Semantic Mapping of Road Scenes

Sunando Sengupta

Thesis submitted in partial fulfilment of the requirements of the award of

Doctor of Philosophy

Oxford Brookes University

2014

Abstract

The problem of understanding road scenes has been on the fore-front in the computer vision community for the last couple of years. This enables autonomous systems to navigate and understand the surroundings in which it operates. It involves reconstructing the scene and estimating the objects present in it, such as 'vehicles', 'road', 'pavements' and 'buildings'. This thesis focusses on these aspects and proposes solutions to address them.

First, we propose a solution to generate a dense semantic map from multiple street-level images. This map can be imagined as the bird's eye view of the region with associated semantic labels for ten's of kilometres of street level data. We generate the overhead semantic view from street level images. This is in contrast to existing approaches using satellite/overhead imagery for classification of urban region, allowing us to produce a detailed semantic map for a large scale urban area. Then we describe a method to perform large scale dense 3D reconstruction of road scenes with associated semantic labels. Our method fuses the depth-maps in an online fashion, generated from the stereo pairs across time into a global 3D volume, in order to accommodate arbitrarily long image sequences. The object class labels estimated from the street level stereo image sequence are used to annotate the reconstructed volume. Then we exploit the scene structure in object class labelling by performing inference over the meshed representation of the scene. By performing labelling over the mesh we solve two issues: Firstly, images often have redundant information with multiple images describing the same scene. Solving these images separately is slow, where our method is approximately a magnitude faster in the inference stage compared to normal inference in the image domain. Secondly, often multiple images, even though they describe the same scene result in inconsistent labelling. By solving a single mesh, we remove the inconsistency of labelling across the images. Also our mesh based labelling takes into account of the object layout in the scene, which is often ambiguous in the image domain, thereby increasing the accuracy of object labelling. Finally, we perform labelling and structure computation through a hierarchical robust P^N Markov Random Field defined on voxels and super-voxels given by an octree. This allows us to infer the 3D structure and the object-class labels in a principled manner, through bounded approximate minimisation of a well defined and studied energy functional. In this thesis, we also introduce two object labelled datasets created from real world data. The 15 kilometre Yotta Labelled dataset consists of 8,000 images per camera view of the roadways of the United Kingdom with a subset of them annotated with object class labels and the second dataset is comprised of ground truth object labels for the publicly available KITTI dataset. Both the datasets are available publicly and we hope will be helpful to the vision research community.

Acknowledgements

I would like to thank my supervisors Professor Philip Torr and Professor David Duce for showing me the concepts of computer vision. Their guidance had helped me to think through the problems during the course of the PhD. I profoundly thank Phil, for his insight into the problems, his enthusiasm, and have learnt a lot professionally from him. I would like to thank Professor Gabriel Brostow and Professor Nigel Crook for examining my thesis and giving their valuable opinion about it.

It was a pleasure to work with some of the best people in the field during my PhD. I would like to thank all my collaborators, Paul Sturgess, Lubor Ladicky, Ali Shahrokni, Eric Greveson, Julien Valentin, Johnathan Warrell, Yalin Bastanlar, Ziming Zhang, Chris Russel and Professor William Clocksin, with whom I have enjoyed working a lot. The Brookes Vision lab would not have witnessed lively heated long discussions without Paul, Mike, Vibhav, Kyle, Ziming, and all the (ex)members of the lab, I thank them all. Finally I would like to thank Paul, Mike and Balasanjeevi for proofreading my thesis.

My time in Oxford was made special with the company of Eric Sommerlade, Jacek Czyz, Marco Cecotti, Mahesh P. and Nethra M.. A special thanks goes to Debati Ghosh for supporting me all the time. And finally, I would like to thank my parents and my brother, who never let me feel that I was 5000 miles away from home.

Oxford

September 2014

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Objective

Computer vision has been applied extensively towards the process of extracting information from images, which is essentially about *what* objects are present in the real world and *where* they are located. Humans and most of the other biological creatures use this capability to assimilate information about their surroundings from the images generated in their retina, enabling them further to take action in response to the events in their vicinity. This is essentially their visual perception [122]. The biological process which gives rise to this visual perception, selectively identifies and passes the information to the brain (e.g. optical lobe in the cerebral hemisphere of the human brain) for processing. Based on the stimuli the brain tries to determine the objects and their relative location and any response if deemed to be necessary. This problem of information capture, dissemination and processing have attracted psychologists, neurologists and more recently computer vision scientists to attempt a reasoning of the overall perception process [91, 122, 183]. The advent of the computer as an information processing machine provided an opportunity to recreate and analyse various models which can mimic the visual perception of the human brain. However the core problem of extracting efficiently and correctly the two factors: 'what' and 'where', from real world images still remains a considerable challenge.

This dissertation proposes techniques for performing semantic mapping and reconstruction of road scenes using computer vision techniques, focussing on the two main factors of what objects are present and where are they located. The thesis begins by proposing a method to generate a dense per-pixel semantic overhead view of a large scale urban region. Then, the method is extended to perform a dense semantically labelled 3D reconstruction of an outdoor scene using only visual data. This thesis, further investigates multiple representations of the structure, namely mesh based and voxel based, and proposes methods to perform accurate object labellings on them. Providing semantic information about the scene in a dense representation, aids autonomous robotic systems to identify the objects in its surroundings and provide an improved object class boundary estimate, required for robotic tasks such as navigation and manipulation.

1.2 Motivation

Understanding the visual perception is a daunting challenge due to the inherent ambiguity in image formation. An image corresponds to the world's projection in the retina, which acts as a stimulus to the sensory receptors. The light that contributes to the image consists of a conflation of several physical phenomena like illumination, reflectance, transmittance and a set of various other physical properties of the objects present in the scene. The objects in the scene are present in different scales/sizes, located spatially at varying distance from the observer, in different orientations, occluding each other and possibly having motion parallax. The final image is the combined effect of all these phenomenon. Such an example can be seen in Fig. 1.1. The image in the top shows a typical urban scene, which comprises humans, vehicles, buildings, road, pavement, etc. The human is closer to the camera, thereby is relatively larger in the image, while the vehicles appear small in the image because they are situated at a large distance from the camera. If we try to stretch the image such that the X axis represents the distance where the objects are located in the scene from the camera and the Y axis denotes approximately the object's actual height. Then we will observe that the human lies towards the left (near the camera) and is of low height while the building is located in the right (the far end from the viewpoint) and much taller. Our aim is to obtain this representation in which we can place the objects in their own scale at the position where they actually occur in the scene. Extracting these individual factors from the images is hard, and often referred as the inverse optics problem [89].

Vision scientists tried to explain the physical process associated with an image through a multitude of ways. One of the approaches to reason about things present in the image was a probabilistic framework where the pixels in the image are associated with an object from a predefined set of objects, based on a classifier response of the features computed around those pixels [158]. A set of training images is used to gather a prior knowledge for each object. The training set is used to learn a classifier which tries to establish a linkage between the image pixels and the real life object. Similarly, for perceiving the physical space, cues like perspective, occlusion and multiple views of the same scene are considered. It is observed by early landscape painters (Leonardo Da Vinci's study on binocular vision dated 16^{th} century [182]) that the objects closer to the person are larger and they have a dissimilar projection at each eye. It was observed by C. Wheatstone [183] that dissimilar perspective projection of the object subtended at each eye, results in the three dimensional perception in the human mind. This property of binocular vision or stereo was later on explored extensively in [122, 123].

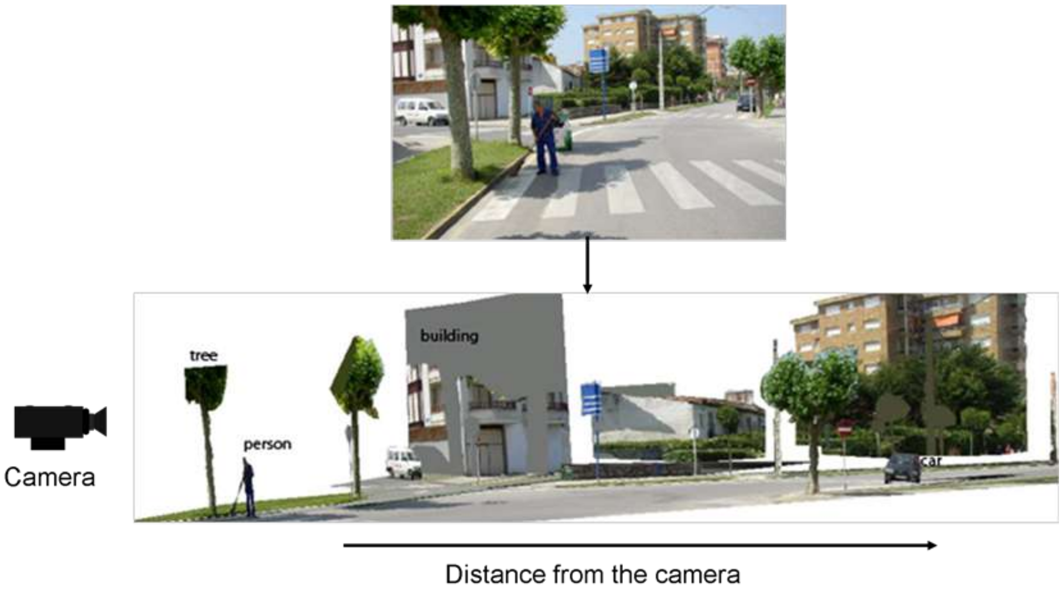

Figure 1.1: *A general description of a scene.* The image (top) describes a general road scene. The image below shows the objects in the relative distance from the viewing point. The person is closer to the camera, while the buildings and cars are farther away. Finding which object classes is present and their positions from the image, is a challenge which we address in this thesis. (Image courtesy: Antonio Torallba, http://6.869.csail.mit.edu/fa13/)

Currently, computer vision has come a long way in terms of its capability: from an explanation of binocular stereo through random-dot stereograms [122] to large working systems [22]. It is being used increasingly in our daily activities via personal devices like smartphones, entertainment consoles (e.g. Microsoft Kinect, PlayStation), health care devices, social care robots [121], assembly line robots in large industrial set-ups [144] and autonomous self-driving vehicles [172]. In all these examples computer vision is playing an increasing role through object recognition, detection, localisation and mapping. Specifically, in the context of self-driving vehicles, technology has progressed from navigating in a desert terrain [25] to navigate in a complex busy street scene [130, 172] and journey through intercontinental highways [21] (see Fig 1.2). It is necessary for these camera enabled driver-less cars to ascertain the drivable region by determining the free road space, handle obstacles like the pedestrians, vehicles, recognise the post/poles and simultaneously be aware of its current position in the road: thus requiring both object perception as well as location awareness. Similarly, for manipulating systems (e.g. robots in assembly lines) it becomes extremely important to identify the correct objects, and simultaneously get situational awareness. Another important application is to aid people with limited physical capability for navigation in the real world [84]. In all these above applications, the task of accurately estimating objects and their locations become primarily important, which is also the goal of this thesis.

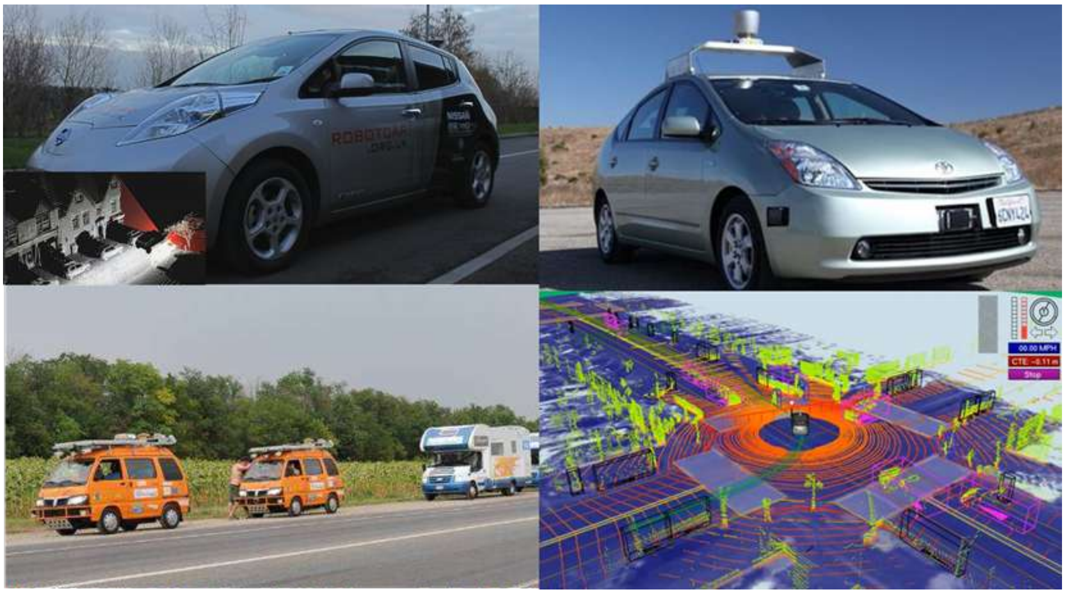

Figure 1.2: *Driverless Cars across the world.* (a) Shows the Oxford University Robot Car [130], which uses multiple sensors like cameras, radars and scanning lasers which enable them to pinpoint the location of the vehicle. (b) Google driverless car [172], one of the most successful autonomous vehicle project uses a multitude of laser range finders, video cameras and laser sensors which is used along prior detailed map information collected using manually driven vehicles. (c) Vehicles participating in the Vislab Intercontinental Autonomous challenge [180], where the challenge was to have a convoy of cars (six in total, four autonomous) to drive from Parma, (Italy) to Shanghai covering 1̃6000Kms, where the convoy is led by a manned control vehicle. Finally (d) shows a view of the laser scan images from a google autonomous vehicle. The laser returns give an approximate idea of the obstacles in the scene. In all these examples a heavy reliance of the laser scans is observed, which drives up the cost of the system. (Image courtesy: (a) Robot Car- http://mrg.robots.ox.ac.uk/robotcar/, (b) Disruptive tech predictions- http://www.dvice.com/archives/2012/05/which_of_these.php (c)Vislab- http://vislab.it/automotive/ and (d) This Is What Google's Self-Driving Car 'Sees' as It Makes a Turn http://mashable.com/2013/05/03/google-self-driving-car-sees/)

However, most of these autonomous systems [172], rely on multiple sensors such as stereo cameras, panoramic vision systems, laser scanners and global positioning systems (GPS), inertial measurement devices, accelerometer, gyroscope, etc. Such complex systems result in a high cost to manufacture and maintain. Ideally to achieve the goal of mass autonomous systems, a pure camera based system for structure determination and perception offer the advantage of being low in cost. Moreover, the images potentially contain a lot of detailed information which helps a vision based system have the additional advantage of being dense in nature in comparison to sparse laser returns from the range finders. This gives an additional motivation to target a system that can generate a dense semantically annotated scene only through visual imagery.

Over the last few years, considerable advances have been witnessed in the area of understanding an image or a sequence of images [23, 69, 111]. Often the computer vision tasks like image segmentation, object recognition, stereo matching are treated as a labelling

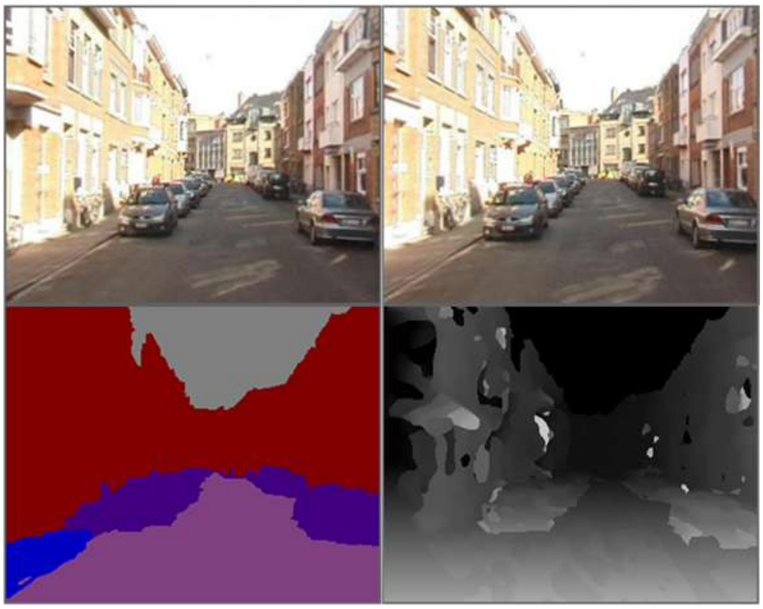

Figure 1.3: *Image Labelling problem*: The top row shows the images taken from a stereo camera mounted on the top of a vehicle, the bottom row shows the object class labelling and the disparity estimation from [111].

problem where the set of image pixels is grouped together, based on their semantics, and associated with a particular label. These labels may correspond to object labels such as road, car, sky and pavement, or depth labels (pixels grouped having similar depth from the camera). These labels are generally structured and are often conditionally dependent on each other. As a result, with increasing number of pixels (from hundreds of thousands up to tens of millions), the interdependency between them makes the problem of labelling quite hard. Most computer vision techniques model these interrelationships among the pixels through pairwise Conditional Random Fields (CRF) and such modelling approaches has proved to be effective for a challenging real world scene problems. Especially in the context of road scenes it was shown by Ladicky et. al in [111] that the relationship between object and depth can be successfully modelled through a CRF representation where both object class labelling and disparity estimation was performed simultaneously. Taking inspiration from this we show in this thesis how object labelling and depth estimation can be employed for applications like semantic mapping, reconstruction and semantic structure recovery.

1.3 Contributions

The main contributions of this thesis are summarized below. We will discuss the relevant contributions in detail at the end of every chapter, and present a consolidated summary in section 7.1.

Dense Semantic Mapping, Chapter 3 : We have presented a technique to generate a dense semantic overhead map (an overhead view or a bird's eye view of an urban region) from a sequence of street level images spanning ten's of kilometres. The proposed method is based on a two stage CRF which aggregates local semantic information into a global inference problem and outputs a map with semantic annotation. A semantically augmented map can add richness to the existing mapping applications [68, 94] through adding object class information. In most of the cases, the mapping applications are associated with overhead satellite imagery which is difficult for determining semantics. On the other hand street images contain detailed information at higher resolution, which can be exploited effectively. In our semantic map, a dense layout of object classes is generated as seen from an overhead view. An example of a semantic map is shown in Fig. 1.4(a).

Large Scale Semantic Reconstruction, Chapter 4 : The semantic mapping is extended in this chpter to generate a dense 3D semantic structure corresponding to the scene. The input to our system is a sequence of stereo images captured from a specialised vehicle at regular interval. The proposed method is capable of performing large scale dense volumetric reconstruction (up to kilometres of urban road scene), using a truncated signed distance representation, with semantic class labels. We perform an on-line volumetric update method to accommodate large tracts of street imagery to facilitate large scale dense reconstruction. A CRF is defined on the images to extract the object class information present in the images. These object labels are accumulated over time in a naive Bayes fashion and projected onto the surface of the reconstruction producing the dense labelled reconstruction. An example of semantic modelling of urban scene is shown in Fig. 1.4(b)

Mesh based inference on structure, Chapter 5 : While performing the object labelling in the image domain, the surface properties are not considered. Here, we have extended our dense semantic labelling of the urban scene to use the structure by performing the inference over the mesh. A CRF is defined on the scene structure which is denoted by a triangulated mesh. By performing labelling over the mesh we solve multiple issues. Firstly, images often contain redundant information where multiple images describe the same scene. As a result, image based labelling tend to be slow by solving the redundant information. Our proposed method is faster in the inference stage by approximately $25\times$ than an image based labelling method. Secondly, often multiple images, even though they describe the same scene result in inconsistent labelling. As we solve only a single mesh, we remove the inconsistency of labelling across images. Finally by virtue of working on meshes, which explicitly encodes

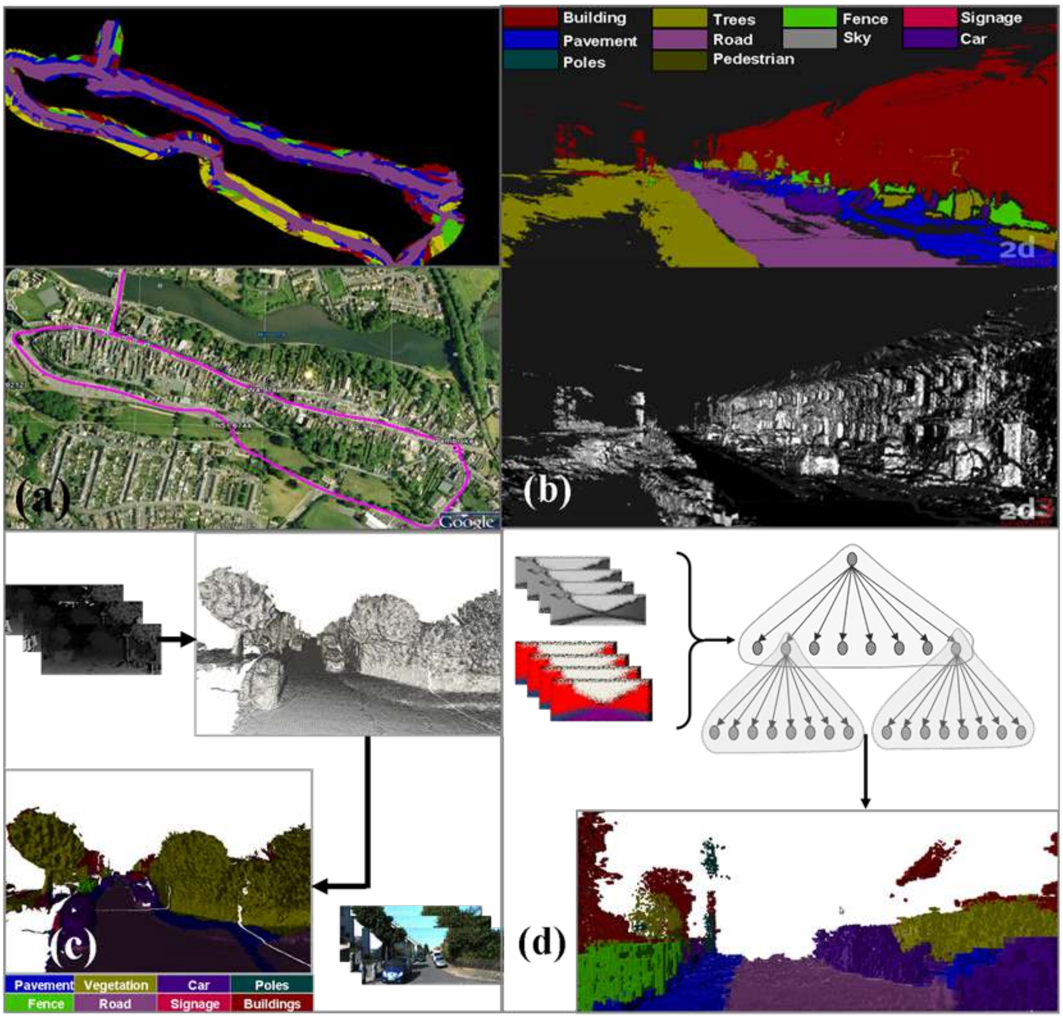

Figure 1.4: *Overview of work contained in the thesis*(a) *Semantic Map*: A bird's eye representation of the urban scene with dense object class labels, (b) *Labelled Reconstruction*: Dense reconstruction from stereo pairs, which is annotated with semantic labels. (c) *Mesh based inference*: A meshed structure is generated from a sequence of stereo image pairs. We compute appearance based scores for each mesh locations in an effective depth sensitive fashion and then perform inference over 3D surface, which enables us to exploit the layout of the objects present in the scene. This helps us in achieving significant speedup in the inference stage, with improved accuracy and consistency in results. (d) *Semantic Octomap*: Our framework can infer labelling and occupancy jointly in an unified manner. We introduce robust P^N constraints over the grouping of voxel volumes indexed through an octree graph.

the object layout information though connectivity, the proposed method is more accurate than object labelling in the image domain. An example of mesh based scene labelling is shown in Fig. 1.4(c).

Octree based hierarchical object and structure computation, Chapter 6 : Finally, we perform a hierarchical labelling and structure computation by introducing a multi-resolution 3D constraint towards the aim of unified scene understanding. The method assigns every voxel in the scene with an object class label or marks them free. The octree representation of the 3D allows voxels to be grouped naturally to form bigger sub-volumes. We propose a hierarchical robust P^N Markov Random Field, defined on the voxels and grouping of voxels in the 3D space, to ascertain the occupancy and semantics of the voxels (see Fig. 1.4(d)).

Datasets, Appendix A : Over the thesis, we have introduced, and made publicly available, two object labelled datasets created from real world data. The Yotta labelled dataset which is used for dense semantic mapping, is created from real world data covering the roadways of the United Kingdom captured by YottaDCL [42]. The capturing process is performed by a specialized vehicle, fitted with multiple cameras. Manual annotations of object class labels has been performed for a representative subset of those images for training and evaluation purpose. Additionally, an aerial image set comprised of satellite views acquired from Google Earth [67] is labelled which corresponds to a subset of the geographical area the region covered by our vehicle (approximately 7.5 kilometres). We have also created ground truth object labels for the publicly available KITTI dataset [62] for performing dense semantic reconstruction. Both the datasets are available online [1] [2] and will be helpful to the vision research community as a whole.

1.4 Publications

The publications related to the thesis are as follows

- Sunando Sengupta, Paul Sturgess, Lubor Ladicky, Philip H. S. Torr: Automatic dense visual semantic mapping from street-level imagery. IEEE/RSJ conference on Intelligent Robots and Systems 2012: pages: 857-862 (*Chapter 3*)

[1] available at http://cms.brookes.ac.uk/research/visiongroup/projects.php

[2] http://www.robots.ox.ac.uk/˜tvg/projects.php

- Sunando Sengupta, Eric Greveson, Ali Shahrokni, Philip H.S. Torr: Urban 3D Semantic Modelling Using Stereo Vision, IEEE International Conference on Robotics and Automation (ICRA), 2013 (*Chapter 4*)

- Sunando Sengupta*, Julien Valentin*, Jonathan Warrell, Ali Shahrokni, Philip H.S. Torr: Mesh Based Semantic Modelling for Indoor and Outdoor Scenes, IEEE conference on Computer Vision and Pattern Recognition, 2013. (**Joint first authors, Chapter 5.)*

  In this work, I was responsible for performing inference on the mesh and labelling the outdoor scene, while the contextual feature learning and indoor scene experiments, not included in this dissertation, was done by co-first author.

- Sunando Sengupta*, Julien Valentin*, Jonathan Warrell, Ali Shahrokni, Philip H.S. Torr: Mesh Based Semantic Modelling for Indoor and Outdoor Scenes. SUNw: Scene Understanding Workshop held in conjunction with IEEE Computer Vision and Pattern Recognition, 2013. (**Joint first authors, Invited paper*)

Other publications

- Ziming Zhang, Paul Sturgess, Sunando Sengupta, Nigel Crook, Philip H.S. Torr: Efficient discriminative learning of parametric nearest neighbor classifiers, IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2012

- Lubor Ladicky, Paul Sturgess, Christopher Russell, Sunando Sengupta, Yalin Bastanlar, William F. Clocksin, Philip H. S. Torr: Joint Optimization for Object Class Segmentation and Dense Stereo Reconstruction. International Journal of Computer Vision 100(2): pages 122-133, 2012 (*Invited paper, IJCV special issue. Chapter 1*)

- Lubor Ladicky, Paul Sturgess, Christopher Russell, Sunando Sengupta, Yalin Bastanlar, William F. Clocksin, Philip H. S. Torr: Joint Optimisation for Object Class Segmentation and Dense Stereo Reconstruction. BMVC 2010 (*BMVA Best science paper*)

1.4.1 Thesis outline

This thesis is comprised of 7 chapters. In the first two chapters we introduce the problem and the CRF notations that are used in this thesis extensively. In chapter 3, method for generating the overhead dense semantic map is described. In chapter 4, a large scale outdoor dense reconstruction with associated semantic annotations is demonstrated. In chapter 5, a mesh based inference for structure labelling is shown and in chapter 6, we have described the hierarchical octree based labelling and structure computation. The thesis finally concludes in chapter 7.

Chapter 2

Background

2.1 Image Labelling Problem

Various computer vision tasks, such as image segmentation [23, 164], stereo matching [111, 151], object recognition [158], can be seen as an image labelling task, where each individual pixel in the image is assigned to a particular object label category. Formally, the image labelling problem is to assign a label from a set of labels $\mathcal{L}$, to a set of discrete sites denoted by $\mathcal{V} = \{1, ..., n\}$ where n is the number of sites [118] corresponding to the image pixels. The label set $\mathcal{L} = \{l_1, l_2, ..., l_k\}$ can take a set of discrete values. For instance, they could be categories such as object class labels: 'vehicles', 'vegetation', and 'buildings', as shown in Fig.2.1. Alternately, they could be a discrete set taken from a continuous range, as encountered in depth/disparity estimation problems: depicted in Fig.2.2 where the image pixels are associated with disparity labels.

2.1.1 Markov Random Fields

Markov Random fields (MRF) provide a probabilistic framework to model the image labelling problem [118] effectively by incorporating the dependencies between the variables. Consider a set of random variables $\mathbf{X} = \{X_1, X_2, ..., X_n\}$, where each variable $X_i \in \mathbf{X}$ corresponds to the image pixel site $i \in \mathcal{V}$. The goal is to assign each random variable $X_i = x_i$ a label $l \in \mathcal{L}$. Any assignment of labels to the random variables is called a labelling, denoted as $\mathbf{x}$, and the total number of such possible configurations are exponentially large $\mathbf{L} = |\mathcal{L}|^n$. The neighbourhood of a random field is given by $\mathcal{N}$ which is comprised of the sets $\mathcal{N}_i, \forall i \in \mathcal{V}$, where $\mathcal{N}_i$ denotes the set of neighbours of the variable X_i in the random

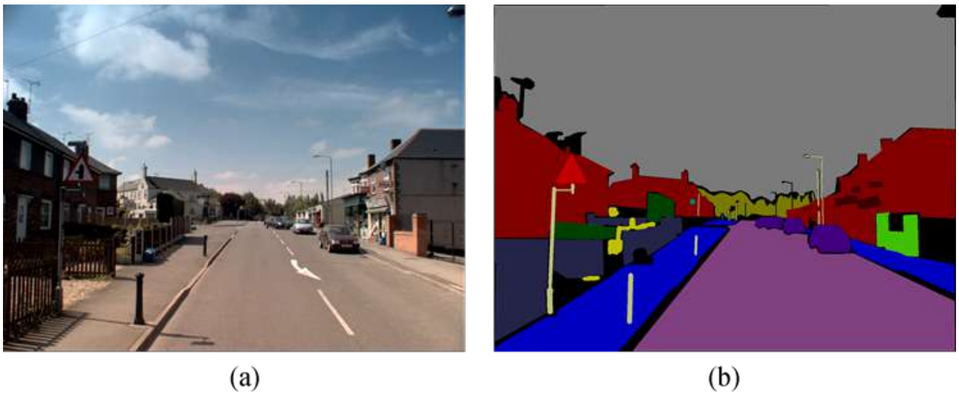

Figure 2.1: *Object class labelling*: An example image encountered in a normal road scene. Every pixel in the input image (a) is assigned to an object class (b) from the set of labels, namely car, road, vehicles and pavement, each denoted by a specific colour.

field. An example of the neighbourhood system can be seen as the 4-neighbourhood, i.e. every image pixel i is connected to its 4 immediate neighbours. A clique is defined as the set of random variables $\mathbf{X}_c \subset \mathbf{X}$ that are conditionally dependent on each other.

