{Q}v + \tilde{e}^T v. \tag{2.42}$$

In a case where the reduced Hessian matrix $\tilde{Q}$ is positive definite then the following linear system provides a unique solution v^* for the sub problem,

$$(Z^TQZ)v^* = -Z^T(QY_0 + e) \tag{2.43}$$

The solution Y^* for the Eq. (2.36) can be obtained by $Y^* = Y_0 + Zv^*$.

2.2.3 CRF Training

CRF models are best known for labelling newly encountered data through the inference process. As we know CRFs are parametric models. Eq. (2.6) indicates a typical CRF model. Learning the perfect model for the interested problem is a challenging task. More detailed description on CRF training is available in [75](Chapter 19.6). Through parameter training we obtain a model that can generate a probability distribution close to the desirable distribution. The optimum parameter set can improve the quality of CRF predictions. Parameter training process requires a set of training sequence $\Omega = \{(z_i, \mathbf{x}_i) \quad i \in \{1, \ldots, N\}$, where $\mathbf{x}_i$ is an observed data sample and z_i is the corresponding labelling or the assignment for each observed data point. y_i is the corresponding label prediction from the CRF model. The training approaches described in the next sections are based on [75] and [11].

Maximum Likelihood-Based Parameter Training

Likelihood function based methods are best known for parameter estimation tasks. The likelihood function states the likelihood of a parameter set $\theta = [\theta_1, .., \theta_K]$ given the training data Ω. We assume that the training data is independent and identically distributed. In this case we can obtain the optimum parameter set θ^* for our model by maximising the likelihood for the given data Ω as denoted in Eq. (3.10c),

$$\theta^* = \underset{\theta}{\operatorname{argmax}}\, \mathrm{L}(\theta|\Omega), \tag{2.44}$$

where L denotes the log likelihood function which is a summation of the likelihood over training samples. In practice log of the actual likelihood is used for the mathematical operations to avoid numerical problems. Likelihood for a general CRF model is denoted by the Eq. (2.45),

$$\mathrm{L}(\theta|\Omega) = \prod_{i=1}^{N} P(\mathbf{z}_i|\mathbf{x}_i, \theta). \tag{2.45}$$

Taking the log of Eq. (2.45) and we can obtain the equation below for the log likelihood. In order to avoid over-fitting, we can add a regularising term to the

log likelihood as follows,

$$l(\theta|\Omega) = \frac{1}{N}\sum_{i=1}^{N}\log\ P(\mathbf{z}_i|\mathbf{x}_i,\theta)$$
$$= \frac{1}{N}\sum_{i}\Big\{\sum_{c}\theta_c^T\psi_c(\mathbf{x}_i,\mathbf{z}_i) - \log\ \mathcal{Z}(\theta,\mathbf{x}_i)\Big\} - \underbrace{\sum_{l=1}^{K}\frac{\theta_l^2}{2\alpha_l^2}}_{\text{Quadratic Regularizer}} \quad . \tag{2.46}$$

Here α_l refer to the penalty applied on θ_l. This regularisation assists to generalise the model for newly encountered data.
The log likelihood is a convex function with respect to θ. Hence a gradient-based algorithm can be used to accomplish the maximisation.The derivative of $l(\theta|\Omega)$ with respect to clique c is denoted by Eq. (2.47),

$$\frac{\partial l}{\partial\theta_c} = \frac{1}{N}\sum_{i}\Big\{\psi_c(\mathbf{z}_i,\mathbf{x}_i) - \frac{\partial\log\mathcal{Z}(\mathbf{x}_i,\theta)}{\partial\theta_c}\Big\} - \frac{\theta_l}{\alpha_l^2}. \tag{2.47}$$

The derivative of the log of the normalising function $\mathcal{Z}(\mathbf{x}_i,\theta)$ can be stated as the expectation of the feature correspond to the clique c according to the model as in Eq. (2.48),

$$\frac{\partial\log\mathcal{Z}(\mathbf{x}_i,\theta)}{\partial\theta_c} = \mathbb{E}[\psi_c(\mathbf{y}_i,\mathbf{x}_i)]. \tag{2.48}$$

The partition function and its derivative computation rely on the inference solution of the model. Conducting these computations in higher dimensional space and performing the computation at each iteration is computationally intractable. Hence approximations to the model are used. One solution is to replace the log likelihood with the pseudo likelihood [9], which is an approximation to the likelihood but is a computationally simpler problem. Otherwise we can deploy the approximate inference methods such as sum product loopy belief propagation.

Loss-Based Parameter Estimation

Loss based models are capable of independently processing the learning criterion from the CRF. This approach is described in [103]. The loss represents the distance between the ground truth and the model solution (MAP solution). In order to compute this distance mean squared error, overall accuracy or F1-measure is used in practice. The ultimate goal is to select θ values so that the expected loss is minimised for the query distribution. Since the actual data distribution is unknown the loss is minimised for a set of known input out put pairs from the true data distribution. This process is referred to as empirical risk minimisation,

$$\theta^* = \arg\min_{\theta} \sum_{j=1}^{N} \ell(\mathbf{z}_j, \mathbf{y}_j^{MAP}). \tag{2.49}$$

Here $\mathbf{z}_j$ is the ground truth for data point j, $\mathbf{y}_j^{MAP}$ is the MAP estimation and ℓ is the loss based on the dissimilarity between $\mathbf{z}_j$ and $\mathbf{y}_j^{MAP}$. As described earlier MAP for CRF models are commonly conducted through approximation algorithms. Therefore, parameter learning must be optimising the approximated model rather than the true CRF model. Loss based methods are efficient to achieve this purpose rather than likelihood models. The loss function can be selected depending on the interested application giving the bias on more critical parameters. Loss minimisation is an optimisation problem, hence gradient-based methods can be used to solve the problem. In practice, automatic differentiation [86] or finite differences [101] generate the required gradients for the operation. Generally batch gradient decent or simulated annealing can be used to optimise the loss. However for large-scale problems, stochastic gradient decent based methods are more popular. In order to avoid over fitting appropriate regularisation should be adopted.

2.3 Gradient Descent for Machine Learning

Gradient descent (GD) [14] is commonly used in machine learning. GD based learning is developed using a greedy, hill-climbing strategy. Fundamentally GD tunes parameters of the model considering the deviation between the actual and the model output. The updated parameters drive the model to a direction where the error of the model declining in the steepest sense.

Consider a supervised learning problem in which $\omega = (x_i, z_i)$ is a example input output pair in the training set Ω where $\mathbf{x} \in \mathbb{R}^{\rho}$ is the input and the $\mathbf{z}$ is the actual label (ground truth). G is the family of functions $g_\theta(\mathbf{x})$ that is parameterised by θ where predicted output $\mathbf{y} = g_\theta(\mathbf{x})$. Then the error of the model can be denoted by a loss function $\ell(\mathbf{y}, \mathbf{z})$. The goal is to find θ^* that minimises the loss $\ell(g_\theta(x), z) \forall \omega$ as indicated in Eq. (2.50),

$$\mathrm{E}(g_\theta) = \int \ell(g_\theta(x_i), z_i) dP(\omega). \tag{2.50}$$

The expected risk $\mathrm{E}(g_\theta)$ represent the accuracy of the model for both existing and the future data. But practically the distribution of $P(\omega)$ is not known. Hence minimising the cost is associated with a selected sample of data points as denoted in Eq. (2.51),

$$\mathrm{E}_N(g_\theta) = \frac{1}{N}\sum_{i=1}^{N} \ell(g_\theta(x_i), z_i). \tag{2.51}$$

However for practical usage the empirical risk $\mathrm{E}_N(g_\theta)$ is minimised in place of the expected risk where G is sufficiently restrictive. GD updates the parameters θ at each iteration using the gradient of the empirical risk.

$$\theta_{t+1} = \theta_t - \eta \frac{1}{N}\sum_{i=1}^{N} \nabla_\theta \ell(g_\theta(x_i), z_i), \tag{2.52}$$

where the learning rate η should be carefully chosen depending on the nature of the problem. As elaborate in [14], with sufficient regularity assumptions, by choosing the initial estimate θ_0 sufficiently close to the optimum and setting the learning rate sufficiently small, linear convergence can be guaranteed. If g is a convex function convergence rate is $O(1/t)$. In cases where g is strongly convex convergence rate tend to be $O(e^{-\rho t})$. In order to improve the optimisation method, learning rate η can be substituted by a matrix B where B should be a positive definite matrix and it should approach the inverse of the Hessian of the loss function at the optimum point,

$$\theta_{t+1} = \theta_t - B_i \frac{1}{N}\sum_{i=1}^{N} \nabla_\theta \ell(g_\theta(x_i), z_i). \tag{2.53}$$

The second order gradient descent is a modification of the Newton algorithm. Under sufficient regularity assumptions, if the θ_0 is selected close to the optimum it can be guaranteed the quadratic convergence. In a case where loss function is quadratic and B is exact, the optimum is obtained with a single iteration.

GD has several advantages such as the applicability in parallel processing models and capability in working incrementally with new data. However GD methods do not always guarantee reaching the global optimum. Hence there are several techniques to overcome GD from settling in a local optimum. One approach is using a convex objective. Alternatively the problem can be randomly initialised and perform GD multiple time to find a best estimate for the optimum.

2.4 Stochastic Gradient Descent

Unlike the GD algorithm, Stochastic gradient descent (SGD) utilise a simpler gradient in place of the gradient of the empirical risk. This section describe the theoretical and the practical aspects of SGD based on the works in [14],[12] and [13]. SGD gradient is computed using a single randomly selected data point during each parameter updating step. The stochastic process $\{\theta_t : t = 1, ..., N\}$

continues on randomly selected data points,

$$\theta_{t+1} = \theta_t - \gamma_t \nabla_\theta \ell(g_\theta(x_t), z_t). \tag{2.54}$$

SGD is proven to optimise the expected risk. With the random selection of data points from the query distribution it develops a more generalised model than GD. Another characteristic of SGD is it avoids the redundant learning occurred in batch gradient descent. Batch GD process similar data points prior to each parameter update which can be avoided in the case of SGD. In large scale machine learning, computing limitations are a major concern. In this context SGD can be efficiently implemented to optimise large chunks of data over batch gradient methods. Although the gradients tend to be noisy, SGD has shown promising results in optimisation. Typically, convergence rate is slower than the GD algorithm. For strongly convex functions, SGD has a convergence rate of $O(1/t)$ and for general convex functions convergence rate is $O(1/\sqrt{t})$.

Setting the learning rate in a SGD algorithm is a delicate issue since the larger values can diverge the algorithm from optimum and smaller values can increase the convergence time. According to convergence analysis in [14], SGD allows different learning rate schedules. Fixed step size and diminishing step size are the most popular choices. Under certain assumptions [14] fixed step size can guarantee to linearly converge to a neighbourhood of the optimal value. But noisy gradients deviate the algorithm from reaching the optimum. In practice selecting a fixed learning rate can be tedious. Usually, if the optimisation process stops showing sufficient progress then the fixed learning rate is replaced with a smaller value and the process is repeated to obtain a better optimum. In order to obtain a quick convergence while not allowing the learning rate to cause the divergence is a tricky task. Bottu *et al* [14] presents a diminishing learning rate schedule for this purpose,

$$\sum_{k=1}^{\infty} \gamma_k = \infty \text{ and } \sum_{k=1}^{\infty} \gamma_k^2 < \infty. \tag{2.55}$$

If a strongly convex function satisfies the assumptions of smoothness and limits for first and second moments [14] then sublinearly decreasing step size can achieve sublinear convergence to the optimum. Considering above results several learning rate schedules have been proposed such as momentum method, adaptive gradient algorithm [89], Adadelta [118], adaptive moment estimation [24] and root mean square propagation [107].

2.5 Sensor Data Processing

This section focuses on pre-processing the data from a camera and laser sensors and extracting useful information. Techniques described here, are used in latter chapters to assist in image classification tasks.

2.5.1 Image Processing

When solving vision problems, operation on the pixel level can be tedious since hundreds of pixels are associated with a single image. Therefore working on units formed by low level pixel grouping can dramatically reduce the computational cost. In addition these units can be more meaningful than a pixel and they also tend to have similar colour and texture that lead to eliminate redundant computations. These units can also assist with preserving the boundaries in the segmentation process. In practice these over segmented units are referred as super pixels.

Super Pixels

Simple Linear Iterative Clustering (SLIC) [1]method involves with over segmentation of the image using a pixel clustering technique depending on colour similarity and vicinity in the image coordinate system. The clustering occurs in a five dimensional space that consist of LAB [67] colour channels and x,y image coordinates. LAB color space represents all perceivable colors using three channels, L lightness, A and B chromaticity layers. Channels are defined using color-opponent theory, which indicate that there are opponent colors, which cannot be perceived at the same time. Channel A is defined from opponent colors green to red and the channel B is extending from blue to yellow. This colour space can provide perceptually uniform values for small colour distances. All the colour and location spaces are normalised to provide equal bias to all components.
Consider an image with m pixels, where required number of super pixels is $\mathcal{M}$. In this case $S = (m/\mathcal{M})^{0.5}$ where S denotes the super pixel size approximately. In the start it requires to extract $\mathcal{M}$ number of cluster centres at constant grid intervals S. Search area for each cluster centre is set to $2S \times 2S$ in the xy plane surrounding a cluster centre. Since the area of the expected super pixel is $S \times S$, it can be assumed that this super pixel lies with in the above search area.The colour(LAB) and location details correspond to a pixel can be denoted by 5D space vector $p_i = [l_i, a_i, b_i, x_i, y_i]$. Instead of computing the Euclidean norm for 5D space vectors, colour based and location-based distances are computed sepa-

rately, in order to reduce violation of boundaries as shown in Eq. (2.56),

$$D_{lab} = \sqrt{(l_c - l_i)^2 + (a_c - a_i)^2 + (b_c - b_i)^2}, \tag{2.56a}$$

$$D_{xy} = \sqrt{(x_c - x_i)^2 + (y_c - y_i)^2}, \tag{2.56b}$$

$$D_r = D_{lab} + \frac{\xi}{S} D_{xy}. \tag{2.56c}$$

Here (l_c, a_c, b_c) refers to the colour metrics of the cluster centre in LAB space. D_{lab} denotes the distance between the cluster centre and the pixel i in the clour space. Coordinates of the cluster center in the image frame is denoted by (x_c, y_c). D_{xy} indicates the distance between the cluster centre and the pixel i in the image coordinate frame. We denote the resultant distance (dissimilarity) between a cluster centre and pixel i by D_r, where ξ adjusts the compactness of superpixels. ξ normally vary between 1 to 20. Properly selected ξ can compromise between the colour contrast and spatial distance to generate super pixels with preserved boundaries.

Algorithm 2: SLIC Super pixels

Input: cluster centre initialisation at regular grid interval
$S : p_c = [l_c, a_c, b_c, x_c, y_c]^T$, RE - residual error
Output: clusters of pixels

```
repeat
    foreach p_k do
        Consider the pixels in the search area 2S × 2S
        Assign the pixel to the cluster p_k if it is the nearest cluster.
    end
    Compute new cluster centers by taking the average of the labxy vector of the
    pixels in each cluster
    Compute the residual error RE (distance between the current and the
    previous cluster centers) for each cluster
until RE < Threshold;
return final clusters of pixels
```

2.5.2 Feature Extraction

Visual data based on camera images contain rich information about the colour and texture of a scene. Most of the time these data includes some redundant and less informative parts. In order to efficiently use image data in applications such as classification it requires reducing the dimensionality of the data. This representative data can be referred to as features. Extracting the most important information from the image input and characterising it in a lower dimensional space is feature extraction. In practice features are expected to have following

properties:

- Capability to enhance the level of machine interpretation of the problem
- Relevance in the context of the application, i.e. features can be used to distinguish between classes
- Less sensitivity to noise
- Robustness to image transformations such as scaling and rotation
- Assisting with the generalisation steps.

However choosing an appropriate feature extraction method for a particular application should be done with a good consideration. Some examples for the commonly used feature extraction techniques in the field of image classification are SIFT descriptor [66], TextonBoost [100], optical flow and deep belief net [73]. Even though some features provide rich information that can enhance the accuracy of the results, computational complexity also rise with the sophisticated nature of the feature. It is important to select a technique that has a higher accuracy and reasonable computational efficiency. In our work we have utilised simple histograms based features including histogram of oriented gradient [121]. Through this we were able to achieve a better level of accuracy combined with fast operation.

Colour Histograms

Colour histograms describe the distribution of pixels in a colour space such as RGB, HSV or LAB. Each colour channel, in a space can be divided into small intervals called bins. The process is termed as colour quantisation. Quantifying the number of pixels contribute to each bin, the histogram can be extracted. The histograms can be generated for the entire image or the image patches. Discrimination power increases with the number of bins. Computational cost restrict the ability to process large number of bins efficiently. One of the advantage with the histograms is its robustness to translation and rotation. Sudden change of scale or view point do not create significant changes in the histogram measure. Histograms contain descriptive knowledge about the underlying image or the image patch. To illustrate, a low contrast image can be identified by a narrow histogram. A histogram slanted in the direction of the high end, suggests a bright image. A bimodal histogram is an evident for a foreground object with a contrasting background. Example are shown in Figure 2.1. In order to extract statistical

features, histogram can be represented in a probabilistic manner as follows,

$$P(g^*) = \frac{m(g^*)}{m}, \tag{2.57}$$

where g^* is the gray level and $P(g^*)$ is a statistic to measure the texture of a image region which is referred as the first order histogram probability, m is the number of pixels in the image region/segment and $m(g^*)$ is the number of pixels in gray level g^*. Different types of informative features such as mean, variance, entropy and energy can be extracted using $P(g^*)$ distribution. To illustrate, the square root of the variance can be used to represent details about the contrast of the image. Energy values describe the intensity distribution.

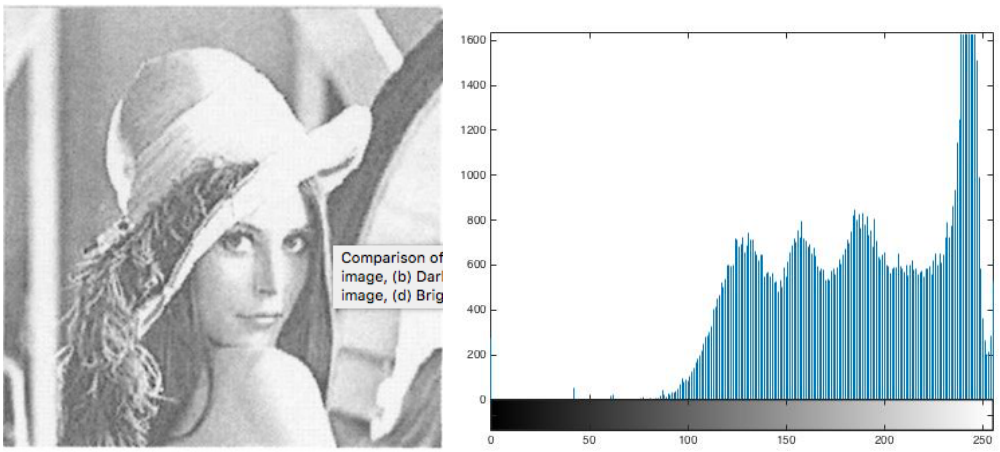
HOG Features

HOG is a visual descriptor that can be efficiently used in image classification and object recognition. The intuition behind the descriptor is that local intensity gradients or edge directions can efficiently illustrate the local texture and object contours.

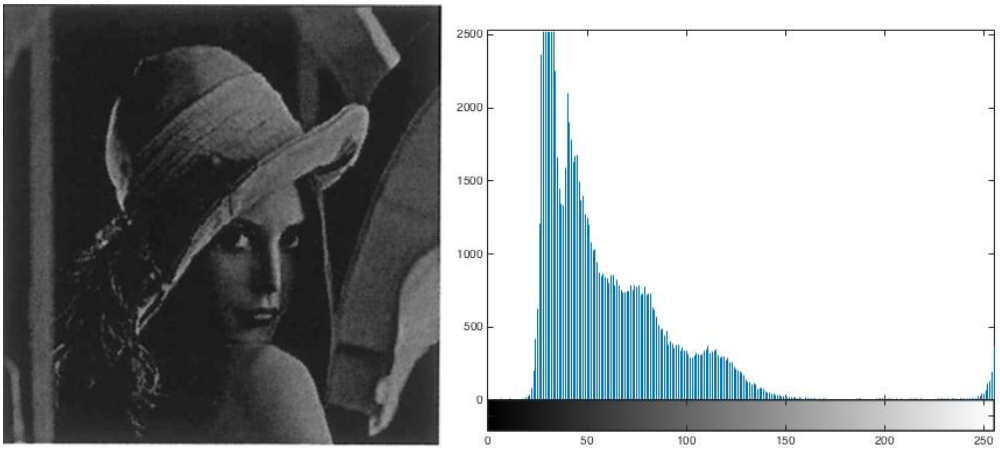
Usually the image is split in to small areas called cells. The cells normally have rectangular or radial shapes. Each cell is associated with a 1-D histogram which consist of constant number of bins divided based on the gradient orientation. The range of the bins is typically from 0 to π radians or 0 to 2π radians. First range is used when the angles are unsigned. Each pixel in a cell, proportionally votes to the corresponding bin based on their gradient orientation. For a particular application we can select appropriate number of histogram channels. Typically for colour images, separate histograms are calculated for each channel. Depending on the application the bias of the pixel vote can be replaced by function of the gradient magnitude. To increase the robustness of the model for lightning changes, shadowing and similar eccentricities, normalisation can be applied. For this purpose normalising constant is derived using the adjacent cells in a defined block. The block can be selected allowing for overlapping. Hence some cells contribute to the overall descriptor multiple times. This overlapping among the blocks preserves the information on local variations. At the end all the normalised histograms correspond to each cell is concatenated form the final descriptor. This descriptor can be a represented as a feature vector to contrast between different object classes.

R-HOG represents the case when the blocks are rectangular. This approach is typically implemented using the count of cells in one block, count of pixels in each cell and count of histogram channels. Circular HOG blocks (C-HOG) can be found in two variants: those with a single, central cell and those with an

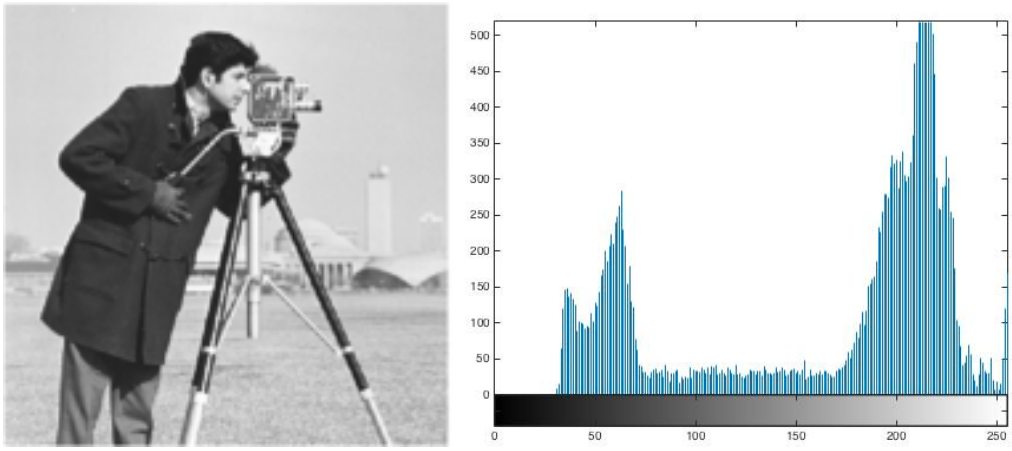
angularity divided central cell. In addition, these C-HOG blocks can be described with four parameters: the number of angular and radial bins, the radius of the center bin, and the expansion factor for the radius of additional radial cells. Figure 2.2 depicts the HOG descriptor for an image from the PASCAL dataset.



(a) Bright Image and Intensity Histogram



(b) Dark Image and Intensity Histogram



(c) Image with contrasting background and Intensity Histogram

Figure 2.1: Figure 2.1a shows a histogram slanted towards high end, Figure 2.1b shows a histogram slanted towards low end and Figure 2.1c depicts is a bimodel histogram

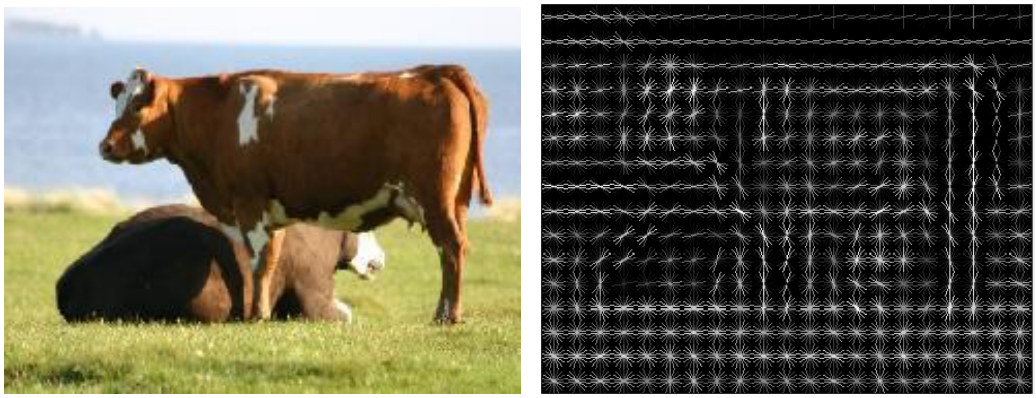


Figure 2.2: The figure shows the histogram descriptor of a image from PASCAL dataset [27]. The gradients are denoted in gray scale. Cells have histogram bins divided in the range of from 0 to 360 degrees. The colour index close to 255 implies larger magnitude of gradients

2.5.3 3D Point Cloud Processing

Laser sensor output is a 3D point cloud which is a set of data points in Cartesian coordinates x,y,z. In addition to the location information each points includes an intensity value. In point cloud processing, single points do not contain sufficient information to measure similarity between each other. Hence local descriptors such as shape descriptors are adopted. Considering the neighbourhood of a point, geometric features correspond to the surface which the point lies on can be obtained. Once this information is associated with the underlying point, the comparison between points during the processing becomes more meaningful because the each point contains more contextual information. Here the expectation is to derive set of geometric features that have similar values for the points lying on the same surface or alike surfaces. These feature outputs also should contrast if the points lie on dissimilar surfaces. Similar to image based features, these geometric features also should have characteristics such as robustness to 3D rotation and translation and less sensitivity to mild noise. Typically laser scanners tend to have varying density in the point cloud. Hence the feature vector should be independent from the density of the corresponding neighbouring volume.

Random Sample Consensus (RANSAC) Algorithm

Ground / non-ground identification is critical in autonomous driving. Since ground has areas such as paved area, road, grass and sand it can be challenging to locate ground by pure manipulation of the visual data. Instead, a laser point cloud can provide additional information about the ground plane. One of the most popular method for ground plane extraction is the Random sample consensus. The main intuition of the method is obtaining a set of model parameters for

a data distribution that can be used to describe the distribution by eliminating the outliers.

To elaborate, RANSAC can also be considered as an outlier detection technique. RANSAC generate a parameter set to represent a model for a set of observed data while removing the effect of the outliers. Outlier can occur not only when there is noisy data but also due to a false hypotheses about the data. The latter is useful in extracting the horizontal ground plane (largest horizontal plane in the image). RANSAC is an iterative method. Usually, the results generated by the method are associated with a probability that indicates the reliability of the result. This probability is increasing with the number of iterations undergone. The RANSAC algorithm was originally proposed in [31]. An overview of the method is presented in [22].

Consider the Algorithm 3, where model type M defines the structure of the model depending on our particular application, i.e. for ground extraction problems, the model can be a horizontal plane (with vertical surface normal) close to the ground. At this stage parameters for the model are unknown. n_0 can depend on the nature of the model and the particular application. In theory, $\bar{S}1$ is named as the consensus set of $S1$. This process is recursively run for T number of iterations. It can be defined a relationship between T and ρ which is the probability that there exist at least a one random sample set which only contains inliers. If the probability of outlier occurrence is ν then the T can be given by:

$$T = \frac{\log(1-\rho)}{\log(1-(1-\nu)^{n_0})}. \tag{2.58}$$

At the end of each iteration, either the existing model is refined or rejected depending on the size of the consensus set. At the end of the iterations, the model containing the highest number of inliers can be selected as the best fitting model. In practice ϵ and γ are selected based on the experimental process considering the particular application and the nature of the data distribution.

Algorithm 3: RANSAC

Input:

M_s -selected model

n_0 - minimum number of data points required to obtain parameters for M_s

P_n - observed data distribution

n_1 - number of observed data points where $n_0 < n_1$

ϵ - a threshold value to decide if a data point fits the model

γ - number of minimum points required to ensure that the model fits well with data

Output:

M_{best} = model best fit the data

Draw a random sample set S_1 (hypothetical inliers) from P_n, $size(S_1) = n_0$
Obtain parameters for the model M_s using $S_1 \Rightarrow M_1$ = Instantiated model
$M_{best} = null$, $\text{Quality}_{best} = 0$
foreach $i = 1$ *to* T **do**
    $\bar{S_1}$ = points fit to model M_1 with a tolerance ϵ
    if $size(\bar{S_1}) > \gamma$ **then**
        Refine model parameters using $\bar{S_1}$
    else
        Discard the existing model
        Redraw a random sample set S_1 (hypothetical inliers) from P_n
        re-estimate parameters for the model type M using $S_1 \Rightarrow M_1$ = model with re-estimated parameters
    end
    $\text{Quality}_{current}$ =number of inliers for the current model
    if $Quality_{best} < Quality_{current}$ **then**
        $M_{best} = M_1$
        $\text{Quality}_{best} = \text{Quality}_{current}$
    end
end
return M_{best}

Euclidean Cluster Extraction

Clustering is the process of grouping the points in an unorganized point cloud into meaningful parts. These smaller groups are easier to process and information extraction from them is more efficient. The main approaches are similarity based clustering and feature based clustering. Similarity based methods define boundaries to subdivide the point cloud based on a proximity measure.

Algorithm 4: Euclidean Cluster Extraction

Input:
$\mathcal{R}$ = Kdtree formulation of the 3D point cloud
$\mathcal{C}_L$ = List of clusters
$Q_{\mathcal{R}}$ = queue of interested points
d_n = radius for point neighbourhood
d_u = upper bound for d_n
Output: $\mathcal{C}$

$\mathcal{C}_L = \emptyset$, $Q_{\mathcal{R}} = \emptyset$
foreach $\rho_i \in \mathcal{R}$ **do**
  add ρ_i to $Q_{\mathcal{R}}$
  foreach $\rho_i \in Q_{\mathcal{R}}$ **do**
    neighbours of ρ_i (with in a sphere of d_n) $\rightarrow \rho_i^k$
    foreach $\varrho \in \rho_i^k$ **do**
      if ϱ *is not already been processed* **then**
        add ϱ to queue $Q_{\mathcal{R}}$
      end
    end
    current set of points in $Q_{\mathcal{R}}$ forms a cluster. add $Q_{\mathcal{R}}$ to list of clusters $\mathcal{C}_L$
    $Q_{\mathcal{R}} = \emptyset$
  end
end
return $\mathcal{C}_L$

This proximity is computed using a metric such as Manhattan or Euclidean distance. Algorithm 4 shows the implementation of the Euclidean cluster extraction [85]. For cluster generation, it is important to discriminate one point cluster from another. For cluster $\mathcal{C}_i = \{\mathrm{r}_i \in \mathcal{R}\}$ to be disjoint from cluster to $\mathcal{C}_j = \{\mathrm{r}_j \in \mathcal{R}\}$, it requires that $\arg\min ||\mathrm{r}_i - \mathrm{r}_j|| \geq d_u$ where d_u is a threshold distance. This means that the distance between every point in $\mathcal{C}_i$ and every point in $\mathcal{C}_j$ should be greater than the given threshold for $\mathcal{C}_i$ and $\mathcal{C}_j$ to be separate clusters. The distance threshold is obtained via approximate nearest neighbour queries. The notion of neighbourhood provides a maximum distance value for a point to be a neighbour of a query point. K-d tree representation is used to structure the data and find the threshold d_u. Once the threshold is found clustering is performed by grouping neighbours.

2.6 Summary

This chapter presented the theoretical groundwork for the semantic segmentation framework proposed in the later chapters. Different classification methods

are used to label the image pixels. In order to utilised the contextual information and solve the image classification problem more resourcefully, conditional random fields are commonly used in image classification. Exact CRF inference is a challenging problem for complicated graph structures with higher tree width. Hence approximate inference methods are discussed in the chapter. We discuss the derivation of quadratic programming relaxation, which is used to solve the CRF inference problem related to the image segmentation model in Chapter 3. This is important because QP turns the problem in to an optimisation problem which facilitates the addition constraint to the inference problem. This characteristic is systematically used to improve the quality of the segmentation in the next chapter.

Optimally learning the CRF parameters can improve the quality of the results. Loss based parameter learning explained in this chapter, has been used in the Chapter 4 to estimate parameters for the semantic segmentation model. This later works also exploits the theory and the practical aspects of the stochastic gradient decent discussed in this chapter in order to optimise the loss function in an online setting.

Theory and techniques described in this chapter have been applied in later chapters to develop an efficient scene-understanding framework using multimodal data. Camera and Velodyne sensors are used to test the framework. Sensor data is processed in parallel. Super pixels are extracted from camera images using SLIC algorithm. Subsequently, a linear discriminate analysis classifier is trained on image super pixels. Simultaneously, ground plane is removed from the laser point cloud using RANSAC to facilitate 3D objects extraction with Euclidean clustering. Lastly a CRF model is proposed to improve image classification using additional label consistency information obtained from the extracted objects. Afterwards, proposed CRF model is extended in to an adaptive model via online parameter training.

Chapter 3

Urban Scene Segmentation with Laser-Constrained CRFs

3.1 Introduction

This chapter presents a novel formulation of CRFs to incorporate "a priori" knowledge in the form of global constraints and thus conduct semantic scene segmentation efficiently. The work reported here was formerly published in IROS[1].

Scene segmentation is a core competency for many robotic tasks. It provides the foundation, which allows a robot to understand and reason about its environment. For navigation in urban environments, such information is critical for safety, as it allows the robot to predict which areas pose a risk due to the presence of dynamic objects. Robots carry many different sensors, such as cameras, laser scanners, RGB-D cameras, etc, which typically observe the environment from slightly different angles due to the physical placement. This variation in viewpoint and diversity of the sensor observations make the optimal sensor fusion challenging. In this chapter, we propose a model, which effectively combines the information from multiple modalities. The method is applied to image segmentation using camera and laser scan data but is general in nature applicable to a wide variety of sensor combinations.

CRFs are efficiently used for image labeling due to several advantages. First, it is a convenient tool for structured prediction, that takes in to account the entire structure of the image for label prediction. Label prediction problem is addressed by formulating it as a probabilistic inference problem that can be solved as a standard optimisation problem. CRFs also encourage similar pixels to have similar labels, establishes spatial regularity and coherency. This reduces the noise in label assignments. CRFs are also capable of higher-level object recognition and longer range labeling relationships representation. CRFs can be built on pixels, superpixels or image patches depending on the application.

It is clear that we want to enforce the validity of the additional information or

[1]Charika De Alvis, Lionel Ott, Fabio Ramos. Urban Scene Segmentation with Laser-Constrained CRFs. In *IEEE International Conference on Intelligent Robots and Systems (IROS)*, 2016

multiple sensor information in the scene labeling problem. As such it is impractical to encode it in the overall cost function directly and solve it efficiently unless the problem is formulated as a standard optimisation problem. Therefore we propose a framework based on a relaxed quadratic program formulation of CRFs for scene segmentation which can conveniently be enforced with a set of global constraints. These global constraints contain "a priori" information about the label consistency. Information from one or multiple sensor modalities can be used to derive these constraints. Since these constraints are applied on the optimisation process, it is important that the accuracy of the constraints are maintained at a higher level. In our work image data is used to build the CRF graph and potential functions while the depth data is used to formulate global constraints over sets of nodes in the CRF. Each constraint encloses nodes belonging to a single object, as determined by the depth data and ensure they take the same label during the optimisation process. The interested optimisation problem is a linear equality constrained problem, which can be easily solved to obtain the MAP solution of CRF model using an efficient gradient-based algorithm introduced in [120].

The main contributions of this chapter are:

- Novel CRF formulation for scene labeling by adding global constraints, that are capable of enforcing label consistency, i.e. nodes on the same object are assigned with the same label
- Demonstration of the powerlessness of the proposed framework for urban scene segmentation by combining image and 3D laser data gathered by a real robotic platforms.

The remainder of this chapter consists of the following sections. Section 3.2 describes related work on the theory and practices in CRF based semantic scene segmentation. Section 3.3 describes enforcing global constraints in scene labelling process. Experimental framework, results and discussions are included in Section 3.4.

3.2 Related Work

CRF Based Semantic Scene Segmentation

As described in Section 2.2 Semantic segmentation problem is commonly formulated as a CRF, where CRF is constructed on pixels or image patches. The CRF model consists of smoothing terms that support label agreement between similar pixels and combines more descriptive terms that model contextual connections between object classes. CRF also has the capability to model long range relationships and global dependencies with in the image. There are several different

approaches to add long range information such as intricate potentials, global image features [108], segment and region based label consistency [33, 46] and global co- occurrences statistics between different classes [59, 61]. Unfortunately, for grid-like graphs, which we are particularly interested in image segmentation, the problem of multi-class classification is NP-hard. Therefore we need to rely on approximate inference algorithms.

The segmentation of the image corresponds to the minimal cut [32] of the CRF. Minimal cut ensures cutting all the edges between the pixels belong to different classes. In computer vision, many successful image segmentation methods are based on graph cuts [16] and its refinements such as normalised cuts [98, 15]. Ladicky et al. [62] propose a improved scene segmentation model based on a novel undirected graphical model, associative hierarchical random fields which is also solved using a graph cut based method. Felzenszwalb and Huttenlocher [29] introduce a predicate to measure evidence for a border between two distinct regions (can be different classes/objects) in an image-based graph. Recently, quadratic programming relaxations [87] have been successfully applied to MAP inference in conditional random fields. At this point it has been proven that QP relaxation is tight and exactly solves the original MAP problem of the CRF.

Semantic Scene Segmentation in Robotics

In robotic applications stereo vision is commonly used for scene segmentation. He and Upcroft [41] introduce a stereo image based system and a non parametric model depend on data to semantically segment images in large scale. The proposed model operate without any offline learning. He *et al.* [42] present the use of both colour and depth cues for automatic segmentation. A similar approach is taken in [97] to automatically labels street scenes through a hierarchical CRF model that is solved using Alpha expansion algorithm [18]. The energy function for the model consists of pairwise potentials correspond to the disparity of the neighbouring nodes in addition to the commonly used color and texture based potentials. A method to semantically segment dense 3D point clouds obtained by RGBD sensors is presented by Hermans et al. [45]. This novel model propagates 2D labels in to 3D space by Bayesian updates and dense pairwise 3D CRF. In robotics, there are several approaches on scene segmentation using multiple modalities, such as camera and 3D laser data. Douillard et al. [23] propose a spatial-temporal CRF method integrating measurements from a conventional $2D$ laser scanner with images from a calibrated camera. Munoz et al. [74] extract features from an image and laser data and use in a classifier to segment the scene. Similar works are done in [116]. Xu *et al.* [115, 116] introduce novel framework for information fusion based on over-segmentation and Dempster-Shafer theory.

Above method generate promising results but confined to a single obstacle class since it lacks ability to identify different objects in the obstacle class. Munoz *et al.* [74] treat two sensor modalities as first class objects and propose a joint inference, that couples the predictions among all modalities. The model enable propagating information through domains during the inference process. This is important because some domains are more apt at predicting certain classes than others, i.e. images can be used to distinguish between elements that are similar in shape but have different texture (e.g. grass , sand vs road), and laser point clouds are useful to differentiate objects that have similar appearance but different in scale (e.g wall and a car with similar color and texture). Visual data based segmentation model introduced by Felzenszwalb and Huttenlocher [29] is extended to used RGBD data is in [104]. This method creates a colored laser point cloud by combining camera images and a 3D Lidar point clouds. Then through an efficient graph-theoretic algorithm segmentation is obtained. In [2] voxels are generated from a 3D urban point cloud and then supervoxels are formed using voxels. Subsequently super voxels are clustered by link-chain method [117] to extract the objects from the point cloud. Later classification of the objects conducted using local descriptors based on color and laser intensity and other geometrical features. A method that exploits both colour and laser based depth information is presented in [119]. This method makes predictions using unimodel classifiiers on the depth and colour data . Subsequently utilise late fusion architecture to fuse unimodel classifiers and the resulting output is post-processed using a CRF. Kundu *et al.* [60] introduce a higher order CRF model for joint inference of 3D structure and semantics in a 3D volumetric model. They use depth and visual information in the form of unary, pairwise and higher order potentials and MAP estimation over a random field. Ladicky *et al.* [105] also formulate an energy minimisation problem of a random field to conduct object class segmentation . In this case also depth and visual information are incorporated as different form of potentials. However both above methods are tested with comparatively low-resolution images. Hence the accuracy of the labeling and the computational time for high-resolution images are unknown.

Deep Network Experts For Semantic Segmentation

Eitel *et al.*[26] extend conventional CNN to use multimodal information, RGB images and depth images. The model trains the network separately for each modality using two parallel streams. Afterwards, fine tunning is done for the two streams jointly to derive a fusion network that can perform the final classification. The model is tested only for short-range indoor foreground object recognition tasks. In [93], authors test different CNN models developed based on the

architecture proposed in [8] which can efficiently exploit RGB data from images and HHA [39] features from lidar points for pedestrian detection task. They also empirically conclude that late fusion of RGB and HHA would yield more accurate detections. Audebert *et al.*[6] performs semantic segmentation by improving the conventional RBG (3 channel) based SegNet [7] model. They use IRRG images with near infrared , red and green channels as input to a SegNet model and also data collected from a aerial laser sensor as input to the another SegNet model. The laser data also formed as a composite image that consists of three channels DSM, NDVI and NDSM[36]. Then using the theory of residual deep learning[43], fusion of the two models is conducted. In[70], it is presented a framework to optimise the image classification by mixing multiple CNNs trained on distinctive modalities. Feature representation of the CNNs is used to train weight of the gating network, which produces the final classification. The framework is tested for pedestrian detection. The main drawback of using deep networks is that it requires large amount of labelled data for training. In contrast our framework can use any conventional classifier such as Naive Bayes or SVM that is learnt with less training examples. In addition, laser-based information is incorporated, enhancing classification in a unsupervised manner. Further, any additional multimodal data that includes label consistency information also can be directly incorporated to our framework with no training.

Higher Order Potentials

Higher order potentials (HOP) introduced in Section 2.2 [53] can encode additional information which can be used to model longer range information within the model. These potentials became popular in CRF based labelling tasks. HOPs and ray potentials [91] enforce soft constraints on the optimisation. However, the complex potential terms associate with additional parameters. This makes the models intricate and time consuming to learn. Kohli et al. [53] use a P^n Potts model-based CRF with HOP for the task of image segmentation and use a graph cut based algorithm to solve the optimisation problem. Tarlow et al. [106] proposed a method with HOP models and belief propagation, adopting a set of potentials for which efficient message passing rules exist. In [55] a dual decomposition based master-slave framework is presented to solve generic higher order Markov random fields.

These soft constraints (higher order potentials) containing long-range information may be violated in the optimisation process, since they are not strictly enforced. Although when we have precise "a priori" knowledge on long-range relationships with in the image, discarding this information can reduce the quality of labelling. Our method on the other hand, strictly enforces the validity of

global constraints during optimisation. In our approach, we demonstrate that the imposed constraints from depth information are satisfied by the solution.

3.3 A Model to Fuse Laser and Visual Information

3.3.1 Overview

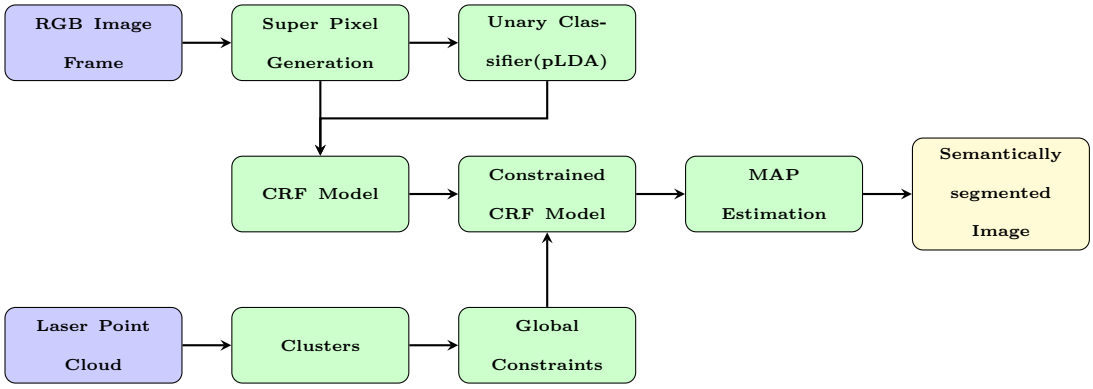


Figure 3.1: Overview of the Global Constraint Based Semantic Segmentation Model

A schematic overview of the processing pipeline is shown in Figure 3.1. In the first step our method preprocess and extracts features from raw images (first row in Figure 3.1) in the following manner:

1. Extracting super pixels from raw images where superpixels contain group of neighboring pixels with proximity in space and color.

2. Extracting features from the superpixels based on colour, texture and location. Subsequently training pseudo linear discriminant analysis classifier (pLDA) [68].

In parallel to training the classifier we process the laser point cloud in the following manner(bottom row in Figure 3.1):

1. Extracting Euclidean clusters from the laser point cloud which contain complete or part of objects (due to occlusion).

2. Mapping the 3D point clusters to the 2D camera frame and locate the superpixels correspond to the each cluster. These sets of super-pixels are used in the form of constraints to impose label consistency on scene classification in the next stage.

We conduct the semantic scene labeling in the last stage using the visual feature base classifier and the laser based constraints (middle row inFigure 3.1) as follows:

1. Building a CRF on the interested image. Integrating the classifier based potentials with the CRF model.

2. Adding the laser based constraints to the inference problem of the CRF.

3. Conducting the inference and obtaining semantic labels for the images.

3.3.2 Superpixel Generation

Commonly image labeling algorithms utilise image pre-processing techniques. To reduce the computing cost and facilitate the method to run in real time, we use an over-segmentation of the image into super pixels. Subsequently image labelling is conducted on superpixel level. This approach increases the possibility to preserve class boundaries in the semantic segmentation process. We generate super pixels using SLIC algorithm. SLIC generate approximately uniform superpixels (in size and shape). Moreover, the method is memory efficient and has a computational complexity of $O(N)$ which is much less compared to the other state of the art methods.
More importantly SLIC superpixels are compact and high in quality. Quality usually links with the capability to adhere to class boundaries. Boundary adherence is evaluated using boundary recall that computes the portion of the class boundaries (given by the ground truth) lying within in at least two pixels of a superpixel border. Higher boundary recall suggest that only few true class boundraies are violated by the superpixels. Under-segmentation error can also be used to assess boundary adherence. This approach consider the reigions of the image correspond to each class and the superpixels enclosed by that region. The quality measure corresponds to the number of pixels of the underlying superpixels that lie across the region boundary (defined by the ground truth image).

3.3.3 Feature Extraction

We build a CRF on the image and conduct inference to obtain the optimum labelling. To this end, we exploit super pixels as nodes in the graph. It requires evaluating the node potentials of the graph in order to perform inference. Therefore a pLDA classifier is trained on super-pixels exploiting simple colour, texture and location features. Location of super pixels in the camera frame, HSV colour histogram per superpixel and HOG features are used for training. HOG features are one of the simplest measures of texture. In this model, HOG features are extracted in RGB space. The combination of two colour spaces assists in overcoming the loss of important information. The classifier provides the posterior

probability of super-pixels/nodes attaining a certain class label. We have used this information as unary potentials in the CRF.