The random field is Markovian if it satisfies the positivity criterion ($Pr(\mathbf{x}) > 0, \forall \mathbf{x} \in \mathbf{X}$) and the Markovianity criterion ($Pr(x_i \mid x_j : j \in \{1, 2, ..n\} \setminus \{i\}) = Pr(x_i \mid \{x_j : j \in \mathcal{N}_i\})$). That is, the prior probability of a labelling to a random variable $X_i = x_i$, depends only upon its neighbouring random variables given by the set $\mathcal{N}_i$. A CRF can be viewed as an MRF globally conditioned on the data. It models the conditional probability of the labelling $\mathbf{x}$ given the data $\mathbf{D}$ assuming the validity of the Markovian property.

The conditional probability of the labelling $\mathbf{x}$ given the data $\mathbf{D}$, is given by

$$Pr(\mathbf{x} \mid \mathbf{D}) = \frac{1}{Z} \prod_{c \in \mathcal{C}} exp(-\psi_c(\mathbf{x}_c)), \tag{2.1.1}$$

where $\mathcal{C}$ is the set if cliques formed by grouping of elements in $\mathcal{V}$. The term $\psi_c(\mathbf{x}_c)$ is the potential function of the clique c where $\mathbf{x}_c = \{x_i : i \in c\}$. The number of elements in each clique $\mathbf{x}_c$ determines the order of the clique. For example, a clique of two elements represents two neighbouring random variables. The term Z is the normalizing constant also known as the partition function. The corresponding Gibbs energy [64, 118] is given as:

$$\begin{aligned} E(\mathbf{x}) &= -\log Pr(\mathbf{x} \mid D) - \log Z \\ &\propto \sum_{c \in \mathcal{C}} \psi_c(\mathbf{x}_c). \end{aligned} \tag{2.1.2}$$

The labelling of the random field will be obtained by maximizing the posterior probability. Thus the most probable labelling or the Maximum a Posteriori (MAP) labelling $\mathbf{x}^*$ is defined as:

$$\mathbf{x}^* = \underset{\mathbf{x} \in \mathbf{L}}{\operatorname{argmax}}\, Pr(\mathbf{x} \mid D) = \underset{\mathbf{x} \in \mathbf{L}}{\operatorname{argmin}} E(\mathbf{x}). \tag{2.1.3}$$

The MAP estimate of $\mathbf{x}^*$ is the labelling which maximises the posterior distribution $Pr(\mathbf{x} \mid D)$ or essentially minimises the energy $E(\mathbf{x})$ [72].

Pairwise CRFs. Most of the image labelling problems in computer vision are formulated as a pairwise CRF on the image pixels. For pairwise CRF, the energy function in Eqn. 2.1.2 can be rewritten as the sum of the unary and the pairwise potential terms:

$$E(\mathbf{x}) = \sum_{i \in \mathcal{V}} \psi_i(x_i) + \sum_{i \in \mathcal{V}, i \neq j, j \in \mathcal{N}_i} \psi_{ij}(x_i, x_j), \tag{2.1.4}$$

where $\mathcal{V} = \{1, 2, ..., n\}$, and $\mathcal{N}_i$ denotes the set of all the random variables forming the neighbourhood of X_i. The unary potential $\psi_i(x_i)$ gives the cost of assignment of a particular labelling x_i to the random variable X_i. The term $\psi_i(x_i)$ is typically computed from colour, texture and location features of the individual pixels and corresponding pre-learned models for each object class [20, 147, 158]. The pairwise term $\psi_{ij}(x_i, x_j)$, imposes a smoothness prior on the labelling, representing the cost of assigning the labels $X_i = x_i$ and $X_j = x_j$ over the neighbourhood $\mathcal{N}_i$. It encourages neighbouring pixels in the image to take the same label and penalises any inconsistency in the labels between x_i and x_j, taking the form of a Potts model:

$$\psi_{ij}(x_i, x_j) = \begin{cases} 0 & \text{if } x_i = x_j, \\ g(i,j) & \text{otherwise}, \end{cases} \tag{2.1.5}$$

where the function $g(i, j)$ is a contrast sensitive function depending on the difference of intensities of the two pixels in consideration for our case. This is typically defined as:

$$g(i, j) = \theta_p + \theta_v exp(-\theta_b ||I_i - I_j||^2), \tag{2.1.6}$$

where I_i and I_j are the colour vectors of the pixels i and j respectively in our experiments. θ_p, θ_v and θ_b are the parameters which are determined through the training/validation data.

Higher Order CRFs. There has been much interest in higher order CRFs' in the recent past. They have been successfully used to improve the results in problems such as image de-noising, restoration [112, 145], texture segmentation [101], object category segmentation [101]. The higher order potential is defined on the cliques comprising more than two elements in the set. These cliques correspond to image segments obtained by multiple unsupervised segmentations [101, 108]. As a result, they are able to model effectively higher contextual information in the image by capturing the fine details including texture and contours of the segments better than the pairwise potentials. The Gibbs energy of our higher order CRF is given by:

$$E(\mathbf{x}) = \sum_{i \in \mathcal{V}} \psi_i(x_i) + \sum_{i \in \mathcal{V}, j \in \mathcal{N}_i} \psi_{ij}(x_i, x_j) + \sum_{c \in \mathcal{C}} \psi_c(\mathbf{x}_c), \tag{2.1.7}$$

where $\mathcal{C}$ denotes the set of all the higher order cliques, ψ_c refers to the potential defined on them and $\mathbf{x}_c$ is the set of all elements in clique c. The cliques are generally obtained through multiple unsupervised segmentation [101]. The hierarchical potential can take the form of a robust P^N model defined as:

$$\psi_c(\mathbf{x}_c) = \min_{l \in \mathcal{L}}(\gamma_c^{max}, \gamma_c^l + k_c^l N_c^l(\mathbf{x}_c)), \tag{2.1.8}$$

where $\gamma_c^l \leq \gamma_c^{max}, \forall l \in \mathcal{L}$ and $N_c^l(\mathbf{x}_c) = \sum_{x_i \in \mathbf{x}_c} \delta(x_i \neq l)$ denotes the number of inconsistent pixels with the label l. This framework allows to integrate multiple image quantisations in a principled manner. To increase the expressiveness of the hierarchical model, conditional dependency between the segments was modelled in [108], where segment level appearance cost and pairwise dependencies between neighbouring cliques was introduced. For further details we refer the reader to [108]. The final labelling is obtained by finding the lowest energy configuration of the CRF. The energy (Eqn. 2.1.4) can be minimised using approximate graph cut based methods such as α-expansion [17, 101]. We will now briefly explain the inference methods.

2.1.2 Graph cut based inference

Finding a computationally feasible MAP estimate for the labelling problem has been of considerable interest over the last four decades. Geman and Geman [65] had proposed a simulated annealing method with gradual temperature reduction to generate the MAP estimate for low level vision tasks such as image deionising. Later on, iterated conditional modes was proposed by Besag [11] to solve the same problem. It was first shown by Greig et. al. [72] that the graph cut algorithm of Ford-Fulkerson for finding the maximum flow in a capacity limited network can be used to evaluate the MAP estimate for a binary image. It was shown later in [15, 16, 168] that efficient graph cut based algorithm with low run-time complexity can be employed for obtaining an optimal solution for binary label problems and an approximate solution with good optimality guarantees [17, 100, 103] for multi-label problems such as image segmentation and stereo.

Move making algorithms Boykov et al. proposed efficient graph cut based α-expansion and $\alpha\beta$-swap algorithms [17, 168] for solving the energy minimisation problem. They belong to the class of move making algorithms. These algorithms work iteratively, by starting from an initial labelling $\mathbf{x}$ and make a series of moves (label changes) which lower the total energy. Convergence is achieved when the energy cannot be decreased further. At each step, the algorithms search in a move space to find the optimal move, which is the one that decreases the energy of the labelling by maximum amount. The α-expansion algorithm finds an approximate map estimate by solving a series of st-mincut problems. At each step, it considers a label $\alpha \in \mathcal{L}$, and allows all the random variables to either retain their current label or change to α. For the $\alpha\beta$-swap algorithm, each move operates on a pair of labels, say α and β, and the move allows to exchange the labels for the random variables which are assigned the labels α and β. Both of the move making algorithms ensure that the move results in a labelling with lower energy than the previous labelling. For further details we refer the reader to [17, 102].

2.2 Dense 3D reconstruction

In this section we first discuss the problem of finding 3D information from a pair of images and then discuss about extending them to a sequence of images.

2.2.1 Stereo

Finding the depth of objects in the scene has been a core subject of research from the early nineteenth century [183], where depth perception in humans was shown to be related to the differing positions of the object in the two eyes. Later on the stereo problem continued to be researched through the works of Grimson [74], Marr and Poggio [123], Baker and Binford [8] etc. Most of these cases showed how edge information in the images can be used for obtaining matches in the left and right images to further reason about their depths.

The core of the stereo problem is the correspondence problem, where given two images of the same real world scene, a pixel in one image needs to be matched to a corresponding pixel in the second image, which describes the same scene element. When the two images differ in horizontal shift of the view points (thereby mimicking the binocular vision) the coordinate shift between the corresponding pixels is known as *disparity*. Knowledge of disparity enables us to perceive the motion between the images and also has a direct relationship with 3D depth [151]. The objects in the scene closer to the camera have higher values of disparity, decreasing uniformly with the depth. The resultant disparity map for all the matching pixels in the image gives a 3D representation of the scene. An example of stereo matching is shown in Fig. 2.2.

Traditionally, to compute disparity between the pair of images, each pixel in first image needs to be matched to another pixel in the second image. The search process is simplified to one dimension if the images are aligned so that the pixels in the right image is shifted in the

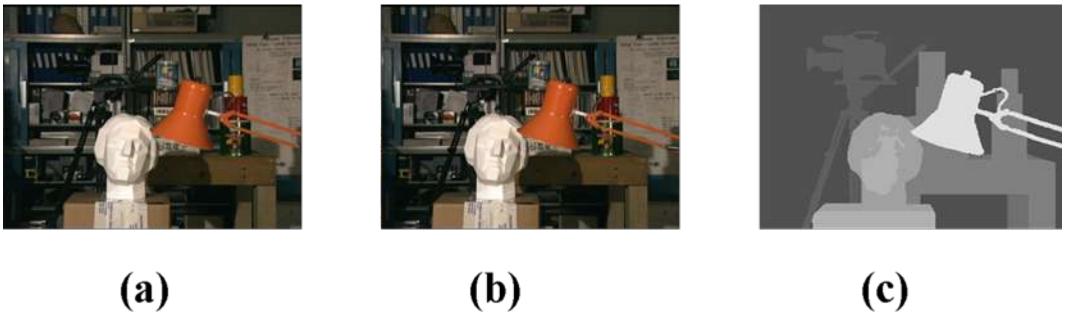

Figure 2.2: *Stereo disparity estimation*: The input images ((a) & (b)) are the ‘Tsukuba’ stereo pairs from Middlebury stereo dataset [151]. Pixels on the left image are matched to the right image ones along the horizontal scan line to generate the disparity image. (c) Shows is the disparity ground truth.

horizontal direction from the left image. This process is known as image rectification which ensures that the pixel correspondence can be determined by searching along the horizontal line in the other image, also known as the scan line. This is shown in Fig. 2.2 where (a) and (b) depict the rectified images. Then the disparity for each pixel in the reference image is computed by comparing the sum of absolute differences (*ssd*) for each pixel in the scan line by considering an image patch centred around the pixel of importance. Evaluating the *ssd* for all the pixels along the scan line gives a cost volume, where the minimum cost corresponds to the true disparity. However, such methods are subject to the noise halo effect [37,151]. To overcome these problems, stereo matching has been has been formulated as a labelling problem in an energy minimisation framework similar to eqn. 2.1.4. The label set corresponds to set of the permissible disparity labels in this case [13, 111]. The image graph comprises nodes corresponding to the image pixels and the edges are given by the nodes 4/8-neighbourhood. The unary score corresponds to the pixel similarity value [12] often measured over a window of $n \times n$ centred around the pixel of interest. The pairwise term encourages the neighbouring pixels in the image to take similar disparity values. Often it is a function depending on the difference between the disparity values of the neighbouring pixels taking the form of a linear truncated model [111]. Similarly, other methods like dynamic programming based approach has been used in [37], filtering the cost volume using a weighted box filter such that depth discontinuity in the disparity image aligns with the edge discontinuity [143] and using image patches of various sizes at object boundaries in [85]. For more details we refer the reader to [169].

2.2.2 Dense 3D reconstruction from image sequences

Recovering 3D from a set of images of a scene has been a long standing research area in computer vision and graphics, and has witnessed multiple approaches in an attempt to solve it. In computer graphics the focus was in generating high quality 3D models of objects in the form of polygonal meshes with detailed information like texture maps, surface normals, reflectance properties, etc. They are mainly used for rendering purposes. However, in computer vision, the focus has been to recover the scene geometry from a set of images [152]. Often these image sources are independent and are taken from the internet [1, 160]. Finding the 3D location from multiple photographs can be done via triangulation if the camera poses are known. Conversely, the camera poses can be determined if the actual 3D location of the points are known. Retrieving the camera pose and recovering the 3D world is also known as Simultaneous Localisation and Mapping (SLAM). To solve both problems, feature matches are established over multiple images and optimised along with the camera pose in a bundle adjustment scheme [2].

For image sequences, estimating the camera pose and the 3D world map, filtering techniques like Extended Kalman Filters have been effectively used [5] in the past. In [41] measurements from image sequences were fused by updating probability estimates for camera extrinsics using Extended Kalman filters. In this work, the tracking and mapping were performed together for all the frames. This was extended by Klien and Murray [99] who proposed a technique employing parallel threads for tracking and mapping, where tracking was performed for all the frames and the mapping was performed using bundle adjustment for selected key frames.

However, SLAM based approaches are used generally to obtain a sparse reconstruction. For dense reconstruction, it is required to estimate the surface information. Early works for surface estimation included methods like space carving [107], voxel based space representation [153], level set based methods [116] and visual hull based methods [32]. Most of these methods follow a two-step approach [181]: first a local depth estimates are obtained using a stereo pair or a group of images followed by fusion of these local depths. The aim of the fusion process is to generate a watertight surface model. The first stage is generally correspondence based stereo estimation (section 2.2.1). The second step, also known as the data fusion step, has been attempted in a multitude of ways: one approach is to use an energy minimization framework (discussed in section 2.1.1) with graph cut based inference [60, 66, 181]. Other methods for the depth data fusion included level-sets based methods [54, 116], deformable models [47, 59] or surface growing methods [76]. Recently, Newcombe et al. [129] proposed a system to generate a dense 3D reconstruction using a handheld RGB-D camera in real-time for a small work space. The dense model is generated by fusing overlapping depth estimates into a volumetric representation using a truncated signed distance function. The final surface is obtained by determining the zero level set in the volume. We follow this approach closely and take it further for outdoor scenes.

2.3 Discussion

In this chapter, we have discussed several existing techniques in computer vision for the task of image labelling; we have considered how the labelling problem can be is equivalent to an energy minimization problem, which can be solved efficiently using graph cut based inference algorithms. Similarly, we have discussed the general 3D reconstruction problem; both sparse and dense, from a set of images. The problem is particularly important for general scene understanding to aid intelligent autonomous systems. To this end, in the next chapter, we will begin exploring how object labelling of individual images in a sequence can be combined to generate a labelled semantic map corresponding to a large scale urban region.

Chapter 3

Dense Semantic Map from Street-Level Imagery

3.1 Introduction

In this chapter, we introduce a computer vision based system to generate a *dense visual semantic map* from a sequence of street level images. We define a semantic map as an overhead, or bird's eye view of a region with associated semantic object labels, such as car, road and pavement. Visual data have been exploited for scene understanding, which is further used for higher level robotic activities. In most cases, robotic platforms are equipped with multiple sensors like cameras for visual cues, laser range scanners for geometric cues, odometry devices for tracking motion and global positioning systems (GPS). The data from these sensors are used for tasks like localisation, mapping, recognition of objects [110, 139] in the scene and navigation in a structured environment [184]. Significant progress has been made to produce the map of the mobile robot's workspace [5, 131]. These maps have rich geometric representation, but lack semantic labels associated with the scene. Similarly, recognising objects in an urban outdoor scene have reached maturity in recent works [111, 185, 187]. In this chapter, we address the issue of augmenting semantics through object class labels in the map to aid autonomous systems where it operates. For example, such a map can easily provide a clue to an intelligent vehicle's navigation by determining drivable regions for autonomous vehicles [39]. Such semantic information can be used to add richness to existing mapping software like Google Earth [68] or Nokia Maps [94]. Moreover, this is particularly useful for asset marking companies (e.g. Yotta [42]) wishing to provide an automatic annotation of street assets (such as street light, drain or road sign) to local authorities.

Incorporating semantics to enhance the environment model via the addition of object class information from range data has been recently shown in [46]. Many LIDAR/range-data based approaches have been demonstrated for perception [82]. However, such systems based on 3D point clouds [28, 45, 124, 131, 132] are sparse in nature and often lack the details for accurate boundary detections. On the other hand, images obtained from devices at ground level are dense and contain a wealth of information allowing us to classify all the individual pixels in the image rather than a sparse set of distance measurements from a laser scan. Another advantage of using visual data is that it can be captured at high frequency and resolution at low cost and it is not adversely affected by surface types, relative to range data. However the images are inherently local in nature. This motivates us to construct a system that reliably aggregates semantic information captured by local sensors into a global map.

A semantic map could be created by labelling the satellite imagery of large areas directly. But this type of data may lack the details that are required for classifying objects of interest. Moreover, it cannot be easily updated in an online fashion.

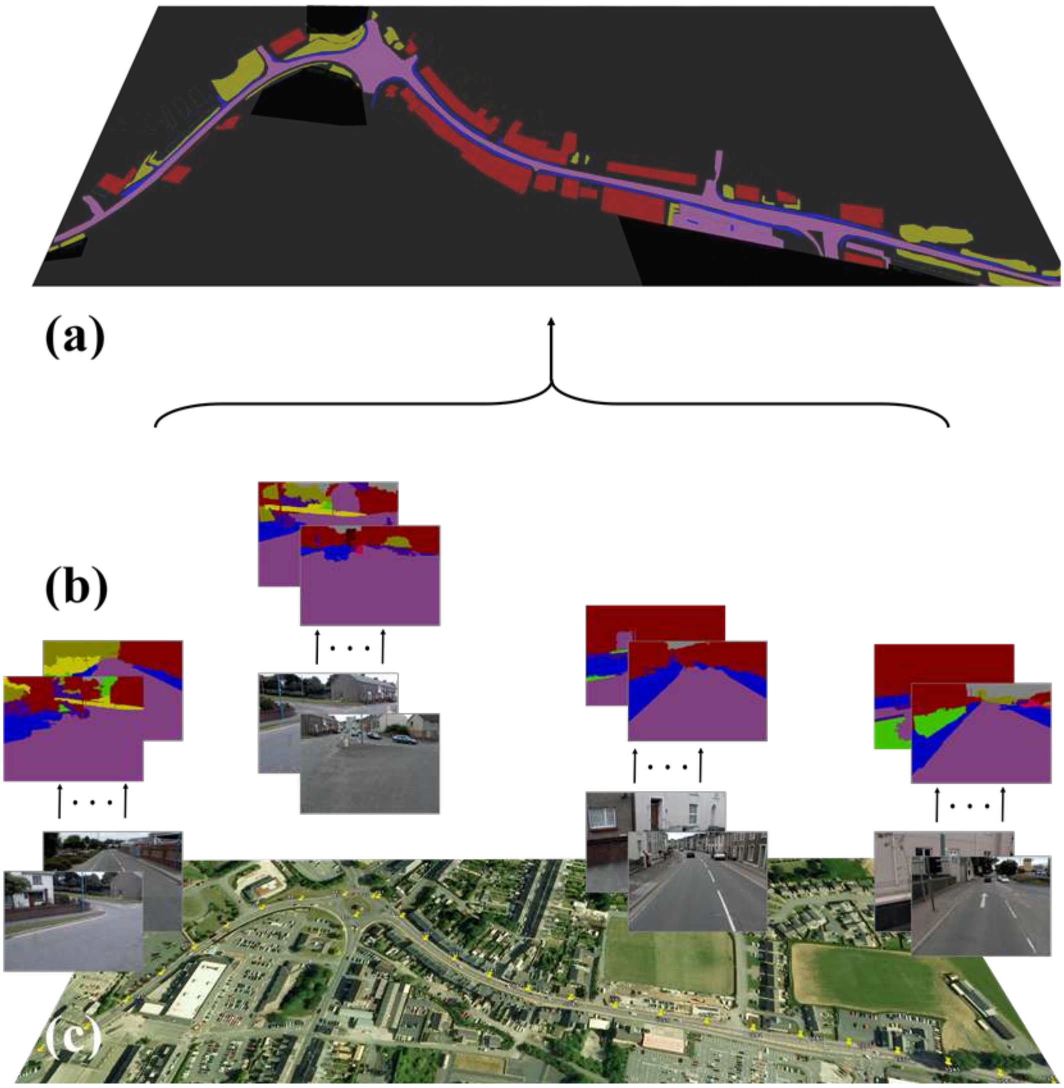

Figure 3.1: *Dense Semantic Mapping overview.* The semantic map (a) is generated from the street level images taken by a vehicle moving in the city. (b) Shows the objects class segmentations from the street level images. These object class information is integrated to generate the Dense Semantic Map. A corresponding overhead view from Google earth is shown below in (c). Best viewed in colour.

We formulate the problem of generating a semantic map by using two layers of conditional random fields (CRF). The first is used to model the semantic image segmentation of the street view imagery treating each image independently. The outputs of this stage are then aggregated over many frames providing a robust object class hypothesis for our semantic map, that is the second random field defined over a ground plane. Each image is related by a simple, yet effective, geometrical function that back projects a region from the street view image into the overhead ground plane map. This is in contrast to existing approaches which use satellite/overhead imagery for classification of urban region [36, 126], allowing us to produce a detailed semantic map for large scale urban area. Moreover by virtue of operating on the street level images, our method can scale from mapping hundreds of metres to tens of kilometres. An example semantic map is shown in Fig. 3.1, where the map is shown along with the Google Earth image of the corresponding region with associated object class labels. For evaluation of our method, we introduce, and make publicly available, a new dataset created from real world data covering the roadways of the United Kingdom captured by YottaDCL [42] using a specialised vehicle, fitted with multiple cameras. The vehicle also has GPS and odometry devices so that its location and camera tracks can be determined accurately. Our qualitative evaluation is performed on an area that covers 14.8 km of varying roadways. We have hand labelled a relatively small, but representative set of these images with per-pixel labels for training and quantitative evaluation. All the views of our 14.8km track are publicly available along with the annotation data[1]. This type of data along with our formulation allows us to aggregate semantic information over many frames, providing a robust classifier. We hope that this stimulates further work in this area. A sample dense semantic map generated by our system is shown in Fig. 3.2 along with the associated Google Earth [67] satellite view and our street-level images. The vehicle track is shown in magenta in the google earth image. The street level images are shown along with the semantic map in the place where they have been captured.

[1] Available from http://www.robots.ox.ac.uk/˜tvg/projects.php

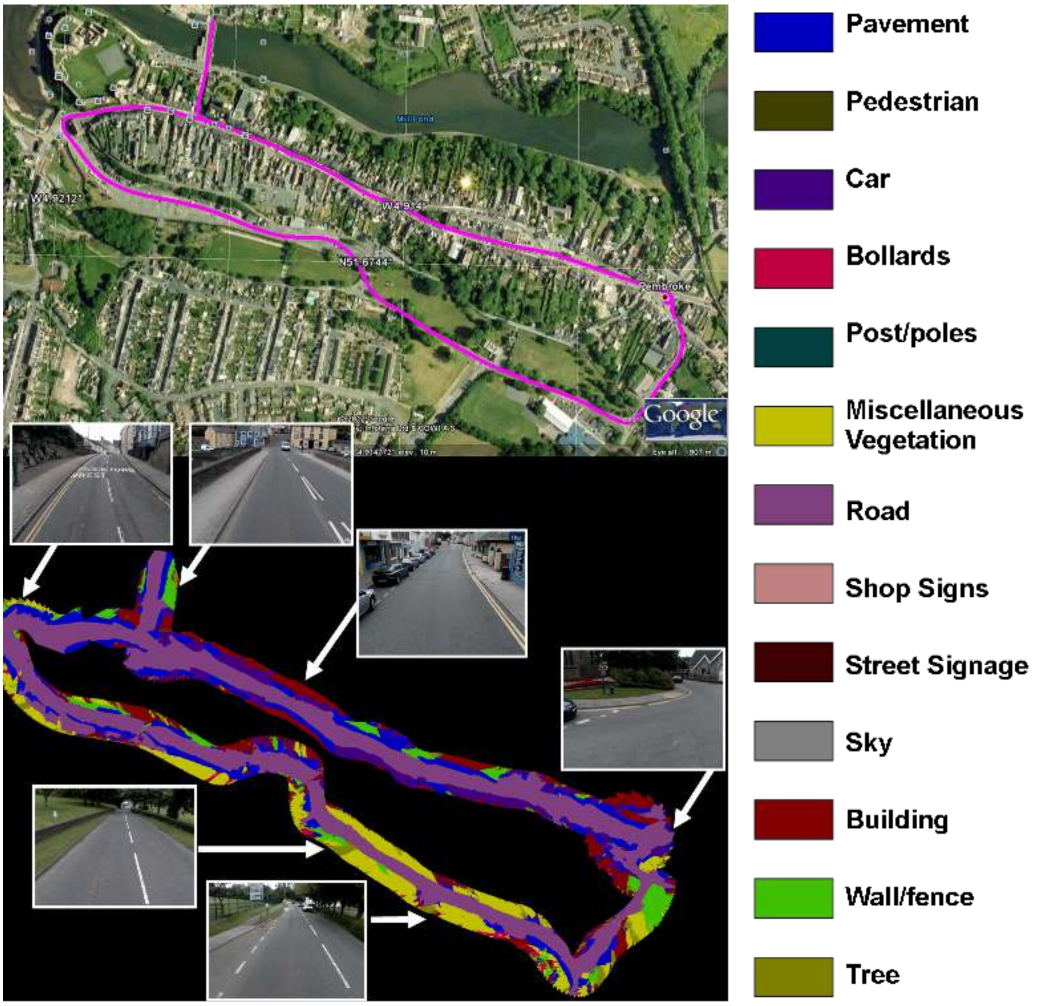

Figure 3.2: *Dense Semantic Mapping generated from our system.* Top: Visualization of the route taken by a specialized van to capture this street level imagery that we use to evaluate our method. The Semantic map (bottom) is shown for the corresponding track along with some example street view images. The palette of colours shows the corresponding class labels. This image is best viewed in colour.

3.1.1 Outline of the Chapter

In the following §3.2 we overview the related literature. Our method for dense semantic mapping is presented in §3.3. We introduce a new large scale road scene dataset in §3.4.1. The qualitative and the quantitative results of our method are shown in section 3.4.2. Summary and discussion are provided in §3.5.

3.2 Related Work

Over the last decade, there has been considerable research on the problem of Simultaneous Localisation and Mapping (SLAM) where the current location of the robot and the general model of the environment is being built simultaneously [5, 7]. However, as the research became mature, the limitations of a map which is void of meaningful semantic interpretation became evident [139]. This has led to a plethora of work in recent years, where information from multiple sensors is used to perform some sort of object classification with the aid of computer vision techniques, thereby improving the map.

Over the recent years, various computer vision techniques have been effectively used to perform the task of object recognition [138, 158], scene classification [113] and object class segmentation [108, 157]. In most of these cases, the problem is posed as a dense labelling problem where every image pixel is assigned a particular class label. In [158], it was shown that incorporating context, shape and appearance in a CRF model was effective for visual recognition and segmentation tasks. Of particular importance to our work presented in this chapter is [108] where it was shown that context can be learnt efficiently via multiple quantisation of image space through a hierarchy of superpixels.

In the context of road scenes, sequences of images taken from moving vehicles provide a rich source of information through high resolution imagery taken at regular intervals. Vision techniques applied to individual images were extended to solve the task of segmentation in the image sequence [23, 114]. In [23], semantic labelling was performed by taking cues from 3D point clouds derived from ego-motion. Similarly, image sequences from calibrated stereo rigs on a moving vehicle were used for 2D object detection, 3D localisation, and tracking in [114]. They tried to generate an overhead view of the region, but the map view was inherently sparse as only point based features were present in the map.

On the other hand, in the robotics domain, multiple sensor modalities like imaging devices along with laser range finders are used for interpreting the scene. In [125], a combined image and laser based system was used to detect obstacles in the road. Visual information from robotic platforms have also been used to determine the drivable regions for autonomous vehicles in [39, 77] for the winning entry of 2005 DARPA grand challenge [173]. It was shown how visual data can be used in conjunction with data from various other modalities to increase the range of the drivable region [39, 77]. This work however was limited in determining the road region and was not applicable to more complex urban scenarios. In [35], a topographical map was generated using a sequence of stereo images which attempted to build an overhead view, but did not consider a per-pixel semantic labelling of the map. Similar approaches have been used for mobile indoor robot scene understanding in [132]. In [139, 140], Posner et al. use overlapping visual information from images for each corresponding laser range data to classify surface types such as concrete, grass, brick, or pavement in an outdoor environment. However, they consider each scans independently. Other works included [176] where laser returns were classified using Associative Markov Networks for outdoor building scenes. The most related work to ours is [46] where a CRF based framework is used to classify a sequence of 3D scans over time and generate a semantic map. However, their method produces a sparse map, as it ignores most of the image data not overlapping with the laser points.