3.3.4 Laser Point Based Clusters

The driving factor of our framework is imposing a priori knowledge as hard constraints on the final labelling problem. We use image information to solve the super-pixel labelling problem. Additionally, laser based information is used as constraints in the optimisation problem. To this end, we obtain the laser based constraints on sets of super pixels by extracting groups of connected points, or objects, from the Velodyne point cloud. This process requires time synchronised camera images and Velodyne point clouds.
To generate the constraints we first perform a ground plane removal step using RANSAC to find the largest plane aligned with the ground. Subsequently, remaining points are grouped using Euclidean cluster extraction [85]. We fine-tune the clustering algorithms so that each cluster contains only points belonging to a single class. From these extracted, we only consider those that include more than 150 points to avoid the issues with noise. The 3D coordinates of the points contain in the selected clusters are then translated into image space coordinates using the extrinsic calibration provided by the dataset. The projected points in the image space then associated with super pixels. In other words, we can obtain super-pixel clusters correspond to complete objects or object parts. The ground plane as well as the retained point clusters after the projection can be seen in Figure 3.4c and Figure 3.4d respectively. Based on this 2D clusters we create constraint sets $\mathbb{C}$ used in the optimisation. All super pixels that correspond to the same laser segment are constrained to be assigned the same label. Super-pixels which do not belong to any of the extracted laser segments are kept unconstrained.

3.3.5 CRF Model For Image Segmentation

Consider a graph $G = (V, \mathbf{E})$ where$V$ and $\mathbf{E}$ denotes the sets of nodes and edges respectively. Each node in set V represent an image superpixel in set S. $\mathbf{y} = \{y_1, y_2, \ldots, y_m\}$ is the set of discrete random variables associated with nodes. Number of nodes is denoted by m. Each random variable y_i is assigned with one of the labels from $L = \{1, \ldots, n\}$ where n is the number of labels. $\mathbb{N}(i) = \{j \in V | (i, j) \in \mathbf{E}\}$ denotes the neighbours of node i. Then a pairwise conditional random field is defined on G such that, the random variables $\mathbf{y}$ conditioned on $\mathbf{x}$ and it satisfy the Markov property with respect to the graph: $P(y_i|\mathbf{x}, y_j)_{j \neq i} = P(y_i|\mathbf{x}, y_j)_{j \in \mathbb{N}(i)}$. Semantic segmentation for the images are obtained through the

inference of the CRF.
$\phi_i(y_i)$ denotes the unary potentials obtained from a linear discriminant analysis classifier described in Section 2.1.1 and $\psi_{ij}(y_i, y_j)$ refers to pairwise potentials between connected nodes.

$$\phi_i(y_i^p) \propto D_E(\mathbf{x}\Phi, \bar{\mathbf{x}}_p\Phi), \tag{3.1}$$

here $\mathbf{x}$ is the newly observed image feature vector described in Section 3.3.3, p refers to the class index, D_E is the Euclidean distance and $\bar{\mathbf{x}}_p$ refers to the mean of the feature vectors correspond to the training samples with label p. Φ is the linear transformation matrix (refer to the Section 2.1.1 for explanation). The edges between nodes are created considering their distance within the image, i.e.:

$$\mathbf{E}(i,j) = \begin{cases} 1 & \text{if } \mathrm{dist}(i,j) < \zeta \\ 0 & \text{otherwise} \end{cases}, \tag{3.2}$$

where ζ is the distance threshold and $\mathrm{dist}(i,j)$ is the Euclidean distance in image coordinates between centres of two super pixels. All super pixels closer than the user defined threshold are connected. Selecting larger values for ζ allows encoding longer range information but also increases the computational complexity. In our experiments ζ was set such that each node is connected to roughly ten neighbouring nodes, which results in a grid like structure. The pairwise potentials ψ_{ij} are derived based on their dissimilarity using colour, texture, and location information of the super pixels, i.e.:

$$\mathrm{dis}(i,j) = \frac{1}{3}\Big[\theta_c||\mathbf{mc}(i) - \mathbf{mc}(j)||_2 + \theta_l||\mathbf{cm}(i) - \mathbf{cm}(j)||_2 + D_B(\mathbf{cl}(i), \mathbf{cl}(j))\Big], \tag{3.3}$$

where $\mathbf{mc}(i)$ is the mean colour vector of the i-th super pixel normalised by θ_c which is the maximum possible color contrast under the HSV color space, $\mathbf{cm}(i)$ is the centre of mass of the super pixel, normalised with θ_l which is the maximum possible distance between two points in the image frame and $\mathbf{cl}(i)$ is the colour histogram for the i-th super pixel, for which similarity with neighboring superpixel is computed using the Bhattacharya distance given by the following equation,

$$D_B(a,b) = \sqrt{1 - \frac{1}{\sqrt{\sum_i a_i \sum_i b_i N_B^2}} \sum_i \sqrt{a_i b_i}}, \tag{3.4}$$

where a and b are two histograms and N_B is the number of bins in the histograms. This results in a similarity value between 0 and 1, with 0 encoding identical super pixels. We use the pairwise potential function to encourage the neighbouring superpixels to have similar labels if disssimilarity between them based on the

colour,texture and location is small as follows:

$$\psi_{i,j}(y_i^p, y_j^q) = \begin{cases} 1 - \text{dis}(i,j)^2 & \text{if } p = q \\ \text{dis}(i,j)^2 & \text{otherwise} \end{cases}. \tag{3.5}$$

The additional *a priori* information about sets of points which belong to the same group is encoded as constraints on the CRF in a later stage. A graphical representation of this structure is shown in Figure 3.2, where nodes are denoted by circles while edges indicate connections between nodes. The set of nodes coloured identically are constrained to take the same label. The unary and pairwise potentials are based on information extracted from the image while the information about groups of nodes is extracted from 3D laser data.

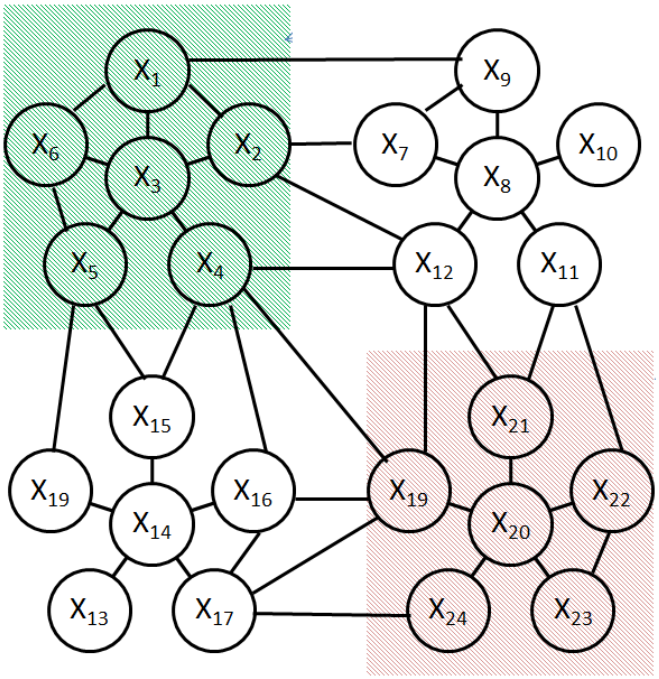


Figure 3.2: Example of the type of CRF structure used in this work. The two shaded areas, A and B, encode sets of nodes which are required to be assigned the same label.

3.3.6 MAP Estimation

Our goal is to find the best label assignment for query variables associated with each node. This assignment is denoted as MAP estimation (Section 2.2.1). The conditional distribution presented in Eq. (2.6) can be reformulated for the unary and pairwise cliques. Taking the log of the likelihood function results in the following conditional log likelihood of the query variables:

$$\log P(\mathbf{y}|S) = \underbrace{\sum_{i \in S} \phi_i(y_i)}_{\text{Unary Potentials}} + \underbrace{\sum_{i \in S, j \in \mathbb{N}(i)} \psi_{ij}(y_i, y_j)}_{\text{Pairwisw potentials}} - \mathcal{Z}(S), \tag{3.6}$$

where $\mathcal{Z}(S)$ is the partition function.

Quadratic Program Formulation

As finding the MAP solution to Eq. (3.6) is NP hard we use QP based approach to solve the inference problem. Refer to Section 2.2.2 for details on the QP relaxation. We start by reformulating the MAP problem as a quadratic integer program [120] as indicated in Eq. (3.7),

$$\begin{aligned}
\text{maximise} \quad & \sum_{i \in S} \sum_{p \in L} \phi_i(y_i^p) \mu_i(y_i^p) \quad + \sum_{\substack{i \in S \\ j \in \mathbb{N}(i)}} \sum_{p,q \in L} \psi_{ij}(y_i^p, y_j^q) \mu_i(y_i^p) \mu_j(y_j^q) \\
\text{subject to} \quad & \sum_{p \in L} \mu_i(y_i^p) = 1 \quad \forall i && \text{(3.7a)} \\
& \mu_i(y_i^p) \in \{0, 1\} \quad \forall i, p, && \text{(3.7b)}
\end{aligned}$$

with the indicator function:

$$\mu_i(y_i^p) = \begin{cases} 1 & \text{if } y_i^p = 1 \\ 0 & \text{otherwise} \end{cases}, \tag{3.8}$$

where y_i^p encodes if node i has been assigned label p. This quadratic program formulation penalizes disagreements between the data via the indicator function, which guides the model to obtain coherent segmentation. Additionally, Equations (3.7a) and (3.7b) enforce that exactly one label is selected for each node. In order to make the NP hard problem solvable we relax the integer requirement of the quadratic program [120] as follows:

$$\begin{aligned}
\text{maximise} \quad & \sum_{i \in S} \sum_{p \in L} \phi_i(y_i^p) \mu_i(y_i^p) \quad + \sum_{\substack{i \in S \\ j \in N(i)}} \sum_{p,q \in L} \psi_{ij}(y_i^p, y_j^q) \mu_i(y_i^p) \mu_j(y_j^q) \\
\text{subject to} \quad & \sum_{p \in L} \mu_i(y_i^p) = 1 \quad \forall i && \text{(3.9a)} \\
& 0 \leq \mu_i(y_i^p) \leq 1 \quad \forall i, p. && \text{(3.9b)}
\end{aligned}$$

This relaxation is tractable and can be solved by standard optimisation algorithms, thus it yields an optimal solution equivalent to the MAP solution for the original problem denoted in Eq. (3.6). Proof is elaborated in [87].

Globally Constrained QP

Optimisation of quadratic programming is a vastly studied area. In this stage we add global level constraints to the maximisation problem. These equality

constraints are derived in Section 3.3.4, resulting in following problem:

$$\text{maximise} \quad \sum_{i \in S} \sum_{p \in L} \phi_i(y_i^p)\mu_i(y_i^p) \quad + \sum_{\substack{i \in S \\ j \in N(i)}} \sum_{p,q \in L} \psi_{ij}(y_i^p, y_j^q)\mu_i(y_i^p)\mu_j(y_j^q) \tag{3.10a}$$

$$\text{subject to} \quad \sum_{i,j \in \mathbb{C}_k} \sum_{p \in L} \mu_i(y_i^p) - \mu_j(y_j^p) = 0 \quad \forall \mathbb{C}_k \in \mathbb{C} \tag{3.10b}$$

$$\sum_{p \in L} \mu_i(y_i^p) = 1 \quad \forall i \tag{3.10c}$$

$$0 \leq \mu_i(y_i^p) \leq 1 \quad \forall i, p, \tag{3.10d}$$

where Eq. (3.10b) enforces that all pairs of nodes i and j in a constraint set $\mathbb{C}_k \in \mathbb{C}$ are assigned the same label.

Inference of Globally Constrained QP

We use the methods described in the Chapter 2.2.2 (Equality Constrained Quadratic Programming Problems) to solve the QP relaxation problem. Consider an objective function similar to Eq. (3.10a) which has constraints set similar to and Eq. (3.10b), multiplying by -1 we can formulate it as a minimisation problem,

$$\text{minimise} \quad \sum_{i \in S} \sum_{p \in L} -\phi_i(y_i^p)\mu_i(y_i^p) \quad + \sum_{\substack{i \in S \\ j \in N(i)}} \sum_{p,q \in L} -\psi_{ij}(y_i^p, y_j^q)\mu_i(y_i^p)\mu_j(y_j^q) \tag{3.11a}$$

$$\text{subject to} \quad \sum_{i,j \in \mathbb{C}_k} \sum_{p \in L} \mu_i(y_i^p) - \mu_j(y_j^p) = 0 \quad \forall \mathbb{C}_k \in \mathbb{C}. \tag{3.11b}$$

Following [56], we can rewrite the minimisation in matrix notation,

$$\text{minimise} \quad \frac{1}{2} Y^T Q Y + b^T Y \tag{3.12a}$$

$$\text{subject to} \quad AY = 0, \tag{3.12b}$$

where $Q \in \mathbb{R}^{mn \times mn}$ is a symmetric matrix which encodes the negative quadratic coefficients (pairwise potentials) and $b \in \mathbb{R}^{mn \times 1}$ negative linear coefficients (unary potentials). $Y = [\mu_1(y_1^1), .., \mu_1(y_1^n), .., \mu_m(y_m^1), .., \mu_m(y_m^n)]^T$ is the indicator variable matrix representing $\mu_i(y_i^p) \forall i \in V$ and $p \in L$. $A \in \mathbb{R}^{e \times mn}$ is a matrix with full row rank which encodes the global constraints from Eq. (3.11b). Here e is the number or independent constraints.

To solve the constrained minimisation problem in Eq. (3.12) efficiently, it can be reduced in to a simpler unconstrained problem. We can use null space method

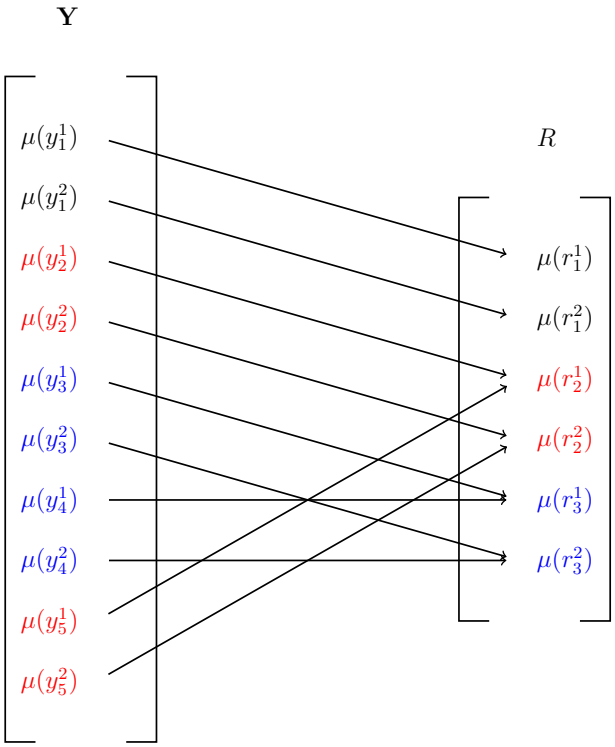


Figure 3.3: The figure shows an example for the mapping from Y to R where number of nodes is 5 and each node can take 2 labels. Assume that based on the global constraints information nodes 2 and 5 are constrained to take same label. Similarly nodes 3 and 4 are constrained together. Image shows that how the dimensionality reduction is occurred. This mapping is achieved through seeking a basis for null space of A accordingly.

[77] to reformulate the constrained minimisation problem in a lower dimensional space.

To this end, we define $Z \in \mathbb{R}^{mn\times(mn-e)}$ where columns of Z are a basis for null space of A as indicated in Eq. (3.13),

$$AZ = 0. \tag{3.13}$$

Then it can be introduced a matrix $R \in \mathbb{R}^{(mn-e)\times 1}$ that contains new set of indicator variables where $ZR = Y$. According to theory if Z^TQZ is positive definite, the equality constrained problem in Eq. (3.12) can be represented by an equivalent unconstrained problem as indicated in Eq. (3.14),

$$\text{minimise} \quad \frac{1}{2}R^T(Z^TQZ)R + (Z^Tc)^TR. \tag{3.14}$$

The equality constraints $AY = 0$ can be rewrite as $A(ZR) = 0$ by substituting for Y since $AZ = 0$ the equality constraint is always satisfied (implicitly) in the minimisation problem Eq. (3.14). This transformation has two benefits: First, the dimensionality of the variable matrix R in the reduced problem is decreased by value e compared to the dimensoinality of Y. This implies that a large num-

ber of constraints makes the optimisation problem easier to solve. Second, the optimisation problem is in an unconstrained form, which again makes it easier to solve.

As mentioned earlier, our main goal is to solve the MAP estimation problem in Eq. (3.10). We obtain an equivalent problem to this MAP estimation by adding the constraints Eq. (3.10c) and Eq. (3.10d) to the unconstrained problem (Eq. (3.14)) as follows:

$$\begin{aligned} \text{minimise} \quad & \frac{1}{2}R^T(Z^TQZ)R + (Z^Tc)^TR && \text{(3.15a)} \\ \text{s.t} \quad & FY = 1 && \text{(3.15b)} \\ & 0 \leq A_k \leq 1 \quad \forall k. && \text{(3.15c)} \end{aligned}$$

With the added constraints in the matrix notation and substituting for $Y = ZR$ we obtain:

$$\begin{aligned} \text{minimise} \quad & \frac{1}{2}R^T(Z^TQZ)R + (Z^Tc)^TR && \text{(3.16a)} \\ \text{s.t} \quad & (FZ)R = 1 && \text{(3.16b)} \\ & 0 \leq (ZR)_k \leq 1 \quad \forall k. && \text{(3.16c)} \end{aligned}$$

Algorithm 5: Globally Constrained CRF

Input: Unary potential array b, Edge potential matrix Q, Laser segment based constraints matrix A, number of laser based constraints e, number of nodes m, number of labels n

Output: Assignment of the nodes $\mathbf{y} = [y_1, .., y_m]$

$\mu_i(r_i^p)-$Indicator variables in the reduced problem:

`// Initialise`

$\mu_i(r_i^p) = \rho_i(r_i^p) / \sum_p \rho_i(r_i^p)$

`// Computation of the null space:`

$Z = null(A)$

`// Unary and pairwise coefficients in the reduced space`

$\rho_i(r_i^p) = (-Z^T c)_{n(i-1)+p}$

$\tau_{ij}(r_i^p, r_j^q) = -(Z^T Q Z)_{(n(i-1)+p),(n(j-1)+q)}$

`// Computing the dimensions of the reduced problem:`

$w = m - e/n$

`// Perform gradient descent`

repeat

  foreach $i \in \{1, \ldots, w\}$ **do**

    foreach $p \in \{1, \ldots, n\}$ **do**

      `// Compute the gradient at node` i

      $q_i(r_i^p) \leftarrow \rho_i(r_i^p) + 2 \sum_{i,j} \sum_q \tau_j(r_i^p, r_j^q) \mu_j^t(r_j^q)$

      $\mu_i^{t+1}(r_i) \leftarrow \frac{\mu_i^t(r_i^p) q_i(r_i^p)}{\sum_p \mu_i^t(r_i^p) q_i(r_i^p)}$

    end

  end

until *convergence*;

`// Array of the relaxed indicator variables in the reduced space`

$R = [\mu_1(r_1^1), .., \mu_1(r_1^n), .., \mu_w(r_w^1), .., \mu_w(r_w^n)]^T$

`// Extract final solution`

$Y = ZR$

$\mu_i(y_i^p) = A_{n(i-1)+p}$

foreach $i \in \{1, \ldots, m\}$ **do**

  foreach $p \in \{1, \ldots, n\}$ **do**

    $y_i \leftarrow p$ if $\mu(y_i^p) = 1$

  end

end

return y

The null space of A can have infinite number of solution matrices. However, we ensure that the selected matrix Z generates a mapping similar to the example shown in Figure 3.3. Depending on the simpler structure of A this is straightforward. This selection of Z makes the mapping from Y to R surjective. Z only contains 0 or 1. Each row contain single element that is equal to 1 and all the other elements are set to zero. The nonzero element locations are selected so that each variable in Y is equivalent to some variable in R. This means that variables in R also vary between 0 and 1 similar to variables in Y. Also when $\sum_p \mu_i(y_i^p) = 1$ and node i is mapped to node j in R then $\sum_p \mu_j(r_j^p) = 1$.
Reversing the transformation from Eq. (3.10) to Eq. (3.12) we can rewrite Eq. (3.14) using element wise notation as a maximisation:

$$\begin{aligned}
\text{maximise} \quad & \sum_i \sum_p \rho_i(r_i^p)\mu_i(r_i^p) \\
& + \sum_{i,j} \sum_{p,q} \tau_{ij}(r_i^p, r_j^q)\mu_i(r_i^p)\mu_j(r_j^q) && \text{(3.17a)} \\
& \sum_{p \in L} \mu_i(r_i^p) = 1 \quad \forall i && \text{(3.17b)} \\
\text{subject to} \quad & 0 \leq \mu_i(r_i^p) \leq 1, && \text{(3.17c)}
\end{aligned}$$

where r_i^p denotes, if label p has been assigned to random variable i. In this manner each random variable i is associated with n number of variables $[\mu_i(r_i^1), .., \mu_i(r_i^n)]$, where $R = [\mu_1(r_1^1), .., \mu_1(r_1^n), .., \mu_{m-e/n}(r_{m-e/n}^1), .., \mu_{m-e/n}(r_{m-e/n}^n)]^T$ with the unary potential $\rho_i(r_i^p) = (-Z^T c)_{n(i-1)+p}$ and the pairwise potential $\tau_{ij}(r_i^p, r_j^q) = -(Z^T Q Z)_{(n(i-1)+p),(n(j-1)+q)}$. Eq. (3.16b) can be formulated as Eq. (3.17b). The objective Eq. (3.17a) is a function of $\mu_i(r_i^p)$, it is indicated by the notation $J(\mu_i(r_i^p))$. We optimise Eq. (3.17) using gradient ascent which can be done efficiently as the gradient can be computed in closed form [120] as follows:

$$\begin{aligned}
q_i(r_i^p) &= \frac{\partial(J(\mu_i(r_i^p)))}{\partial \mu_i(r_i^p)} \\
&= \rho_i(r_i^p) + 2\sum_{i,j} \sum_q \tau_j(r_i^p, r_j^q)\mu_j(r_j^q),
\end{aligned} \tag{3.18}$$

$$\mu_i^{t+1}(r_i^p) = \frac{\mu_i^t(r_i^p) q_i(r_i^p)}{\sum_q \mu_i^t(r_i^q) q_i(r_i^q)}. \tag{3.19}$$

Zhang *et al* [120] conduct the optimisation process by implicitly maintaining the constraints Eq. (3.17b) and Eq. (3.17c) using a combination of fixed point iteration and gradient ascent as indicated in Eq. (3.19). If $-(Z^T Q Z)$ is negative definite, then gradient ascent guarantee to converge to a global maximum, oth-

erwise it may converge to a local maximum. It is possible to conduct convex approximation as described in Chapter 2.2.2 by modifying the reduced hessian matrix. However in our case original values yield striking solutions in practice without the convex approximation. Once the algorithm has converged we can extract the values of original indicator variables $\mu(y_i^q)$ and thus the MAP label assignments to the y_i variables. To this end we transform the solution for $\mu(r_i^p)$ obtained from Eq. (3.17) back into the form of Eq. (3.10) using $A = ZR$. A is a column vector whose entries correspond to the values of the $\mu(y_i^p)$. The optimal assignment to each node $i \in V$ is found by selecting the label $p \in L$ for which $\mu_i(y_i^p) = 1$ holds. Generally, $\mu_i(y_i^p)$ values converges to 0 or 1. Otherwise label is assigned to the class which has the largest $\mu_i(y_i^p)$ value.

This procedure is summarised in Algorithm 6. The required inputs are the values of the unary and the pairwise potentials of the reduced problem Eq. (3.17). $\mu_i(r_i^p)$ variables are initialised using local potentials ρ. Then the gradient (Eq. (3.18)) is computed and used to update the solution iteratively until convergence is achieved. Here $\mu_i^t(r_i^p)$ refers the value of $\mu_i(r_i^p)$ during the t^{th} iteration. Finally, the solution is extracted and returned.

3.4 Experiments

3.4.1 Experimental Set up and Feature Selection

In this section we present experimental evaluation of our proposed framework on the task of urban scene segmentation. We use the KITTI dataset [35] as it provides typical urban data. The dataset was captured by driving around the city of Karlsruhe, Germany. Importantly, the datasets contain both colour images and Velodyne point clouds. The image information is used to build the CRF model structure and potential functions while the Velodyne data is used to construct global constraint sets.

Type	Description	Dimensionality
Texture	RGB gradient magnitude histogram	$50 \times 3 = 150$
	RGB gradient orientation histogram	$50 \times 3 = 150$
Colour	RGB mean	3
	RGB std	3
	HSV histogram	$50 \times 3 = 150$
Location	Super pixel image coordinates	$25 \times 2 = 50$

Table 3.1: Features used for the unary potential of the CRF based on a discriminant analysis classifier applied to super pixels.

We start by extracting super pixels from the image using SLIC which forms an

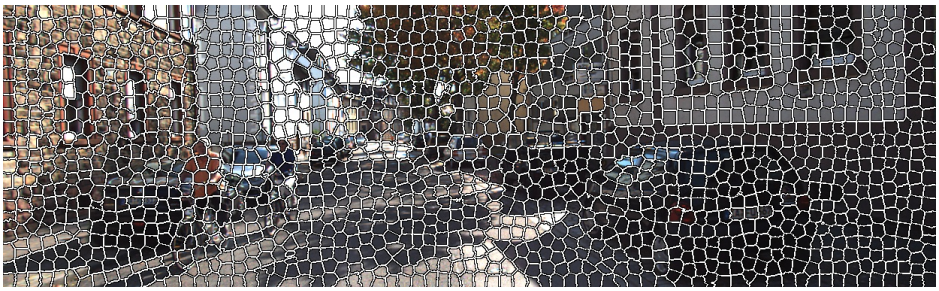
over segmentation of the original image. From each 375×1242 image we extract roughly 1600 super pixels, shown in Figure 3.9b. We have selected the average super pixel size to ensure the homogeneity of the over segmentation is preserved in a higher level. Each of these super pixels represents a node in our CRF and the goal is to label them with one of the seven different classes: vehicle, pedestrian & cyclist, buildings, ground, sky, vegetation, or unknown. Due to the low sample size of pedestrians and cyclists in the data set they are assigned to same class

The unary potentials ϕ_i are obtained from the posterior of a pseudo linear discriminant analysis classifier as described in the previous chapter. The classifier is trained on 100 manually labeled images (training set) from the *drive*_0048, *drive*_0091 and *drive*_0106 of KITTI dataset using colour, texture, and location features, shown in Table 3.1. The magnitude and the orientation of the gradient at each pixel are computed using the Piotr's Computer Vision Matlab Toolbox [82] for each colour channel separately. Standard practice is to use equal sized square cells to create gradient magnitude and gradient orientation histograms. Therefore, 2D bounding boxes are drawn centering at the centroid of each super pixel. By analysing the sizes of the generated superpixels, dimensions of the bounding box can be set to 15×15 since it approximate the size of the super pixels. Magnitude and orientation histograms are computed on each bounding box and normalised locally. Our final model is tested on 100 labeled images (validation set) from *drive*_0093, *drive*_0095, *drive*_0021, *drive*_0059 and *drive*_0043. Labeling was taken place with the assistance of the Image Annotation Tool [4].

CRF parameters

Weight values corresponding to the unary and pairwise terms of the CRF model can be learnt from the available ground truth data (Section 2.2.3). However in the experiment section we are evaluating the constrained CRF method on data recorded in diverse environments (areas with high pedestrian density, urban areas, rural areas, areas with different illumination conditions). Therefore it would not be feasible to obtain one global parameter set that can perform optimally in all different situations. To address this problem we develop a framework to adaptively learn these parameters for changing environments in the next chapter. During the experiments on this section we have given unit weight to all the unary and pairwise terms after a simple grid search. These suboptimal parameters add reasonable quality to the CRF classification to demonstrate the improvement achieved through the laser constraints addition.

(a) Original Image



(b) Superpixel Generation

(c) Ground Plane Removal

(d) Laser Point Cloud Clustering

Figure 3.4: Display of a typical scene from the KITTI dataset. (a) shows the raw image, (b) overlays the super pixels extracted from the image, (c) depicts the projection of the RANSAC plane in to the 2D image frame, (d) shows the global constraints extracted from the 3D laser point projected into the image space, each colour represents a single segment. Labels of the segments are unknown at this stage.

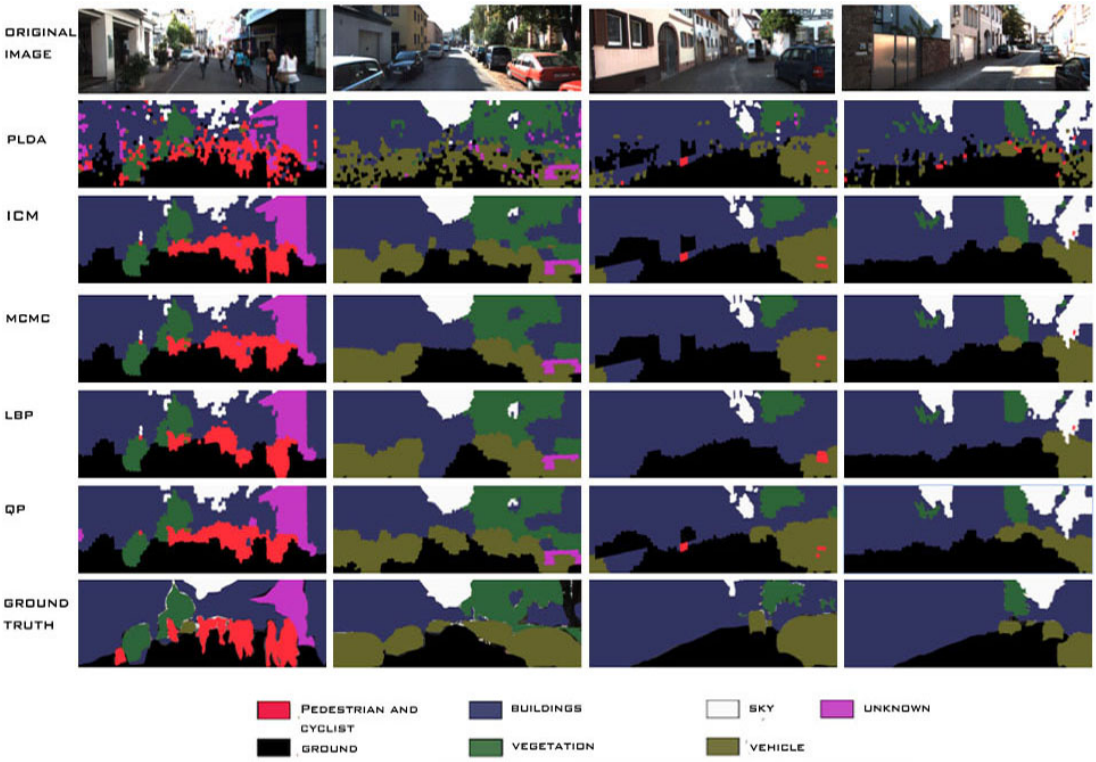


Figure 3.5: This figure shows exemplary scene segmentation results obtained on KITTI images. From top to bottom we have: each row containing original image and the labeling conducted by respective methods, discriminant analysis classifier,iterative conditional modes, markov chain monte carlo, loopy belief propagation, quadratic programming, and finally ground truth labels.

3.4.2 Scene parsing using visual information and laser based hard constraints

In the following we compute solution for the MAP problem denoted in Eq. (3.6) using approximated inference methods such as iterative conditional modes algorithm (ICM) [10], markov chain monte carlo (MCMC), loopy belief propagation (LBP). Here the MAP problem only consist of image based potentials. No laser information in used in this context. Subsequently, we compare the results from the state of the art methods with the results obtained from solving the QP problem (Eq. (3.9)) using the gradient based method described in Eq. (3.18) and Eq. (3.19). This showcase the quality of the segmentation results obtained by pure visual data. Thereafter, we introduce the global constraints obtained from the Velodyne points to demonstrate the enhancement of the segmentation quality due to the combination of multiple sensors inputs. Later we compare the results obtained from our contained model (Eq. (3.10)) with a graph-cut based HOP method [52].

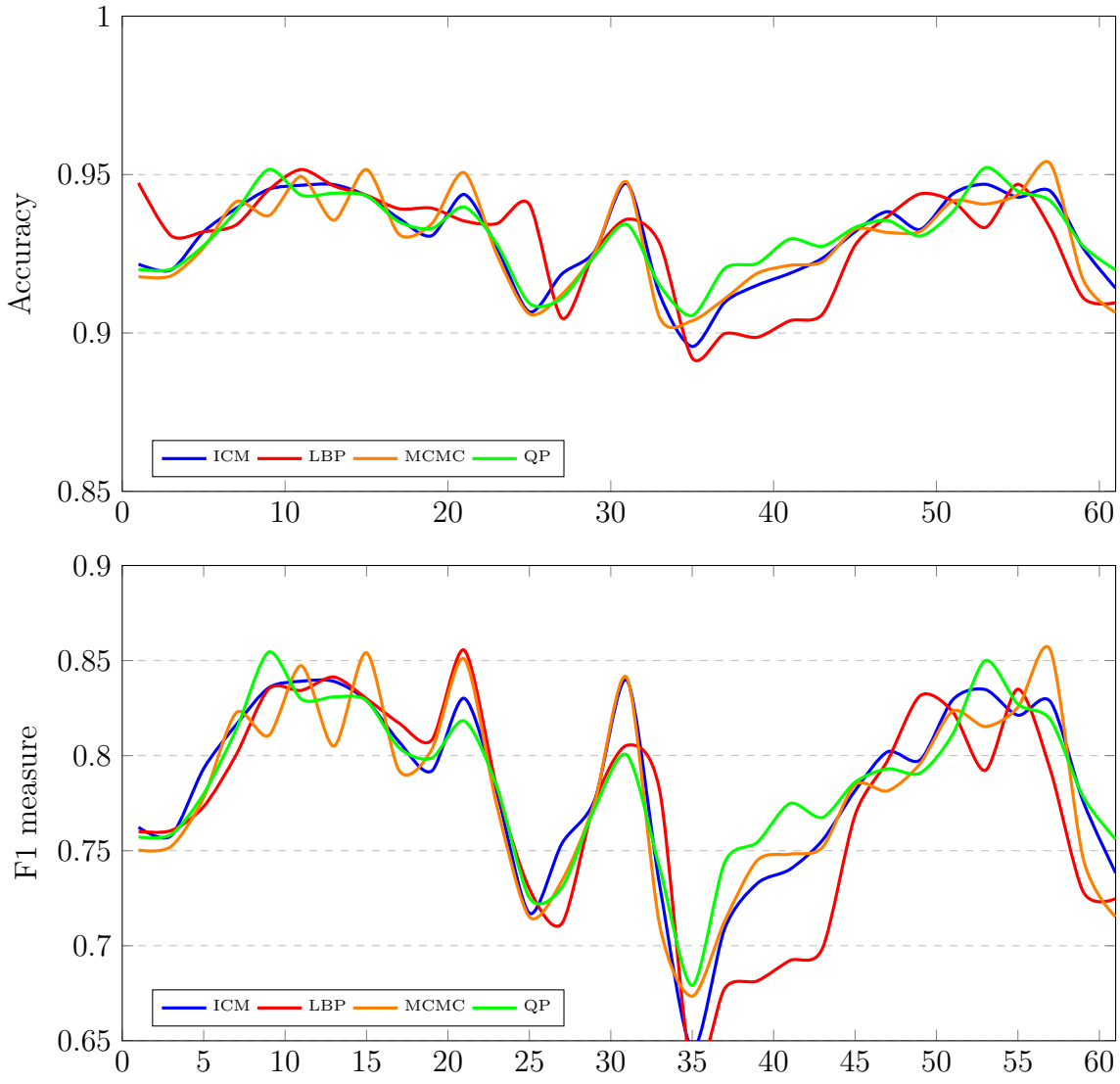


Figure 3.6: This figure shows scene labeling accuracy of *drive*_0093 (in KITTI dataset) obtained using different CRF inference algorithms.

Visual Information Only Segmentation

In Figure 3.5, we present results from five methods, (i) discriminant analysis classifier which provides the unary potentials of the CRF, (ii) loopy belief propagation using the UGM toolbox [94], (iii) ICM method using UGM toolbox, (iv) MCMC method using UGM toolbox, and (v) quadratic programming solution [120]. Exemplary results together with the original image and ground truth labels are shown in Figure 3.5. The first row shows the original colour images while the second row shows the most likely class of the discriminant analysis classifier which is used as the unary potentials of the CRF. As to be expected the classifier output is noisy and incorrect in several places. All the other CRF based solutions produce a much cleaner and consistent result compared with the raw classifier result. However, there are still segmentation errors present due to effects such

as shadowing and illumination changes. The quantitative evaluation results on the validation set, shown in Table 3.2, further demonstrates the improvements and also indicates that the QP based solution outperforms the LBP and MCMC methods. Where ICM method has a slightly better performance over all the other methods. Figure 3.6 indicates the accuracy and F1 measure plots for the validation set. All these analysis prove that the basis on which our method is built is capable of producing high quality segmentation results before any additional constraints are added, which will be evaluated next. It also conveys that the quality of the segmentation is accurate compared to the state of the art.

Laser Constrained Segmentation

In this section we explore the impact additional constraints, extracted from Velodyne data, have on segmentation results by comparing our method to a HOP based method by Kohli et al. [52]. The higher order potentials penalize label inconsistencies between nodes identified to be part of a single segment in the 3D data. Both methods use uniform weight parameters for the unary, pairwise, and higher order potentials, where applicable.

Some exemplary results are shown in Figure 3.7 with the original image shown on the far left, followed by the result of the HOP based method in the second column, then our method, and finally the hand labeled ground truth. Inspecting the results we can see that the HOP based method struggles to correctly identify distant objects, especially when cars or walls are involved. Additionally, the results our method obtains appear more uniform with less spurious classifications. This difference in behavior is explained by the way the additional 3D information is used. While our method enforces the constraints the HOP based method is allowed to violate them. The examples in Figure 3.8 show the benefit of using the hard constraints rather then soft constraints. The first two rows showcase this for a single wall while the third row shows the result of this in a scene populated by pedestrians. The first two columns show the original image and the segment extracted from the Velodyne data. Due to the visual appearance of these areas the classifier fails to pick the correct class in some parts of the 3D segment. The HOP based method fixes some classification errors, however, cannot fix every single one. In the case of the pedestrian scene the HOP method even misclassifies all pedestrians. Our method on the other hand is forced to assign a single class to the entire segment and as such the correct class is assigned even to the areas where the classifier makes mistakes.

For a quantitative analysis we compute average precision, recall, accuracy, and F1-score for the different methods on validation set. The F1 score is a combined matric of precision and recall. The highest quality of the classification is achieved

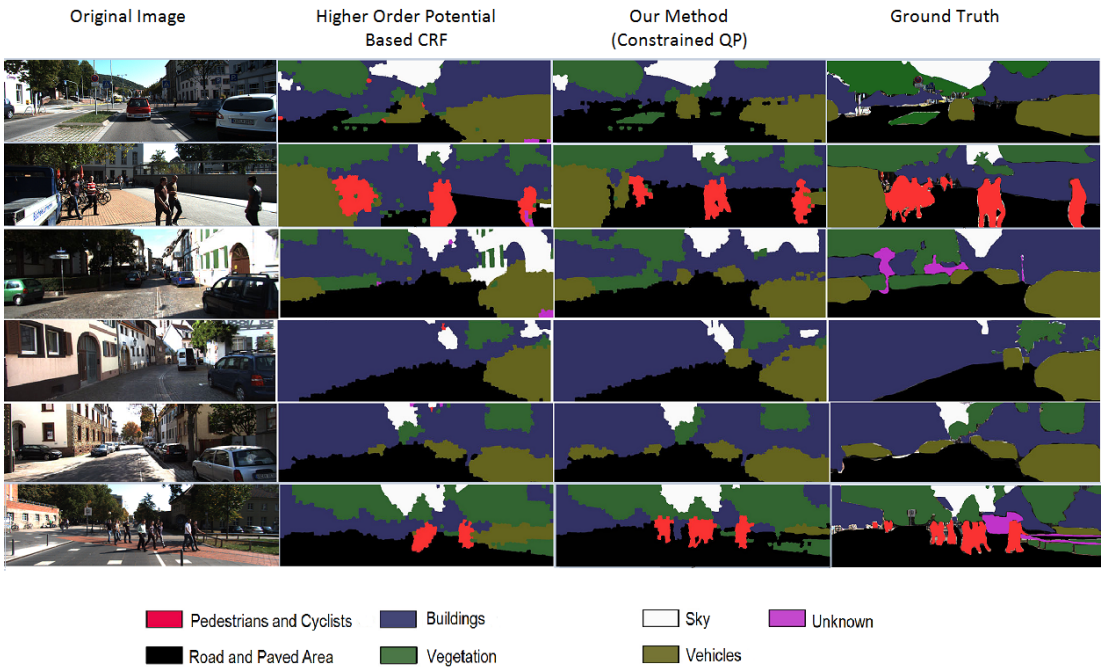


Figure 3.7: This figure shows results obtained on KITTI urban scenes using the HOP based method and our proposed method together with the original image and the hand labelled ground truth. We can see that our method (Constrained QP) performs better then the HOP based method at segmenting distant and objects cast in shadows.

when the F1 measure is 1 and poorest case is zero. Higher F1 score is an indication of both high precision and high recall. Since we operate in a multiclass setting, micro average F1 score is used to give an equal bias to all the classes which is indicated by $(\sum_{i=1}^{n} \mathrm{F1}_i)/n$ where $\mathrm{F1}_i$ denotes the F1 score for i^{th} class. As we can see in Table 3.2 the addition of global constraints in our method allows it to significantly outperform the other methods lacking this information and even the HOP method, using the same information, does not provide the same benefits. Improvement of the precision,accuracy and F1 measure in CQP compared to the HOP method is around 2%. Where recall has increased by 3%. Results show that adding constraints based on simple information about which areas belong to a single object allows the segmentation to be more accurate. This is good news, as this type of information is readily available in robotic systems. Looking at the performance of the individual classes in Table 3.3 we can see that "cyclist & pedestrian" class is the hardest one. This is explained by the fact that instances of this class occur infrequently and as such the classifier has a harder time at classifying them correctly. Furthermore, this class has the smallest appearance in the Velodyne data and as such will only be detected at close range. The other classes exhibit similar performance, which is not surprising, given that they occur frequently in the data and cover larger areas of the scene. Numerous colours and areas with different textures (glasses, wheels, and body have different texture)

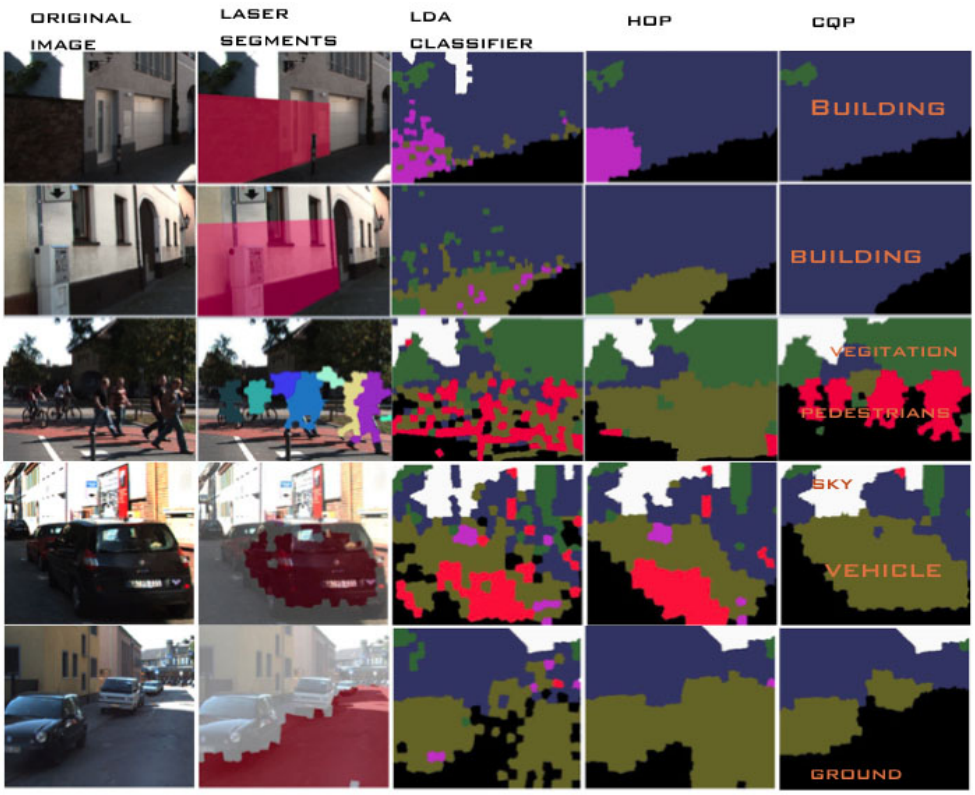
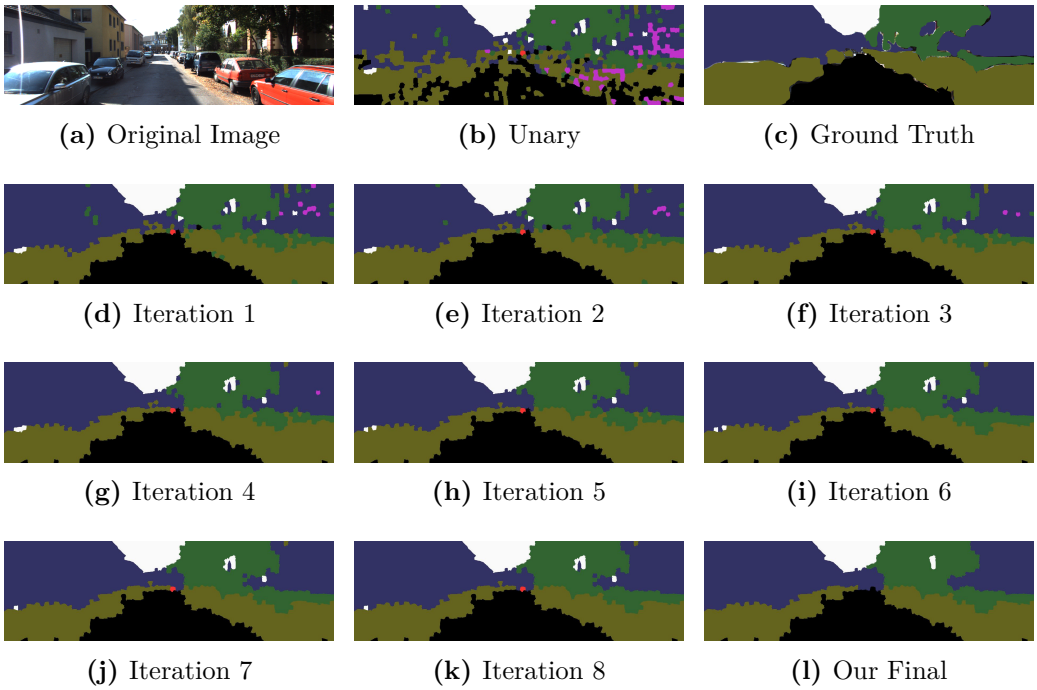


Figure 3.8: This figure shows examples from KITTI dataset to convey the benefit of enforcing hard constraints . The highlighted areas in the image show continuous 3D segments extracted from Velodyne data. The classifier output in these areas is noisy and wrong due to visual ambiguities. While the HOP based method fails to correct this our method succeeds in classifying the entire area correctly, as it is forced to assign a single class to each of the laser based segments.

associated with the vehicle class, cause difficulties to identify the superpixels belong to that class accurately. In the table, we can see that improvement of the accuracy and F1 measure from HOP to CQP, correspond to the vehicle class is around 4% and 8% respectively. This amount of dramatic improvement is obtained by the label consistency information enforced with the hard constraints. Apart from that, constraints also help to separate vehicles from walls and the ground.