In this chapter, we propose a method to perform a dense per-pixel semantic map from multi-view street-level imagery. We use the street level images as a local source of information and exploit the local context through a CRF defined on each individual images. The semantics associated with each image are linked globally via the second ground plane CRF. In our work, we associate every ground plane pixel with an object class label giving us a dense labelling of the ground plane making this work unique to previous sparse laser based temporal scene classification [46, 139]. Also by defining multiple CRF's, we are able to harness the temporal as well as the local information effectively.

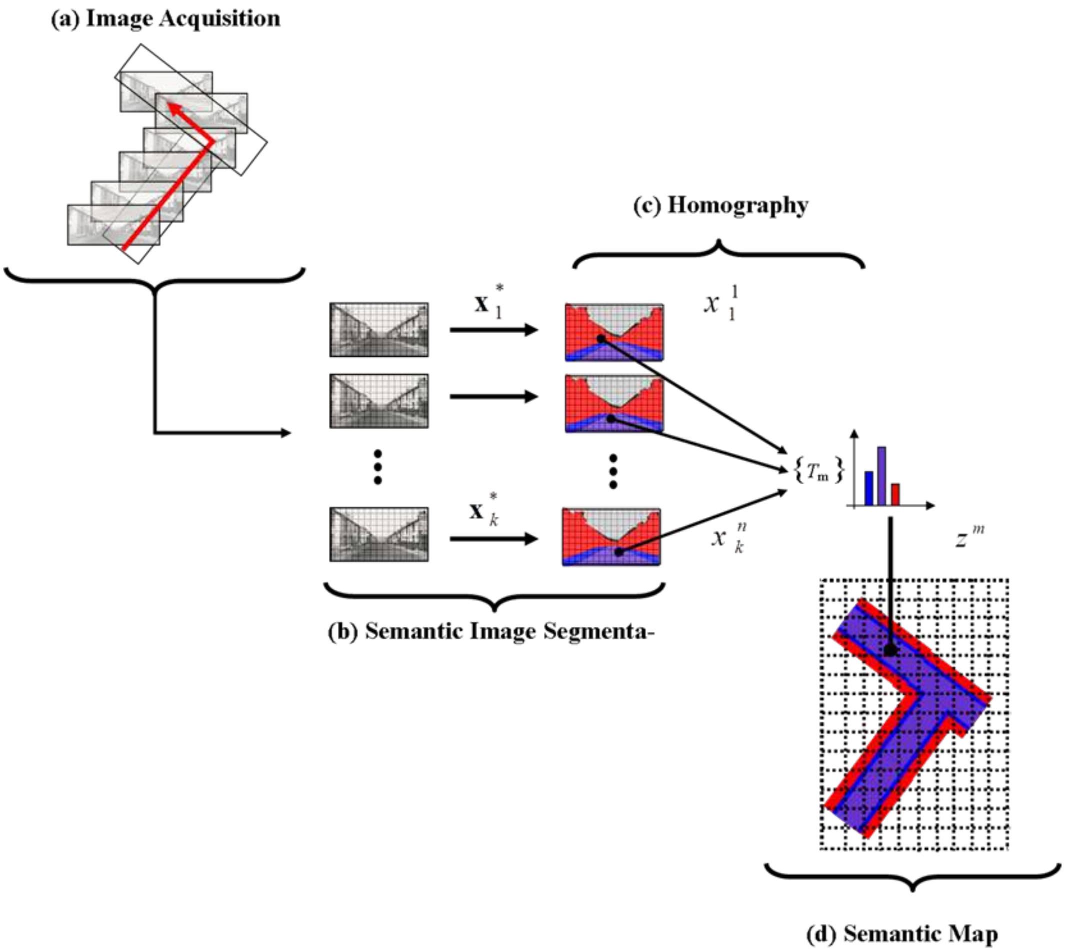

Figure 3.3: *Overview of our Algorithm* (a) Each image is treated independently and each pixel in the image is classified, giving a labelling $\mathbf{x}^*$ (§3.3.1). (b) A homography gives the relationship between the pixel on the ground plane to a pixel in a sub-set of the images, where the ground pixel is visible (§3.3.2). T_m denotes the set of the image pixels that correspond to the ground plane pixel z_m. A histogram of labels are generated from the pixel set which is used to obtained the unary measure for the ground plane pixel. (c) The CRF produces a labelling of the ground plane, or a semantic map (§3.3.3).

3.3 CRF Model for Dense Semantic Mapping

We model the problem of dense visual semantic mapping using two distinct conditional random fields (CRF). A graphical overview of the proposed system is shown in Fig. 3.3. The first CRF works with a sequence of images taken from synchronized cameras on the vehicle (Fig. 3.3(a)). The object class information present in the local images is extracted (Fig. 3.3(b)) and used to update the second CRF which is defined over the ground plane. The two layers are linked via a homography (Fig. 3.3(c)). The output of the final CRF is the semantic map. The two stage process enables us to model spatial contextual relations in both the image domain as well as on the ground plane. We chose to use CRFs as promising results have been demonstrated on street-level imagery [111, 164].

In this section we describe the street level image segmentation framework in (§3.3.1) and the ground plane mapping framework in (§3.3.3). The homography linking the local street images and the ground plane is described in (§3.3.2).

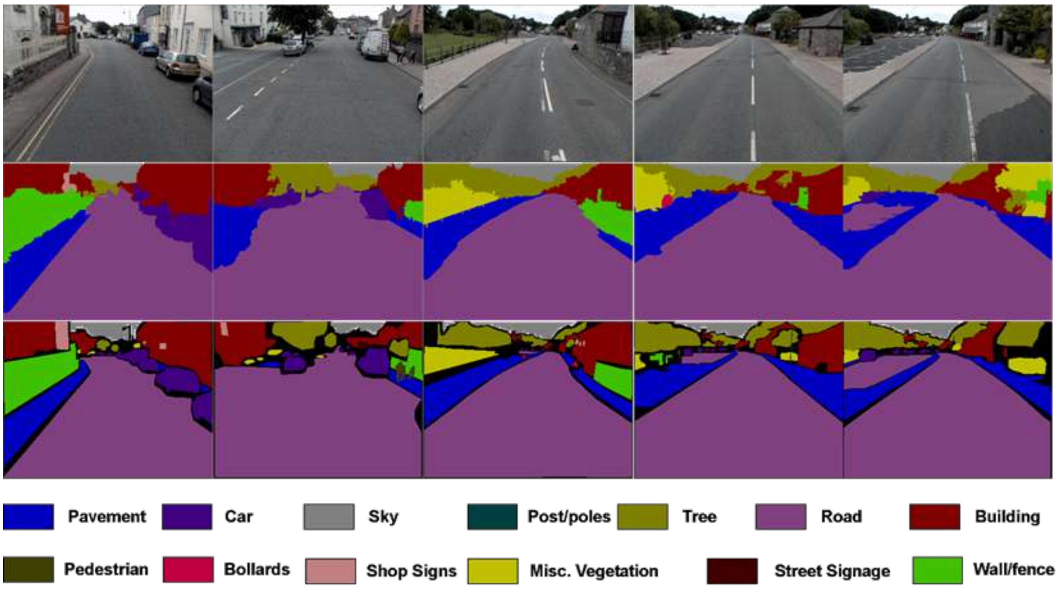

Figure 3.4: *Semantic Image Segmentation*: The top row shows the input street-level images followed by the output of our first CRF model. The last row shows the corresponding ground truth for the images. This image is best viewed in colour.

3.3.1 Street level Image Segmentation

For this part of our semantic mapping system we use an existing CRF based computer vision system [108] that is competitive to state-of-the art for general semantic image segmentation [52] and leading the field in street scene segmentation within the computer vision community [164]. We give a brief overview of their method here with respect to our application.

Consider a set of discrete random variables $\mathbf{X} = \{X_1, X_2, \ldots, X_n\}$, where each variable $X_i \in \mathbf{X}$ takes a value from a pre-defined label set $\mathcal{L} = \{l_1, l_2, \ldots, l_k\}$. Our label set is $\mathcal{L}$ = {pavement, building, road, vehicle, tree, shop sign, street-bollard, misc-vegetation, pedestrian, wall-fence, sky, misc-pole, street-sign}. The random field is defined over a lattice $\mathcal{V} = \{1, 2, \ldots, N\}$, where each lattice point corresponds to an image pixel, $i \in \mathcal{V}$, which is associated with the corresponding random variable X_i. Let $\mathcal{N}$ be the neighbourhood system of the random field defined by sets $\mathcal{N}_i, \forall i \in \mathcal{V}$, where $\mathcal{N}_i$ denotes the set of all neighbours (usually the 4 or 8 nearest pixels) of the variable X_i. A clique c is defined as a set of random variables $\mathbf{X}_c$ which are conditionally dependent on each other. This framework combines features and classifiers at different levels of the hierarchy (pixels and superpixels), resulting a strong hypothesis for a particular random variable to take any specific object class label. The Gibbs energy [118] for the street-level image is

$$E^S(\mathbf{x}) = \sum_{i \in \mathcal{V}} \psi_i^S(x_i) + \sum_{i \in \mathcal{V}, j \in \mathcal{N}_i, i \neq j} \psi_{ij}^S(x_i, x_j) + \sum_{c \in \mathcal{C}} \psi_c^S(\mathbf{x}_c). \tag{3.3.1}$$

Unary potential The unary potential ψ_i^S describes the cost of a single pixel taking a particular label. It is learnt as the multiple-feature variant [108] of the TextonBoost algorithm [158].

Pairwise potential The 4-neighbourhood pairwise term ψ_{ij}^S takes the form of a contrast sensitive Potts model:

$$\psi_{ij}^S(x_i, x_j) = \begin{cases} 0 & \text{if } x_i = x_j, \\ g(i,j) & \text{otherwise,} \end{cases} \tag{3.3.2}$$

where, the function $g(i,j)$ is an edge feature based on the difference in the colours of neighbouring pixels [15], defined as:

$$g(i,j) = \theta_p + \theta_v \exp(-\theta_\beta ||I_i - I_j||_2^2), \tag{3.3.3}$$

where I_i and I_j are the colour vectors of pixels i and j respectively. θ_p, θ_v, $\theta_\beta \geq 0$ are model parameters which can be set by cross validation. Here we use the same parameters as [164] as we have a limited amount of labelled data and our street views are similar in style to theirs. The intensity-based pairwise potential at the pixel level induces smoothness of the solution encouraging neighbouring pixels to take the same label.

Higher Order Potential The higher order term $\psi_c^S(\mathbf{x}_c)$ describes potentials defined over overlapping superpixels obtained using multiple unsupervised meanshift segmentations [34]. The potential models the likelihood of pixels in a given segment taking similar label and penalizes partial inconsistency of superpixels. The costs are learnt using a bag-of-words classifier. The higher order term itself comprises of a segment unary and pairwise terms. This level of generalization allows to select the best possible segment from multiple overlapping segments. The higher order term is given as

$$\psi_c(\mathbf{x}_c) = \min_{\mathbf{y}}(\sum_{c \in \mathcal{S}} \psi_c^P(x, y_c) + \sum_{c,d \in \mathcal{S}} \psi_{cd}(y_c, y_d)) \tag{3.3.4}$$

where y_c is an auxiliary variable for the clique c, while c and d are the neighbouring segments and $\mathcal{S}$ denotes the total number of segments. $\mathbf{y}$ is the set of all the auxiliary variables corresponding to the overlapping segments. The pairwise term is responsible for encouraging smoothness over the neighbouring segments (cliques). We refer the reader to [108] for more details. The energy minimization problem is solved using the graph-cut based alpha expansion algorithm [17].

3.3.2 Homography

In this section we describe how the image and the ground plane are related through a homography. We use a simplifying assumption that the world comprises a single flat ground plane. Under this assumption, we use the camera pose and estimate a frontal rectangular ground plane region that maps to a patch in the corresponding image assuming a constant camera height. For computing the camera viewing direction we refer to [80]. Each ground plane pixel z_i in the ground plane patch is back-projected into the k^{th} image X^k, as $x_j^k = P^k z_i$, where P^k is the corresponding camera matrix. This allows us to define the set T_i of the valid ground plane to image correspondences (as shown in Fig. 3.3(b)). Fig. 3.5 shows an example ground plane patch and a registered corresponding image region. Any object such as the green pole (Fig. 3.5(a)), that violates the flat world assumption will thus create an artefact on the ground plane that looks like a shadow, seen in the circled area in Fig. 3.5(a). If we have only a single image, then this shadowing effect would cause the pole to be mapped onto the ground plane incorrectly as seen in Fig. 3.5(b). If we have multiple views of the scene, then we will have multiple shadowing effects. These shadows will overlap where the violating object meets the ground, as seen in Fig. 3.5(c). This is precisely the part of the object which is necessary for building an overhead map. Also the flat ground around the object, shown in blue, will be correctly mapped in more views. This means that if we use voting over many views we gain robustness against violations of the flat world assumption.

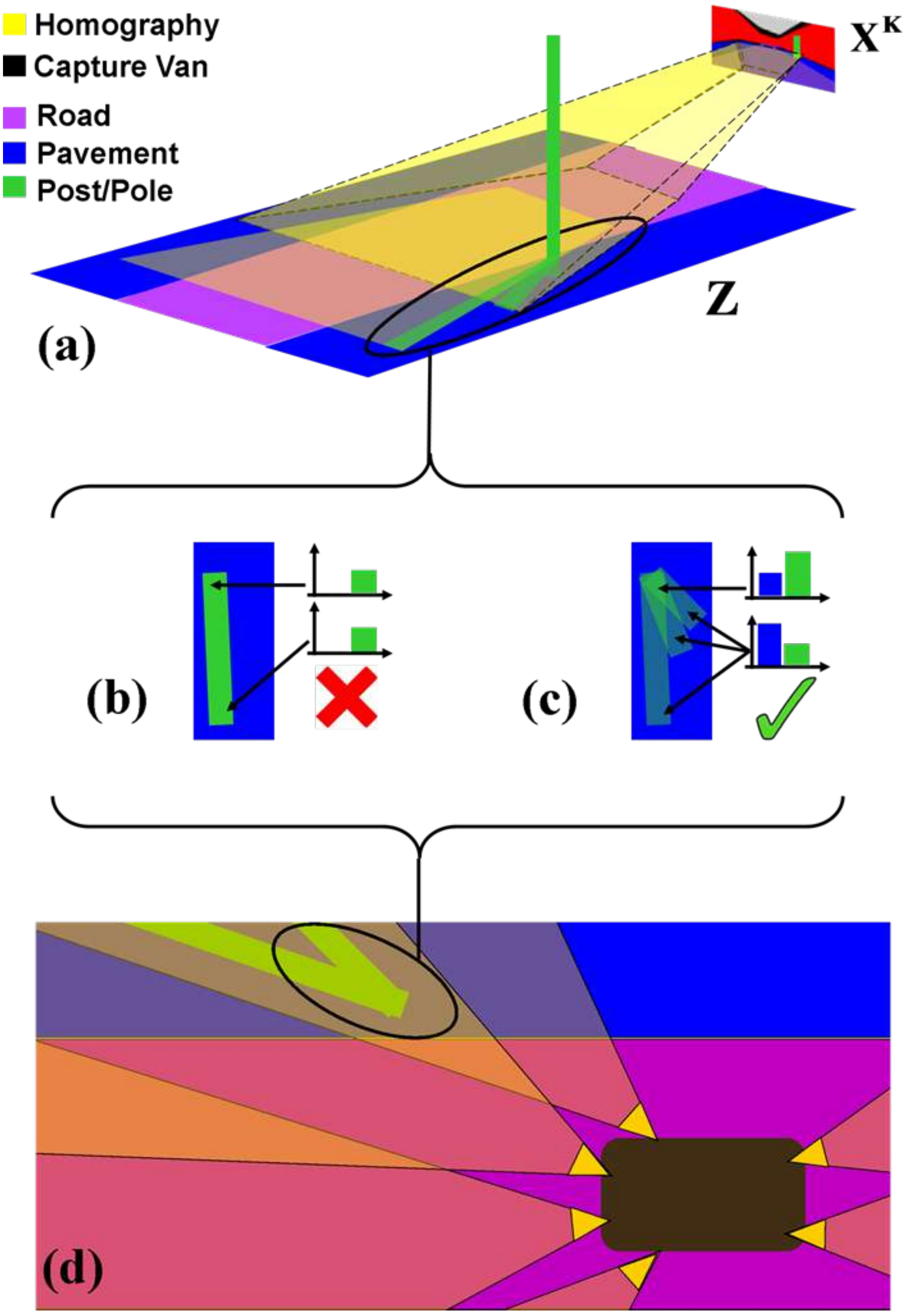

Figure 3.5: *Flat World Assumption:* In order to find a mapping between the local street level images to our global map (a) we use the simplifying assumption that the world is flat. The camera's rays act like a torch, depicted in yellow. The green pole, which violates this assumption creates an artefact on the ground plane that looks like a shadow, seen in the circled area in (a). With a single image, the shadowing effect would cause the pole to be mapped onto the ground plane incorrectly as seen in (b). With multiple views of the scene, we have multiple shadowing effects. These shadows overlap where the violating object meets the ground, as seen in (c). Each view votes onto the ground plane, resulting in more votes for the correct label. (d) Shows the effect of the multiple views from different cameras. This image is best viewed in colour.

3.3.3 Dense Semantic Map

Our semantic map represents the ground plane as if it were being viewed from above and is formulated as a pairwise CRF. The energy function for the map E^M is defined over the random variables $\mathbf{Z} = \{Z_1, Z_2, ..., Z_N\}$ corresponding to the ground plane map pixels as

$$E^M(\mathbf{z}) = \sum_{i \in \mathcal{V}} \psi_i^M(z_i) + \sum_{i \in \mathcal{V}, j \in \mathcal{N}_i} \psi_{ij}^M(z_i, z_j), \tag{3.3.5}$$

where $i \in \mathcal{V}$ is the ground plane pixel corresponding to the random variable Z_i and $\mathcal{N}_i$ is the set of its neighbouring ground plane pixels. The label set is kept same as that of the image domain (§3.3.1), except for the *sky* that should not be on the ground plane.

Unary potential The unary potential $\psi_i^M(z_i)$ gives the cost of the assignment: $Z_i = z_i$. The potential is calculated by aggregating label predictions from many semantically segmented street-level images (§3.3.1) given the registration (§3.3.2). The unary potential $\psi_i^M(z_i = l)$ is calculated as:

$$\psi_i^M(z_i = l) = -log(\frac{1 + \sum_{t \in T_i} \delta(x_t = l) c_t}{|L| + \sum_{t \in T_i} c_t}) \tag{3.3.6}$$

where $|L|$ is the number of labels in the label set. T_i denotes the set of image plane pixels with valid registration via the homography as defined in 3.3.2. The factor c_t is used to weigh the effect of the image plane pixel contributing to the ground plane. Ideally, we would like to learn the weights, but due to the lack of the training data, we weight all the pixels uniformly. This kind of unary potential represents a form of naive Bayes probabilistic measure which is easy to update in an online fashion. This voting scheme captures information across multiple views and time inducing robustness in the system. The ground plane unary generation is explained in Fig. 3.6.

Pairwise potential The pairwise potential $\psi_{ij}^M(z_i, z_j)$ represents the cost of the assignment: $Z_i = z_i$ and $Z_j = z_j$ and takes the form of a Potts model:

$$\psi_{ij}^M(z_i, z_j) = \begin{cases} 0 & \text{if } z_i = z_j, \\ \gamma & \text{otherwise,} \end{cases} \tag{3.3.7}$$

The pairwise potential encourages smoothness over the ground plane. A strong hypothesis for a particular label in the ground plane position is likely to be carried over to its neighbours. Thus, for uncertain regions, this property will help in getting correct labelling.

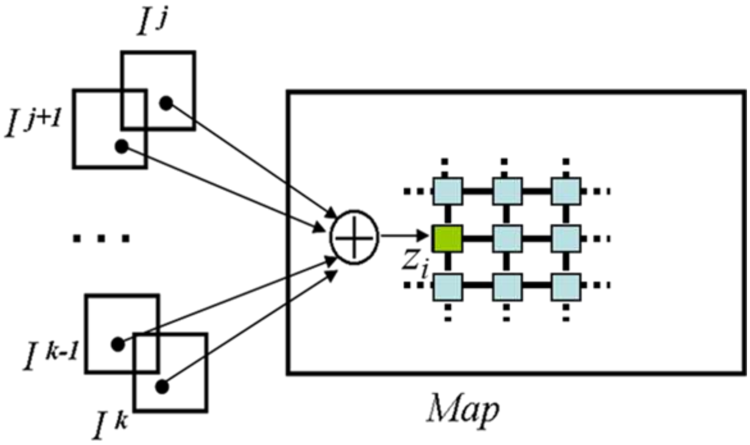

Figure 3.6: *Semantic Map Unary*: Each ground plane point z_i is seen from a set of images $I^j, I^{j+1}, ..., I^k$. The corresponding image pixels form the set T_i, which contribute to the hypothesis for the ground plane unary $\psi_i^M(z_i)$. The label prediction from all these image points are fused according to Eqn. 3.3.6 to the unary hypothesis for the ground plane point.

The pairwise parameter γ, is set manually on unlabelled validation data.

3.4 Experiments

For qualitative and quantitative evaluation of our method, we are introducing a new dataset that is 14.8 kilometre long, captured along the roadways in the UK. We have performed a manual annotation for a representative set of images which is used for training and evaluation purpose. The images have been selected in such a way that all the object classes are represented adequately in both the training and the test set and do not contain overlapping scenes.

3.4.1 Yotta Pembrokeshire Dataset

Our dataset is comprised of two sets of images. The first set is composed of street level images captured from a moving vehicle, and the second set comprises of the overhead aerial images (obtained from Google earth) of the same region. Both the image sets have corresponding ground truth object class labellings which is made publicly available for the computer vision community[1]. We now describe the dataset in details.

[1] available at http://www.robots.ox.ac.uk/˜tvg/projects/SemanticMap/index.php

3.4.1.1 Yotta Pembrokeshire Street Images

Our street image set comprises of a 8000 long sequence of images captured from each camera in the Pembrokeshire area of the United Kingdom. The images are captured at an interval of 2 yards (approximately 1.8 meters) making it particularly hard for any sparse point cloud reconstruction using traditional SLAM based methods [46]. The total length of the sequence spans approximately 15 kilometres of roadways. The images are captured by Yotta[2] using a specialised road vehicle with 6 mounted cameras, 2 frontal, 1 either side and 2 rear facing. The van that has been used to capture the data is depicted in Fig. 3.7 along with some of the captured images. The images at full resolution are 1600×1200. This area contains urban, residential, and some rural locations, making it quite varied and challenging real world dataset. The camera parameters, computed from the GPS and odometry information, are also provided. We have manually selected a small set of 86 images from this sequence for ground truth annotation with 13 object classes, 44 of the labelled images are used for training the potentials of E^S and remaining 42 are used for testing. The 13 labels are road, building, vehicle, pedestrian, pavement, tree, misc-vegetation, sky, street-bollard, shop-sign, post-pole, wall-fence, street-signage.

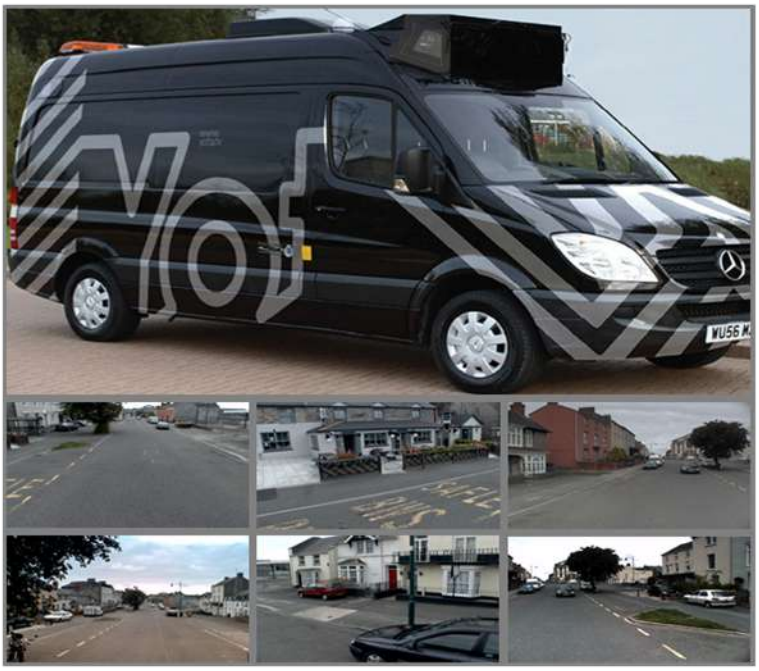

Figure 3.7: *YottaDCL van:* The van used to capture the data by YottaDCL [42] with 6 mounted cameras, 2 frontal, 1 either side and 2 rear facing. Some images captured by the van is also shown in the bottom.

[2] This data has been used in the past for real-world commercial applications and has not been designed for the purpose of this paper. Please see http://yotta.co.uk//

3.4.1.2 Yotta Pembrokeshire Aerial Images

Our aerial image set is comprised of satellite views that are acquired from Google Earth [67] and correspond to a subset of the geographical area the region covered by our vehicle (approximately 7.5 kilometres). We annotate these images with the ground truth object labels. A 'kml' track [92] is generated from the GPS data of the vehicle, enabling us to acquire the overhead views from google earth of the vehicle's track. We have used the initial part of the vehicle track(4.5km) for training and the remaining (3km) for testing purposes, which enables us to separate the training and testing scenes. Few samples of overhead aerial images with corresponding ground truths are shown in Fig. 3.8.

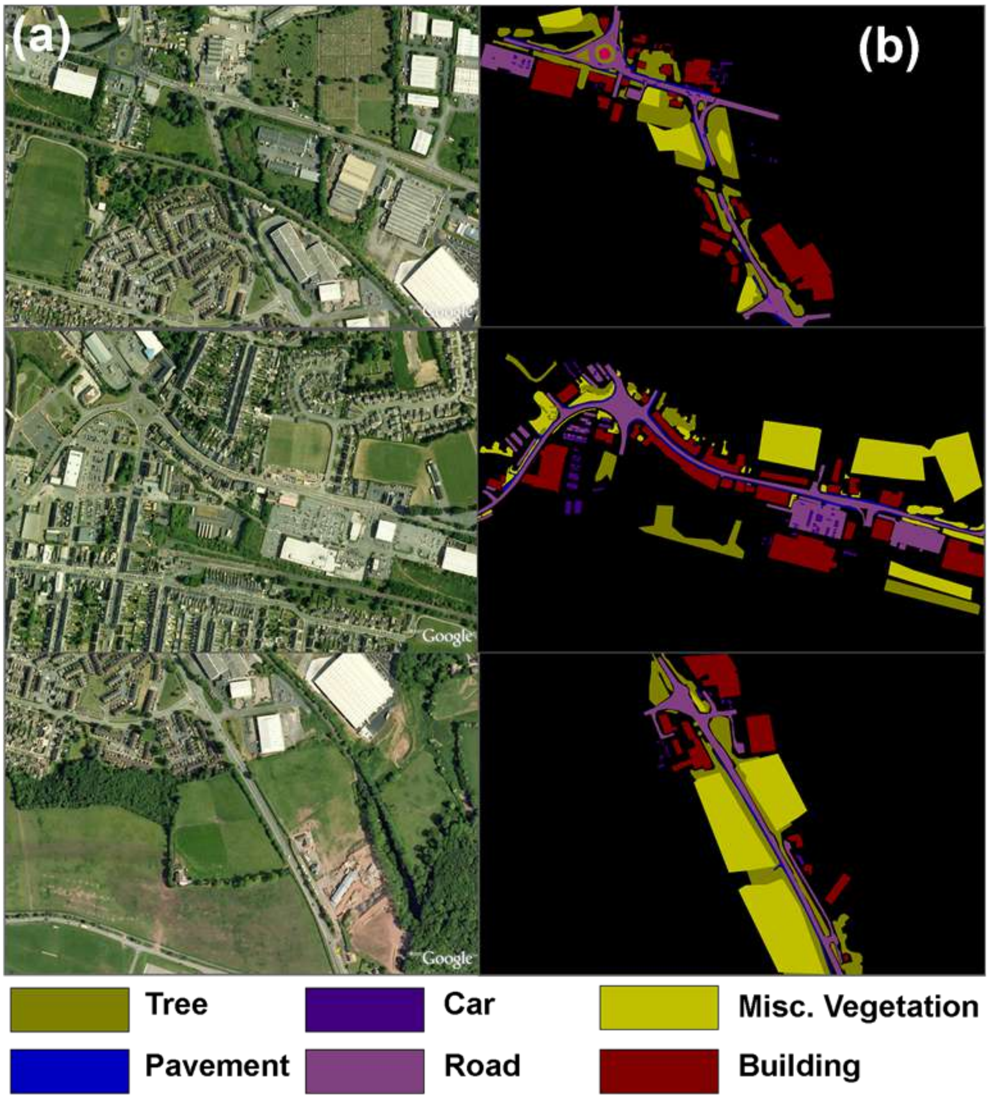

Figure 3.8: Object class ground truth for aerial images. (a) The overhead images are obtained from Google earth corresponding to the Pembroke City. (b) Ground truth object class labels to the corresponding overhead images. The ground truth labels are used to learn the baseline classifier which performs object class segmentation on the aerial imagery.