Method	Average Precision	Average Recall	Average Accuracy	F1 Score
Discriminant Analysis Classifier	0.7027 ± 0.045	0.5127 ± 0.061	0.8826 ± 0.045	0.5927 ± 0.057
Iterative Conditional Modes	0.7671 ± 0.032	**0.8023** ± 0.048	**0.9185** ± 0.044	**0.7843** ± 0.061
Markov Chain Monte Carlo	0.7629 ± 0.038	0.7999 ± 0.041	0.9116 ± 0.024	0.7810 ± 0.062
Loopy Belief Propagation	0.7435 ± 0.051	0.7197 ± 0.081	0.9024 ± 0.053	0.7314 ± 0.067
Quadratic Programming Relaxation	**0.8001** ± 0.032	0.7645 ± 0.048	0.9150 ± 0.053	0.7818 ± 0.040
Higher Order Potentials	0.8319 ± 0.073	0.8143 ± 0.067	0.9278 ± 0.022	0.8230 ± 0.070
Constrained Quadratic Programming	**0.8549** ± 0.079	**0.8424** ± 0.078	**0.9507** ± 0.025	**0.8482** ± 0.076

Table 3.2: Quantitative evaluation of various segmentation methods for the selected drives of KITTI dataset. The first row shows the results from the unary classifier and the next four rows represent the results from state of the art CRF inference algorithms using only image based information. The last two rows show the results for methods using additional information obtained from 3D Velodyne scans. The HOP method incorporates this information as an additional potential, while our method (Constrained Quadratic Programming) enforces the validity of this additional information as constraints. We can see that the addition of the 3D information improves the performance compared to the image only based solutions. However, actively enforcing the constraints allows our method to outperform the HOP based method.



(a) Original Image **(b)** Unary **(c)** Ground Truth

(d) Iteration 1 **(e)** Iteration 2 **(f)** Iteration 3

(g) Iteration 4 **(h)** Iteration 5 **(i)** Iteration 6

(j) Iteration 7 **(k)** Iteration 8 **(l)** Our Final

Figure 3.9: The image flow depicts intermediate solutions of the gradient-based optimisation of the constrained CRF model. Segmented images up to 8^{th} iteration are shown consecutively. Final solution is shown in the last row (25^{th} iteration).

Quality Measure	Average Precision		Average Recall		Average Accuracy		F1 Score	
Method	HOP	CQP	HOP	CQP	HOP	CQP	HOP	CQP
Cyclists & Pedestrians	0.7689	**0.7700**	0.5134	**0.5334**	0.9670	**0.9772**	0.6153	**0.6302**
Ground	0.8431	**0.8554**	0.9747	**0.9775**	0.9569	**0.9789**	0.9032	**0.9124**
Vegetation	0.8284	**0.8440**	0.5161	**0.5359**	0.9468	**0.9473**	0.5931	**0.6192**
Buildings	**0.8431**	0.8420	0.8448	**0.8838**	0.8652	**0.9103**	0.8382	**0.8568**
Sky	0.7519	**0.7877**	**0.7690**	0.7564	0.9723	**0.9780**	0.7265	**0.7461**
Vehicles	0.8485	**0.9089**	0.7101	**0.8058**	0.9031	**0.9413**	0.7614	**0.8543**

Table 3.3: Quantitative evaluation of the performance on a per class for HOP method and our method CQP(For the KITTI dataset). All the major 6 classes excluding the unknown class are separately evaluated for the quality of segmentation.

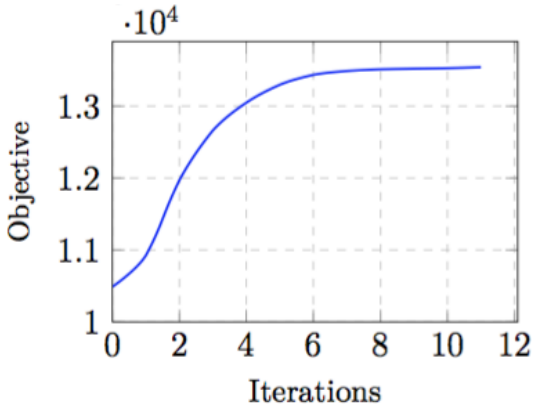


Figure 3.10: The graph denotes the change of the objective ($J(\mu_i(r_i^p))$) value of the input image shown in the Figure 3.9 corresponding to iterations of the gradient based method.

The inference process of the proposed method is shown in Figure 3.9. It is evident that the algorithm fixes most of the issues with the unary classifier and enforces label consistency information on the segmentation at the end of the iteration 1. During next iterations it eliminates the noise in the labeling and the posterior probabilities for the random variables correspond to the labels are converged to zeros and ones. After the 8^{th} iteration we can obtain a fairly well labeled image, but it takes approximately 25 iterations for random variables to completely stabilise. The graph in the Figure 3.10 indicate the variation of magnitude of the objective which is equivalent to the MAP of the CRF model. It can be seen that the at the 8^{th} iteration it reaches to a considerably maximum level and increment of the objective value after this point is trivial.

The performance of both constrained QP and HOP can be improved by training the weight parameters of the potential functions, which encodes knowledge about class relationships and object co-occurrence statistics. The advantage of our method is, that it only requires unary and pairwise potentials while HOP has

additional higher order potentials, which can be harder and time consuming to learn. This makes the proposed method easier to fine tune as there are fewer parameters involved.

Ford Campus Vision and Lidar Dataset

For further validation of the model, we tested the method on Ford campus vision and lidar dataset [79] that has cameras with spherical vision. This dataset is gathered by an autonomous ground vehicle testbed, developed using a customised Ford $F-250$ pickup truck. The vehicle consist of multiple sensors including a Point Grey Ladybug3 omnidirectional camera system [102] and a Velodyne $HDL-64E$ lidar [37]. The truck has drawn around the Ford Research campus and downtown Dearborn, Michigan to collect the dataset. Visual data has been captured using only half resolution (1600×600) of the full capacity of the camera. During our experiments we utilised the time-registered data in dataset 1 correspond to ladybug camera and Velodyne. The ladybug camera usually creates spherically distorted images. Transformations provided by the camera manufacturer are used to recover the distorted points by ultimately constructing flat images. However due to the nature of a ladybug camera, usually the proportions of the objects are not realistic. Especially the area covered by the classes such as vehicles, trees, buildings and pedestrians are small. Since these objects have a very low sample size, it was hard to distinguish between them through classification, to the contrary ground and sky has an enormous amount of training data. Also, these classes suffer from the low resolution which deteriorates the capability to extract rich texture features. As we know, texture is highly important in training an accurate classifier especially for the critical classes such as vehicles and pedestrians. All these reasons make it challenging to train an accurate local classifier for this dataset. Meanwhile, poorly distributed classes also lack contextual information. However, we can see that constraint addition still able to do a considerable enhancement of the quality of segmentation. Hence it is evident that the a priori information addition can be highly useful even in the situations with a poorly trained local classifier.

Method	Precision	Recall	Accuracy	F1 Measure
pLDA classifier	0.6872	0.6645	0.8432	0.6757
QP	0.7462	0.7320	0.9011	0.7390
CQP	**0.7651**	**0.7537**	**0.9251**	**0.7593**

Table 3.4: Quantitative evaluation on the Ford vision and lidar dataset. The values depict the overall precision, recall, accuracy and $F1$ measure for QP and CQP methods. The result shows that CQP method is distinguishably improved over QP method.

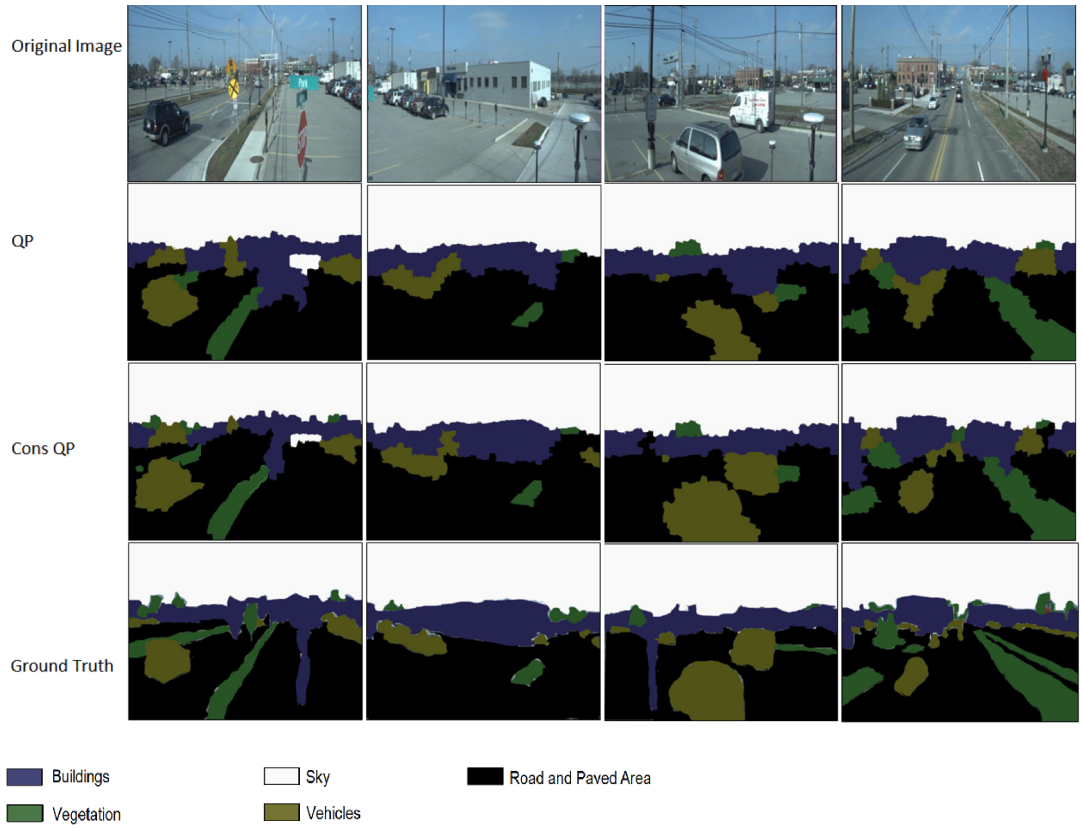


Figure 3.11: Example images from Ford vision and lidar dataset. This shows that the CQP does a considerable improvement to the pure image based QP solution.

Runtime Comparison

We start by comparing the runtime required to solve the constrained quadratic program of Eq. (3.10) directly using NLOPT BOBYQA [48] compared to our proposed framework. As we can see in Figure 3.12, directly solving the quadratic program is not feasible for problems of interesting size. On the other hand, our method scales very favourably with the problem size. Additionally, while typically increasing the number of constraints makes the problem harder and thus slower to solve, our method becomes faster with more constraints. This is caused by the fact that constraints reduce the size of the actual problem we solve. This means that adding more domain knowledge allows us to improve the quality of the result as well as speed up the computation.

A typical CRF derived from the KITTI images used in the experiments consists of 1600 nodes, each of which can have one of seven different labels, which means we have on the order of 11 200 random variables. Solving this CRF using the quadratic program formulation Eq. (3.9) (with no laser based constraints) takes around 2 s while the belief propagation based solution takes 0.5 s. Including the constraints we can reduce the number of nodes to around 400 which results in a much smaller number of variables, around 2800. Solving this inference problem using gradient based method takes around 0.07 s. All computations were

performed on an Intel Core i5 3.20 GHz processor with MATLAB and C++ implementations of the algorithms. Besides the reduction of the number of variables involved our method also requires fewer iterations to converge, around 25, compared to 70 for the purely image based quadratic program. These two advantages, reduction in number of variables and faster convergence gives our method a significant computational advantage.

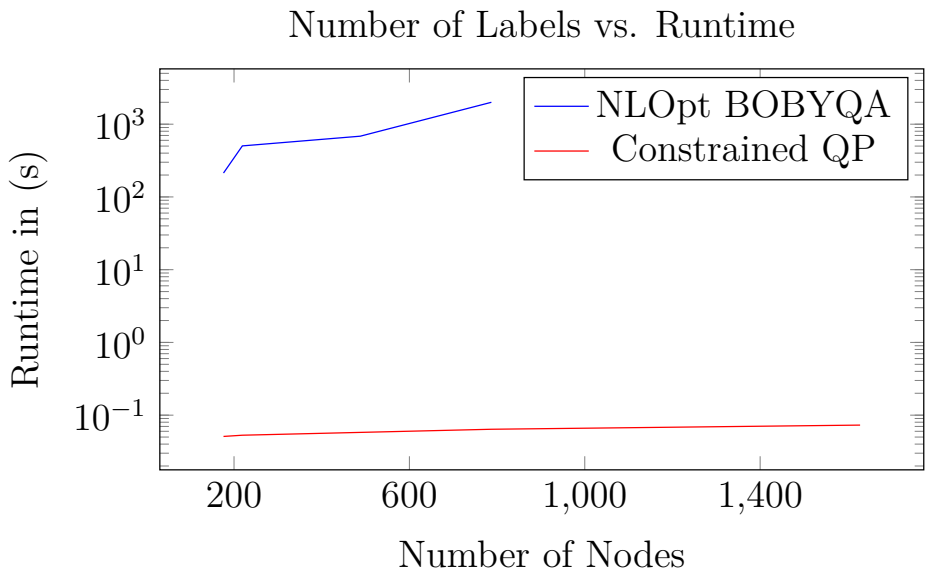


Figure 3.12: The plot shows the time (log scale) needed to find a solution as a function of the number of nodes in the CRF. NLopt BOBYQA solving the problem directly scales very poorly while our proposed method is scaling much more favourably.

3.5 Summary

In this chapter we presented a novel image segmentation method based on a conditional random field with additional global constraints which encode *a priori* information about groups of nodes having the same label obtained from a secondary sensor. This CRF is formulated as a relaxed quadratic program whose MAP solution is found using gradient descent based optimisation. We evaluate our method on data from the KITTI and Ford datasets. Each image is pre-processed into super pixels which provide the unary and pairwise potentials of the CRF. The global constraints on sets of super pixels are obtained from Velodyne data. The results show that the addition of these hard constraints significantly improves on the solution obtained without constraints. Runtime comparisons show how black box solvers do not scale for this problem and how our formulation exploits constraints in a way which simplifies the problem. Finally, the proposed method is general and capable of encoding other forms of constraints, such as relative positioning of classes with respect to each other.

Chapter 4

Online Learning for Scene Segmentation With Laser-Constrained CRFs

4.1 Introduction

In the previous chapter a semantic segmentation model was proposed that can efficiently combine the information from multiple sensors. That model processes each image independently of all the other images. In this chapter, the goal is to extend the model to a real time scene segmentation framework so that it can be used in autonomous driving. The content of the chapter is accepted for publishing in ICRA [1].

Formerly, we used CRF models to perform scene labelling as they can integrate local classifiers and smoothness of labels depending on the context. However, efficient combination of this information is challenging, especially in the context of autonomous driving, where the robot's environment is continuously changing. The ability to efficiently combine features is more important in dynamic than in static cases. Adaptively and continually learning the CRF parameters is therefore coupled with the changes in the data distribution. However, CRF parameter learning can be painstaking due to complex correlations between variables , cost involved with inference and gradient computations. Stochastic Gradient Descent (SGD) algorithms have become an appealing alternative in online learning settings since they use a gradient calculated at a single point or small subset of the data, instead of the full gradient.

Training CRFs is commonly conducted with fully labelled images. In some cases partially labelled images are used to train CRFs since it also helps to overcome issues such as parameter overfitting and over-estimation. However in autonomous navigation learning is done continuously as new data encountered with no ground truth available. In this scenario parameter learning is not a trivial problem. Therefore, it is important to do online parameter-learning by eliminating the use of manually labelled images. In this setup, we are searching for the best set of

[1] Charika De Alvis, Lionel Ott, Fabio Ramos. Online Learning for Scene Segmentation With Laser-Constrained CRFs. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2017

parameters for the CRF model to accurately do the image classification of the current images from a continuous sequence. In other words, we are interested in finding the best local estimate rather than a global optimum for all past data. We use information from the camera and laser sensors as a reference to compute the loss function. By minimising this loss, we obtain a best set of parameters for the current data distribution (the true input data distribution is non-stationary). We use a SGD-based approach for the optimisation because it facilitates getting the best parameter set based on recent data. We demonstrate the performance of the online parameter learning on the CQP model proposed in [5]. We use real world street scene data from the KITTI dataset [35]. To summarise the main contributions of the chapter are:

1. Development of a model to learn CRF parameters by eliminating the need for ground truth labels. The model derives the reference labels for learning by pre-processing the sensor information.

2. A stochastic gradient based method to continuously update the parameters while making the method robust to non-stationary data observed during long trajectories.

3. Evaluation of the methods on real urban datasets.

4.2 Related Work

A number of approaches have been proposed to efficiently estimate the parameters of CRF models. Verbeek *et al.* [111] introduce a method for learning CRFs from datasets with unlabelled nodes by marginalising out the unknown labels and by maximising the log-likelihood of the known nodes by gradient ascent. Tsuboi *et al.* [109] present a similar work for training CRF using partially annotated corpora for natural language processing. In [44] the authors develop a hybrid model for exploiting incompletely labelled data that combines a generative topic model for image appearance with discriminative label prediction. These methods target only offline learning of CRF parameters.

For large scale learning problems it is imperative that the algorithms scale well. Due to the high cost associated with CRF training models, SGD methods have replaced batch learning in online settings. The momentum method [83] is commonly used to help SGD to accelerate in relevant directions and dampen oscillations. Selecting an ideal learning rate for SGD can be challenging. ADAGRAD [25] is a first-order method that can efficiently adapt learning rates. This method exploit separate learning rates for each dimension where large gradients result in smaller

learning rates and the vice versa. Similar to second order methods, ADAGRAD can balance the progress in each dimension over-time. This method continuously decreases the learning rates based on the accumulated gradients over time, which also provide the effects similar to annealing. ADADELTA [118] is a recent improvement which reduces the sensitivity to the hyperparameter selection. This method uses a learning rate similar to ADAGRAD, but accumulation of the gradients is not done from the beginning of the sequence instead it uses a moving window which only considers recent set of gradients. This formulation prevents the continual decline of the learning rates allowing to grow the learning rate when necessary. SGD has a slower convergence rate compared to the batch gradient descent methods. To overcome this issue Schmidt *et al.* [95] apply stochastic average gradient algorithm which combines the characteristics of deterministic and stochastic models to train CRFs. They show that this algorithm converges with a smaller number of iteration than SGD. However, despite this advantage it is difficult to apply SAG algorithm to models with sophisticated features and a large number of labels when the number of training examples is small.

In our research, we are interested in online learning for autonomous navigation. Schraudolph *et al.* [96] propose a scalable, stochastic quasi-Newton method for online convex optimisation. Schaul *et al.* [92] propose a method to automatically tune learning rates to minimise the expected error at (any)time t. The framework performs well in non-convex problems. The method is based on local gradient variations across samples. In this framework, learning rates have the freedom to grow or decline to make the model robust in non-stationary problems. Our framework also uses a similar technique to change learning rates to adapt to changing data distributions but focus on exploiting the robot sensor data.

Fathi *et al.* [28] propose an incremental self-training algorithm for object segmentation in a video, where they iteratively label the least uncertain frame and update similarity metrics. This self training video segmentation provide higher accuracy for foreground identification problems. The approach of [72] consists of a self-learning algorithm for ground detection. The system automatically learns to identify salient features which correspond to the ground class. New observations are labelled by outlier rejection using the past data. Vijayanarasimhan *et al.* [112] demonstrate a method to reduce the human effort in video annotation. They choose k frames for manual labelling to ensure that automatic pixel level label propagation occurs with minimal expected error. They minimise the effort required for labelling and correcting propagation errors. All these methods require some amount of labelling of the data which is difficult to obtain in a real time navigation task. Our framework omits the need to use labelled data in the learning process. Instead it attempts to exploit existing sensor information to

constraint the problem and adjust the parameters accordingly.

4.3 An Adaptive Model to Parse image Sequences

4.3.1 Overview

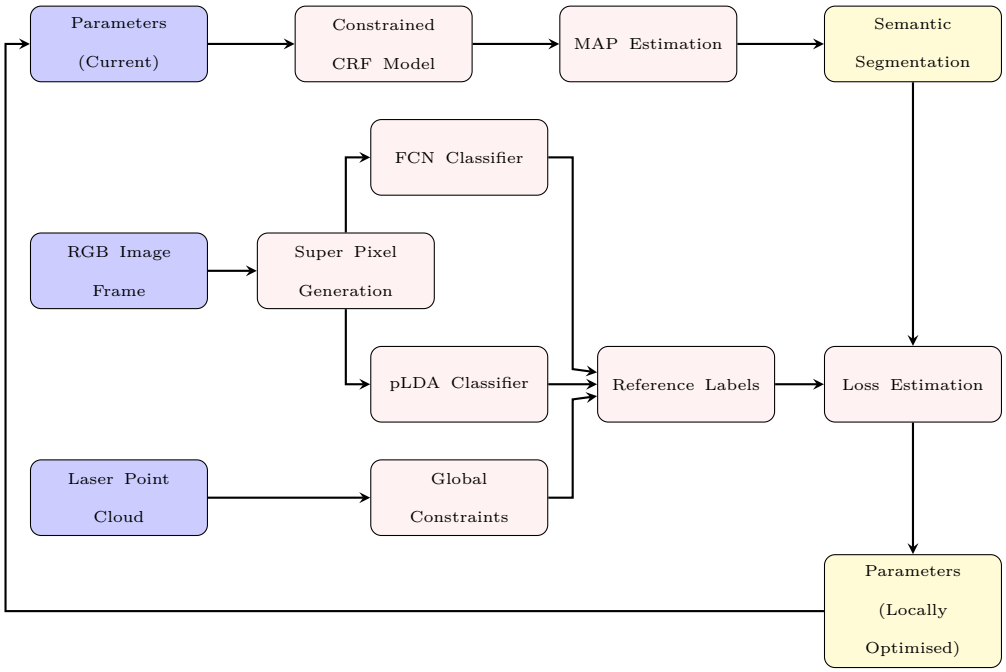


Figure 4.1: This flow diagram summarizes the main parts of the framework. As indicated processing and feature extraction of visual and laser data is conducted independently of each other. Later this sensor information has been used to derive reference labels for the training images. These reference labels are used to minimise the mislabeling loss while optimising the model parameters.

Figure 4.1 shows the pipeline of our learning framework based on CQP. The first stage of the framework involves pre-processing the data and extracting visual and depth based features.

1. Superpixel generation and training a pLDA classifier (described in Chapter 3.3.3) for super pixels and obtaining posterior probability for possible labelling.

2. Obtaining label prediction for foreground objects using a pre-trained Fully Convolutional Network based (FCN)[65] classifier.

3. Extracting the laser based segments (described in Chapter 3.3.4).

As mentioned in Chapter 3, we initialise by setting all the CQP parameters to 1. Firstly, using MAP estimation of the CQP model we obtain the best labelling for the superpixels. Secondly, we combine all the information extracted from

sensors to generate reference labels for the image superpixels. Thirdly the loss is computed depending on the dissimilarity between the current set of optimal labels and the reference labels. Stochastic gradient decent is used to update the parameter set by optimising the loss. The learning process continues in this order in an online setting while providing locally optimum parameters for the CQP.

Convolutional Neural Networks for Scene Classification

Convolutional Neural Networks [49] are revolutionising the field of image classification. An image can be directly fed into CNN architectures, which permits to encode many important properties in the classification process. CNN is encompassed with single or multiple convolutional layers followed by fully connected layers to the end. There are also intermediate pooling layers. Layers of CNN operate in three dimensions: width, height and depth. The main advantage of CNN is that it has considerably less number of parameters than a conventional neural network with the same number of hidden layers. To illustrate, the input volume for a CNN is *image width* x *image height* x *3* (for RGB colors) and usually its output volume tend to be *1* x *1* x *number of classes*. This is a single vector that gives the class prediction for the image. CNN has been extended for prediction in a pixel level. Long *et al.* proposed fully convolutional networks (FCN) [65] that can be trained to do pixel to pixel segmentation. FCNs are designed by replacing last fully connected layers in CNN, by convolutional layers that can classify each image pixel. Consequently, the size of the output volume becomes *image width* x *image height* x *number of classes*. Therefore it can provide score for the class of each pixel. We have used a FCN classifier along with the pLDA classifier to improve the local classification.

Laser Constraints

The constraints required by the CQP model as well as generating the reference labels, are extracted from the Velodyne scans. Section 3.3.4 describes the process of laser based constraints extraction. Then constraints are mapped in to the image frame and associated with superpixels. Subsequently it can be identified groups of superpixels that have label consistency within a group.

4.3.2 CRF Based Scene Segmentation Model

The main objective of this chapter is to improve the accuracy of the CQP method proposed in the previous chapter by adaptive learning. CQP conducts scene segmentation by enforcing a set of global constraints during the optimisation which makes it more computationally efficient while providing capacity for performing

in real time. Furthermore, CQP achieves high accuracy in scene segmentation by utilising only unary and pairwise potential terms. This is attractive because parameter learning of pairwise CRFs is a extensively studied area.

CRF Model Building

Refer to Section 3.3.5 for the definition of the CRF model. In this stage unary potentials used in the CQP model are derived by combining two local classifiers. The pseudo linear discriminant analysis classifier [71] trained in our previous work (3.3.5) is combined with a fully convolutional net based classifier.
pLDA classifier prediction on a node labeling is denoted by $\phi_i(y_i^p)$. The pre-trained FCN classifier has a higher accuracy in identifying classes such as vehicles and pedestrians of the KITTI dataset. Our goal is to segment the images into the following seven classes: pedestrians and cyclists, ground, vegetation, buildings, sky, vehicles, and unknown. Pedestrians and vehicles classes are associated with label number 1 and 6 respectively. If the FCN classifier recognised that a superpixel belong to class 1 or 6 then $f(y_i^p) = 1$ where $p \in [1, 6]$. The set $\mathbf{a} = [k|f(y_k^1) = 1]$ denotes the nodes which are recognised as pedestrians by the FCN classifier. Similarly for the vehicle class, $\mathbf{b} = [j|f(y_j^6) = 1]$.

Now we introduce the formulation of unary potentials $\psi_i(y_i^p)$ for the CQP model by averaging pLDA and FCN classifier outputs. (Note: Averaging is done only for the nodes which are recognised as pedestrians or vehicles by FCN. Rest of the nodes are assigned with the values from pLDA classifier alone),

$$\psi_k(y_k^1) = 0.5(\phi_k(y_k^1) + 1) \quad k \in \mathbf{a}, \tag{4.1a}$$

$$\psi_k(y_k^p) = 0.5\phi_k(y_k^p) \quad k \in \mathbf{a} \text{ and } p \in [2, 3, 4, 5, 6, 7], \tag{4.1b}$$

$$\psi_j(y_j^6) = 0.5(\phi_j(y_j^6) + 1) \quad j \in \mathbf{b}, \tag{4.1c}$$

$$\psi_j(y_j^p) = 0.5\phi_j(y_j^6) \quad j \in \mathbf{b} \text{ and } p \in [1, 2, 3, 4, 5, 7], \tag{4.1d}$$

$$\psi_j(y_j^p) = \phi_j(y_j^p) \quad j \in V \quad j \notin \mathbf{a}, \mathbf{b} \quad p \in L. \tag{4.1e}$$

The selection of the pairwise potential matrix ψ for this setting is given below:

$$\psi_{i,j} = \begin{cases} 1 & \text{if } i = j \\ 0.01 & \text{otherwise} \end{cases}. \tag{4.2}$$

This model is chosen to encourage the neighbouring super pixels to take same labels.

Constrained Quadratic Programming

The labelling for $\mathbf{y}$ is attained with a maximum a posteriori (MAP) estimation of the conditional log-likelihood $log(P(\mathbf{y}|S))$ To solve the inference problem efficiently Zhang *et al* [120] proposed the quadratic programming relaxation which is introduced in Eq. (3.9). To improve the accuracy of label assignment we added global constraints over the optimisation problem which is denoted in Eq. (3.10) (CQP model). This relaxation includes the parameters corresponding the unary and pairwise terms.

Previously, CQP parameter values linked to all the unary and pairwise potentials were set to 1. In this chapter we are interested in an adaptive model which has the ability to update its parameters according to the context. The parameter set Θ contains 21 CRF parameters, of those 14 are associated with pairwise potentials while the remaining seven are linked to unary potentials. All nodes in the CRF use the same set of parameters,

$$\theta_{pq}^{pair} = \begin{cases} \theta_p^{\text{ondiag}} & \text{if } p = q \ p, q \in L \\ \theta_p^{\text{offdiag}} & \text{otherwise} \end{cases}, \tag{4.3a}$$

$$\Theta = [\theta_1^{\text{unary}}, ...\theta_n^{\text{unary}}, \theta_1^{\text{ondiag}}, ...\theta_n^{\text{ondiag}}, \theta_1^{\text{offdiag}}, ...\theta_n^{\text{offdiag}}]. \tag{4.3b}$$

We modify the CQP problem in Eq. (3.10) by adding the parameter variables which results in the following optimisation problem:

$$\begin{aligned} \text{maximise} \quad & \sum_{i \in S} \sum_{p \in L} \theta_p^{unary} \psi_i(y_i^p) \mu_i(y_i^p) \\ & + \sum_{\substack{i \in S \\ j \in N(i)}} \sum_{p,q \in L} \theta_{pq}^{pair} \psi_{ij}(y_i^p, y_j^q) \mu_i(y_i^p) \mu_j(y_j^q) && \text{(4.4a)} \\ \text{subject to} \quad & \sum_{p \in L} \mu_i(y_i^p) = 1 \quad \forall i && \text{(4.4b)} \\ & \sum_{i,j \in \mathbb{C}_k} \sum_{p \in L} \mu_i(y_i^p) - \mu_j(y_j^p) = 0 \quad \forall \mathbb{C}_k \in \mathbb{C} && \text{(4.4c)} \\ & 0 \leq \mu_i(y_i^p) \leq 1 \quad \forall i, p. && \text{(4.4d)} \end{aligned}$$

Here $\mathbb{C}$ is the set of laser segments. y_i^p encodes if node i has been assigned label p and the label assignment to each node is represented by the indicator function $\mu_i(y_i^p)$. Chapter 3.3.6 describes efficient means of solving the CQP problem using a dimensionality reduction technique and gradient ascent based algorithm. When the gradient ascent convergences $\mu_i(y_i^p)$ provides the most probable label assignment to the super pixels.

4.3.3 Online Learning

In this section we present a framework to optimise the parameters Θ of the unary and pairwise potentials of the CQP model in an online fashion. We optimise the loss function, detailed next, using stochastic gradient descent (SGD) which allows for fast and continue update of the parameters.

Loss Function

Our goal is to minimise the difference between the predicted probability of the label assignment $\mathbf{r}$ to nodes in G against the reference labels $\mathbf{z}$ extracted in a self-supervised manner from image and laser data by selecting the optimal CRF parameters Θ, i.e.:

$$\Theta^* = \arg\min_{\Theta} l(\mathbf{z}, \mathbb{C}, \mathbf{r}). \tag{4.5}$$

where Θ^* is the set of optimal parameters we wish to find and l is the loss function we need to optimise. $\mathbf{r}$ is the matrix that consist of the label prediction from the CQP model where $\mathbf{r}_{kp} = \mu_k(y_k^P) \quad k \in V$ and $p \in L$. Ideally, we would compare the predicted result to ground truth labels, as typically done in parameter learning. However, as we operate in the online setting we do not have access to such ground truth labels for the newly observed data. Therefore we generate labels for regions of high confidence on the image based on laser scanner data, fully convolutional net (FCN) [65] classifier outputs, and pLDA classifier [71] results.

Our loss function is comprised of several components which are introduced next. The classifier and the laser constraints contain information about recognising classes. Some classes are easier to recognise using the FCN classifier and vice versa. Hence in order to get accurate reference labels it is important to extract maximum amount of information correspond to each class. The classes are associated with a label index. 1- Pedestrians and Cyclists, 2 - Ground , 3 - Vegetation, 4- Buildings, 5- Sky , 6 - Vehicles and 7- Unknown. The loss function consist of 3 components.

First component l_{agree} provides a measure of deviation from reference labels. Here we consider the super pixels that we can predict their labels with a higher confidence and compare with the labels predicted from the CQP model . Then

loss is assigned proportional to the dissimilarity,

$$l_{\text{agree}} = \sum_{S_j \in S_{\text{agree}}} \sum_{k \in S_j} \lambda_j ||\mathbf{r}_k - \mathbf{z}_k||^2 \tag{4.6a}$$

$$S_{agree} = [S_1, S_2, S_3, S_4, S_5, S_6] \tag{4.6b}$$

$$S_1 = \{k | f(y_k^1) = 1 \text{ and } v_k = 1 \text{ and } k \notin \text{ Ransac Plane } \} \tag{4.6c}$$

$$S_2 = \{k | \phi_k(y_k^2) = 1 \text{ and } k \in \text{ Ransac Plane } \} \tag{4.6d}$$

$$S_3 = \{k | \phi_k(y_k^3) = 1\} \tag{4.6e}$$

$$S_4 = \{k | \phi_k(y_k^4) = 1 \text{ and } v_k = 1 \text{ and } k \notin \text{ Ransac Plane } \text{ and } f(y_k^4) = 0\} \tag{4.6f}$$

$$S_5 = \{k | \phi_k(y_k^5) = 1 \text{ and } v_k = 0 \text{ and } f(y_k^5) = 0\} \tag{4.6g}$$

$$S_6 = \{k | f(y_k^6) = 1 \text{ and } v_k = 1 \text{ and } k \notin \text{ Ransac Plane } \} \tag{4.6h}$$

$$\mathbf{r}_k = [\mu_k(y_k^p)]_{1 \times n} \tag{4.6i}$$

$$\mathbf{z}_{kp} = \begin{cases} 1 & \text{if } k \in S_p \\ 0 & \text{otherwise} \end{cases} \tag{4.6j}$$

$$\mathbf{z}_k = [\mathbf{z}_{kp}]_{1 \times n}. \tag{4.6k}$$

Here v_i is a binary variable to indicate if there are any 3D laser points are mapped on to the superpixel i. As we can see each frame we process will have a varying number of reliable labels at our disposal since set S_{agree} is changing from frame to frame. S_j denotes the set of nodes where we are confident that true label should be i based on the self supervised labeling process. $\mathbf{z}_{kp} = 1$ if only $k \in S_p$ where $p \in L$ in all the other cases $\mathbf{z}_{kp}$ is set to zero, i.e. S_1 contains the super pixels that are labelled as pedestrians based on the sensor information. We consider 3 factors in finding superpixels belong to pedestrian class.

- The FCN classifier has a higher accuracy for this class therefore FCN classifier should predict the superpixel as a pedestrian
- The super pixel should contain projected laser points on it. This implies that this super pixel can not be in the sky or in a very high level from the ground and
- The super pixel can not belong to the ground plane.

If a certain superpixel undergo these requirement it can be labelled as a pedestrian. Similar rules are used in finding the super pixel sets for the other classes.

The second component of the loss function l_{differ} is used in cases where we have knowledge that a certain label assignment is not possible, i.e. a super pixel that is observed by the laser cannot be sky. Here we add a loss if the CQP predictions

assign a label to a less probable class (based on the sensor based knowledge),

$$l_{\text{differ}} = \sum_{\mathcal{S}_j \in S_{\text{differ}}} \sum_{i \in \mathcal{S}_j} \alpha_j ||\mathbf{r}_k \cdot \mathbf{z}_k^b||^2 \tag{4.7a}$$

$$S_{\text{differ}} = \{\mathcal{S}_1, \mathcal{S}_2, \mathcal{S}_5, \mathcal{S}_6\} \tag{4.7b}$$

$$\mathcal{S}_1 = \{k | f(y_k^1) = 0\} \tag{4.7c}$$

$$\mathcal{S}_2 = \{k | v_k = 1 \text{ and } k \notin \text{Ransac Plane}\} \tag{4.7d}$$

$$\mathcal{S}_5 = \{k | v_k = 1\} \tag{4.7e}$$

$$\mathcal{S}_6 = \{k | f(y_k^6) = 0\}. \tag{4.7f}$$

Here $\mathcal{S}_j$ denotes the set of nodes where we are confident that true label is unlikely to be j based on the sensor inputs. $\mathbf{z}_{kp} = 1$ if only $k \in \mathcal{S}_p$ where $p \in L$ in all the other cases $\mathbf{z}_{kp}$ is set to zero, i.e. $\mathcal{S}_1$ denotes the set of superpixel in which the true label is very unlikely to be pedestrians. If FCN classifier prediction is zero for a certain superpixel which means that the super pixel is not in the foreground of the image. This implies that this super pixel have a least probability to be a pedestrian. On similar basis $\mathcal{S}_2, \mathcal{S}_5$ and $\mathcal{S}_6$ are found correspond to ground, sky and vehicle class respectively.
Finally for parts where we have point cloud segments we assign a loss, l_{laser}, if CQP prediction violates the label consistency obtained by the laser segments,

$$l_{\text{laser}} = \lambda_l \sum_{\mathbb{C}_j \in \mathbb{C}} \sum_{k \in \mathbb{C}_j} ||\mathbf{r}_k - \mathbf{r}_{k+1}||^2. \tag{4.8}$$

Now combining all the three loss components we obtain the total loss for an image frame,

$$l = l_{\text{agree}} + l_{\text{differ}} + l_{\text{laser}}. \tag{4.9}$$

Putting these parts together with a regularizer to prevent over fitting we obtain the following optimisation problem:

$$\Theta^* = \min_{\Theta} \sum_w l + ||\exp(\Theta)||^2, \tag{4.10}$$

where each w refer to input image index. This type of function is amenable to optimisation using stochastic gradient descent. For our method we propose to use ADAGRAD which is described in the next section.

Stochastic Gradient Decent For Loss Optimisation

As we operate in an online setting where we continuously obtain new observations, standard batch gradient optimisation methods are not applicable due to the unbounded size of the data to be processed. Instead, we use stochastic gradient descent (SGD) which operates on a single observation at a time to optimise the parameters using the loss function presented in Eq. (4.5).

For each image we compute the stochastic gradient of the loss function with which we update the parameter vector Θ. To this end, we form a mini-batch composed of the last consecutive B number of image frames and perform B number of parameter update steps (1 step per each frame). Mini-batch size is dependent on the rate at which images are received, resolution of the image and the required frequency of the semantic segmentation. The values of Θ learnt with this stochastic process are adapted to the current context of the continuous image sequence, however, also retain information from the past.

As the learning rate has a big impact on the speed of convergence and quality of the result we employ ADAGRAD which uses individual learning rates for each parameter based on past data. The basic equations of ADAGRAD have the following form:

$$\mathcal{G} = \sum_t \Lambda_t \Lambda_t^T. \tag{4.11}$$

where $\Lambda_t = \bigtriangledown l(\mathbf{z}, \mathbb{C}, \mathbf{r})$ is the point gradient at iteration t. The gradient is calculated using central finite differences. With this we can update the parameter set Θ as follows:

$$\Theta := \Theta - \eta Diag(\mathcal{G})^{-0.5} \cdot \Lambda, \tag{4.12}$$

where η is the global learning rate, Λ the current gradient, and t the iteration number.

Even though, ADAGRAD works well in typical large scale problems there are some drawbacks when using it in an online setting. The main one is that the entire gradient value history is accumulated which results in a continuously decreasing step size. In an online settings this means that at some point the parameters would no longer adapt to changes in the environment. One possible solution is to use a constant fixed learning rate which would always allow for changes in the environment to be reflected in the parameters. However, selecting a suitable fixed learning rate is not trivial and would require a lot of testing for different scenarios which clearly is not ideal. So in order to preserve good learning rates of ADAGRAD while still being able to adapt to changes we adopt a procedure

Algorithm 6: Online Learning Algorithm

// cqp(..)- Function which provides the MAP estimation for CQP
// t - Iteration number
// w - Image frame index
// δ - Threshold value for minimum step size change per iteration
// υ - Threshold value for number of iterations
// $\mathcal{G}$ - Accumulated gradient
// Θ_w - Parameter set correspond to w^{th} image frame
Input: η-Global learning rate , I - Input image , B - Mini batch size
Output: Label assignment $\mathbf{y}$
// Initialisation

$w = B + 1$
$\mathcal{G} = 0$
$\Theta_w = [1]_{1\times 21} \quad \forall w \in [1, ..B]$
// SGD parameter optimisation
while *Images available* **do**
  // Select past B frames and shuffle
  foreach $t \in \{w - B, .., w\}$ **do**
    $\mathbf{r} \leftarrow \text{cqp}(\Theta, I_t)$
    $\mathbf{z} \leftarrow$ self supervised reference labels of image I_t
    $\mathbb{C} \leftarrow$ laser constraints correspond to the image I_t
    loss=$l(\mathbf{z}, \mathbb{C}, \mathbf{r})$
    // gradient of the loss function
    $\Lambda \leftarrow \frac{dl}{d\Theta}$
    // gradient accumulation
    $\mathcal{G} \leftarrow \mathcal{G} + \Lambda\Lambda^T$
    // updating the parameters
    $\Theta_w \leftarrow \Theta_w - \eta Diag(\mathcal{G})^{-1/2} \cdot \Lambda$
  end
  // Decide when to reset the step size
  if $\forall \tau \in [t : t - \upsilon]; abs(\Theta_\tau - \Theta_{\tau-1}) < \delta$ **then**
    $\mathcal{G} = 0;$
  end
  $\Theta_{w+1} \leftarrow \Theta_w$
  $\mathbf{y} \leftarrow \text{cqp}(\Theta_{w+1}, I_{w+1})$
  $w \leftarrow w + 1$
end
return y

similar to [92]. The basic idea is to have a decaying learning rate, but at opportune moments increase this learning rate again to allow quicker adaptation. In our case once the learning rate has become sufficiently small for a number of iterations we set $\mathcal{G} = 0$ which discards all previously accumulated gradient information. This effectively increases the learning rate and allows the optimiser to adapt to changes if necessary. In the case the distribution has changed the gradient will be non-zero and pull the solution to a different local minimum. Similarly, if the data distribution hasn't changed the gradients will be close to zero and the algorithm will not change the parameters.

An overview of the steps involved in our algorithm are summarised in Algorithm 6. For each new image the last B images, typically 5 to 10, are used to perform the intermediate updates of Θ using the loss value for each individual image (lines 8 to 14). Next we decide whether or not to reset the step size which allows us to keep adapting to changes (lines 17 to 19). Finally, we replace the current parameters with the updated parameters obtained which can be subsequently used to classify the next incoming image (lines 20 to 22).

4.4 Experiments

4.4.1 Model Building

In this section we present experimental evaluation of our proposed framework for online learning of CRF parameters. We compare the results obtained using CQP with no parameter learning with those obtained using CQP with adaptive parameters. We use the KITTI dataset [35] as it provides typical image and laser scanner data collected in urban environments. The data which was collected in the city of Karlsruhe, Germany using a vehicle equipped with the cameras and a Velodyne laser scanner provides a variety of scenes and environmental conditions.

The range of the loss function weight values α and λ are chosen through a grid search. Subsequently, fine-tuning is done manually using training set of 100 labelled images from `drive_0091`. Intuition behind fine-tuning is the importance of the each loss component and the reliability of the corresponding reference label, i.e. if the scene classification model assigns ground label to a unlikely super pixel that is highly critical for autonomous driving application. Hence weight on this loss (error) component should be larger, Pedestrians appear less and cover a small area of the image hence the loss component is smaller compared to the other classes. Therefore it requires a larger weigh value on this component to have an equal bias. α values are the weights on each class for the loss correspond to unlikely labelling. As mentioned in Chapter 4.3.3, we do not define this loss component for 3-vegetation and 4-building classes because the information from

α_1	α_2	α_3	α_4	α_5	α_6	
0.47	8.00	0.00	0.00	0.20	0.30	
λ_1	λ_2	λ_3	λ_4	λ_5	λ_6	λ_l
1.50	0.47	0.50	0.75	0.50	3.00	2.00

Table 4.1: Overview of the loss function parameters.

the sensors is not sufficient to have solid decisions about these classes on wrong labelling. Therefore α_3 and α_4 are set to zero. λ and α values are summarised in Table 4.1. The base learning rate was set as $\eta = 0.037$ which was obtained from the experiments on the training set.

We use the same features HSV colour histograms, RGB Hog features, and pixel coordinates as in the previous chapter to train the pLDA classifier using the selected training set. The FCN classifier is a publically available pre-trained model (`pascal-fcn32s-dag`) [110] for recognising the foreground objects in an outdoor scene. The model is trained using the Pascal dataset[27]. We used Matconvnet [17] environment to run the FCN framework to obtain the class predictions. We have used the `drive_0021`, `drive_0043`, `drive_0071`, `drive_0038`, `drive_0093` and `drive_0095` for testing the algorithm. Every 10^{th} frame in these sequences are manually labeled for quantitative analysis of the online CQP model.

4.4.2 Results

In the following we present results comparing CQP using fixed parameters and CQP using parameters that are adapted online using our proposed method. An overview of the typical behaviour and performance of the proposed algorithm is shown in Figure 4.2. The top image shows how the online adaptive CQP maintains a higher overall accuracy in comparison to CQP using fixed parameters. This is clearly visible in the areas where CQP has drops in accuracy which the online CQP manages to avoid as it adapts to the changes and as a result doesn't drop as much in terms of accuracy. The middle and bottom graph evaluate the accuracy on the parts of the image for which we have obtained reference labels (second) and those where we have had no reference label information (third). As to be expected the result for areas where we had label information is better then for those where we lack label information, however, the difference is relatively small. Overall the shapes and trends are quite similar which is a good indication that the parameter training done on the labelled parts of data influences the parameters of classes of data without labels in a positive way. One interesting case are the two drops in performance around the frame #200 and #350. In the first instance this drop is present in both cases with and without reference labels

and as a result the online CQP manages to mitigate it. By contrast the second instance only occurs in the part with no reference labels and as such no parameter adaptation happens because of it and both the online CQP and fixed CQP reduce in accuracy. This again demonstrates that parameters updated based on the parts of the data with reference labels improves the performance in areas where we have not obtained reference labels.

Figure 4.3 shows selected images of the sequence . Figure 4.4 and Figure 4.5 show the corresponding reference labels of images A,B,C and D. In the instances A, B and D the errors are mainly due to the wrong classification of the ground class. Unexpected noise in the image data(blur images) and changes in lighting can reduce the accuracy of the local classifier. However when the class prediction is less accurate in a certain area the error may propagate to a large area through the enforcement of the laser based global constraints. These occasional reductions in performance in the CQP can be overcome through the online parameter learning. The parameters will assist the CQP to find a solution closer to the previous adjacent image frames without a large deviation and maintain the consistency of the classification under real world conditions.