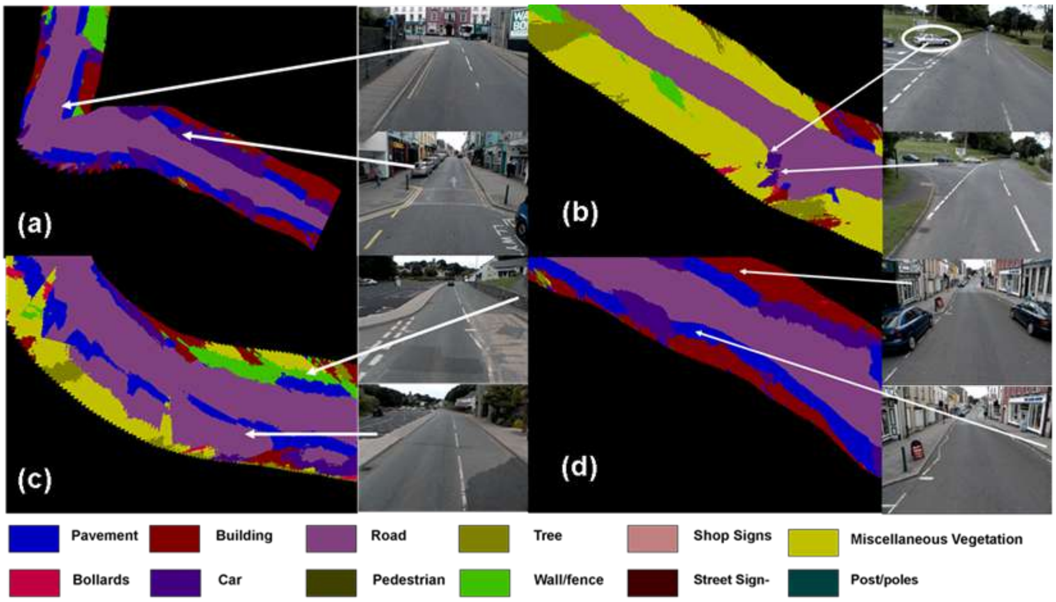

Figure 3.9: Overhead Map with some associated images. The arrow shows the positions in the map where the image was taken. In (a) we see from the image there is a T-junction, which is also depicted in the map. The circle in the image (b) shows the car in the image, which has been labelled correctly on the map. Similarly, in (c) the fence and the carpark from the images are also found in the map. In (d) arrows showing the pavement and the buildings being mapped correctly. Best viewed in colour.

3.4.2 Results

Fig. 3.4 shows the output of the first level of our CRF framework. The first row shows the street-level images captured by the vehicle. The second and the third row show the semantic image segmentation of the street images and the corresponding ground truth. For computational efficiency, we have down sampled the original image size into 320×240. Fig. 3.9 shows the map and corresponding street imagery. The arrows show the positions on the ground plane map where the image objects are located. We can see in Fig. 3.9(a) a T-junction and cars in both the map and images. Similarly, in Fig. 3.9(b) the cars are shown in the map. In Fig. 3.9(c) a car-park (modelled as road) and a fence are shown in both map and the images. Finally in Fig. 3.9(d) we see the buildings and the pavement in the map. In the semantic map we do not have the sky as it lies above the horizon in the street images and are not captured in the ground plane-image registration. Since the images are captured every two yards, the classes like roads, pavements, building, fence, vehicle, vegetation, tree which have long range contextual information span across multiple images. Thus those classes appear more frequently in the map, unlike the smaller object classes like pedestrian, post-pole, street-signage. This is witnessed in the qualitative results shown in Fig. 3.9. Fig. 3.14 shows another output of our algorithm, where the map spans a distance of 7.3km.

The vehicle track (white track on Google Earth) and the semantic map output (overlaid image) are both shown. The map building for this example takes 4000 consecutive frontal street-level images as the input.

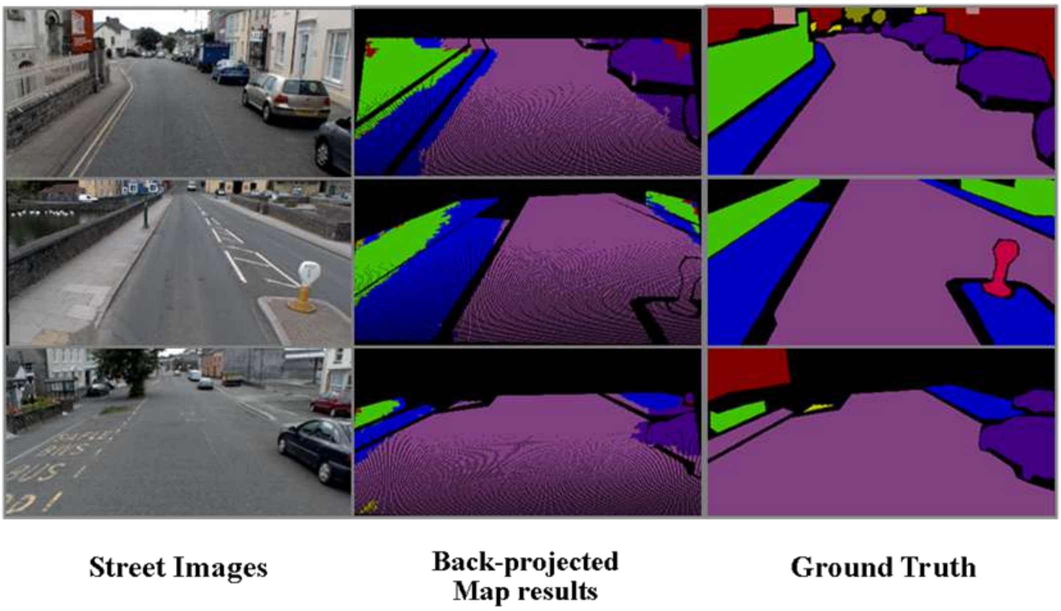

Figure 3.10: Evaluation method. The first column shows the street images, the second column shows the back-projected ground plane map results, while the final column show the ground truth object class labels for the corresponding images.

For the purpose of quantitative evaluation of object level classification, we have projected the semantic map back into the image domain using the same homography estimation (§3.3.2). Our evaluation set is a representative subset of the entire sequence, comprised of 42 images. We measure the number of correctly labelled pixels over all classes on the re-projected results into the image as illustrated in Fig. 3.10. The first column shows some of the street level images from our YottaDCL dataset, the back projected map results are shown in the second column and the image level ground truth is displayed in the last column.

We have compared our semantic map method with a baseline method which is trained using the overhead Google earth satellite images (see §3.4.1.2). The baseline method is a CRF based classifier, similar to the street level image labeller described in §3.3.1. We evaluate the baseline method on the the google earth test image set, to provide an indicative performance measure compared to our semantic map method. Few sample outputs of the baseline method are depicted in Fig. 3.11. The first column shows the aerial images, followed by the ground truth object class and the output of the baseline method. It is observed that often the road and the pavement are mislabelled in the final output. Similarly, in cases where the image contains lot of buildings, the baseline classifier is unable identify the road.

For a more direct comparison of the baseline method with our semantic map, we generate a *plan view*, from the from the Yotta street level images (see Fig. 3.12(a)), by projecting them using the homography described in §3.3.2. This is because the satellite images obtained from Google earth and street level images are captured at different times, they contain different objects in the scene. This plan view is classified using the baseline classifier, and back projected into the image domain for quantitative measurements. The input plan view and output segmentation from the baseline method for the plan view is shown in Fig. 3.12.

Our results are summarised in table 3.1. We have shown quantitative results on two metrics, $Recall = \frac{\text{True Positive}}{\text{True Positive + False Negative}}$ and $Int.$ vs $Union = \frac{\text{True Positive}}{\text{True Positive + False Negative + False Positive}}$ measures. ‘Global’ refers to the overall percentage of pixels correctly classified, and ‘Average’ is the average of the per class measures. In this evaluation we have not considered the class ‘sky’ as this is not captured in the model. Furthermore, in the overhead projected view the classes like ‘pedestrian’, ‘shop-sign’, ‘post-pole’ and ‘street-signage’ are not represented because of their size and upright orientation, and hence omitted from evaluation. The first row describes the class-wise performance of our semantic map evaluated. The second row shows the classification accuracy of the baseline method while operating on the plan view (generated from Yotta street images). As expected, the baseline classification of the plan view suffers from lack of details. Our semantic map generated from street level images has the advantage of capturing local semantic and contextual details, which are then aggregated into building the semantic map which is more expressive. However, as we are using the ground plane homography (§3.3.2) for registration, the class ‘trees’ are not captured often due their height, unlike the class ‘misc. vegetation’ which is present closer to the ground level. This explains why our method fares inferior for the class ‘trees’ as compared to other classes. Overall, our method achieves a global accuracy of 82.9%, an average recall of 57.8% and average Int. vs Union score of 41.4%.

It is worth noting that the baseline method is trained with the satellite images, and is less effective in classifying the plan view generated from the street images. Hence, we additionally report the performance of the baseline method for segmenting the overhead satellite images (see table 3.2). This evaluation is not directly comparable to our method as the satellite images are captured in different times and have different objects in the scene, and serves as an indicative performance measure. The baseline classifier (satellite images), achieves a global accuracy of 80.1%, an average recall of 50.1% and average Int. vs Union score of 40.3% while evaluating the overhead satellite images. The aerial images are par-

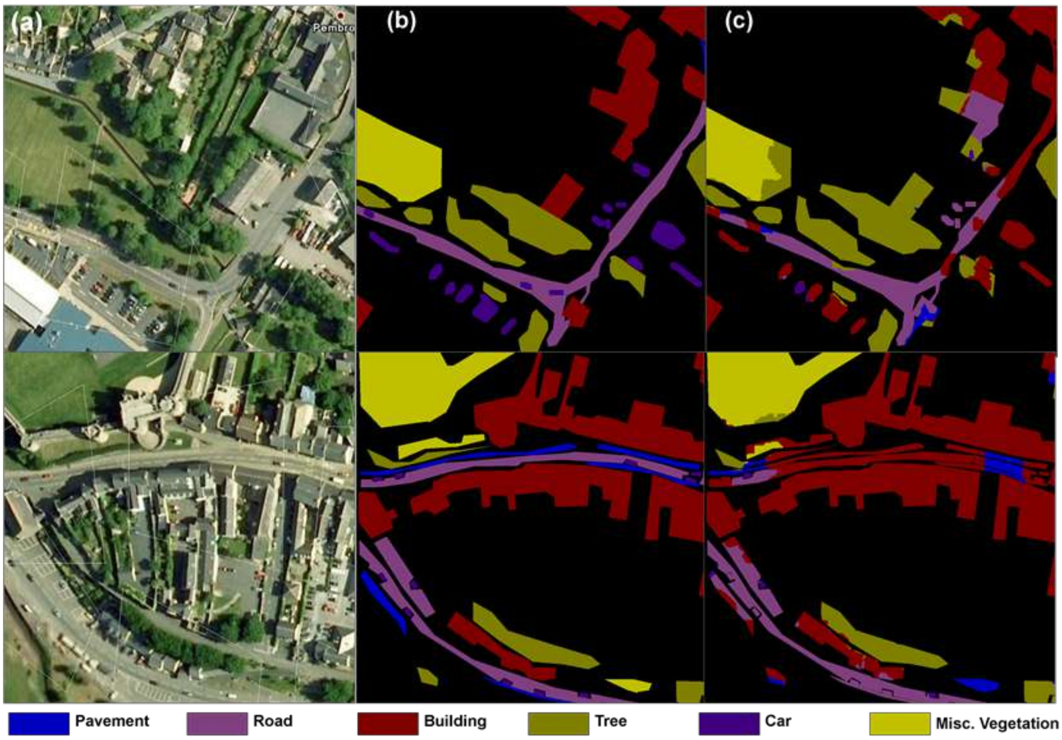

Figure 3.11: Semantic Map (baseline): Indicative performance of the baseline method when evaluated over aerial images. (a) Over head aerial images captured from Google earth, (b) corresponding object class ground truth and (c) object class segmentation. (Best viewed in colour)

ticularly good for classifying ‘trees’ which appear very distinct in the overhead images. However, in the satellite images it is hard to determine the boundary of the road and the pavement, which results in the low accuracy of the pavement. Similarly, due to the nature of the upright orientation of the fence and small size of the class ‘car’ in the satellite images, they are not classified correctly by the baseline method. These observations depict the importance of exploiting local details, available in abundance in the street images for the task of generating a detailed overhead semantic map for a road scene.

3.5 Discussion

We have proposed a framework to generate a semantically labelled overhead view of an urban region from a sequence of street-level imagery and introduced a new dataset towards this application. Our approach, by virtue of operating on street images, allows us to produce a detailed semantic map of a large scale urban area, which is difficult for the methods operating on overhead satellite views. Furthermore, by operating on the local street level images, our method can scale easily from mapping hundreds of metres to tens of kilometres.

Table 3.1: Semantic Evaluation: Pixel-wise percentage accuracy on the test set. We compare the semantic map, generated from street images and the baseline method using classifiers learnt from overhead imagery.
†: Average computed for classes ‘Building’ and ‘Road’, hence not directly comparable.

Method	Building	Tree	Car	Road	wall/fence	Pavement	Misc. Vegetation	**Average**	**Global**
Recall measure									
Semantic Map (ours)	35.4	5.1	77.9	89.9	72.7	62.3	62.5	57.8	82.9
Baseline Map (Plan view evaluation)	17.8	n/a	n/a	36.9	n/a	n/a	n/a	27.4†	22.4
Intersection *vs* union									
Semantic Map (ours)	26.9	3.8	51.0	85.7	36.7	42.8	42.7	41.4	
Baseline Map (Plan view evaluation)	3.1	n/a	n/a	32.0	n/a	n/a	n/a	17.5†	

Table 3.2: Indicative performance evaluation of Baseline method: Pixel-wise percentage accuracy on the overhead satellite test set images(§3.4.1.2).

Method	Building	Tree	Car	Road	Pavement	Misc. Vegetation	Wall/Fence	**Average**	**Global**
Recall measure									
Baseline Map (Satellite Images)	89.6	96.1	16.1	49.3	15.4	85.1	n/a	50.1	80.1
Intersection *vs* union									
Baseline Map (Satellite Images)	74.9	59.7	15.1	42.2	9.6	84.1	n/a	40.8	

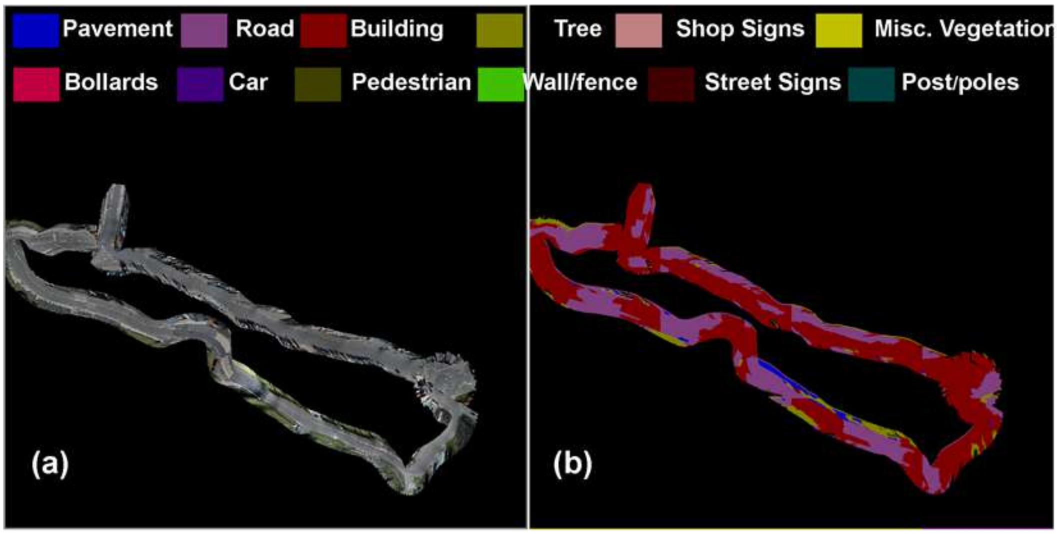

Figure 3.12: Semantic Map (baseline). (a) Over head image generated by projecting the street level imagery via the homography as defined in 3.3.2, corresponding to the Pembroke City. (b) Semantic map, generated from classifiers trained with overhead imagery.

We formulated the problem using two conditional random fields. The first one performs semantic image segmentation locally at the street level and the second one updates a global semantic map. We have demonstrated results for a track of tens' of kilometres showing object labelling of the map. Potentially, this can be extended to work for a much larger scale comprising the whole of the UK. However, this poses a challenge of accommodating the variety of visual data for classification, taken across geographical locations spanning thousands of kilometres of roadways.

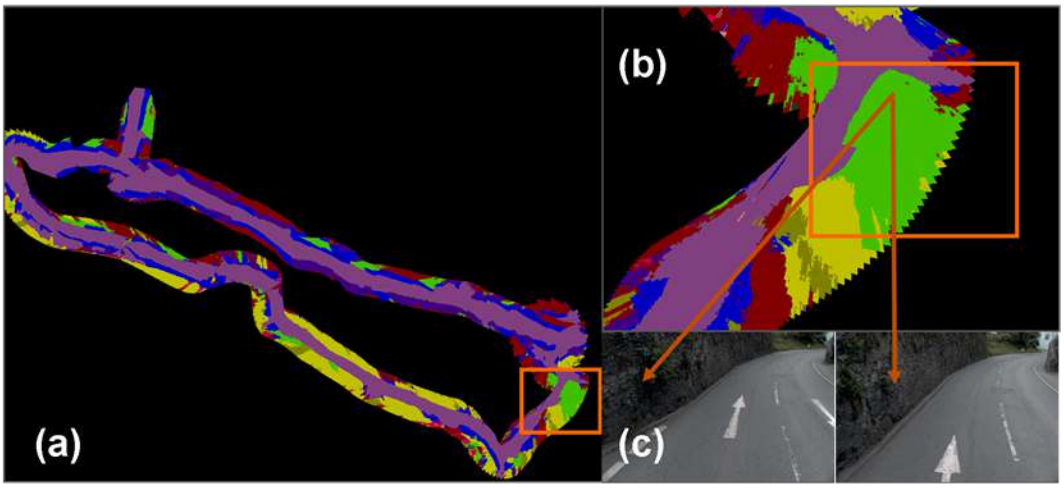

Figure 3.13: Artefacts produced via the flat world assumption: (a) Shows an urban semantic map, (b) shows a magnified part of the map, showing a bend in the road and (c) shows the corresponding street level images at that particular location. The map shows a fence which being upright, violates the flat world assumption. As we do not have sufficient camera views to map the fence and the area beyond the fence, a shadow is created in the final map. Ideally, the fence should have been displayed in the map corresponding to the ground area it covers in the actual world.

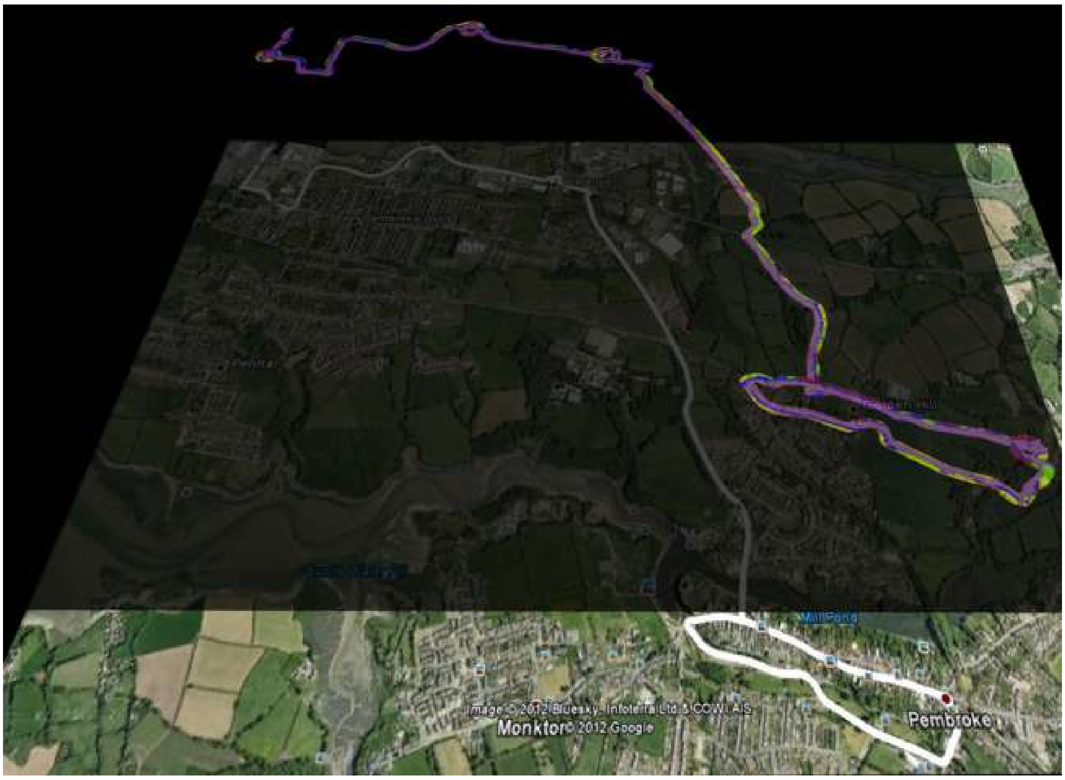

Figure 3.14: *Map output 2* Semantic Map shows a road path of 7.3km from the Pembroke city, UK. The Google Earth image (shown only for visualization purpose) shows the actual path of the vehicle (in white). The top overlaid image shows the semantic map output corresponding to the vehicle path. Best viewed in colour.

In this work, we have relied on the flat world assumption simulating the ground plane: which creates some undesirable artefacts as shown in Fig. 3.13. The magnified figure shows a part of the final map with a shadow generated by a fence. Ideally, the fence should have been displayed in the map corresponding to the ground area it covers in the actual world. However, as we do not have sufficient camera views to map the fence and the area beyond it, a shadow is created in the final map. This kind of undesirable artefacts motivates us to go beyond the flat world mapping and create a dense semantic 3D model using the multiple street view images. In the next chapter, we propose a method to generate such dense representation with associated semantic labels, from a sequence of stereo images.

Acknowledgements

I would like to thank Manish Jettwa from Yotta DCL, for providing the basis for the dataset including the images and the camera calibrations.

Chapter 4

Urban 3D Semantic Modelling Using Stereo Vision

4.1 Introduction

While the last chapter dealt with the generation of a ground plane semantic map of a large scale urban region, in this chapter we propose a method to generate a dense semantic structure of the urban scene. By a semantic structure we mean a 3D reconstruction of the scene with annotated semantic labels. Classification of a scene along with spatial information of the present objects has been a long standing goal of the computer vision community. Such interpretations are useful for various robotic applications to perform high level reasoning of their surroundings, such as autonomous navigation in urban areas with collision avoidance [31]. Recent entries in the Darpa urban challenge [40] have shown how visual perception from multiple sensor modalities can help in the task of autonomous navigation [115]. In all their work, it has been shown that the robots need to understand, plan their path and recognise the objects in the scene [44, 106, 171]. In this chapter, a computer vision algorithm is presented which uses images from stereo cameras mounted on a vehicle to generate a dense 3D semantic model of an urban environment. As the vehicle moves around the scene, our method is able to generate a 3D semantic representation in an online fashion to enable reconstruction over a long street image sequence. The output of our algorithm is the 3D surface model in a meshed representation, where every mesh face is associated with a particular object category, such as road, pavement, or vehicle.

Currently, laser based systems are the mainstay of most robotic systems which try to navigate autonomously through an urban area. These laser systems provide sparse scans of the scene which are used for 3D depth perception [45, 115, 117, 131]. As a result, the reconstructed scene is sparse and ineffective for accurate boundary predictions [155] and proves to be inefficient for robotic manipulation tasks. To obtain an accurate understanding of the scene, a dense, metric representation is required [129]. We show that such a representation can be obtained using a vision based system. Moreover, compared to normal cameras, laser sensors are expensive and power hungry, can interfere with other sensors, and have a limited vertical resolution [63].

Our approach is illustrated in Fig. 4.2. The input to our system is a sequence of rectified stereo images. We use a robust visual odometry method with stereo feature matching to track the camera poses. The estimated camera poses are used to fuse the depth-maps generated from stereo pairs, producing a volumetric 3D representation of the scene. The fusion process is made online to handle long image sequences. In parallel, the pixels in the input

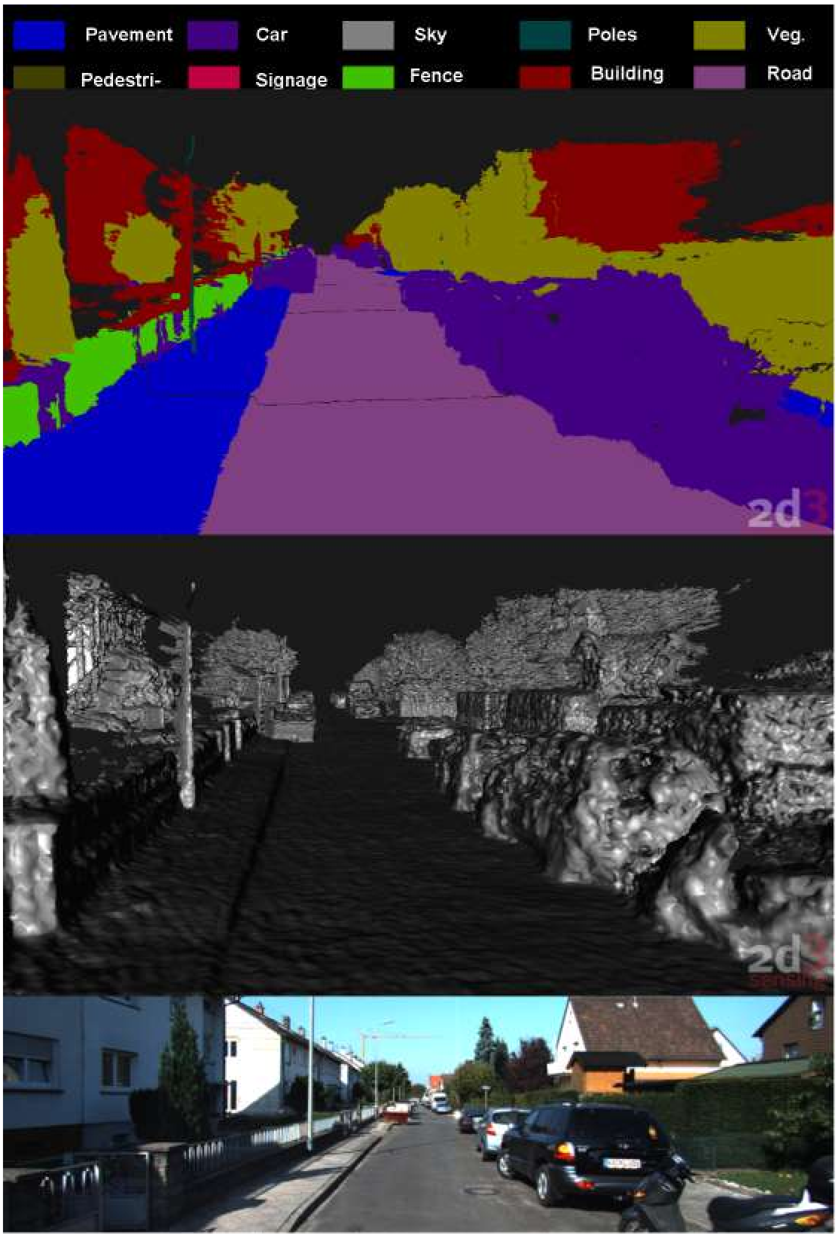

Figure 4.1: 3D *semantic reconstruction.* The figure shows a sample output of our system. Dense 3D semantic reconstruction along with class labels is shown in the top and the surface reconstruction is shown in the middle. Bottom image shows one of the image corresponding to the scene.

views are semantically classified using a CRF model. The label predictions are aggregated across the sequence in a robust manner to generate the final 3D semantic model. A sample output of our system is shown in Fig. 4.1. The top image shows the semantically labelled dense reconstruction and the middle image shows the corresponding structure. To evaluate our method we manually labelled object class ground truth of a representative subset of the KITTI [62] dataset. We evaluate both object labelling and odometry results.

4.1.1 Outline of the Chapter

In § 4.2 we briefly describe some of the related work in object labelling and surface reconstruction. In § 4.3 we describe the large scale dense surface reconstruction from stereo images. § 4.3.1 describes our robust visual odometry method using both stereo image pairs while § 4.3.2 describes our large scale online surface reconstruction. § 4.4 describes the semantic model generation and § 4.5 describes the experiments and the datasets. The chapter concludes with a discussion in § 4.6.

4.2 Related Work

Object recognition has been extensively researched in the field of computer vision [4, 23, 69, 111, 164] towards addressing the problem of object recognition for urban road scenes. Vision based techniques have been used for determination of road regions as early as [43] and for general robotic navigation in an urban environment and recognise the objects in the scene [7,39,173,184]. These algorithms work in the image domain where every pixel in the image is classified into an object label such as car, road, pavement, etc. In [55], the sequence of images taken from a moving vehicle were used to improve the object classification to perform a geometrically consistent segmentation. The pixel based classification approach was extended to perform a coarse 3D representation in the form of blocks in [75], and with a stixel representation in [137]. In [137] stixel representation was shown to assist drivers through the obstacle detection in the road. Object class segmentation in the image domain was extended to generate a semantic overhead map of an urban scene from street level images [155]. The most closely related to our work is [111], where a joint representation of object labelling and disparity estimation is performed in the image domain. However, none of these methods deliver a dense and accurate 3D surface estimation.

On the other hand, stereopsis has been considered as an important source of depth perception by presenting two different perspectives of the world scene similar to biological vision [183]. Computing depth from stereo is a traditional problem in computer vision, where the disparity between the two views has been used to provide a meaningful depth estimate [122]. Fusion of multiple depth estimates from range and image data corresponding to a scene, for generating surface or a 3D model, has been extensively studied in the field of computer graphics [38,162,181]. Other shape recovery methods are based on visual hull [32], implicit surface [10] or through a set of point based representations [3]. Recently, Newcombe et al. [129] proposed a system for dense 3D reconstruction using a hand-held

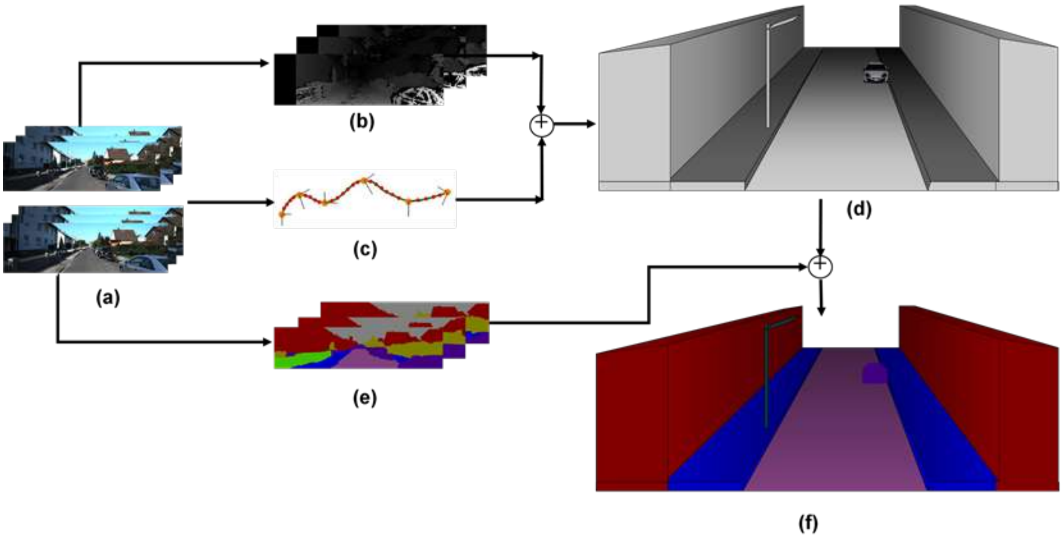

Figure 4.2: *System Overview.* (a) Shows the input to our method which is a sequence of rectified image pairs. The disparity map (b) is computed from the images and (c) is the camera track estimation. The outputs of (b) and (c) are merged to obtain a volumetric representation of the scene (d). (e) Shows the semantic segmentation of the street images which is then fused into a 3D semantic model of the scene (f). Best viewed in colour.

camera in real-time. The dense model is generated from overlapping depth-maps computed using every image pixel instead of sparse features, thus adding more details to the final model. Geiger et al. [63] proposed a method for reconstruction of road scenes from stereo images. Their method uses a point cloud representation which is updated by averaging the estimated 3D points and as a consequence, can quickly suffer from accumulated drift.