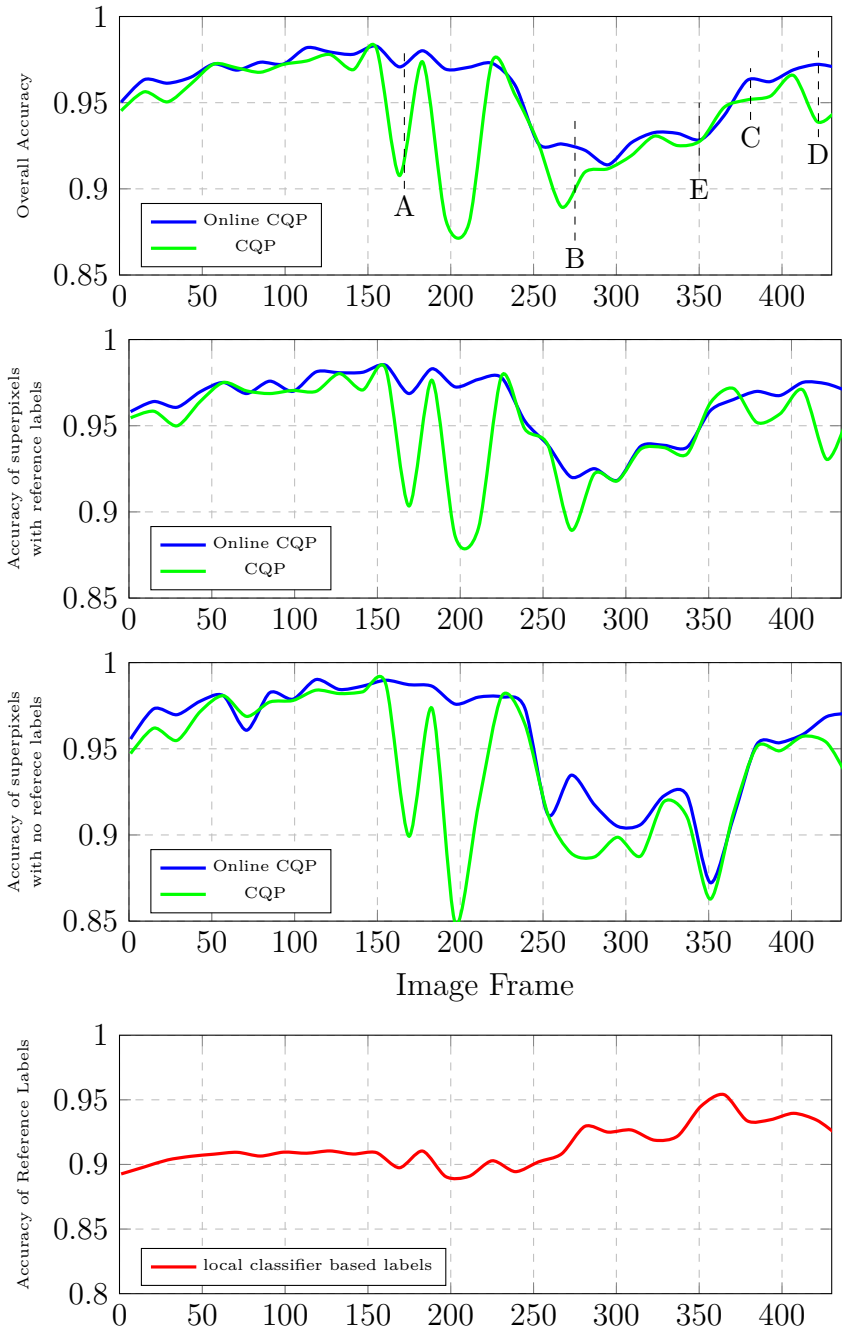


Figure 4.2: Accuracy on a per frame basis for the `drive_0093` dataset from KITTI. (Top) overall accuracy of each frame (second) accuracy of the super pixels for which we extracted reference labels in a self-supervised manner, and (third) accuracy for super pixels without label information. Overall the online CQP is able to adapt the parameters to prevent drastic reduction in accuracy. Comparing the (second) and (third) graphs one can see that even though the parameters are learned only on data from the (second) the changes have a positive impact on the (third) graph. The (bottom) graph depicts the reference labels correspond to each frame. These labels are derived from local classifier outputs.

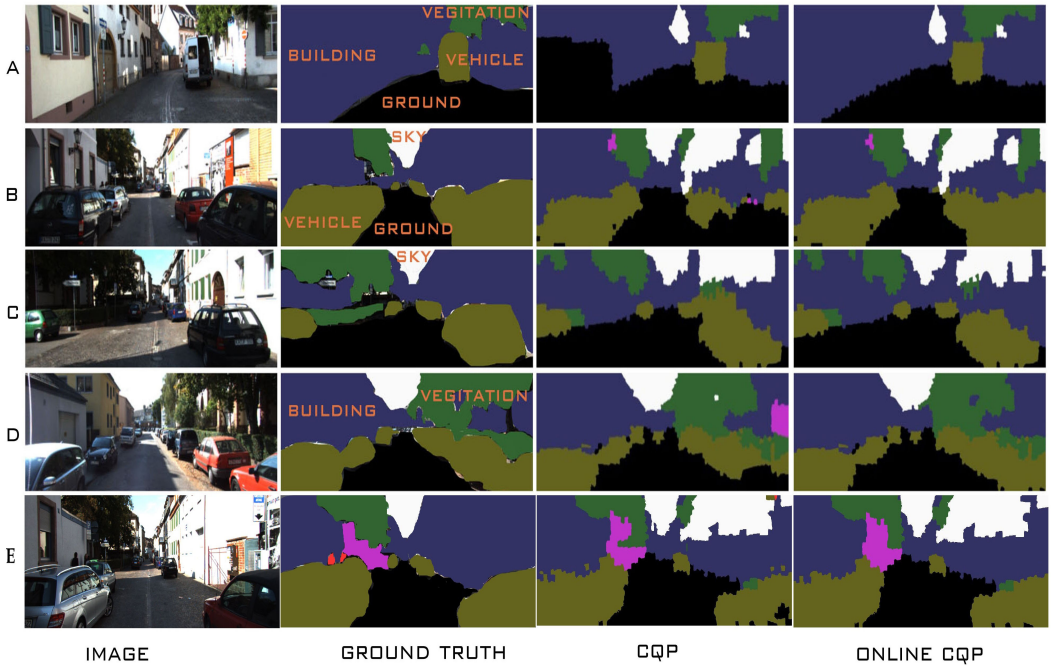
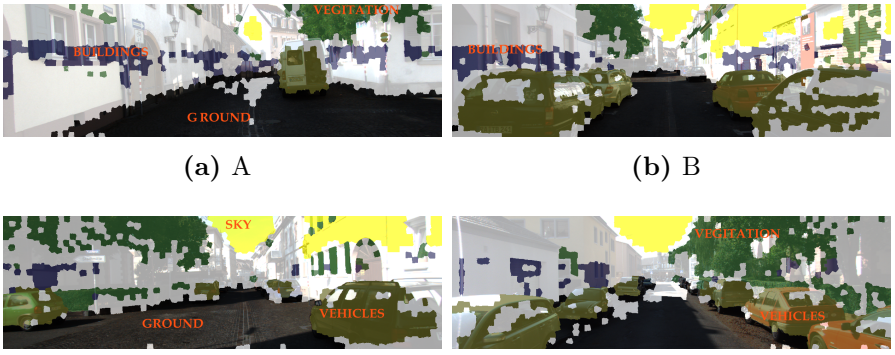


Figure 4.3: This figure contains the image frames corresponding to the marked points A,B,C,D in the top graph of Figure 4.2.The result of online CQP is be suggestive of overcoming the random errors in CQP solution due to illumination and noise



(a) A **(b)** B

(c) C **(d)** D

Figure 4.4: Images shows the set of the reference labels derived from the label agreement of pLDA classifier, FCN classifier and laser segments correspond to the image A,B,C,D.

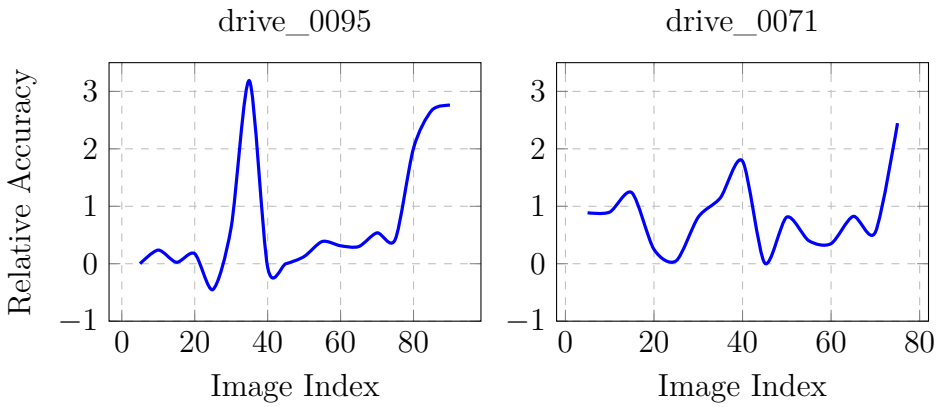
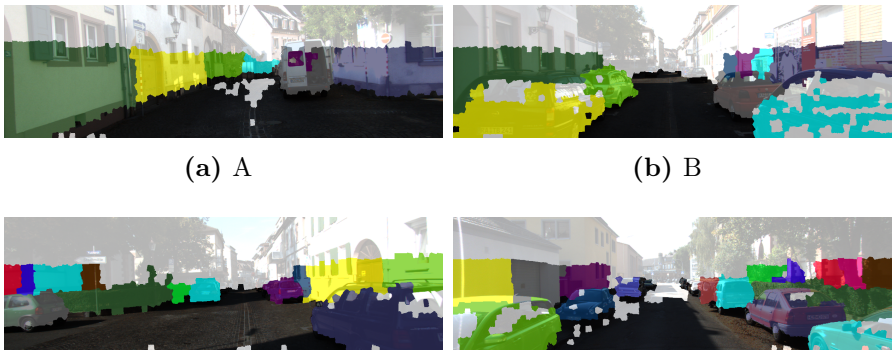


Figure 4.6: The plots show the relative accuracy, i.e. difference in absolute accuracy values, between CQP using fixed parameters and online CQP. A positive value indicates that the accuracy of online CQP is better then that of CQP. We can see how online CQP is outperforming CQP in almost all cases. Big spikes in the relative accuracy can be explained by a drop in accuracy of CQP that online CQP managed to adapt to in time.



(a) A **(b)** B

(c) C **(d)** D

Figure 4.5: Images shows the set of the reference labels related to label consistency correspond to the image A,B,C,D . Each color shows where the label assignment should be consistent. These constraints are derived from laser segments.This label consistency is encouraged in learning process.

The same type of improvements can be observed in other datasets. Figure 4.6 shows the relative change in accuracy between CQP and online CQP, i.e. a positive value indicates that online CQP is performing better then CQP using fixed parameters. From these plots we can see the constant gain in accuracy where the spikes stem from sudden drops in accuracy in CQP which online CQP manages to mitigate. These results are also verified in the comparison of several performance metrics on multiple datasets in Table 4.2. The table shows how online CQP consistently improves on the results obtained by CQP. This improvement is typically in the 2% to 3% range, but in a few cases the gain is as much as 6%.

Next we are going to look at the per class performance to see the impact online CQP has on those. Looking at Figure 4.7 we can see that for very simple classes such as “ground” there is barely any improvement. For more complex and varied

Quality Measure	Average Precision		Average Recall		Average Accuracy		F1 Score	
Method	CQP	Online CQP	CQP	Online CQP	CQP	Online CQP	CQP	Online CQP
Dataset0071	0.8440	0.8676	0.8793	0.8987	0.9493	0.9562	0.8237	0.8481
Dataset0095	0.8501	0.8938	0.9350	0.9393	0.9496	0.9642	0.8435	0.8884
Dataset0038	0.7454	0.7860	0.7135	0.7780	0.9317	0.9441	0.7284	0.7814
Dataset0093	0.8534	0.8767	0.8390	0.8624	0.9503	0.9601	0.8458	0.8692

Table 4.2: Quantitative comparison of CQP and online CQP on the test set. Online CQP shows an improvement over the results of CQP for the image sequences collected under different environmental conditions, i.e. The dataset `drive_0071` has a high concentration of pedestrians and `drive_0095` is collected in an area with higher vehicle density.

Quality Measure	Average Precision		Average Recall		Average Accuracy		F1 Score	
Method	CQP	Online CQP	CQP	Online CQP	CQP	Online CQP	CQP	Online CQP
Pedestrians	0.8066	0.7994	0.5797	0.7279	0.9233	0.9346	0.5184	0.6375
Ground	0.7223	0.7805	0.9095	0.9379	0.9145	0.9363	0.8335	0.8687
Vegetation	0.8923	0.8949	0.8830	0.8658	0.9705	0.9707	0.8622	0.8624
Buildings	0.9527	0.9515	0.9210	0.9219	0.9050	0.9124	0.8768	0.8888
Sky	0.6674	0.6911	0.8885	0.8960	0.9868	0.9877	0.7435	0.7521
Vehicle	0.8701	0.8987	0.8801	0.8955	0.9123	0.9345	0.8751	0.8955

Table 4.3: Class wise accuracy, precision, recall, and F1 score for CQP and online CQP on the `drive_0093` dataset. Different metrics are improved for different classes which is dependent on what makes a class hard to classify correctly. However, across the board the F1 score increases, indicating that online CQP manages to improve on hard aspects of the classification without sacrificing other areas.

classes this changes. In the case of the "pedestrian and cyclists" class there is mostly no change, however, when CQP makes large errors the online CQP method maintains good accuracy. Looking at the "vegetation" and "buildings" classes we can see that online CQP has a somewhat smoother curve while exhibiting a positive accuracy offset over CQP. These impressions are also verified by the numerical evaluation presented in Table 4.3 for `drive_0093`. For hard classes such as "Cyclists & Pedestrians" the precision is not improving, however, recall improves significantly which also reflects in the F1 score. Depending on the class some metrics remain unchanged while others gain and as a result the F1 score improves across the board. As such the online CQP method manages to improve on the challenging metrics for each class without degrading others.

In longer range autonomous navigation the environment changes smoothly rather then abruptly we evaluated the ability of our proposed method to quickly adapt to changes in the data. To this end we selected two very different datasets, `drive_0093` which contains mainly vehicles and `drive_0071` which has data captured in a pedestrian zone. These two datasets were processed one after the other as if they were one continuous data stream. In Figure 4.8 we show the evolution

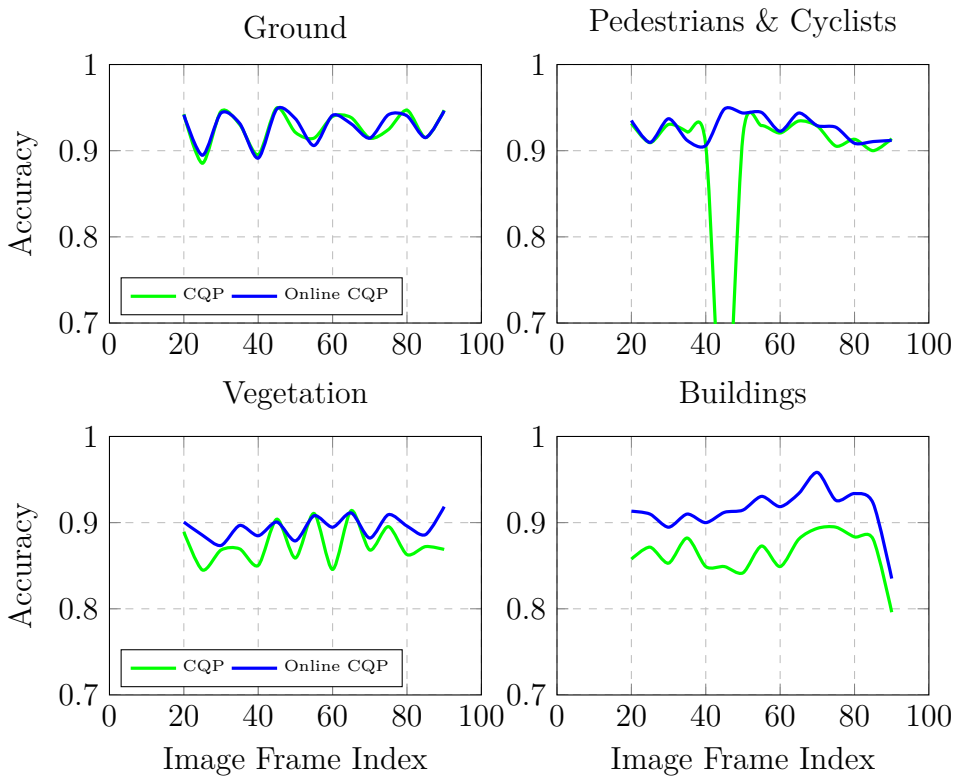


Figure 4.7: Accuracy of CQP and online CQP on a per class basis for the `drive_0038` dataset. For easy classes there is little difference, however, in more complex ones we can see online CQP retaining good accuracy when CQP drops significantly as seen in "Pedestrians & Cyclists" or has a constant performance offset as in the "Buildings" class.

of the unary potential parameters of online CQP (top) and on-diagonal pairwise parameters (bottom) as we process the data. All parameters start with a value of 1 and we can see how they quickly move to mostly stable values different from 1. Then around iteration 800 the first dataset ends and the second one starts being processed. We can see abrupt jumps in the values indicating that the SGD method is able to quickly change parameters if needed. After this short period of rapid changes all parameters settle again. The actual direction in which the parameter values move is not necessarily indicative of the scene composition as the parameter interact in complex ways inside the CQP method itself.

The importance of being able to quickly adapt to changes, even if this is a rare occurrence, is demonstrated in Figure 4.9 which compares the accuracy of the first 50 frames after we switch the datasets. We compare the results of CQP using the same fixed parameters, online CQP which contentiously adapts its parameters and partial online CQP which adapts the parameters until the dataset changes, i.e. the parameters at point "A" in Figure 4.8 are used. This allows us to evaluate how important the ability to adapt quickly is. We can see that both online CQP methods outperform CQP which is in line with the

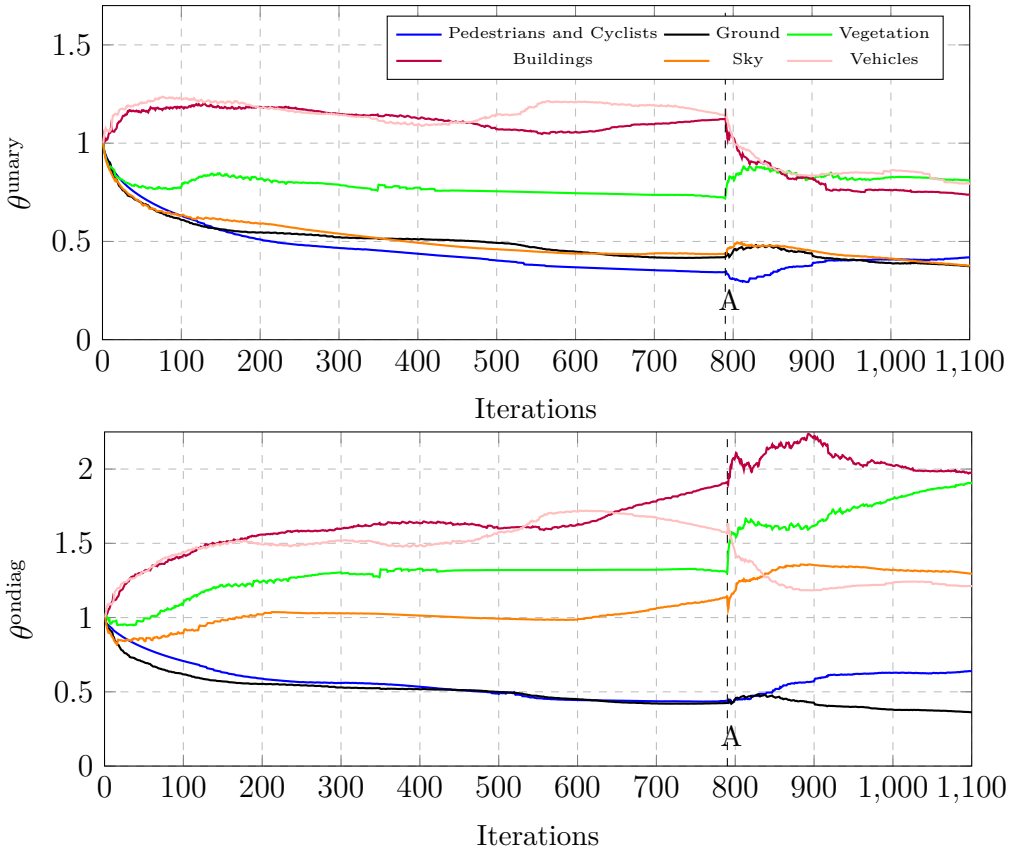


Figure 4.8: The top and bottom plots show the change of parameters correspond to unary and pairwise potentials with adaptive learning. The test is done for `drive_0093`. At iteration A data sequence `drive_0071` is fed to the framework which has different lightning conditions and class distribution than the previous one. Plots clearly depict that after this sudden change on input data, parameters dramatically change to adapt the situation.

previous results. The interesting part is the comparison of the two online CQP methods. In several areas we can observe that the lack of adaptability results in degraded performance, for example around frame 30 and 40. As such being able to react quickly to changes in the environment is important to prevent errors to accumulate over time.

All computations were performed on an Intel Core-i5 3.20GHz processor with MATLAB implementations of the algorithms. Each parameter update step using a single image requires 70 ms. As the parameter updates are independent of the segmentation itself it is possible to perform the segmentation at a higher frequency then the parameter updates. Furthermore, mini batch size B considered in a single update step can be chosen in a wide range. As we can see in Figure 4.10 the performance stays very stable with 10 or more images used. This means that longer range information, from older images, does not negatively impact the adaptation capability of the algorithm.

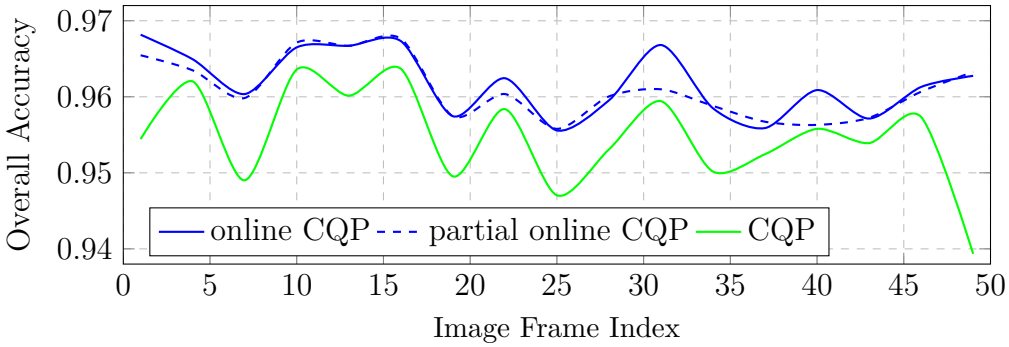


Figure 4.9: Accuracy of the first 50 frames after the new dataset was introduced. Online CQP continues to adapt, while partial online CQP continues to use the parameters used at the end of the first dataset (at iteration A) while CQP uses the same initial parameters. We can see how the continued adaptation allows online CQP to improve over the partial online CQP. As seen previously both versions of online CQP outperform CQP using fixed initial parameters.

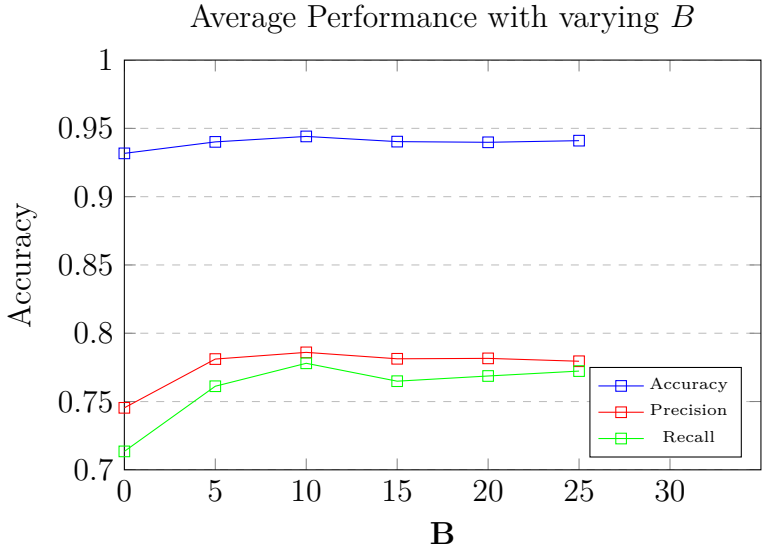


Figure 4.10: The plots shows the quality of the segmentation of the image sequence `drive_0038` with online CQP with varying number of images B considered in each update step.

4.5 Summary

In this chapter we presented a method that learns the parameters of the unary and pairwise potentials of a CRF in an online manner. This enables the algorithm to adapt the parameters based on the current situation which is advantageous in a life-long learning scenario where it the environment is expected to change over time. This is achieved by formulating the selection of the optimal parameters as a loss function using reference labels that are obtained in a self-supervised manner. This loss function is updated efficiently using stochastic gradient descent with continuously adapting learning rates. In experiments conducted using data from the KITTI dataset we demonstrate the benefit in regards of scene segmentation performance of a CRF that continuously adapts it's parameters over one with fixed parameters. Furthermore, we demonstrated that the proposed method can quickly adapt to changes in the environment.