In this chapter, we propose a method to merge the dense per-pixel object labelling from stereo images and dense surface reconstruction to generate a dense 3D semantic model of an outdoor road scene. We exploit a CRF based method to extract semantics from the street level images and aggregate in a simple yet effective method to generate model labellings.

4.3 Large Scale Dense Semantic Reconstruction from Stereo Images

Our semantic 3D reconstruction pipeline consists of three main stages. The first stage is the camera pose estimation from a sequence of rectified stereo images (Fig. 4.2-c). The second stage is the volumetric 3D reconstruction and surface meshing (see Fig. 4.2-d) for arbitrarily long image sequences. In this stage, the camera poses estimated from the previous stage are used to fuse the depth maps generated from stereo disparity in an online fashion into a global truncated signed distance function (TSDF) volume. Then a meshed representation of the surface is generated by using the marching tetrahedra algorithm on the volume [136]. The final stage involves semantically labelling the structure with appropriate object class labels. (Fig. 4.2-f). It comprises of two steps where first we extract object label information locally from each image frame, and then we fuse the labels to generate a semantically consistent structure.

4.3.1 Robust stereo camera pose estimation

In this section we describe the camera pose estimation using stereo vision. In general, camera tracking requires: a) camera calibration parameters, which model the lens and sensor properties, and b) the extrinsic camera parameters, which indicate its position and orientation in space as the stereo rig moves, with 6 degrees of freedom (3 rotation and 3 translation parameters) [150]. We assume calibrated stereo cameras positioned on a rigid rig, so that the relative locations of the stereo pair are fixed as the vehicle moves. Estimating the extrinsic parameters involves mainly the following steps: feature matching, estimating the initial camera pose and finally, bundle adjustment of the estimated camera pose. We now discuss these steps in detail.

4.3.1.1 Stereo Feature matching

To generate feature matches, we use both the stereo and egomotion information. For stereo matching we try to find potential matches across the epipolar line based on the sum of squared difference score of an 8×8 patch surrounding the candidate pixel (for an image of resolution 1241×376). The candidate pixel is determined using a corner detector based on [79]. Each potential match is cross-checked to ensure it lies in a valid range (i.e. minimum depth/maximum disparity magnitude) and the fact that points must be in front of both the cameras. The image matches are cross-checked for both left-right and right-left pairs and the agreed matches are kept. After the list of stereo matches is obtained, we perform frame to frame matching for both left and right images. The basic approach is similar to the stereo matching framework, except that we do not rely on epipolar constraints. Also, as the frames are captured at a frequent interval from the moving vehicle, we assume that features may only move a certain number of pixels from frame to frame, enabling us to reduce our search space in the image domain for the egomotion case.

Once the matches are computed, the corresponding feature tracks are generated. All the stereo matches which also have corresponding frame-to-frame matches are kept in the track. Having this agreement between both the stereo and ego-motion helps to generate an initial camera estimate, which is further refined using the bundle adjuster accurately.

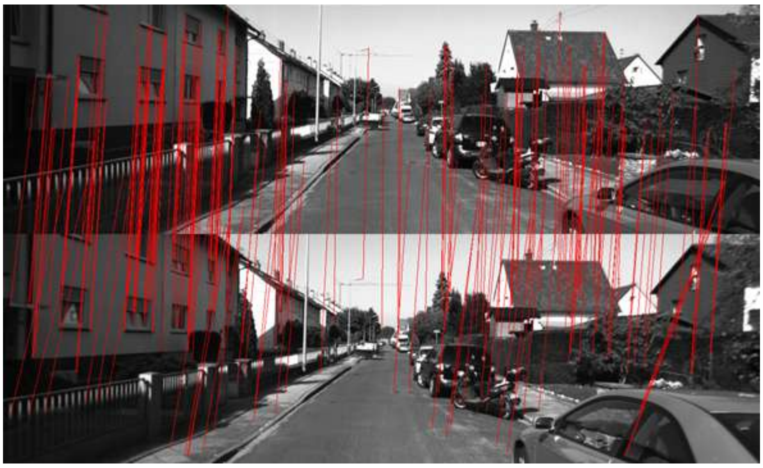

Figure 4.3: *Feature matching.* The red lines show the feature matches between the left (top) and right (bottom) images of the stereo pair, for a particular frame in the sequence.

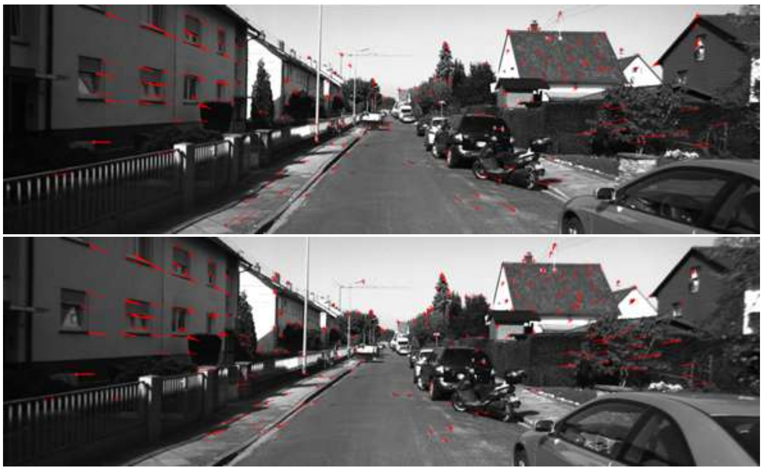

Figure 4.4: *Frame by Frame matching.* Consecutive frame to frame matching for left (top) and right (bottom) images. The matches are indicated by the red lines.

4.3.1.2 Bundle adjustment

Once we generate the feature track database, visible in multiple frames, we estimate the initial camera poses and the 3D points using [174]. *Bundle adjustment* refers to the general approach of performing non-linear minimisation of an objective function that minimises re-projection errors of the 3D points obtained from the feature tracks [48], where the parameters to the bundler are the 3D coordinates of the points (relative to the first camera), and the camera pose parameters (translation and rotation for each frame) [177]. For a globally optimal solution, we would need to simultaneously adjust all of the camera poses and 3D points for every frame. However, since we will be dealing with street-level sequences of arbitrary length, this approach is computationally infeasible, and may be unnecessary anyway, since the structure of the world near to the current frame is of more interest. As a result, we use a bundle method where our optimiser estimates camera poses and the associated features viewed by the last n cameras, leading to lower accumulated drift by reducing noise over n frames. In our experiments we set $n = 20$ which we found to be a good compromise between speed and accuracy.

Iterative initialisation

The merits of the bundle adjustment procedure depends on the initial estimate of the solution provided to it [48, 73]. Before performing the refinement, it is important to initialise the parameters to values that are within the sphere of convergence of the objective function. In practice, this means that we must estimate camera poses and 3D feature point locations as accurately as possible before performing any bundle adjustment. We do this by adding one frame at a time, then bundle adjusting, adding the next frame, bundle adjusting again, and so on. In this way, we expect to perform 19 bundle adjustments on problem sizes increasing from 2 to 20 frames before our full bundle window is initialised. From the 20^{th} onwards, bundle adjustment is performed over the last 20 sets of camera pose parameters, and the associated 3D features viewed by these cameras. This set corresponds to the size of the sliding window in our bundle adjustment process.

An alternative technique would be to initialise each new frame using the previous frame pose estimate, and the frame to frame pose estimation technique, without the intermediate bundle adjustments. However, this method will suffer from drift, and after 20 frames, the pose parameters might be outside the optimiser's sphere of convergence. In our experiments, we observed that it is better to use the iterative bundle method to avoid this risk, at the expense of a few extra bundler iterations.

Conditioning bundle adjustment parameters

The convergence and the quality of the estimated parameters using the bundler is sensitive to the conditioning of the parameters [177]. The set of parameters includes the 3D positions of the feature track points, the camera rotation matrix and the camera centres. As we are dealing with road scenes, the feature tracks can correspond to the scene elements situated from a couple of meters upto hundreds of metres away from the camera. Thus to accommodate a wide range of depths, we use a 4D homogeneous representation, with a local 3D parametrisation during bundle adjustment. The local parametrisation simply involves holding the largest element constant, and adjusting the remaining three elements, followed

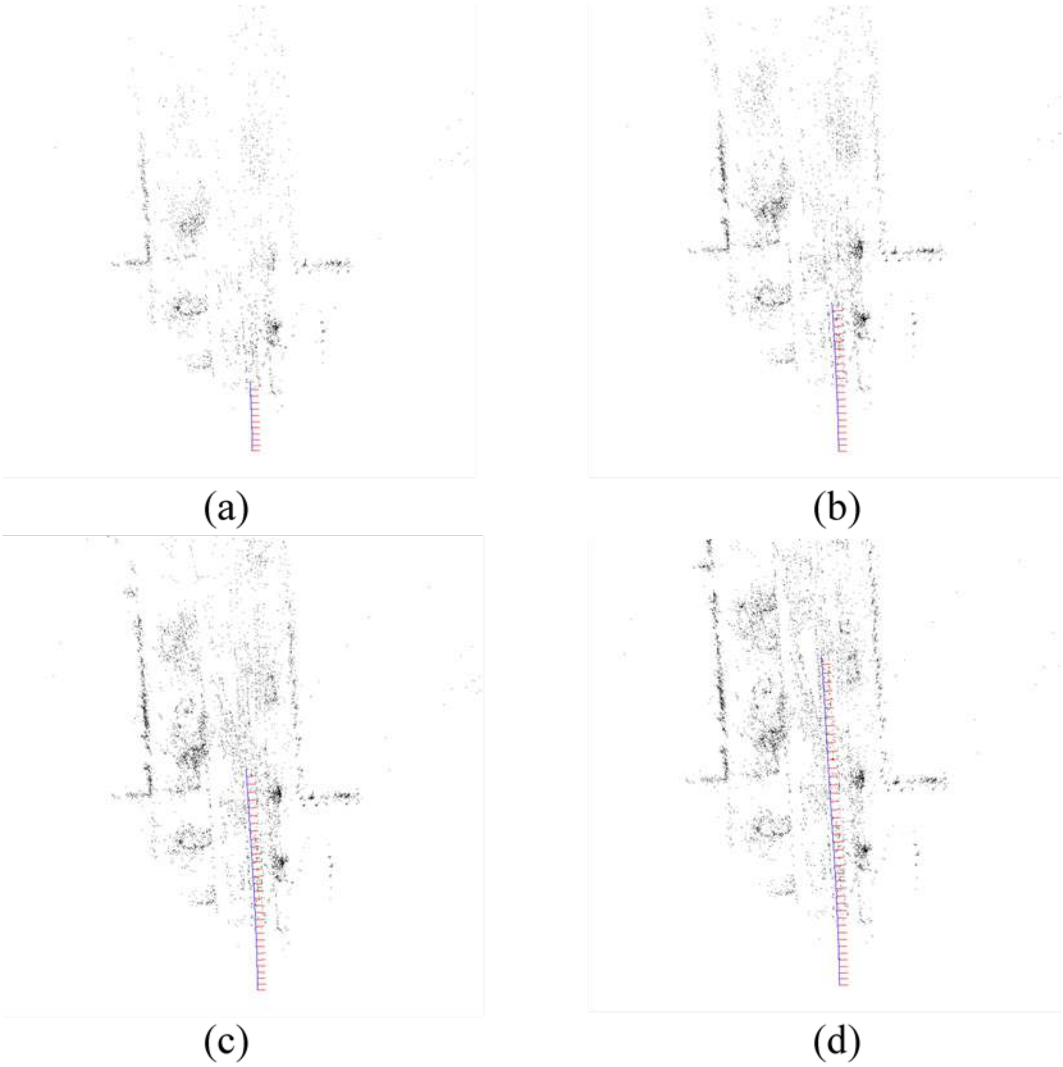

Figure 4.5: Bundle adjustment results, showing camera axes and 3D track points.

by a renormalisation step (such that the 4-vector always has a magnitude of 1) after each adjustment iteration. For representing the camera rotation, we use modified Rodrigues parameters (MRP's), as these do not suffer from gimbal lock (whereas Euler angles, suffer from the problem [71]). They are also a minimal 3-parameter representation, and are fairly well normalised to approximately unity for the expected range of rotations. The MRP's actually represent delta rotations from the orientation computed during initialisation, so are typically small corrections only. Finally, the camera centres are represented in a similar fashion as 3D feature points, using the 4-vector representation, with a 3D local parametrisation, although this isn't strictly necessary for camera positions, as they are within a well-defined range (unlike 3D track points, which may be near infinity). The 4-vector normalised representation of 3D track points and camera centres provides an implicit conditioning for these parameters, and the delta rotations represented as MRP's provides implicit conditioning for the rotations.

The objective function is the sum of squares of reprojection errors, so each residual (a 2D vector) may be conditioned by mapping the image extents to unity, in a coordinate system centred on the image centre. A good approximation of this is obtained by simply using image plane coordinates (i.e. subtracting the principal point and dividing by focal length). This ensures that our objective function will start off near unity, depending on the number of residuals and the average initial residual error. The final optimization is performed using the Levenberg-Marquardt optimiser [80].

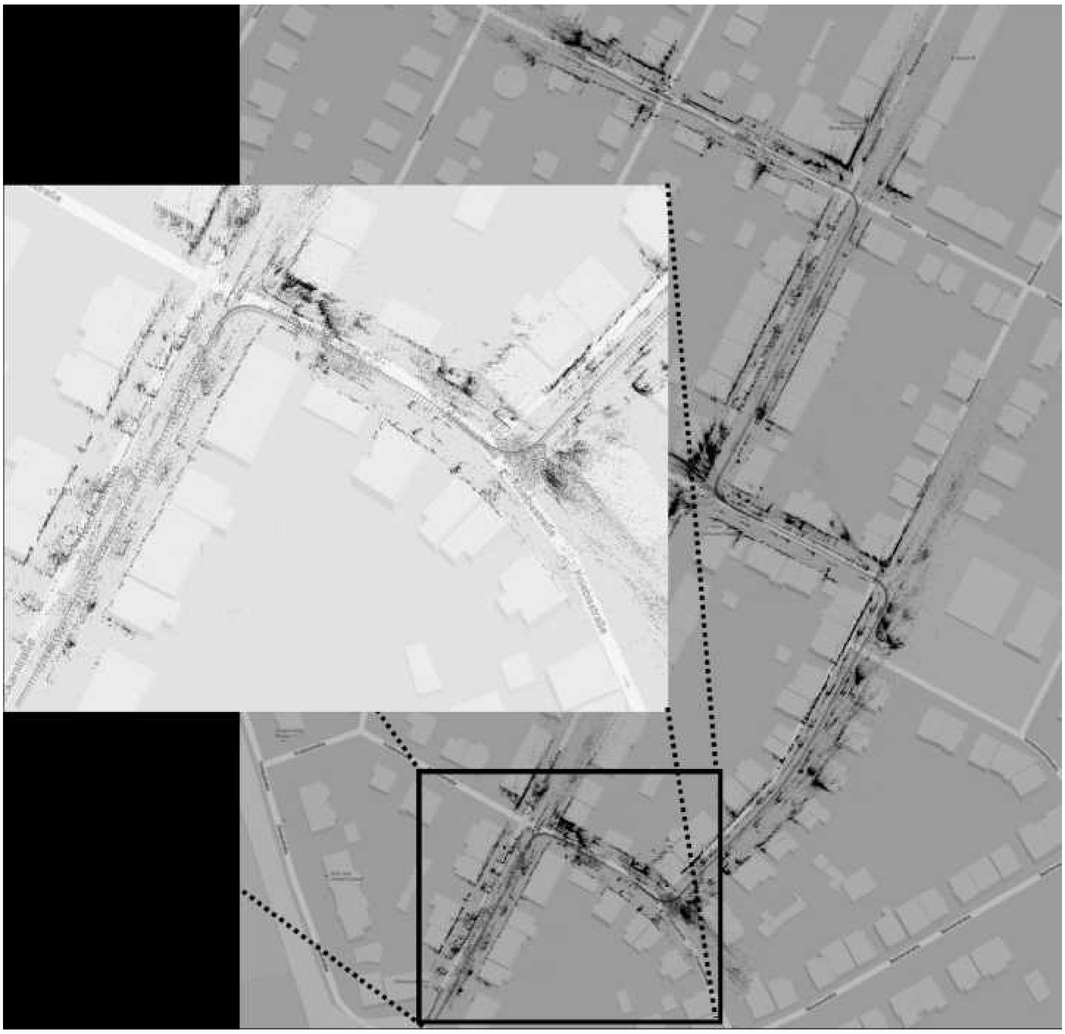

Figure 4.6: Bundle adjustment results for 1000 frames, showing camera centres (in red-dotted line) and 3D points, registered manually to the Google map.

The sample results of bundle adjustment after 10, 20, 30 and 40 frames have been processed (using sequence 01 from the KITTI dataset [62] as visualised in Fig. 4.5). The results of a longer portion of the sequence are shown in Fig. 4.6, along with the Google Map image for the region. The bundle results were aligned to the map manually, as only relative coordinates are generated in the bundle adjuster. The high degree of accuracy of the bundle adjuster can be seen visually from the reconstructed 3D points along the front of many buildings, which do not drift significantly even after more than a thousand frames have been processed.

4.3.2 Surface reconstruction

This section discusses the surface reconstruction from individual depth maps generated from a sequence of stereo pairs. In most cases, reconstruction of a large scale scene has been tried using a sparse set of feature points estimated from the bundle adjustment technique [1]. However, often a sparse structure is not sufficient for various tasks like precise object boundary prediction [155] and accurate modelling of the scene. An example of a sparse structure obtained using bundle adjustment from the stereo image sequence [62] is shown in Fig. 4.7 where we can visualize the sparsity of reconstructed points. To generate a dense structure, we compute individual depth maps from the stereo pairs, and fuse them into a surface model using the estimated camera poses from the bundle adjustment procedure. These depth estimates are fused into a Truncated Signed Distance volume [38], and the final meshed structure is recovered using marching tetrahedra algorithm [136]. The individual steps are described in detail below.

Depth Map Generation

To estimate the depth maps, the pixel correspondence need to be established between the pair of images. Pixels from two images are said to correspond if they represent the same scene element. Given a rectified image pair, the pixel correspondence between the left and right image pair can be determined by searching along the horizontal scan line, giving us the disparity value for the particular pixel. The rectification process makes the two image planes coplanar and the corresponding epipolar lines parallel to the horizontal image axes [61]. A depth map is computed from the disparity image as: $z_i = B.f/d_i$, where z_i and d_i are the depth and the disparity corresponding to the i^{th} pixel respectively. The terms B and f are the camera baseline and the focal length, which are obtained from the camera matrix. The disparity is computed using the Semi-Global Block Matching (SGBM) method [85]. The depth values are clipped based on a permissible disparity range. Some example of stereo images used in our experiments is shown in Fig. 4.8.

Fig. 4.9 shows an example stereo depth obtained using the semi-global block matching method of [85] using the publicly available implementation [18]. The colour indicates the depth with red regions are closer to the camera and the blue is far away from the viewing point. This depth map is used an as input to the TSDF based surface reconstruction method. We can observe the artefacts in the disparity image, due to the challenging nature of the

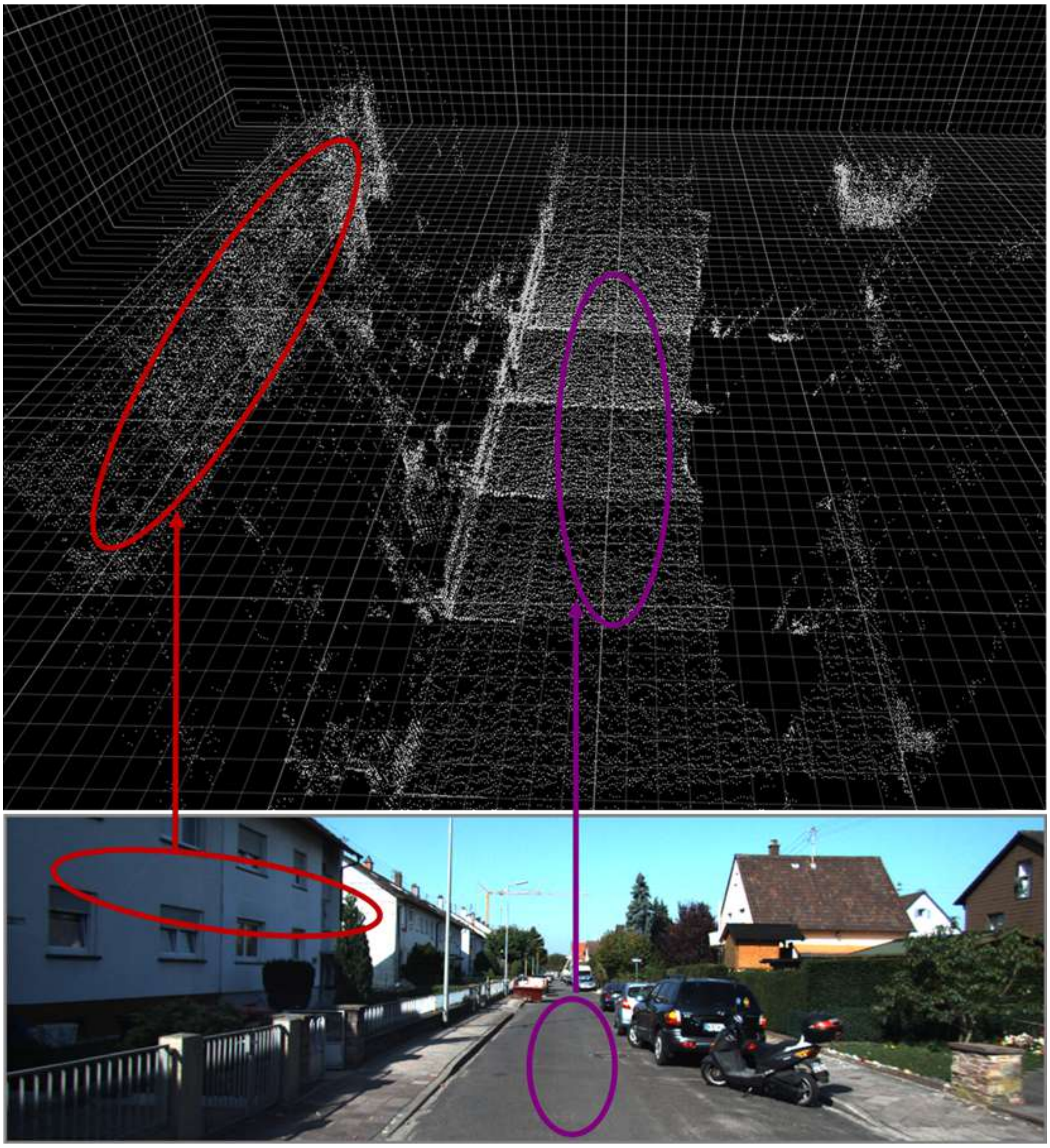

Figure 4.7: Sparse structure of the outdoor scene generated using bundle adjustment. The red ellipse shows the region corresponding to the building and the magenta ellipse shows the road region.

scene with high variation in illumination, shadows and shiny surface. As a result estimating surface from a single depth map will be prone to noise, making it necessary to fuse the information from multiple depth-maps for an improved surface estimation.

4.3.2.1 TSDF Volume Estimation

Each depth map with estimated camera parameters is fused incrementally into a single 3D reconstruction using the volumetric TSDF representation [38]. A signed distance function corresponds to the distance to the closest surface interface (zero crossing), with positive values corresponding to free space, and negative values corresponding to points behind the surface. Such a representation allows for an efficient registration of multiple surface

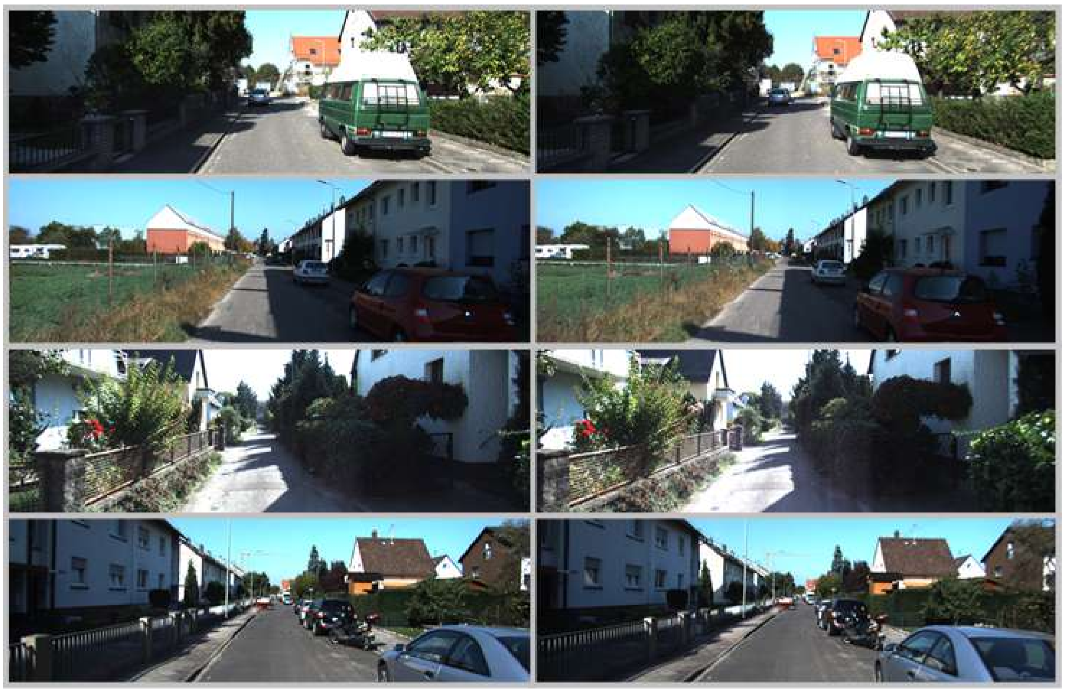

Figure 4.8: Examples of the input stereo imagery from the KITTI [62] dataset.

measurements, by globally averaging the distance measures from every depth map at each point in space.

We assume that the depth of the true surface lies within $\pm\mu$ of the observed values from the depth maps. So the points that lie in the visible space at a distance greater than μ are truncated to μ. The points beyond μ in the non-visible side are ignored. This is expressed in equation 4.3.1, where, R_k is a depth-map, x is the image projection of the voxel p via the camera projection matrix M, r is the distance of the point p along the optical axis of the camera from the camera centre and F_{R_k} is the TSDF value at point p in space computed as:

$$\begin{aligned} F_{R_k}(p) &= \Psi(R_k(x) - r), \\ x &= Mp, \\ \Psi(\eta) &= \begin{cases} \min(1, |\frac{\eta}{\mu}|)\,\mathrm{sgn}(\eta) & \text{iff} \quad \eta \geq -\mu \\ null & otherwise \end{cases}. \end{aligned} \tag{4.3.1}$$

The TSDF values are computed for each depth map. They are merged using an approach similar to [129] where an averaging of all TSDF's is performed. This smooths out the irregularities in the surface normals of the individual depth estimates computed from a single stereo pair. The global fusion of all depth-maps in the volume can be computed by weighted averaging of the individual TSDF measurements from each depth-map. [129] re-

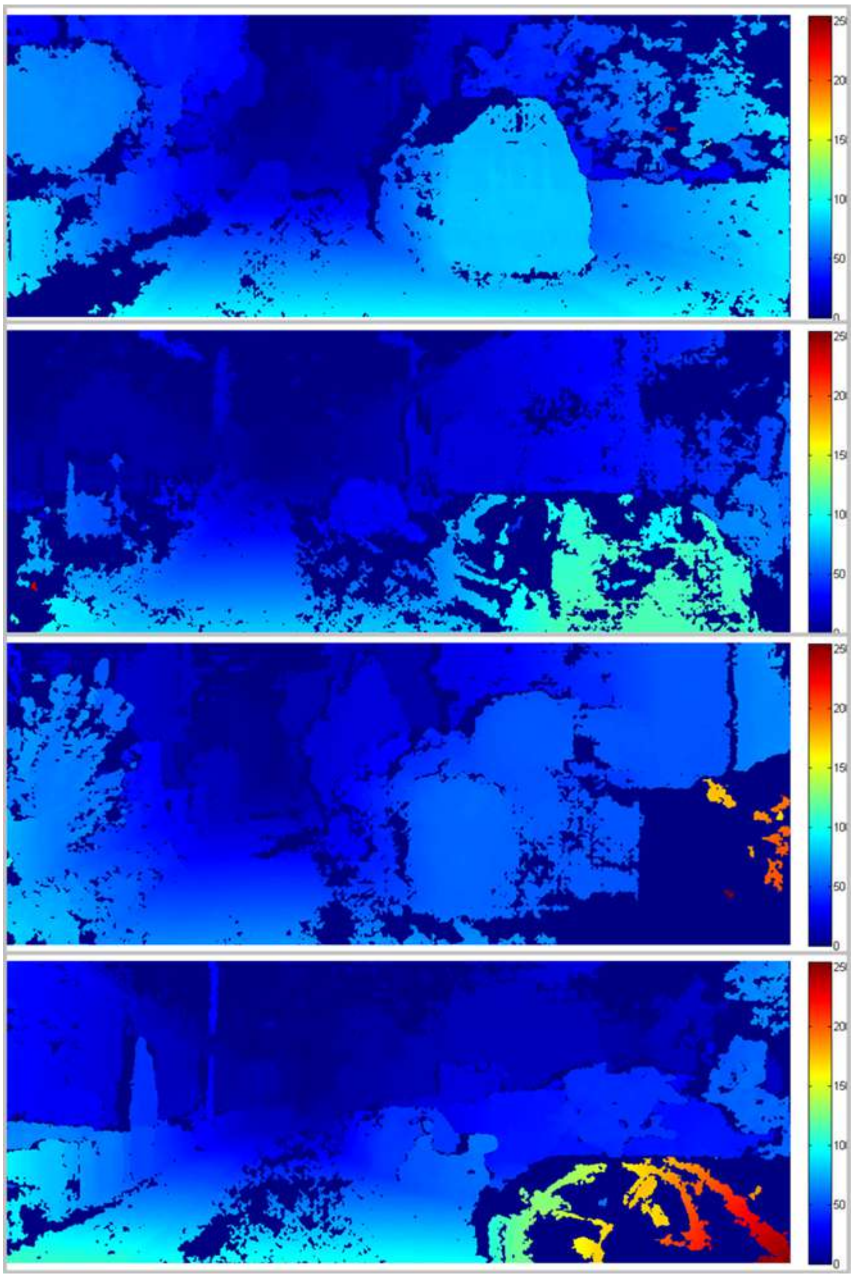

Figure 4.9: Depth-maps computed using semi global block matching algorithm [85] for the stereo pairs shown in Fig. 4.8. The colour indicates the depth with red regions are closer to the camera and the blue is far away from the viewing point. The depth maps are the the input to the TSDF based surface reconstruction method. Best viewed in colour.

ports that simple addition also produces desirable results which is our preferred option due to the irregularities in the surface normals of the individual depth-maps which are averaged out when fused across multiple frames. This is visualised in Fig. 4.10, where we show the surface generated from a single depth map in (a), and a surface generated after fusing multiple depth maps in (b). (c) Shows the image and the yellow boxes show the corresponding region in the image and the surface. The surface contains a large number of holes when generated from a single view, particularly along the viewing direction. In the zoomed in part, we can see the smoothness of the surface in the case where multiple depth maps are fused. Still, sometimes few holes are visible in the model along the line of sight, which is due to the sparsity of the input frames as the images are taken approximately at every 1 meter distance when the vehicle is in motion. However, in this approach the quality and accuracy of the final model can be improved by using more input imagery and depth-maps.

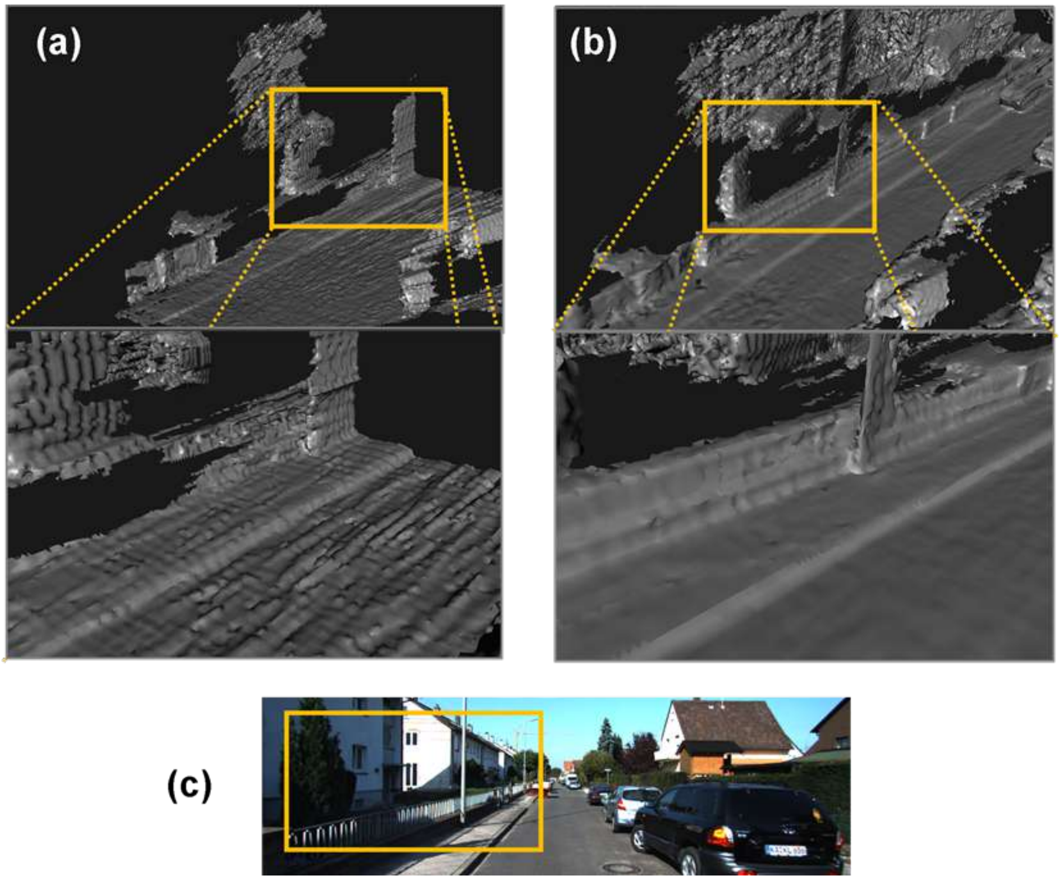

Figure 4.10: (a) Surface estimated from a single depth map, (b) surface generated after fusing multiple depth maps. (c) Corresponding image of the scene. The yellow boxes show the region in the image and the surface. The surface contains a large number of holes when generated from a single view, particularly along the viewing direction. In the zoomed in part, we can see the smoothness of the surface in the case where multiple depth maps are fused.

Sequential world scene update

The 3D models produced by our framework are reconstructed metrically where the distance measurements on the model correspond to real-world measurements. As we are reconstructing road scenes which can run from hundreds of meters to kilometres, it becomes difficult to fuse distance estimates due to memory limitations. To overcome this, we use an online grid update method. We consider an active $3 \times 3 \times 1$ grid of volumes at any time of the fusion. Each volume is further discretised into $n \times n \times n$ set of voxels. The value n controls the granularity of the resultant voxels. In our experiments, we have used $n = 150$. The active volume corresponds to $15 \times 15 \times 10$ metre^3 in the real world ($width \times length \times height$). The sampling resolution of the TSDF space is defined based on the scene geometry: higher sampling rates result in denser and more detailed models, but also higher computational and memory requirements. We allow for only one volume in the vertical direction, assuming minor elevation changes compared to the direction of motion. For every new depth map, the grid is updated in a sliding window fashion. As the vehicle track goes out of range of a volume to the next volume in the same column/row, the previous row/columns (or both for diagonal camera motion) of volumes is written to the disk and a new set is initialised. This allows us to handle arbitrarily long sequence without losing any granularity of the surface. This is explained visually in Fig. 4.11. The amber volumes indicate the active $3 \times 3 \times 1$ grid of volumes. The camera motion is shown by the arrow. As the camera moves from one volume to another, the row of green volumes is written off. The red volumes indicate the set which will be initialised in the next grid update.

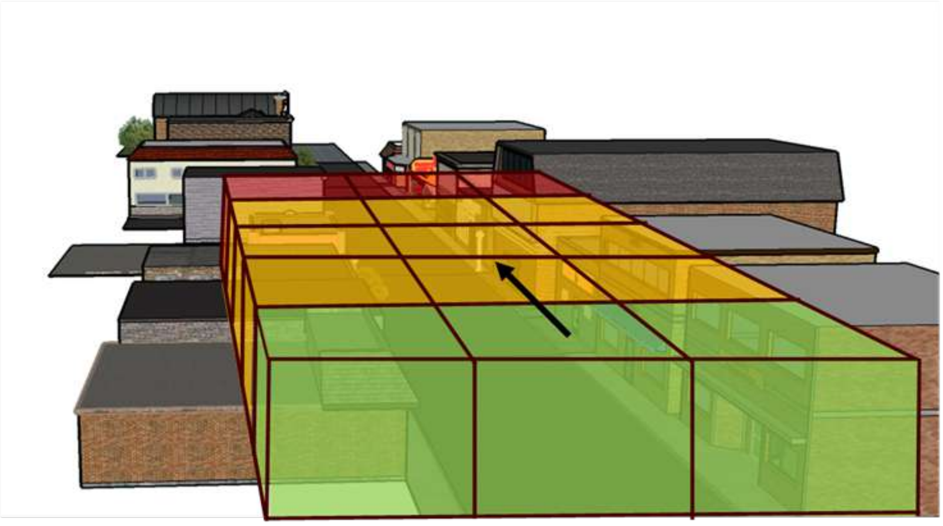

Figure 4.11: *Sequential world scene update.* The amber volumes indicate the active $3 \times 3 \times 1$ grid of volumes. The camera motion is shown by the arrow. As the camera moves from one volume to another, the row of green volumes is written off. The red volumes indicate the future set.

Surface estimation In order to obtain a complete meshed surface, we first infer an iso-surface from the TSDF field by finding all the zero crossings. Then we use the "Marching Tetrahedra" algorithm [136] to extract a triangulated mesh of the zero valued iso-surface. This algorithm has an improvement over the Lorensen and Cline [120] "Marching Cubes" algorithm being consistent topologically by resolving certain ambiguities in the cube configuration and hence resulting in a crack free surface. For further details of the Marching

Cubes/Tetrahedra, we refer the reader to [120, 136].

Fig. 4.12 shows an example output of the surface reconstruction and Fig. 4.13 shows the textured model. The reconstructed model captures fine details which is evident with the pavements, cars, road and vegetation. Once the mesh is obtained, the faces of the mesh are associated with object class labels which is described in the next section.

4.4 Semantic 3D Reconstruction of the World

In this section, we describe the object labelling method that we have used to annotate the reconstructed structure. A Conditional Random Field (CRF) based approach has been used to perform a per-pixel classification of the street level images similar to [108] followed by aggregation of the class labels to annotate the reconstruction.

4.4.1 CRF based image segmentation

Consider a set of random variables $\mathbf{X} = \{X_1, X_2, \ldots, X_n\}$, where each variable $X_i \in \mathbf{X}$ takes a value from a pre-defined label set $\mathcal{L} = \{l_1, l_2, \ldots, l_k\}$. The random field is defined over lattice $\mathcal{V} = \{1, 2, \ldots, N\}$, where each lattice point, or pixel, $i \in \mathcal{V}$ is associated with its corresponding random variable X_i. A labelling $\mathbf{x}$ refers to any possible assignment of labels to the random variables and takes values from the set $\mathbf{L} = \mathcal{L}^N$. For our application, the label set is $\mathcal{L}$ = {pavement, building, road, vehicle, vegetation, signage, wall/fence, sky, post/pole}. Let $\mathcal{N}$ be the neighbourhood system of the random field defined by sets $\mathcal{N}_i, \forall i \in \mathcal{V}$, where $\mathcal{N}_i$ denotes the set of all neighbours (usually the 4 or 8 nearest pixels) of the variable X_i. A clique c is defined as a set of random variables $\mathbf{X}_c$ which are conditionally dependent on each other. The corresponding Gibbs energy [118] for an image is given as:

$$\begin{aligned} E(\mathbf{x}) = \sum_{i \in \mathcal{V}} \psi_i(x_i) + \sum_{i \in \mathcal{V}, j \in \mathcal{N}_i} \psi_{ij}(x_i, x_j) + \\ \sum_{i \in \mathcal{V}, j \in \mathcal{N}_i} \psi^d_{ij}(x_i, x_j) + \sum_{c \in \mathcal{C}} \psi_c(\mathbf{x}_c), \end{aligned} \tag{4.4.1}$$

The energy minimisation problem is solved using a graph-cut based Alpha Expansion algorithm [17]. This framework combines features and classifiers at different levels of the hierarchy (pixels and superpixels), with the unary potential capturing the pixel level information and the higher order term forcing consistency among a group of pixels denoted by a superpixel. We now describe the constituent potentials in the energy function.

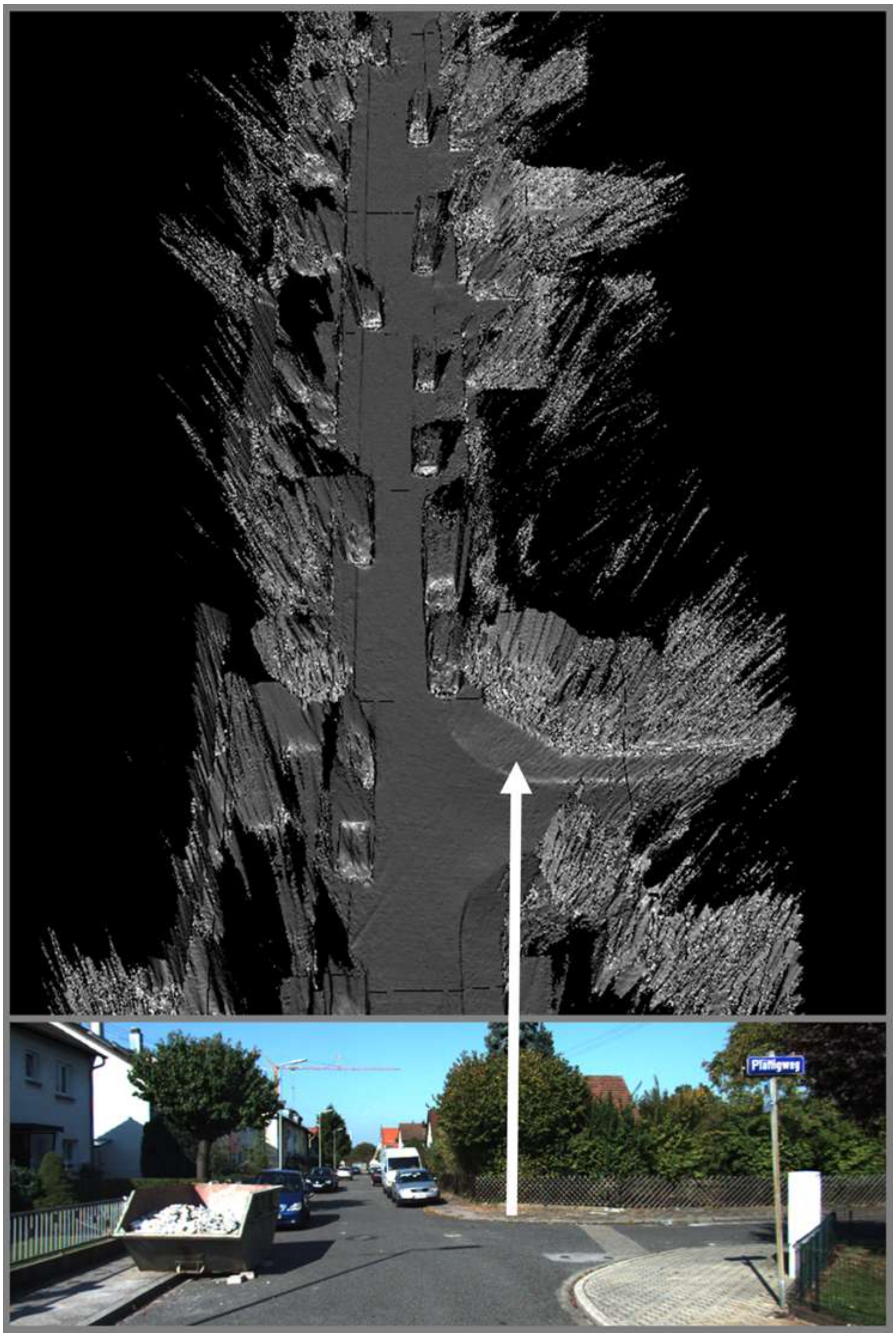

Figure 4.12: Volumetric surface reconstruction: Top figure shows the 3D surface reconstruction over 250 frames (KITTI sequence 15, frames 1-250) with street image shown at the bottom. The arrow highlights the relief of the pavement which is correctly captured in the 3D model.

Unary potential The unary potential ψ_i describes the cost of a single pixel taking a particular object class label. The form of the unary potential is the negative logarithm of the normalised output of a boosted classifier. We use the multi-feature variant of the TextonBoost algorithm [108].

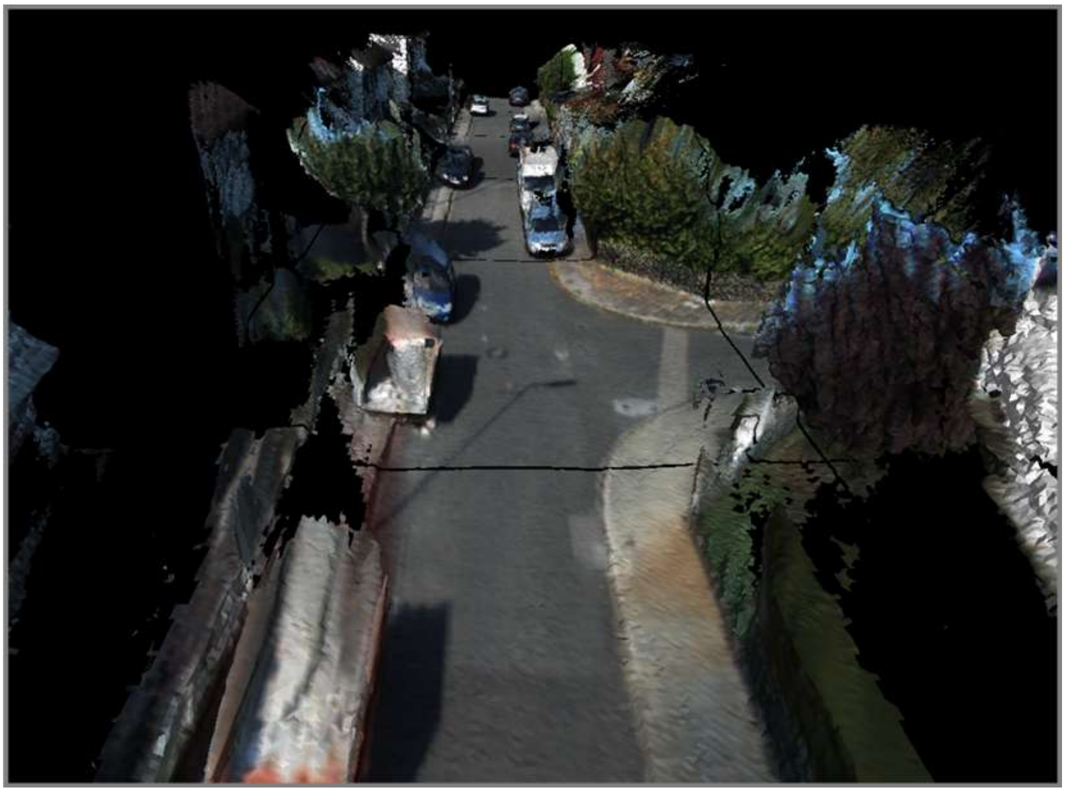

Figure 4.13: Textured reconstructed model. The details in the form of street layout, buildings, parked cars and trees and lamp post are evident in the model. Note the lack of surface in areas are that are hidden from the stereo camera views such as the shadow like areas behind the skip. Having more input depth-maps from different viewpoints will reduce the unobserved areas.

Pairwise potentials The pairwise term ψ_{ij} is an edge preserving potential which induces smoothness in the solution by encouraging neighbouring pixels take the same label. It is defined over a neighbourhood of eight pixels taking the form of a contrast sensitive Potts model:

$$\psi_{ij}(x_i, x_j) = \begin{cases} 0 & \text{if } x_i = x_j, \\ g(i,j) & \text{otherwise,} \end{cases} \tag{4.4.2}$$

where the function $g(i,j)$ is an edge feature based on the difference in colours of neighbouring pixels [108], defined as:

$$g(i,j) = \theta_p + \theta_v \exp(-\theta_\beta ||I_i - I_j||_2^2), \tag{4.4.3}$$

where I_i and I_j are the colour vectors of pixels i and j respectively. θ_p, θ_v, $\theta_\beta \geq 0$ are model parameters set by cross validation. The disparity potential $\psi^d_{ij}(x_i, x_j)$ takes the same form as the pairwise potential,operating on the disparity image, where neighbouring pixels with similar disparity are encouraged to take same labels. This takes the form of

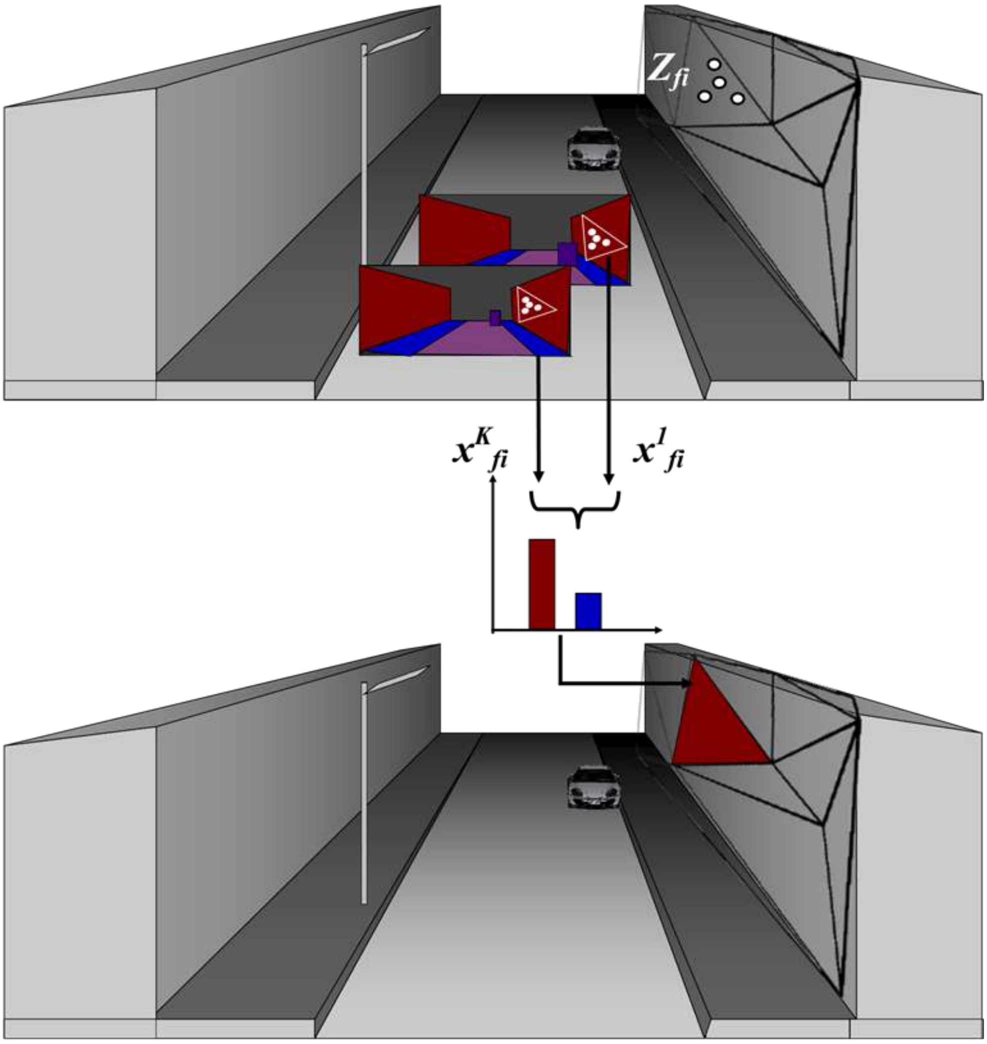

Figure 4.14: *Label fusion scheme.* The white dots Z_{f_i} are the sampled points on the face of a mesh triangle. The corresponding image points $x^j_{f_i}$ are obtained by projecting the face points onto the labelled street image. The mesh is labelled with the class labels with the most votes on the mesh face.

$$\psi^d_{ij}(x_i, x_j) = \begin{cases} 0 & \text{if } x_i = x_j, \\ d(i,j) & \text{otherwise,} \end{cases} \tag{4.4.4}$$

where the function $d(i, j)$ takes a similar form as Eqn. 4.4.3 but operates on the difference of the disparity value for the pixels i and j. Adding information from both image and disparity domain helps us to achieve more consistent results (we give equal importance to both these terms). An alternative potential based on the full depth map could be considered, however the back projected points can be sparse in the image domain, which are not suited for the dense per-pixel inference.

Higher Order Potential The higher order term $\psi_c(\mathbf{x}_c)$ describes potentials defined over overlapping superpixels as described in [108]. The potential encourages the pixels in a given segment to take the same label and penalises partial inconsistency of superpixels, capturing a longer range contextual information.

4.4.2 Semantic Label Fusion

Once the street level image segmentations are obtained, the label predictions are fused as follows: for each triangulated face f in the generated mesh model, we randomly sample s points (Z_{f_s}) on the face. Instead of considering all the points on the face triangle, sampling a few random points is fast, provides robustness to noise and avoids aliasing problems. This is better than considering only the face vertices labels in the image because in the final model the face vertices' labels are likely to coincide with the object class boundary. The points are projected back in to K images using the estimated camera pose (P^k), resulting in a set of image points ($x^k_{f_s}$). The label predictions for all those image points are aggregated and the majority label is taken as the label of the face in the output model. The label histogram Z_f for the face f is given as:

$$Z_f = \frac{1 + \sum_{s \in Z_{f_s}} \sum_{k \in K} \delta(x^k_s = l)}{|L|} \tag{4.4.5}$$

where l is a label in our label set $\mathcal{L}$ and $|L|$ is the number of the labels in the set. This provides a naive probability estimation for a mesh face taking label l. We set $s = 10$ in our experiments. The fusion step is illustrated in Fig. 4.14. The white dots in the model (top) are projected back to the images. The class label prediction of all those image points are aggregated to generate a histogram of the object label distribution and the final label prediction is made corresponding to the maximal object class label.

4.5 Experiments

4.5.1 KITTI semantic labelled dataset

To demonstrate the effectiveness of our proposed system, we have used the publicly available KITTI dataset [62] for our experiments. The images are 1241×376 at full resolution. They are captured using a specialised car in urban, residential and highway locations, making it a diverse and challenging real world dataset. We have manually annotated a set of 45

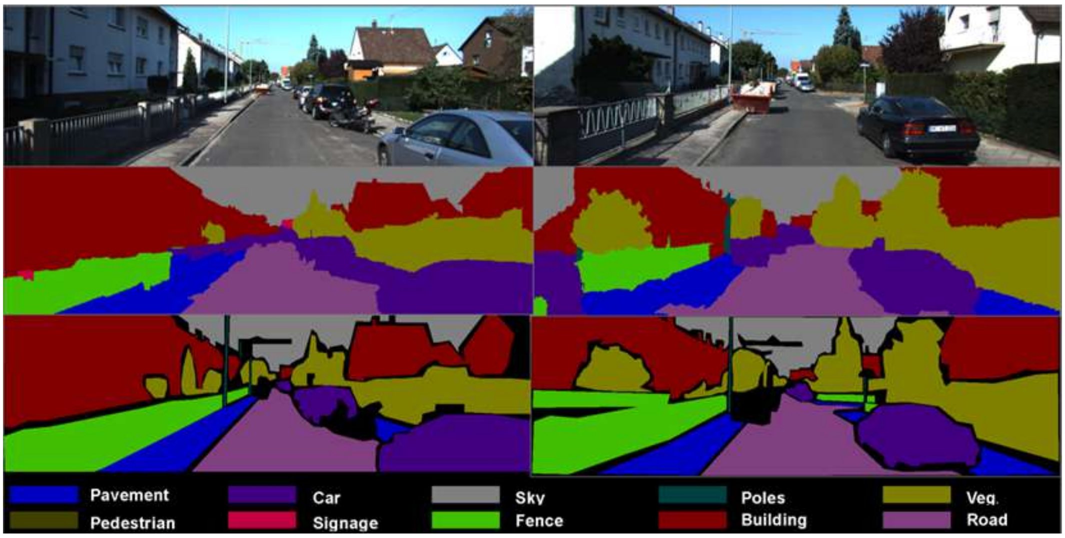

Figure 4.15: *Semantic image segmentation*: The top row shows the input street-level images and the middle row shows the output of the CRF labeller. The bottom row shows the corresponding ground truth for the images. These semantic labels are the intermediate results for our system which are aggregated to label the final model.

images for training and 25 for testing with per-pixel object class labels. The images were selected so that each of the object class is represented adequately. The training and the test images were selected from the training and the test image sequences available from [62]. The class labels are road, building, vehicle, pedestrian, pavement, tree, sky, signage, post/pole, wall/fence. We have made these labels publicly available for the vision community[1].

4.5.2 Results

We evaluate our camera pose estimation using two metrics, translation error (%) and rotation error (degrees/m) over the increasing number of frames with the ground truth provided by [62] for sequence 8 (see table 4.1). We have evaluated our sliding window bundle method (*full*) and a *fast* variant of that. The *fast* method performs the Levenberg-Marquardt minimisation for two successive frame pairs to estimate camera pose and the feature points. As expected the average error for the *full* method reduces with increasing number of frames. Also the absolute magnitude of error for the *fast* method is higher than the *full* method.

Our *full* bundle method runs in around 3.5 seconds per frame on a single core machine. However the *fast* method runs at approximately 4 fps. The feature extraction takes about 0.02 seconds, feature matching (both stereo and frame to frame) takes 0.2 seconds per frame. For disparity map extraction we use OpenCV implementation of semi-global block matching stereo [85] which takes around 0.5 seconds for the full sized 1280 × 376 image.

[1] available at http://www.robots.ox.ac.uk/˜tvg/projects/SemanticUrbanModelling/index.php

Table 4.1: Odometry results: Translational and rotational error with increasing number of frames.