Chapter 5

Conclusion

This thesis proposes a framework to efficiently combine multiple modalities available on robotic platforms to conduct street scene understanding with the ability to adapt to changes for long-term navigation. The semantic scene segmentation problem is formulated as a MAP estimation of a pairwise CRF. The solution to the MAP problem is sought using a QP formulation. QP formulation provides the flexibility to combine the information from multiple modalities by allowing to incorporate global constraints about label consistency. This addition of global level information increase the accuracy of the segmentation and the computational cost of inference problem. The adaptability to changing input data is achieved through parameter learning. SGD based approach is used in parameter learning while exploiting a self-supervised set of reference labels. Despite the fact that the framework is proposed for scene segmentation, it is general in nature. Therefore the proposed methods can be used to combine arbitrary types of a priori information about the optimum solution for any problem that can be formulated as a MAP estimation of a pairwise CRF. Furthermore, the online learning process described here can be applied to any model that need to adjust to changing input data given that model has sufficient amount of sensor/input data to interpret its environment. This chapter summarises the main contributions of the thesis and discusses interesting extensions to the proposed model.

5.1 Summary of Contributions

5.1.1 Constrained Quadratic Programming Inference

In Chapter 3, we proposed a method to enhance the quality of semantic scene segmentation by incorporating a priori information (possibly from multiple modalities) in the form of constraints. There we formulated the scene understanding problem as a pairwise conditional random field to obtain optimum labelling through the MAP estimation. Subsequently, we expressed the MAP problem as a QP problem which permits adding additional constraints to enforce label consistency. Then we presented an efficient gradient-based method to solve the constrained QP problem. The proposed inference process reduced the dimension-

ality of the inference problem dramatically which allows the segmentation to be performed in real time.

5.1.2 Integration of Visual and Depth Information

In Chapter 3, we demonstrate an application of the proposed CQP model. We address the issue of combining camera-based visual data and laser based depth data for outdoor scene segmentation. Since these modalities have different viewpoints (due to the physical location), operate in different domains and also lacks unique correspondence. These facts make it challenging to fuse this information directly. We use visual features to compute the potential terms of the CQP model while extracting label consistency constraints from the laser point cloud. In this manner, the visual and depth information is represented in a single domain which results in scene labelling that is robust to illumination changes and occlusions. The efficiency of the model is tested on a dataset gathered by a real robotic platform equipped with colour cameras and Velodyne laser scanners.

5.1.3 Self Supervised Parameter Learning

Parameter learning is vital in enhancing model robustness to changing input data distribution. However, learning in an online setting is difficult because the nature of new observations is unknown because no labelled data is available in online settings. Therefore, we introduce a method in Chapter 4 which generates reference labels for the data in a self-supervised manner which allows it to be used for parameter learning. These reference labels are obtained by exploiting the predictions from a discriminant analysis classifier and fully convolutional net, which are combined with laser-based label consistency information. These reference labels are utilised to optimise the CQP parameters by minimising the loss due to the inaccurate predictions from the CQP model. This learning process improve the autonomy of CQP method in long-term navigational tasks because it requires a minimum amount of human supervision.

5.1.4 Robust Parameter learning for non-stationary data distributions

In real world navigation, robots may encounter entirely new scenes, e.g. areas with large crowds, areas with heavy traffic and environmental changes due to different weather conditions. Scene understanding becomes problematic due to this diversity in the situations, which necessitates adapting the scene segmentation model accordingly. Therefore, we proposed a method to preserve the adaptability of parameter learning in long term navigation. We optimised the loss com-

puted using the reference labels, by implementing a stochastic gradient descent method that can continuously seek the best set of parameters to the current circumstances. These locally optimised parameters maintain the segmentation accuracy in a higher level. To obtain quality results it requires carefully handling the SGD learning rates. We let the learning rates to automatically decrease the learning rate using the ADAGRAD method to reach local optimum and also provide the ability to increase the learning rate when the input data distribution changes so the parameters can escape the local optimum. The learning is done with minimum level of human supervision.

5.2 Future Work

In this section, we discuss areas of possible future developments based on the methods proposed in this thesis.

5.2.1 Local Classification

In our proposed method, we used a linear discriminant analysis classifier combined with a pre-trained fully convolutional net classifier. The quality of the image segmentation can be improved by training a new fully convolutional net classifier for urban street scenes. An advanced unary classifier can increase the number of superpixels that have reliable reference labels for loss computation as well as the accuracy of each labelling which again increase the accuracy of parameter learning.

5.2.2 Global Constraints

In our model, we only exploit label consistency information in the form of global constraints. Nevertheless, constraints can be modified to impose other information such as object co-occurrence statistics. Furthermore, the model performance can be tested for the cases where it needs to combine modalities other than camera and laser, i.e. Adding infrared image based constraints along with laser constraints might lead to interesting results. The thesis has focused on establishing an efficient inference method for QP problems with linear equality constraints that is used in semantic image segmentation. Still, it would be interesting to discover efficient means of inference in linear inequality constrained QP problems in this context. Equally, it might be useful to expand the inference algorithm to solve nonlinear constraints as well. Sophisticated constraints will facilitate imposing more informative priori information about the final segmentation which is harder to do using CRF potentials.

5.2.3 Long Term Autonomy

The learning rate of SGD utilised in the thesis is set to decrease over time and increase when the change in parameters is below a threshold. This technique assists in adjusting the parameters for unknown environments. It is important to investigate other possible methods to identify when there is a change in the input data distribution so the learning rate can be increased accordingly to facilitate learning the new distribution. It would be useful if we can derive a measure of the distribution change so we can increase learning rate more meaning fully, i.e. Selecting the magnitude of the learning rate increment depending on the sharpness of the distribution change. In [92], a method was proposed to increase the SGD learning rate when there is an abrupt change in the input data distribution to facilitate parameter learning. Even though, in real world navigation problems the change of the data distribution is gradual, it might be possible to adapt the above method to improve the learning rate selection process.

Another interesting area to investigate is building a strong relationship between consecutive frames over time. The constrained CRF model can be extended to a hierarchical model connecting multiple images over time. Spatio-temporal connections will allow the segmentation model to get the use of the past knowledge as well as enable addition of temporal constraints that can increase the quality of the segmentation.

Bibliography

[1] R Achanta, A Shaji, K Smith, A Lucchi, P Fua, and S Süsstrunk. Slic superpixels compared to state-of-the-art superpixel methods. *IEEE transactions on pattern analysis and machine intelligence*, 34(11):2274–2282, 2012.

[2] A Aijazi, P Checchin, and L Trassoudaine. Segmentation Based Classification of 3D Urban Point Clouds: A Super-Voxel Based Approach with Evaluation. *Remote Sensing*, 5(4):1624–1650, 2013.

[3] I Aleksander and H Morton. An introduction to neural computing. 240, 1990.

[4] J M Álvarez, A M López, T Gevers, and F Lumbreras. Combining priors, appearance, and context for road detection. *IEEE Transactions on Intelligent Transportation Systems*, 15(3):1168–1178, 2014.

[5] C Alvis, L Ott, and F Ramos. Urban scene segmentation with laser-constrained crfs. In *International Conference On Intelligent Robots and Systems*, 2016.

[6] N Audebert, B Le Saux, and S Lefèvre. Semantic segmentation of earth observation data using multimodal and multi-scale deep networks. In *Asian Conference on Computer Vision*, pages 180–196. Springer, 2016.

[7] Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *arXiv preprint arXiv:1511.00561*, 2015.

[8] R Benenson, M Omran, J Hosang, and B Schiele. Ten years of pedestrian detection, what have we learned? *arXiv preprint arXiv:1411.4304*, 2014.

[9] J Besag. Statistical analysis of non-lattice data. *The statistician*, pages 179–195, 1975.

[10] J Besag. On the statistical analysis of dirty pictures. *Journal of the Royal Statistical Society. Series B*, pages 259–302, 1986.

[11] A Blake, P Kohli, and C Rother. *Markov random fields for vision and image processing*. Mit Press, 2011.

[12] L Bottou. Large-scale machine learning with stochastic gradient descent. In *Proceedings of COMPSTAT'2010*, pages 177–186. Springer, 2010.

[13] L Bottou. Stochastic gradient descent tricks. In *Neural Networks: Tricks of the Trade*, pages 421–436. Springer, 2012.

[14] L Bottou, F E Curtis, and J Nocedal. Optimization methods for large-scale machine learning. *arXiv preprint arXiv:1606.04838*, 2016.

[15] Y Boykov and G Funka-Lea. Graph Cuts and Efficient ND Image Segmentation. *International Journal of Computer Vision*, 70(2):109–131, 2006.

[16] Y Boykov and M Jolly. Interactive Graph Cuts for Optimal Boundary & Region Segmentation of Objects in ND Images. In *IEEE International Conference on Computer Vision*, 2001.

[17] Y Boykov and V Kolmogorov. An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. *IEEE transactions on pattern analysis and machine intelligence*, 26(9):1124–1137, 2004.

[18] Y Boykov, O Veksler, and R Zabih. Fast approximate energy minimization via graph cuts. *IEEE Transactions on pattern analysis and machine intelligence*, 23 (11):1222–1239, 2001.

[19] R G Cowell. *Probabilistic networks and expert systems: Exact computational methods for Bayesian networks*. Springer Science & Business Media, 2006.

[20] N Cristianini and J Shawe-Taylor. *An introduction to support vector machines and other kernel-based learning methods*. Cambridge university press, 2000.

[21] F Delbos and J C Gilbert. *Global linear convergence of an augmented Lagrangian algorithm for solving convex quadratic optimization problems*. PhD thesis, INRIA, 2003.

[22] K G Derpanis. Overview of the ransac algorithm. *Image Rochester NY*, 4(1):2–3, 2010.

[23] B Douillard, D Fox, and F Ramos. A Spatio-Temporal Probabilistic Model for Multi-Sensor Object Recognition. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2007.

[24] J Duchi, E Hazan, and Y Singer. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 12(Jul): 2121–2159, 2011.

[25] J Duchi, E Hazan, and Y Singer. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 12(Jul): 2121–2159, 2011.

[26] A Eitel, J Tobias Springenberg, L Spinello, M Riedmiller, and W Burgard. Multimodal deep learning for robust rgb-d object recognition. In *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on*, pages 681–687. IEEE, 2015.

[27] M Everingham, S M A Eslami, L Van Gool, C K I Williams, J Winn, and A Zisserman. The pascal visual object classes challenge: A retrospective. *International Journal of Computer Vision*, 111(1):98–136, January 2015.

[28] A Fathi, M Balcan, X Ren, and J Rehg. Combining self training and active learning for video segmentation. In *In Proceedings of the British Machine Vision Conference*, volume 29, pages 78–1, 2011.

[29] P Felzenszwalb and D Huttenlocher. Efficient Graph-Based Image Segmentation. *International Journal of Computer Vision*, 59(2):167–181, 2004.

[30] M Fischler and R Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981.

[31] M A Fischler and R C Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981.

[32] L R Ford and D R Fulkerson. Maximal flow through a network. *Canadian journal of Mathematics*, 8(3):399–404, 1956.

[33] C Galleguillos, A Rabinovich, and S Belongie. Object categorization using co-occurrence, location and appearance. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8. IEEE, 2008.

[34] R S Garfinkel and G L Nemhauser. *Integer programming*, volume 4. Wiley New York, 1972.

[35] A Geiger, P Lenz, C Stiller, and R Urtasun. Vision meets Robotics : The KITTI Dataset. *The International Journal of Robotics Research*, 32(11):1231–1237, 2013.

[36] Markus Gerke. Use of the stair vision library within the isprs 2d semantic labeling benchmark (vaihingen). 2014.

[37] C Glennie and D Lichti. Static calibration and analysis of the velodyne hdl-64e s2 for high accuracy mobile scanning. *Remote Sensing*, 2(6):1610–1624, 2010.

[38] N Gould, M E Hribar, and J Nocedal. On the solution of equality constrained quadratic programming problems arising in optimization. *SIAM Journal on Scientific Computing*, 23(4):1376–1395, 2001.

[39] S Gupta, R Girshick, P Arbeláez, and J Malik. Learning rich features from rgb-d images for object detection and segmentation. In *European Conference on Computer Vision*, pages 345–360. Springer, 2014.

[40] W Hager. The dual active set algorithm and its application to linear programming. *Computational Optimization and Applications*, 21(3):263–275, 2002.

[41] H He and B Upcroft. Nonparametric semantic segmentation for 3d street scenes. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2013.

[42] H He and B Upcroft. Automatic object segmentation of unstructured scenes using colour and depth maps. *Computer Vision, IET*, 8(1):45–53, 2014.

[43] K He, X Zhang, S Ren, and J Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[44] X He and R Zemel. Learning hybrid models for image annotation with partially labeled data. In *Advances in Neural Information Processing Systems*, 2009.

[45] A Hermans, G Floros, and B Leibe. Dense 3D Semantic Mapping of Indoor Scenes from RGB-D Images. In *IEEE International Conference on Robotics & Automation*, 2014.

[46] A Ion, J Carreira, and C Sminchisescu. Probabilistic joint image segmentation and labeling. In *Advances in Neural Information Processing Systems*, pages 1827–1835, 2011.

[47] F V Jensen. *An introduction to Bayesian networks*, volume 210. UCL press London, 1996.

[48] S Johnson. The NLopt nonlinear-optimization package. http://ab-initio.mit.edu/nlopt.

[49] Alex K, I Sutskever, and G Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems 25*, pages 1097–1105. Curran Associates, Inc., 2012.

[50] D Kahle, T Savitsky, S Schnelle, and V Cevher. Junction tree algorithm. *STAT*, 631, 2008.

[51] R Kindermann and L Snell. *Markov random fields and their applications*. 1980.

[52] P Kohli, L Ladicky, and P Torr. Graph cuts for minimizing robust higher order potentials. In *International Conference on Computer Vision and Pattern Recognition*, 2008.

[53] P Kohli, L Ladicky, and P Torr. Robust Higher Order Potentials for Enforcing Label Consistency. *International Journal of Computer Vision*, 82(3):302–324, 2009.

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

[55] N Komodakis and N Paragios. Beyond Pairwise Energies: Efficient Optimization for Higher-Order MRFs. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2009.

[56] E Krogstad. *Optimeringsteori Quadratic Programming Basics*. PhD thesis, NTNU, 2012.

[57] F R Kschischang, B J Frey, and H-A Loeliger. Factor graphs and the sum-product algorithm. *IEEE Transactions on information theory*, 47(2):498–519, 2001.

[58] M P Kumar, V Kolmogorov, and P H Torr. An analysis of convex relaxations for map estimation. *Advances in Neural Information Processing Systems*, 20: 1041–1048, 2007.

[59] S Kumar and M Hebert. A hierarchical field framework for unified context-based classification. In *IEEE International Conference on Computer Vision*. IEEE, 2005.

[60] Abhijit Kundu, Yin Li, Frank Dellaert, Fuxin Li, and James M Rehg. Joint semantic segmentation and 3d reconstruction from monocular video. In *European Conference on Computer Vision*, pages 703–718. Springer, 2014.

[61] L Ladicky, C Russell, P Kohli, and P H S Torr. Graph cut based inference with co-occurrence statistics. In *European Conference on Computer Vision*, pages 239–253. Springer, 2010.

[62] L Ladicky, C Russell, P Kohli, and P Torr. Associative Hierarchical Random Fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(6): 1056–1077, 2014.

[63] L D Landau and E M Lifshitz. Statistical physics, vol. 5. *Course of theoretical physics*, 30, 1980.

[64] T Li, S Zhu, and M Ogihara. Using discriminant analysis for multi-class classification: an experimental investigation. *Knowledge and information systems*, 10 (4):453–472, 2006.

[65] J Long, E Shelhamer, and T Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3431–3440, 2015.

[66] D G Lowe. Object recognition from local scale-invariant features. In *The proceedings of the seventh IEEE international conference on Computer Vision*, volume 2, pages 1150–1157. IEEE, 1999.

[67] D Margulis. *Photoshop LAB color: The canyon conundrum and other adventures in the most powerful colorspace*. Peachpit Press, 2005.

[68] MathWorks. Discriminant Analysis, 2016. URL `https://au.mathworks.com/help/stats/discriminant-analysis.html`.

[69] K A McShane, C L Monma, and D Shanno. An implementation of a primal-dual interior point method for linear programming. *ORSA Journal on computing*, 1 (2):70–83, 1989.

[70] O Mees, A Eitel, and W Burgard. Choosing smartly: Adaptive multimodal fusion for object detection in changing environments. In *Intelligent Robots and Systems (IROS), 2016 IEEE/RSJ International Conference on*, pages 151–156. IEEE, 2016.

[71] S Mika, G Ratsch, J Weston, B Scholkopf, and K Muller. Fisher Discriminant Analysis with Kernels. In *IEEE Signal Processing Society Workshop Neural Networks for Signal Processing*, 1999.

[72] A Milella and G Reina. Adaptive multi-sensor perception for driving automation in outdoor contexts. *International Journal of Advanced Robotic Systems*, 11, 2014.

[73] A Mohamed, G Dahl, and G Hinton. Deep belief networks for phone recognition. In *Nips workshop on deep learning for speech recognition and related applications*, volume 1, page 39, 2009.

[74] D Munoz, J Bagnell, and M Hebert. Co-Inference for Multi-Modal Scene Analysis. In *European Conference on Computer Vision*, 2012.

[75] K P Murphy. *Machine learning: a probabilistic perspective*. MIT press, 2012.

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

[77] J Nocedal and S Wright. Numerical optimization. Springer Science & Business Media, 2006.

[78] S Nowozin, C H Lampert, et al. Structured learning and prediction in computer vision. *Foundations and Trends® in Computer Graphics and Vision*, 6(3–4):185–365, 2011.

[79] G Pandey, J R McBride, and R M Eustice. Ford campus vision and lidar data set. *The International Journal of Robotics Research*, 30(13):1543–1552, 2011.

[80] C H Papadimitriou and K Steiglitz. *Combinatorial optimization: algorithms and complexity*. Courier Corporation, 1982.

[81] J Pearl. *Probabilistic reasoning in intelligent systems: networks of plausible inference*. Morgan Kaufmann, 2014.

[82] D Piotr. Piotr's computer vision matlab toolbox. `https://github.com/pdollar/toolbox`.

[83] N Qian. On the momentum term in gradient descent learning algorithms. *Neural networks*, 12(1):145–151, 1999.

[84] J R Quinlan. Generating production rules from decision trees. In *International Joint Conference on Artificial Intelligence*, volume 87, pages 304–307. Citeseer, 1987.

[85] Bogdan R and S Cousins. 3D is here: Point Cloud Library (PCL), May 9-13 2011.

[86] Louis B Rall. Automatic differentiation: Techniques and applications. 1981.

[87] P Ravikumar and J Lafferty. Quadratic programming relaxations for metric labeling and markov random field map estimation. In *Proceedings of the 23rd international conference on Machine learning*, pages 737–744. ACM, 2006.

[88] H Robbins and S Monro. A stochastic approximation method. *The annals of mathematical statistics*, pages 400–407, 1951.

[89] D E Rumelhart, G E Hinton, and R J Williams. Learning representations by back-propagating errors. *Cognitive modeling*, 5(3):1, 1988.

[90] Radu Bogdan Rusu. Semantic 3d object maps for everyday manipulation in human living environments. *KI-Künstliche Intelligenz*, 24(4):345–348, 2010.

[91] N Savinov, H Christian, M Pollefeys, and Z Eth. Discrete Optimization of Ray Potentials for Semantic 3D Reconstruction. 2015.

[92] T Schaul, S Zhang, and Y LeCun. No more pesky learning rates. *International Conference on Machine Learning*, 28:343–351, 2013.

[93] J Schlosser, C Chow, and Z Kira. Fusing lidar and images for pedestrian detection using convolutional neural networks. In *Robotics and Automation (ICRA), 2016 IEEE International Conference on*, pages 2198–2205. IEEE, 2016.

[94] M Schmidt. UGM: A Matlab toolbox for probabilistic undirected graphical models, 2007. URL `http://www.cs.ubc.ca/~schmidtm/Software/UGM.html`.

[95] M Schmidt, R Babanezhad, M Ahmed, A Defazio, A Clifton, and A Sarkar. Non-uniform stochastic average gradient method for training conditional random fields. *arXiv preprint arXiv:1504.04406*, 2015.

[96] N Schraudolph, J Yu, and S Günter. A stochastic quasi-newton method for online convex optimization. In *International Conference on Artificial Intelligence and Statistics*, pages 436–443, 2007.

[97] S Sengupta, E Greveson, A Shahrokni, and P Torr. Urban 3d semantic modelling using stereo vision. In *IEEE International Conference on Robotics & Automation*, 2013.

[98] J Shi and J Malik. Normalized Cuts and Image Segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):888–905, 2000.

[99] M I Shlezinger. Syntactic analysis of two-dimensional visual signals in the presence of noise. *Cybernetics and systems analysis*, 12(4):612–628, 1976.

[100] J Shotton, J Winn, C Rother, and A Criminisi. Textonboost: Joint appearance, shape and context modeling for multi-class object recognition and segmentation. In *European conference on computer vision*, pages 1–15. Springer, 2006.

[101] Gordon D Smith. *Numerical solution of partial differential equations: finite difference methods*. Oxford university press, 1985.

[102] FLIR Integrated Imaging Solutions. Ladybug3 1394b, 2016. URL `https://www.ptgrey.com/ladybug3-360-degree-firewire-spherical-camera-systems`.

[103] V Stoyanov and J Eisner. Minimum-risk training of approximate crf-based nlp systems. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 120–130. Association for Computational Linguistics, 2012.

[104] J Strom, A Richardson, and E Olson. Graph-Based Segmentation for Colored 3D Laser Point Clouds. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2010.

[105] P Sturgess, C Russell, S Sengupta, Y Bastanlar, W Clocksin, and P H S Torr. Joint Optimization for Object Class Segmentation and Dense Stereo Reconstruction. *International Journal of Computer Vision*, 100(2):122–133, 2012.

[106] D Tarlow, I Givoni, and R Zemel. HOP-MAP: Efficient Message Passing with High Order Potentials. In *International Conference on Artificial Intelligence and Statistics*, 2010.

[107] T Tieleman and G Hinton. Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude. *COURSERA: Neural Networks for Machine Learning*, 4(2), 2012.

[108] T Toyoda and O Hasegawa. Random field model for integration of local information and global information. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(8):1483–1489, 2008.

[109] Y Tsuboi, H Kashima, H Oda, S Mori, and Y Matsumoto. Training conditional random fields using incomplete annotations. In *Proceedings of the 22nd International Conference on Computational Linguistics-Volume 1*, pages 897–904. Association for Computational Linguistics, 2008.

[110] A Vedaldi and K Lenc. Matconvnet: Convolutional neural networks for matlab. In *Proceedings of the ACM International Conference on Multimedia*, 2015.

[111] J Verbeek and W Triggs. Scene segmentation with crfs learned from partially labeled images. In *Advances in Neural Information Processing Systems*, volume 20, pages 1553–1560. MIT Press, 2008.

[112] S Vijayanarasimhan and K Grauman. Active frame selection for label propagation in videos. In *European Conference on Computer Vision*, pages 496–509. Springer, 2012.

[113] M J Wainwright and M I Jordan. Graphical models, exponential families, and variational inference. *Foundations and Trends® in Machine Learning*, 1(1-2): 1–305, 2008.

[114] M Welling. Fisher linear discriminant analysis. *Department of Computer Science, University of Toronto*, 3:1–4, 2005.

[115] P Xu, F Davoine, J Bordes, Z Huijing, and T Denoeux. Information Fusion on Oversegmented Images : An Application for Urban Scene Understanding. In *International Conference on Machine Vision Applications*, 2013.

[116] P Xu, F Davoine, J Bordes, Z Huijing, and T Denœux. Multimodal Information Fusion for Urban Scene Understanding. *Machine Vision and Applications*, 27(3): 331–349, 2016.

[117] K Yoshida and S Tadokoro. *Field and Service Robotics: Results of the 8th International Conference*, volume 92. Springer, 2013.

[118] M Zeiler. Adadelta: an adaptive learning rate method. *arXiv preprint arXiv:1212.5701*, 2012.

[119] R Zhang, S Candra, K Vetter, and A Zakhor. Sensor Fusion for Semantic Segmentation of Urban Scenes. In *IEEE International Conference on Robotics & Automation*, 2015.

[120] Y Zhang and T Chen. Efficient inference for fully-connected CRFs with stationarity. *IEEE Conference on Computer Vision and Pattern Recognition*, 2012.

[121] Q Zhu, M Yeh, K T Cheng, and S Avidan. Fast human detection using a cascade of histograms of oriented gradients. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 2, pages 1491–1498. IEEE, 2006.