	Trans. Error (%)		Rot. error (degs/m)	
Length (frames)	fast	full	fast	full
5	12.2	12.15	0.035	0.032
10	11.84	11.82	0.028	0.026
50	8.262	8.343	0.021	0.018
100	4.7	4.711	0.019	0.013
150	3.951	3.736	0.017	0.01
200	3.997	3.409	0.015	0.009
250	4.226	3.209	0.013	0.007
300	4.633	3.06	0.012	0.007
350	5.057	2.939	0.011	0.006
400	5.407	2.854	0.01	0.004

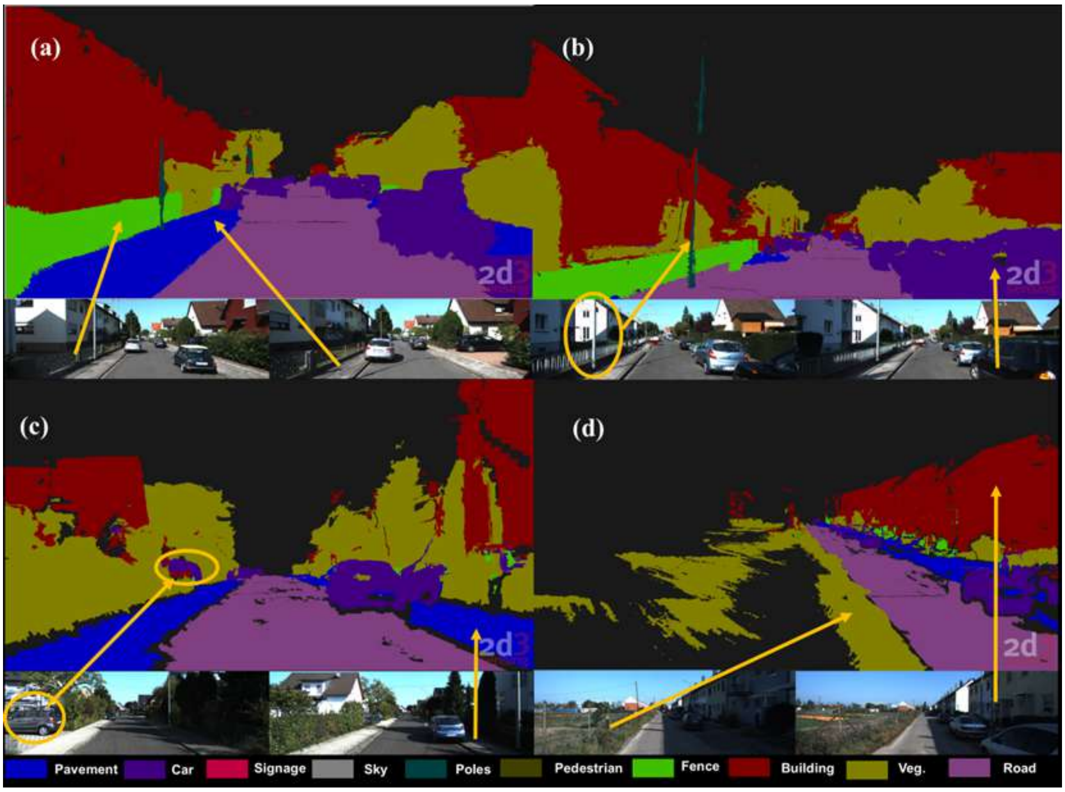

Figure 4.16: *Closeup view of the 3D model.* The arrows relate the image locations and the positions in the 3D model. In (a) we see the fence and the pavement in both the image and the model. (b) show that the generated model is able to capture thin objects like posts. The circle in the image (c) shows the car in the image and the final model. In (d) arrows shows the vegetation and the buildings respectively.

The TSDF stage is highly parallelisable as each voxel in the TSDF volume can be treated separately. Currently, our implementation considers around 30 million voxels per $3 \times 3 \times 1$ grid of TSDF volume, running on a single core. All these steps can be optimised using a GPU implementation [149].

Fig. 4.15 shows the qualitative results of the street level image segmentation using our

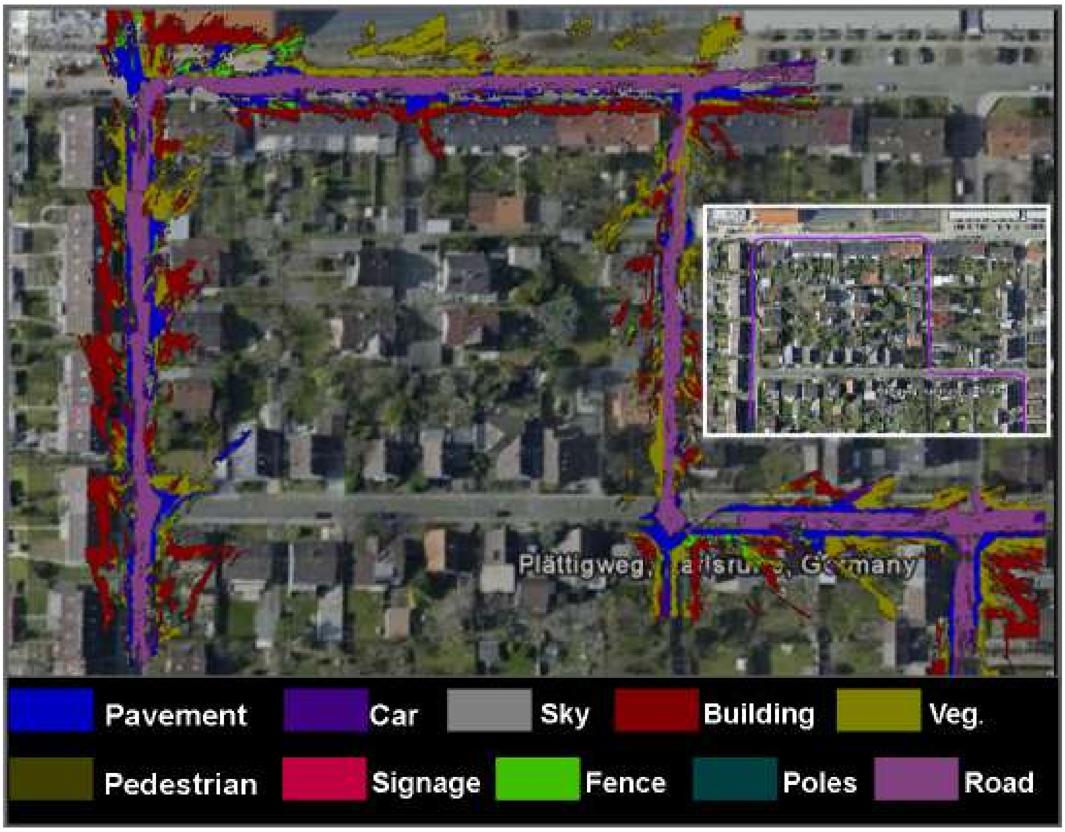

Figure 4.17: Semantic model of the reconstructed scene overlaid with the corresponding Google Earth image. The inset image shows the Google earth track of the vehicle.

CRF framework. The first row shows the street-level images captured by the vehicle. The second and the third column show the semantic image segmentation of the street images and the corresponding ground truth. A qualitative view of our 3D semantic model is shown in Fig. 4.16. The arrows relate the positions in the 3D model and the corresponding images. We can see in (a), both fence and pavement are present in the model as well as the associated images. The model can capture long and thin objects like posts as shown in (b). The circle in the image (c) shows the car in the image, which has been captured correctly in the final model. In (d) arrows show the vegetation and the car respectively. In Fig. 4.17, a large scale semantic reconstruction, comprising of 800 frames from KITTI sequence 15, is illustrated. An overhead view of the reconstructed model is shown along with the corresponding Google Earth image. The inset image shows the actual path of the vehicle (manually drawn).

Next we describe the quantitative evaluation. For object level classification, we use the approach similar to §3.4.2, where we project the reconstructed semantic model back into the image domain and compare with the ground truth labellings. The model points are projected back using the estimated camera poses. Points in the reconstructed model that are far away from the particular camera ($>$ 20m) are ignored. The projection is illustrated in Fig.

4.18, where the top (a) shows the semantic model, (b) shows an image of the corresponding scene. The model labels are back projected using the estimated camera poses and shown in (c), while (d) shows the corresponding ground truth images for the object class labels. We show quantitative results on two metrics, *Recall*=$\frac{\text{True Positive}}{\text{True Positive + False Negative}}$ and *Intersection* vs *Union*=$\frac{\text{True Positive}}{\text{True Positive + False Negative + False Positive}}$ measures, for both street image segmentation and semantic modelling. Our results are summarised in table 4.2. Here, 'Global' refers to the overall percentage of pixels correctly classified, and 'Average' is the average of the per class measures. In this evaluation we have not considered the class 'sky' which is not captured in the model. Due to lack of test data we have also not included the classes 'pedestrian' and 'signage' in our evaluation. We compare the accuracy of the structures generated from the *full* and the *fast* sliding window bundle adjustment methods, along with the street level image labelling. We observe a near similar performance of object labelling for the two semantic models (generated from fast and full sliding window bundle adjustment). This is due to the following reason: the object labelling is performed in the image domain and then aggregated into the 3D model. This approach does not effectively utilise the 3D scene during labelling, motivating us further to perform labelling in the 3D which we discuss in the next chapter. We have also shown the performance evaluation for the street image segmentation. Due to the errors in estimation of the camera, the performance of the semantic model is marginally lower than the street image segmentation. This specially affects the thin object classes like 'poles/posts' where small error in projection leads to large errors in the evaluation. Classes like 'vegetation', where the surface measurement tends to be noisy, have increased error in classification. Our system is designed to model static objects in the scene, which causes an adverse effect when considering moving objects such as cars which is reflected in the results.

To evaluate the accuracy of the structure, we use the ground truth depth measurement from Velodyne lasers as provided in [62]. The depth measurements from both the Velodyne lasers (δ_i^g) and our generated model (δ_i) are projected back into the image and evaluated. We measure the number of pixels that satisfy $|\delta_i - \delta_i^g| \geq \delta$, where δ is the allowed error in pixels. The results of our method are shown in Fig. 4.19. δ ranges between 1 to 8 pixels. It can be noted that the estimated structural accuracy at $\delta = 5$ pixels is around 88%, which indicates the performance of the structure estimation.

Table 4.2: Semantic Evaluation: Pixel-wise percentage accuracy on the test set. We compare the Image labelling, two method for generation of Semantic model (full sliding window and fast sliding window)

Method	Building	Vegetation	Car	Road	wall/fence	Pavement	Pots/Pole	**Average**	**Global**
Recall measure									
Street Image Segmentation	97.0	93.4	93.9	98.3	48.5	91.3	49.3	81.68	88.4
Semantic Model full (back projected)	96.1	86.9	88.5	97.8	46.1	86.5	38.4	77.15	85.1
Semantic Model fast (back projected)	95.0	89.0	87.0	98.0	46.0	88.5	39.1	77.4	85.4
Intersection *vs* union									
Street Image Segmentation	86.1	82.8	78.0	94.3	47.5	73.4	39.5	71.65	
Semantic Model full (back projected)	83.8	74.3	63.5	96.3	45.2	68.4	29.1	65.8	
Semantic Model fast (back projected)	80.0	77.5	62.6	96.8	45.2	71.4	32.3	66.5	

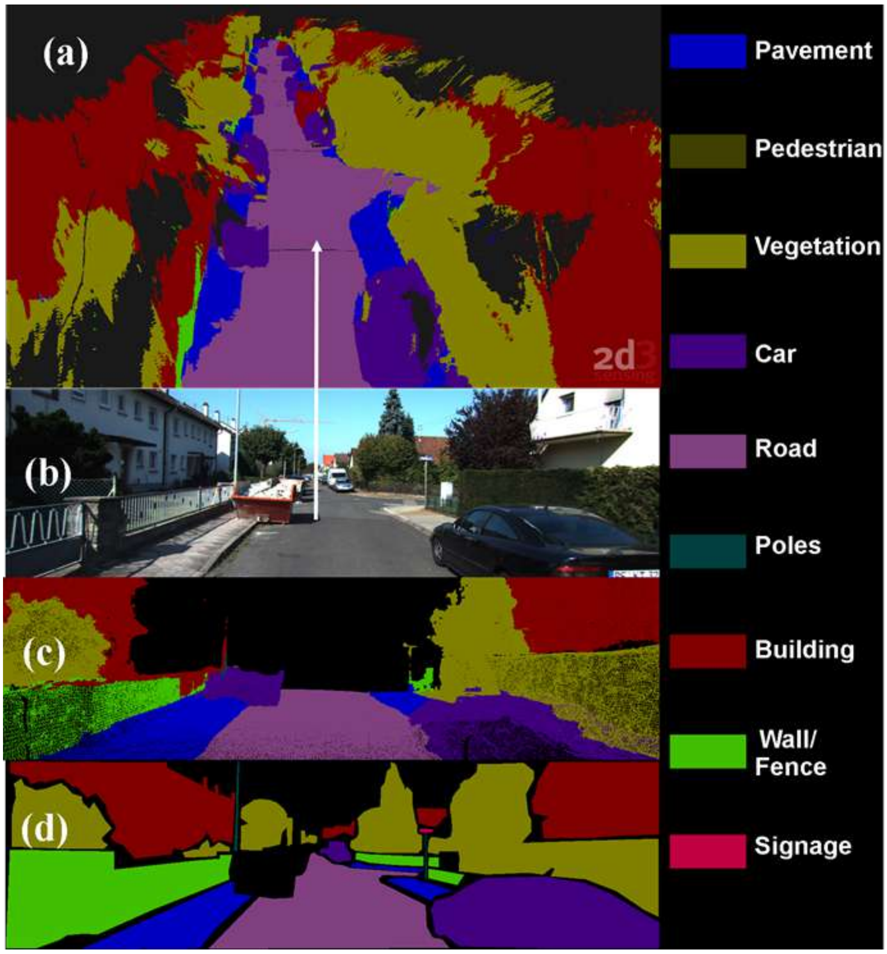

Figure 4.18: *3D semantic model evaluation.* (a) Shows the 3D semantic model. (b) Shows a test image, (c) shows the corresponding image with labels back-projected from the 3D model and (d) ground truth image.

4.6 Discussion

We have presented a novel computer vision-based system to generate a dense 3D semantic reconstruction of an urban environment. The input to our system is a stereo video feed from a moving vehicle. Our system robustly tracks the camera poses which are used to fuse the stereo depth-maps into a TSDF volume. The iso-surface in the TSDF space corresponding to the scene model is then augmented with semantic labels. This is done by fusing CRF-based semantic inference results using the input frames. We have demonstrated desirable results both qualitatively and quantitatively on a large urban sequence from the KITTI dataset [62]. We have also generated a labelled dataset comprising object class labels for the same, and made them publicly available.

The current method is highly parallelisable, where the fusion of the depth maps into the voxels and the corresponding voxel updates can be performed independently. This makes it very suitable for leveraging the parallel architecture of the GPU for surface generation.

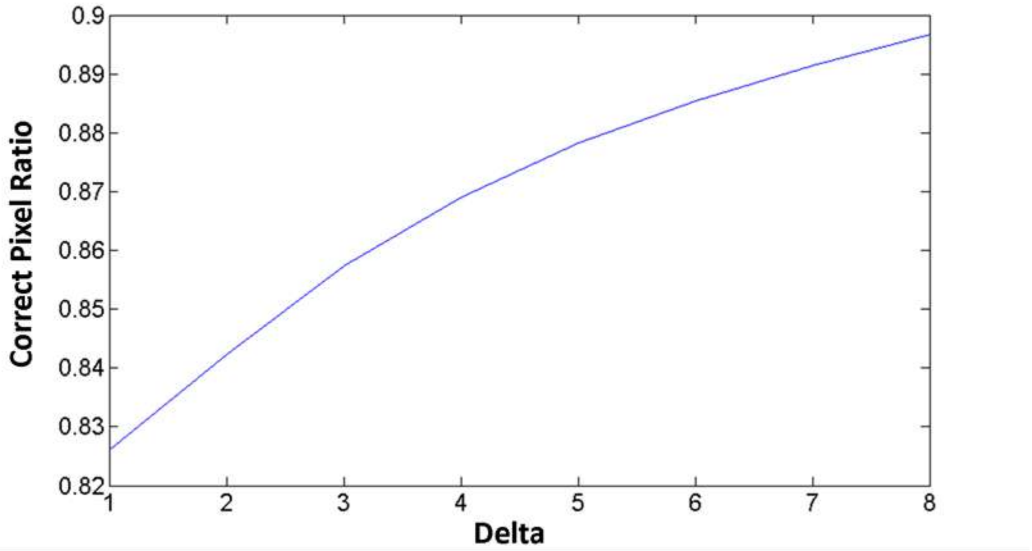

Figure 4.19: *Model depth evaluation.*

However the main drawback of the proposed approach is that the labelling is performed in the image domain. Firstly, this slows the entire process as we need to perform labelling over all the images. This is because, while creating an accurate 3D model of the scene, a large number of views/images are required. In most of the cases, the views are overlapping and contain redundant information to process. This causes a significant slowdown of the system in the labelling phase. Secondly, labelling on individual images separately results in inconsistency even when they depict the same part of the scene, and finally this method does not effectively utilize the structure of the scene. In the next chapter, we show how to exploit the scene structure for labelling, making the overall system accurate and efficient.

Acknowledgements

I would like to thank Eric Greveson and Ali Shahrokni from 2d3 Sensing Ltd. for discussions and helping with the design and implementation of the bundle-adjustment part for estimating camera poses. The work was done partly during an internship at 2d3 Sensing Ltd.

Chapter 5

Mesh Based Inference for Semantic Modelling of Outdoor Scenes

5.1 Introduction

In the last chapter, we showed how a scene can be reconstructed from stereo images annotated with object class labels to give a semantic meaning to the model. However, the labelling was performed on a local image level and then aggregated over the sequence in a naive Bayes fashion to generate the corresponding semantic model. Often a large number of overlapping images describing the scene, are required to perform an accurate reconstruction. As a result, when object labelling is performed on these images, it results in performing inference on redundant information, making the labelling process slow. Also, treating the images independently often result in inconsistent object class labels. This is explained in Fig. 5.1, where consecutive images of a road scene sequence are shown with inconsistent object class labels. Finally, when an object class labelling is performed on the images, the geometric layout of the objects in the scene is not captured. This is because an image is essentially a projection of 3D into a 2D plane. For example, in a road scene image, pixels denoting objects like poles/post might be adjacent to pixels representing a building, while in the scene they are far apart. This additionally motivates us to perform labelling over the structure to overcome the shortcomings of image based object recognition.

In this chapter, we propose a method to generate object labelling in 3D. Our method builds a triangulated mesh representation of the scene from multiple depth estimates. The meshed representation of the scene enables us to capture the surface and the geometric layout of the objects in the scene. In contrast to the point cloud based scene modelling, a mesh captures the scene connectivity and provides a dense representation of the scene through the faces in the mesh. Our goal is to semantically annotate the mesh. This form of annotated 3D representation is necessary to allow robotic platforms to understand, interact and navigate in a structured indoor environment [14, 83, 163] or outdoor scenes [7, 39, 184]. Towards this aim, we define a CRF over the mesh and perform inference on the mesh, providing us a consistent labelling as compared to the previous chapter, where we performed the labelling on the images and then combined them to give a labelled structure. Our method generates object hypothesis from multiple images and combines them in a robust weighted fashion. Moreover, by virtue of working on meshes, the proposed approach is *highly efficient* in the inference stage. We demonstrate the robustness of our framework for large scale outdoor scenes on KITTI odometry dataset [62]. We observe a significant speed-up in the inference stage by performing labelling on the mesh (25×), and additionally achieve higher accura-

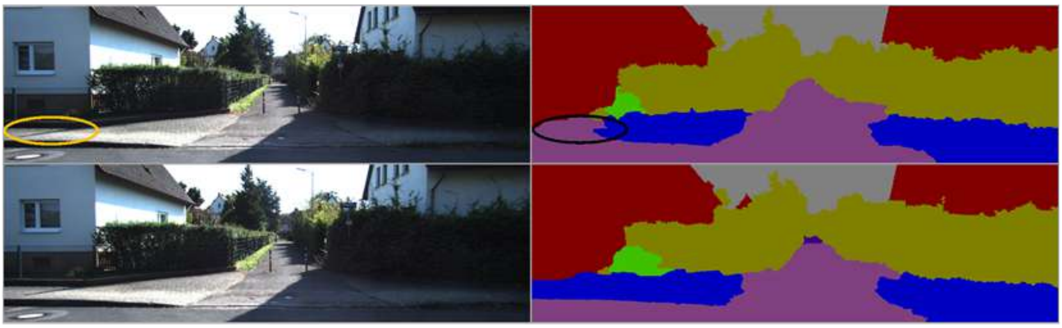

Figure 5.1: Consecutive images in a sequence with inconsistent object labels. The left column shows two consecutive input images from the sequence and the right column show the corresponding object labels. The pavement in the top image have been mislabelled (highlighted in the yellow ellipse) while correctly segmented in the bottom image. Also, we can see when the images are taken in a sequence, often have redundancy in information.

cies. Furthermore, we also demonstrate how the meshed scene with semantic information can be used for autonomous navigation or for synthesizing new objects in a scene.

5.1.1 Outline of the Chapter

In § 5.2 we briefly describe some of the related work in object labelling and surface reconstruction. In § 5.3 we describe the CRF inference in the mesh representing a scene. § 5.4 describe the experiments and the chapter concludes with a discussion in § 5.5.

5.2 Related Work

Over the last four decades there has been a significant development in visible surface reconstruction from a sequence of images. Most of the early work included the reconstruction from intrinsic images by Barrow and Tenenbaum [9] and Marr's 2.5 representation [122]. In these approaches, an elevation map, orientation was generated by exploiting the grey level changes in the images. In other early attempts, range data [6], stereo [58] and shading/lighting information [87, 175] were used to depict the surface of the object in consideration. More recently, accurate whole scene 3D representation was targeted with voxel representations [38, 153, 167], polygonal meshes [57], level sets [54, 56, 97]. In the voxel based representation, the entire 3D space is regularly sampled and an occupancy grid is generated indicating the free or occupied space. Often, a set of image silhouettes are fused and the free space is carved out giving the occupied region in the final 3D reconstruction [50, 96]. The level set methods minimise a set of constraints on the free space of the region, by starting initially with a large volume and shrinking inwards. Similarly a number of methods associate a cost on the 3D volume and then extract the surface by defining a volumetric MRF [148, 181] or recover the structure by matching features across a sequence of images [170]. In most of these methods the final 3D structure was represented either as a set of point clouds or a set of polygonal surface meshes. Such a mesh based representation has the advantage of being dense, but compact, and capture the details of the real scene more accurately.

The problem of semantic object labelling has been studied extensively and has seen major improvements [108, 158, 179]. However, most of these algorithms work in the image domain, where every pixel in the image is classified with an object label. With respect to outdoor urban scenes, most labelling methods concentrated on classification in the image domain [23, 111, 164] or using a coarse level 3D interpretation [49]. In [154] a semantic 3D reconstruction is generated, but the object labelling is performed in the image domain, and then projected to the model resulting in slow inference and inconsistent results. On the other hand, most of the mesh segmentation methods are based on simple geometric constraints on the input mesh [30, 90, 98, 105, 119]. In these approaches,partitioning the mesh was based on geometric criteria like concavity, skeleton topology, fitting shape primitives are applied or geometric features like shape, diameter, curvature tensor, geodesic distances are used to partition the mesh. The advent of inexpensive depth sensors has made it possible to reconstruct indoor scenes from streams of depth and RGB data [129]. Attempts to label such indoor scene reconstructions has been made in [159] where labelling is performed in the image domain assisted with a 3D distance prior to aid their classification. This was extended in [127] where object surface and support relations of an indoor scene were used as a prior to perform indoor scene segmentation. However both of these methods work in the image domain and ignore the structural layout of the objects in the scene. Similarly, approaches like [104] were proposed to label indoor scene point clouds. However, the method uses geometric clustering based on the distance in the real world to establish pairwise connections. This can produce inconsistency along object boundaries as the pairwise connections do not follow the geodesic distance and hence might result in incorrect neighbourhood.

In this chapter, we tackle the problem of semantic scene reconstruction for outdoor scenes in 3D space by performing labelling over meshed structure effectively utilising the object layout and mesh connectivity. Our approach to semantic mesh segmentation is illustrated in Fig. 5.2: (a) shows the input to our method is a sequence of stereo images. The depth estimates are generated from these individual stereo images and are fused into a volu-

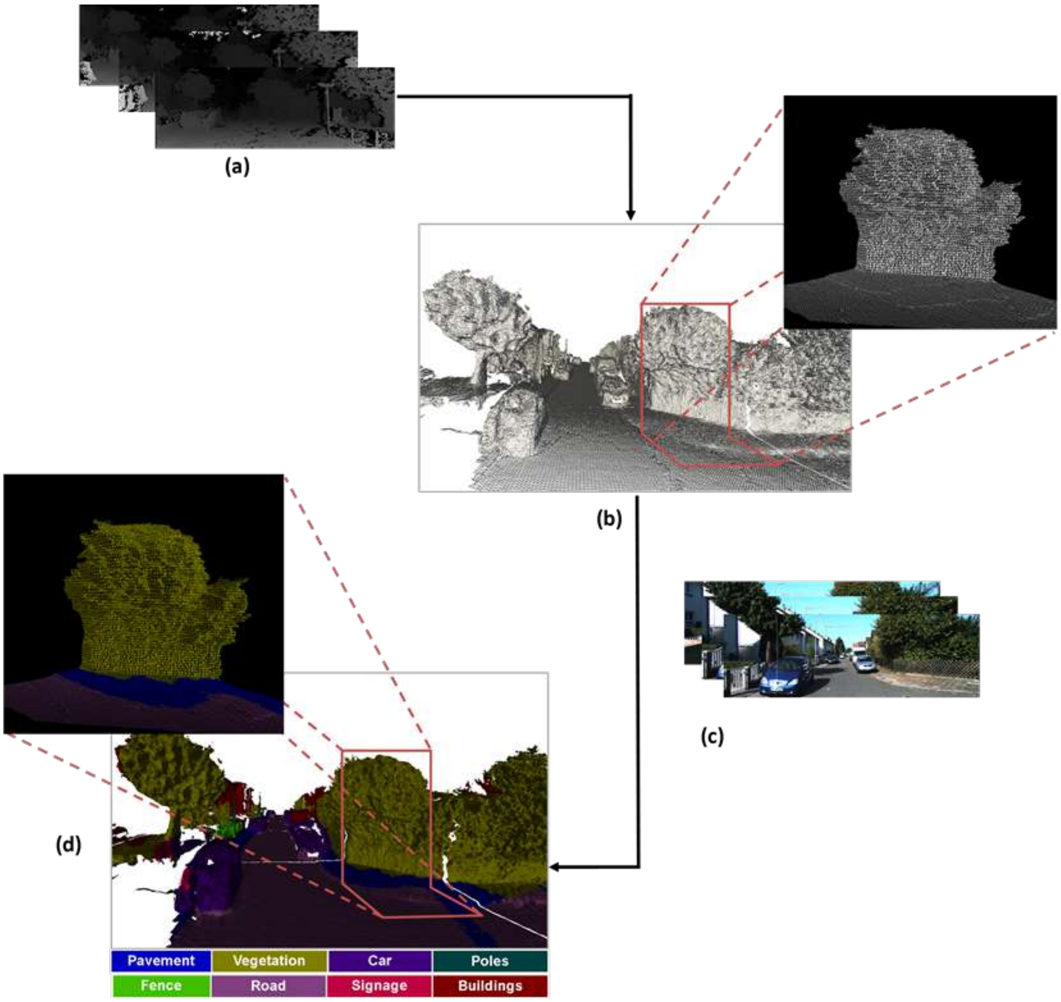

Figure 5.2: Semantic Mesh Segmentation: (a) shows the input to our system which is a sequence of images with depth estimates (here we show the images of a street scene captured using a stereo camera mounted on a vehicle [62]), (b) the depth estimates are used to generate a mesh based representation of the scene (enlarged in the inset image), (c) the system combines image level information and geometric connectivity information from mesh to perform object class labelling, (d) object class labelling is performed by establishing a conditional random field (CRF) on the mesh, with local neighbourhood interactions defined by the neighbourhood for each vertex in the mesh. (Best viewed in colour)

metric consistent scene. (b) A triangulated mesh is generated from the volumetric 3D space by the method of [136], enabling us to capture the scene's inherent geometry. The enlarged inset image of the mesh shows how the surface of the scene is captured in the mesh. (c) Appearance cues from the street level images are fused into a CRF model that is defined over the over the scene mesh (d), effectively exploiting the scene geometry. In the next section we explain our mesh based object labelling method in details.

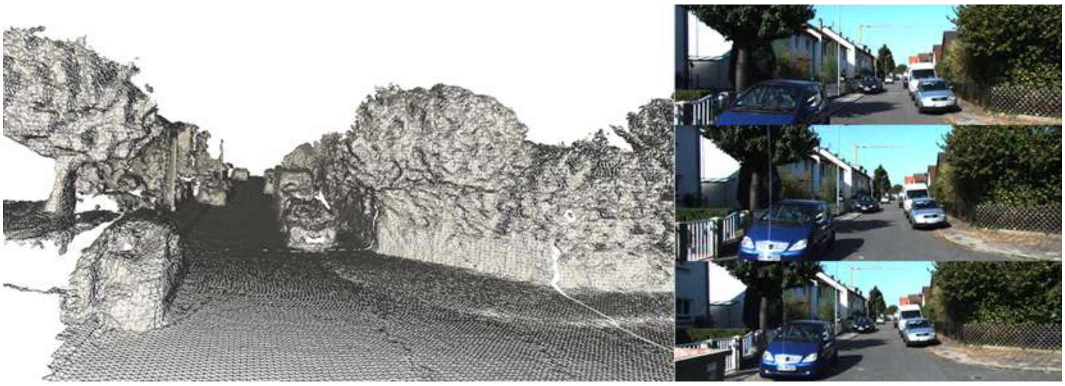

Figure 5.3: Meshed Representation of the scene along the corresponding images.

5.3 Mesh Based Inference

In this section we describe the mesh labelling approach. We first describe the mesh estimation process followed by the CRF model definition on the mesh.

5.3.1 Mesh Estimation

To estimate the meshed representation, we use the framework described in the previous chapter (§4.3.2) operating on a sequence of stereo image pairs for outdoor scenes. The depth estimates are fused incrementally into a single 3D reconstruction using the volumetric TSDF representation [38]. A signed distance function corresponds to the signed distance to the closest surface interface (zero crossing), with positive increasing values corresponding to free space and negative decreasing values corresponding to points inside the surface. The representation allows for the efficient registration of multiple surface measurements, obtained from different depth maps by averaging the distance measures from every depth map at each point in space. This smooths out the irregularities in the surface normals of the individual depth maps. In order to obtain a complete meshed surface, we first infer an iso-surface from the TSDF field by finding all the zero crossings. A triangulated mesh corresponding to the zero valued iso-surface is extracted using [136, 186]. Fig. 5.3 shows an example output of meshed scene representation along with the images of the scene.

5.3.1.1 CRF energy model

We use a Conditional Random Field (CRF) based approach, defined over a mesh structure, to perform the semantic labelling of the mesh. The mesh is given by $\mathcal{M} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ is the set of mesh vertices and $\mathcal{E}$ denotes the edge set defining the mesh connectivity.

Consider a set of random variables $\mathcal{X} = \{X_1, X_2, \ldots, X_N\}$, where each variable $X_i \in \mathcal{X}$ is associated with a mesh vertex $i \in \mathcal{V}$. Our aim is to associate each random variable X_i with a label $l \in \mathcal{L} = \{l_1, l_2, \ldots, l_k\}$. Let $\mathcal{N}$ be the neighbourhood system of the random field defined by sets $\mathcal{N}_i, \forall i \in \mathcal{V}$. $\mathcal{N}_i$ denotes the set of all the adjacent vertices of vertex v_i given by the mesh edges $\mathcal{E}$.

The corresponding Gibbs energy of the mesh is given as:

$$E(\mathbf{x}) = \sum_{i \in \mathcal{V}} \psi_i(x_i) + \sum_{i \in \mathcal{V}, j \in \mathcal{N}_i} \psi_{ij}(x_i, x_j) \tag{5.3.1}$$

The most probable or maximum a posteriori labelling $\mathbf{x}^*$ of the CRF corresponds to the minimum energy of the graph. The energy minimization problem is solved using the graph-cut based alpha expansion algorithm [17], giving us the labelling of the mesh.

Unary potential The unary potential ψ_i describes the cost of a mesh vertex v_i taking a particular object class label. This is derived from the images which define the scene. Each mesh vertex is associated with a set of $\mathcal{K} = \{1, 2, ..., K\}$ images, if the vertex is visible from that view. Given the estimated camera pose P^k of any particular image, the vertex can be associated with a particular image pixel $\tau_i^k, k \in \mathcal{K}$. This allows us to determine a set of image pixel registrations associated with any mesh vertex. For each of these image pixels, a classifier score is obtained using the multi-feature variant [108] of the TextonBoost algorithm [158]. It is worth noting that classifier response could be computed for the pixels which have at least one vertex association in the mesh further speeding up the unary evaluation. This is particularly beneficial as we do not have to compute the response for all the pixels in the image, e.g. the sky pixels in the image which are generally not associated with any mesh vertex. Let the classifier responses for the registered pixel be denoted as $\mathcal{H}(\tau_i^k)$. The individual pixel's class wise probability score is computed as $g(\tau_i^k) = \frac{e^{\mathcal{H}(\tau_i^k)|_l}}{\sum_{l \in \mathcal{L}} e^{\mathcal{H}(\tau_i^k)|_l}}$. In order to obtain the unary potential for the mesh vertex, we combine the classwise probability scores for every pixel τ_i^k as:

$$\psi_i(x_i) = -log \left[f(\{g(\tau_i^k)\}_{k=1,..,K}) \right]. \tag{5.3.2}$$

The function f, effectively combines evidence from multiple pixels to a particular mesh vertex. We examine three different approaches in combining the probability scores. In the first approach, we take the average of the scores for each individual pixels associated with the mesh location (method 'average'). In the second approach, we combine by taking the class-wise maximum (method ‘max’). In the third approach, we compute the 3D distance d_i^k of a particular pixel from the camera P^k and weight the pixel probability values inversely with the distance and then take average for every pixel registered to the particular vertex. As a result, the image pixels which are closer to the camera are weighted more than the pixels that are lying at the farther distance from the viewing point. This is because the pixel information corresponding to the objects far from the camera are often ambiguous, unlike the close objects which are distinct in the image. We call this method as ‘weighted’ in our experiments.

Pairwise Potential The pairwise potential takes the form of a generalised Potts model:

$$\psi_{ij}(x_i, x_j) = \begin{cases} 0 & \text{if } x_i = x_j \\ g(i,j) & \text{otherwise,} \end{cases} \tag{5.3.3}$$

where $g(i,j) = \theta_p + \theta_v \exp(-\theta_\beta ||Z_i - Z_j||^2)$ and Z_i and Z_j are the 3D location of the vertices v_i and v_j respectively. The model parameters θ_p, θ_v, $\theta_\beta \geq 0$ are set by cross validation on unlabelled data. This form of pairwise potential encourages mesh vertices that are close in the 3D space to take similar labels, thereby encouraging smoothness over the mesh. A strong hypothesis for a particular object label towards a mesh vertex is likely to be carried forward to the neighbouring mesh vertex. As the connectivity is defined by the edges in the mesh, the label propagation respects the contours and surface of the objects.

5.4 Experiments

Outdoor dataset For evaluation of our method on outdoor scenes, we use the KITTI odometry dataset [62]. Stereo images are captured using a specialised car in urban, residential and highway locations, making it a varied and challenging real world dataset. We use the object class labellings for the KITTI dataset as introduced in § 4.5.1. The labelled set is comprised of 45 images for training and 25 for testing with per-pixel class labels. The class labels are road, building, vehicle, pavement, tree, signage, post/pole, wall/fence.

For reconstruction from a stereo image sequence, the camera poses are estimated using the approach mentioned in §4.3.1. This comprises of two main steps, namely feature matching and bundle adjustment. We assume calibrated cameras are positioned on a rigid rig. To obtain the feature correspondences, we perform both stereo matching (between left and right pairs) and frame by frame matching (consecutive images for both left camera and right camera). Once the matches are computed, the corresponding feature tracks are generated similar to §4.3.1. All the stereo matches and the corresponding frame-to-frame matches are kept in the track. As the road scene images are subject to high glare, reflection and strong shadows, it is important to generate good feature matches for accurate camera estimation. Having this agreement between both the stereo and ego motion helps bundle adjustment to estimate the camera poses and feature points more accurately. Given the feature track database, the camera poses and the associated feature points are estimated using a Levenberg-Marquardt optimiser. As we are dealing with street-level sequences of arbitrary length, a global optimization is computationally infeasible, and may be unnecessary, since only the structure of the world near to the current frame is of interest in many applications. Hence, we use a sliding window approach for the bundle adjustment with a window size of n frames. In our experiments we set $n = 20$ which we found to be a good compromise between speed and accuracy. The estimated camera poses are used to fuse the depth estimates (computed from stereo disparity) in an approach similar to [129]. Finally, surface reconstruction is performed using the marching tetrahedrons algorithm [136] which extracts a triangulated mesh corresponding to the zero valued iso-surface. We release these meshes along with ground truth labels for further research in the community[1].

Fig. 5.4 demonstrates the qualitative results of our approach for the outdoor sequence from the KITTI dataset [62]. The scene is reconstructed from 150 stereo image pairs taken from the moving vehicle. The reconstructed model in Fig. 5.4 comprises 703K vertices and 1.4 million faces. Visually, we can see the accuracy of the reconstructed semantic model. A close up view of the semantic model is shown in Fig. 5.9. The reconstructed model captures fine details which are evident on the pavements, cars, road, fence, etc. The arrows relate positions in the 3D model and the corresponding images. Fig. 5.9(a) shows the fence, hedge and the pole (encircled) in the image and the reconstructed model. Fig. 5.4(b) shows the right side of the street in the reconstructed model, showing the car, pavement and the trees in both the images and the model. In all the above cases, we can see the effectiveness of our method in handling large scale stereo data sequence to generate a semantic model of the scene.

The quantitative results of our reconstructed semantic model are summarised in Table 5.1. The evaluation is performed by comparing with ground truth labelled mesh. They are generated by projecting the image ground truth labels into the mesh using the estimated camera poses, and taking a vote on the maximally occurring ground truth label for a partic-

[1] available at http://www.robots.ox.ac.uk/˜tvg/projects/

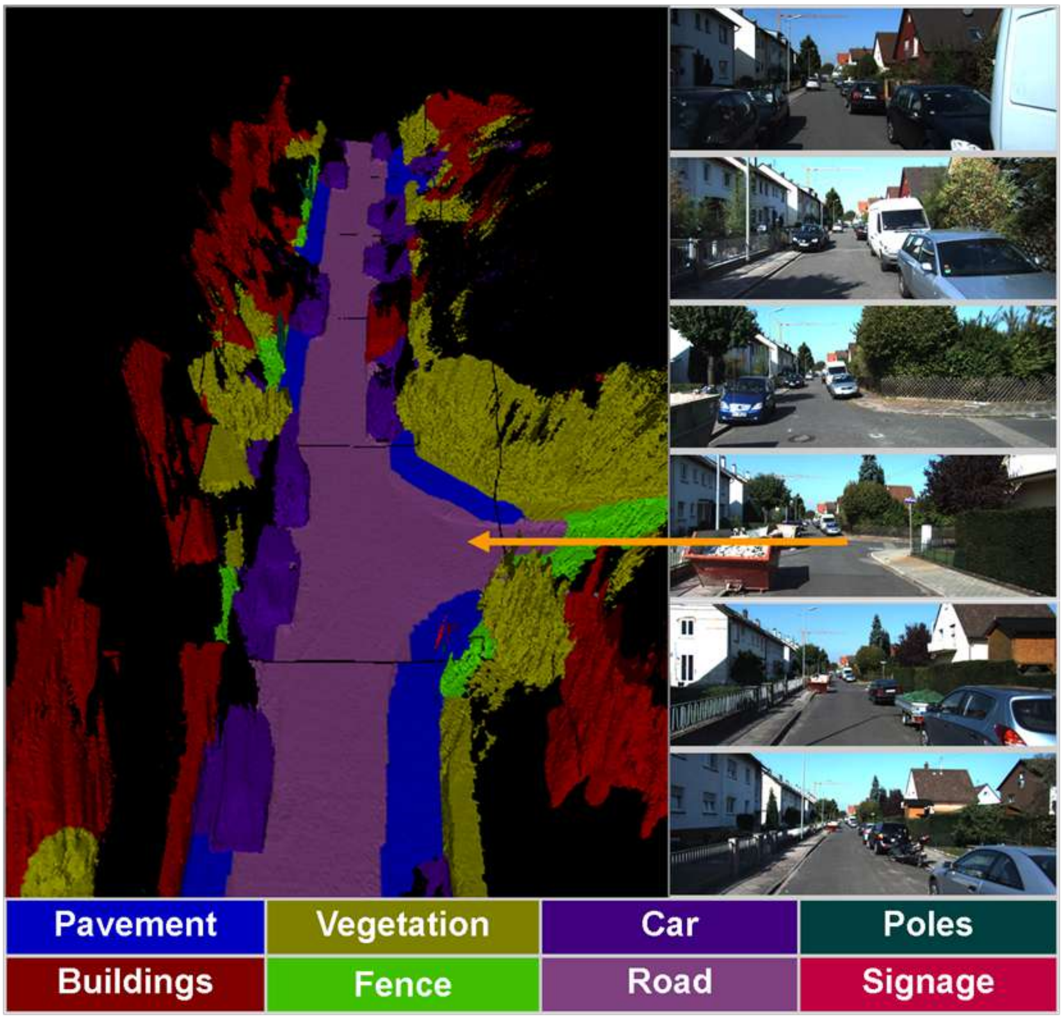

Figure 5.4: Reconstruction of an urban sequence sequence from KITTI dataset [62]. The right column shows the sequence of stereo images that are used for reconstruction purposes. The vehicle traverses around 200 meters capturing 150 stereo image pairs. The arrow shows the bend in the road in both the image and the model (best viewed in colour).

Table 5.1: Semantic Evaluation for outdoor scene KITTI dataset [62]

Method	Building	Vegetation	Car	Road	wall/fence	Pavement	Signage	Pots/Pole	**Average**	**Global**
Infernece (Unary+pairwise)										
Recall measure										
Chapter 4 (Unary+pairwise)	**97.2**	84.5	77.9	96.2	36.6	75.2	**24.5**	31.7	65.4	83.9
Mesh (Average)	96.39	84.83	**79.33**	96.16	41.50	74.03	23.60	38.1	66.74	84.30
Mesh (Max)	95.63	80.07	76.04	95.48	34.40	57.41	13.75	**55.56**	63.63	81.40
Mesh (Weighted)	96.45	**85.08**	77.17	**97.33**	**44.01**	**77.59**	23.91	39.49	**67.63**	**84.67**
Intersection *vs* Union										
Chapter 4 (Unary+pairwise)	81.6	72.3	65.3	90.5	34.7	61.3	**19.8**	29.1	56.8	
Mesh (Average)	81.73	72.47	66.67	90.08	39.31	60.78	18.50	**30.71**	57.35	
Mesh (Max)	81.87	68.73	57.00	88.57	31.40	48.29	12.5	28.77	52.14	
Mesh (Weighted)	**81.97**	**73.12**	**67.51**	**92.05**	**41.78**	**62.15**	17.29	27.07	**57.87**	
Unary										
Recall measure										
Mesh Unary (Average)	96.25	85.10	**78.5**	95.55	39.4	75.5	**24.50**	**40.0**	66.8	84.0
Mesh Unary (Max)	95.0	83.8	74.5	96.6	38.72	78.06	8.4	37.70	64.15	82.95
Mesh Unary (Weighted)	96.45	**85.4**	76.8	**96.9**	**42.7**	**78.52**	24.36	39.36	**67.57**	**84.5**
Intersection *vs* Union										
Mesh Unary (Average)	81.7	72.65	65.6	90.3	37.3	60.8	**18.6**	**30.7**	57.2	
Mesh Unary (Max)	81.8	70.87	61.6	91.3	36.3	**63.4**	6.3	19.2	53.86	
Mesh Unary (Weighted)	**82.1**	**73.4**	**67.25**	**91.6**	**40.6**	62.1	16.7	25.86	**57.4**	

ular mesh location. ‘Global’ refers to the overall percentage of the mesh correctly classified, and ‘Average’ is the average of the per class measures. We compare the mesh based methods presented in this chapter with the method presented in chapter 4 (§4.5.2) where the labelling is performed in image domain, with the unary and pairwise potential added in the model for image level segmentation and then projected on to the mesh. We observe an overall increase in the measures of global accuracy, average recall and average intersection-union scores for mesh based labelling approach (methods ‘average’ and ‘weighted’). This can be attributed to the following reasons. Firstly, the mesh captures the layout of the objects in the scene accurately. Thus pairwise connections in the 3D mesh are in accordance with the structure of the objects and the scene, while in the image domain, the connections between adjacent pixels in the images might violate the objects in the scene. For example, in the images there might be a pairwise connection between pixels denoting poles and building, even though they are well separated in the scene. Secondly, the process of determining mesh unaries by combining evidence from multiple street level images results in better classification for each mesh location and removes inconsistency present in the individual image labelling. Also, we observe that while fusing the image level classifier responses, performing a weighted combination achieves a better accuracy than averaging the pixel scores. Taking the maximum score while fusing the image pixel unary is prone to noisy predictions, resulting in lower accuracy. The effect of performing inference on the mesh for object labelling is visualised in Fig. 5.5. We show the ground truth labelled mesh in 5.5(a), the unary only labelling in (b) and inference results with unary and pairwise turned on in (c). Visually the results can be seen as more consistent and smooth, as highlighted in the yellow ellipse, where a part of the road is mislabelled as pavement in the unary only labelling. By performing inference with unary and the pairwise turned on, the road part get correctly labelled. The table 5.1 also shows the results on unary only labelling, where we observe a lower global and classwise average scores in comparison to the mesh based inference. However, it is worth noting that due to the void sections in the ground truth mesh around the object boundaries, the mesh inference quantitative results are not significantly better than the unary only results.

Table 5.2: Timing performance on outdoor scenes (in seconds). The scene is comprised on 150 images. The image level timings are obtained by summing up the inference time for every frame in the scene.

Image Level Inference [108]	Ours
940±31s	35.2±.3s

We also evaluate the timing performance for the inference stage for an outdoor scene

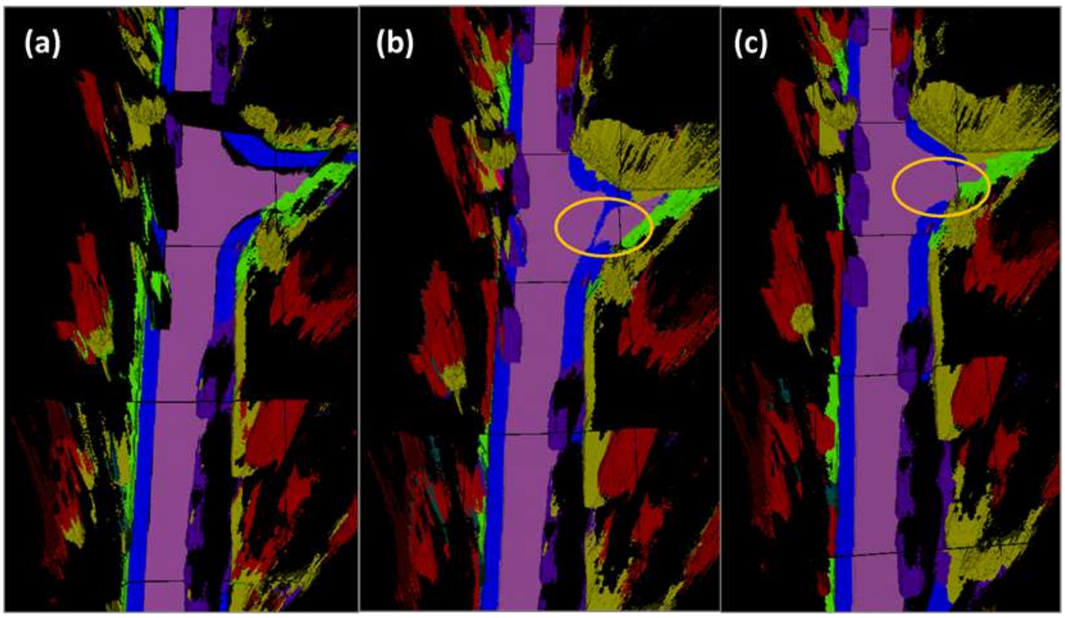

Figure 5.5: (a) Ground truth labelled mesh, (b) labelled mesh with unary potential (no inference) and (c) mesh labelling using Unary and pairwise. We can see the smoothness in the solution where we observe that a part of the road is labelled as pavement in the unary mesh (shown in yellow ellipse). The inference over the mesh penalises such inconsistency and helps to recover the part of the road correctly. Best viewed in colour.

reconstruction. We compare with [108] for inference in the image domain. Our scene is reconstructed from 150 images, the size of each image being 1281 $\times$ 376. As we are trying to reconstruct an outdoor scene which is not limited in its extent, we need to have a larger number of mesh faces and vertices to describe the scene with sufficient detail. The reconstructed mesh has around 704K vertices and 1.27 million faces. We observe a significant speedup (25x) at the inference stage when performed on the mesh (see Table 5.2) using a single core on a machine with Intel Core2 Quad CPU 2.83 Ghz. It is worth noting that our method needs to estimate the mesh to perform the inference. However, this stage can be speeded up by performing TSDF fusion on a GPU [149].

We also evaluate the timing performance for meshes of varying size and compare with the image based labelling. We create meshes of the same scene, with increasing granularity so that the mesh captures finer details of the scene. In Fig. 5.6, we evaluate inference times for meshes with increasing number of vertices from 700K till approximately 3.95 million vertices. As expected the inference time increases linearly, but still remains significantly lower than the image based labelling. In terms of labelling accuracy for meshes with increasing size, we report the global, class-wise average recall and average Int.*vs* Union scores in the table 5.3. We create multiple ground truth labelled meshes of increasing size by projecting the image ground truth class labels into the mesh. We observe a marginal dip

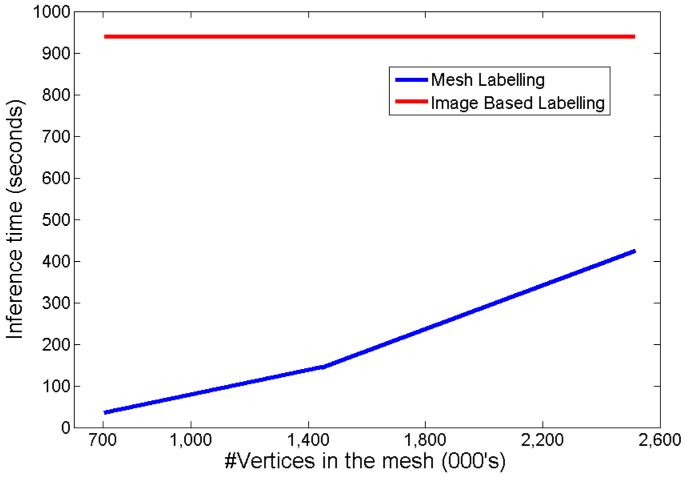

Figure 5.6: Times for image based inference and mesh based inference. We can see the significant speedup achieved by performing inference in 3D scene. The blue curve denotes the inference time for meshes with increasing size. The red curve denotes the image based labelling for the same scene comprising 150 frames.

in accuracy with the increased size of the mesh, which we believe is due to the fact increased granularity in the mesh results in smaller triangulated faces and closer vertices. This affects the pairwise cost, and the parameters for pairwise potential (we use the same parameters for each case) need to be adjusted to reflect the change in the mesh granularity.

Table 5.3: Evaluation for Meshes with varying number of Vertices.

Mesh Vertex count	Global Accuracy	Recall Average	Int.*vs* Union
704K	84.66	67.63	57.87
1.44 Million	84.36	66.89	57.12
2.51 Million	84.07	66.69	56.91
3.95 Million	83.93	66.6	56.65

Scene Editing The semantic mesh representation of the scene enables different modalities for scene editing. Mesh based scene editing has been attempted [29] where interactive editing is performed for moving objects. In our case the knowledge of object labels enables us to edit the scene and place the objects relevant to the scene. Given the labelled scene mesh, we can separate the regions of the scene as shown in Fig. 5.7 where we can accurately estimate the road and the pavement regions. We fit a plane onto the road segment, which allows us intuitively to insert objects like cars (shown in Fig. 5.8) onto the road. The object can be replicated or moved anywhere based on the permissible semantics. As the mesh is generated in a metric scale, the newly inserted vehicle can follow the path as specified.

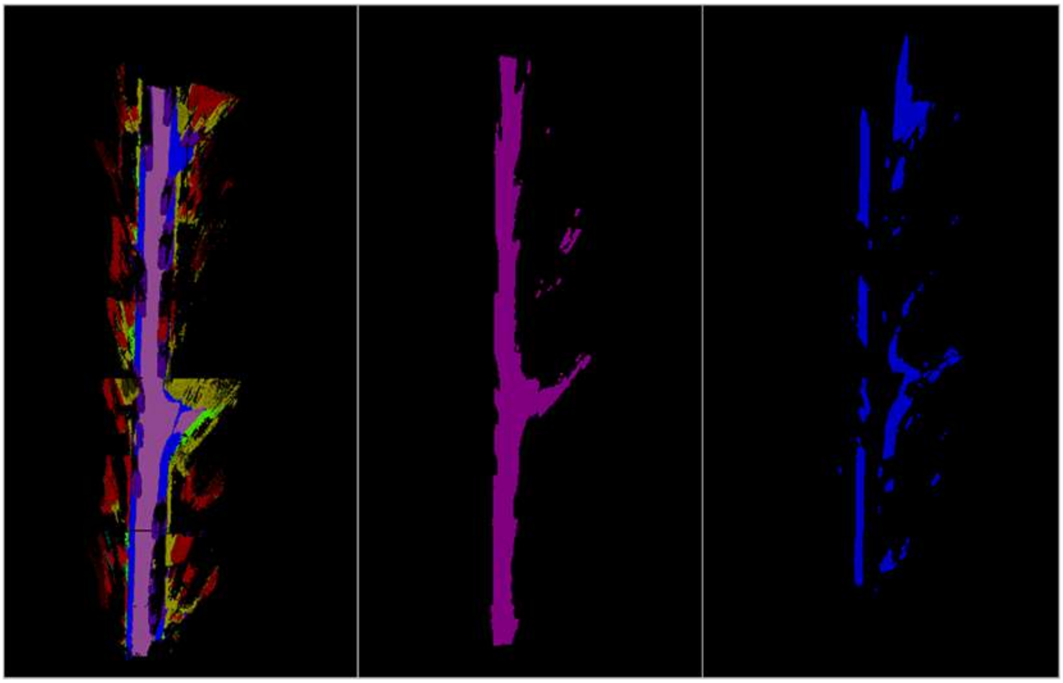

Figure 5.7: Bird's eye view of the scene. One the left is the entire reconstructed scene viewed from the top. The middle part shows the region corresponding to the road (magenta) and right shown the pavement regions (blue). (Best viewed in colour)

This has an application in autonomous vehicle navigation where the path can be determined by the extent of the road. In this example (Fig. 5.8) we allow the vehicle to follow a predetermined path.

5.5 Discussions

We have presented an efficient framework to perform 3D semantic modelling applicable to outdoor road scenes. We formulated this problem in 3D space, thereby capturing the object

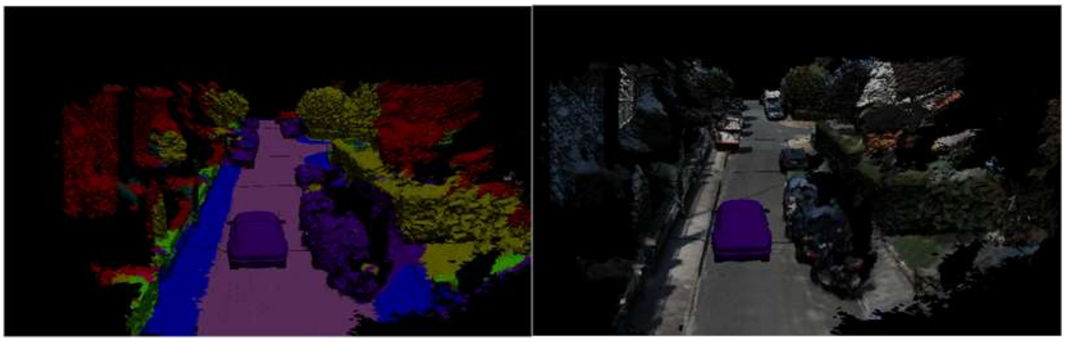

Figure 5.8: The left shows an additional object (car) being placed on the road. The right shows the corresponding textured scene. (Best viewed in colour)

layout in the scene accurately. We show that performing labelling over the meshed structure improves labelling performance. We present various unary generation scheme for the mesh locations, and observe that performing a weighted combination of individual pixel classifier scores achieves a better accuracy than averaging the pixel scores. Furthermore, having inference on the mesh results in a magnitude improvement in the inference speed (˜25x) and achieve higher accuracy for outdoor road scenes. To facilitate the training/evaluation of our model, we have created ground truth labelled meshes for KITTI outdoor scene dataset with object class labelling. We have released these ground truth labelled meshes to the community for further research.

However, the proposed method has the requirement of estimating a meshed representation of the scene. The mesh is generated by fusing the depth estimates in a TSDF volume and performing marching tetrahedra on the fused volume. This can be made efficient by a GPU implementation [149]. Similarly, the unary fusion which is performed independently for each mesh location can be parallelised easily.

In the next chapter, we incorporate multi-resolution higher order constraints for scene labelling. We perform the scene labelling on a probabilistic octree representation of the scene, allowing us to incorporate higher order constraints and describe the scene where each occupied voxel is associated with an object class label.

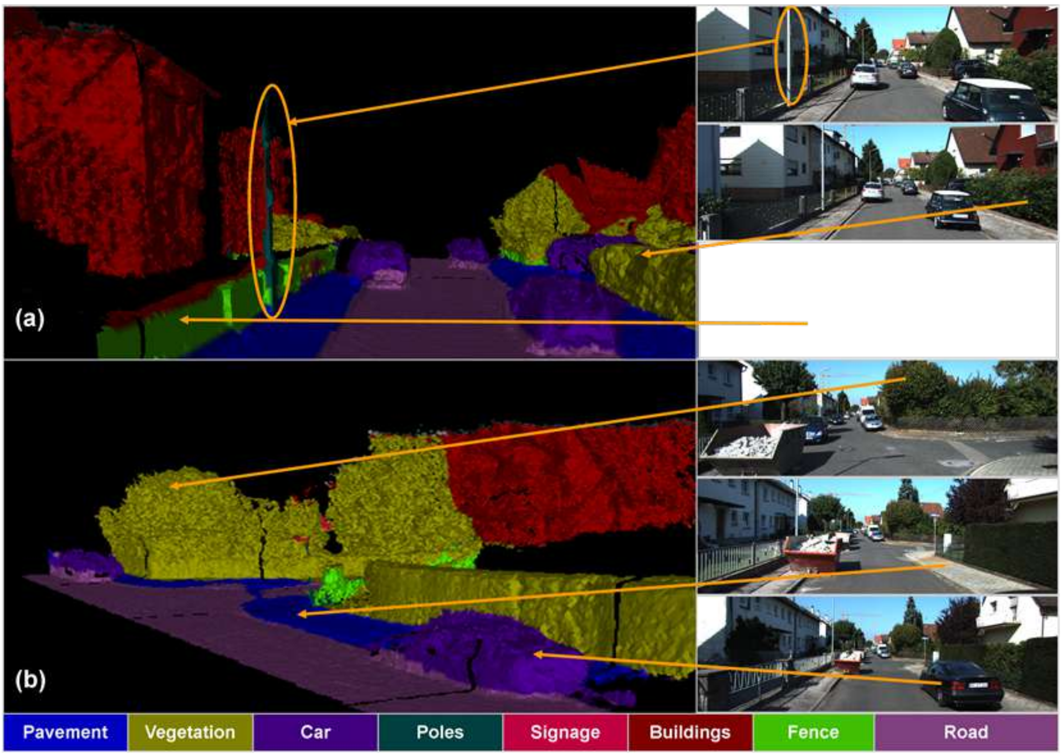

Figure 5.9: Closeup view of reconstructed semantic model of an urban sequence from KITTI dataset. The arrows relate the position in the model and the associated image.(a) shows the fence, hedge and the post (encircled) in the image and the reconstructed model. (b) shows the right side of the street in the reconstructed model, showing the car, pavement and the trees in for both the cases. (View arrows from bottom, best viewed in colour.)

Chapter 6

3D Scene Understanding Modelled as a Hierarchical CRF on an Octree Graph

6.1 Introduction

In the previous chapter, we demonstrated how semantic object labelling can be performed accurately and efficiently over a meshed representation of the scene. This is due to the fact that the mesh captures the surface of the scene accurately. However, there are two limitations in the mesh based labelling. Firstly, once the mesh is generated, it is not updated while labelling is performed [178]. The second limitation, is more restricting due to the very nature of the mesh representation: it is not capable to map the free space in the scene. The notion of the free space is required by many autonomous systems to plan a collision free path in the scene [53]. Similarly, other representations such as point-clouds, elevation maps and multi-level surface maps also suffer from the same problem. The point clouds cannot model the unknown areas or free space and are not memory efficient. The elevation/ multi-level surface maps do not account for the free space. However a voxel based volumetric representation overcomes this drawback, by mapping the entire scene, including 'occupied', 'free' and 'unknown' areas. Motivated by these two factors, we propose in this chapter a method to perform object class labelling in a volumetric representation and jointly determine the free/occupied space to create a truly labelled scene.

Representing the 3D scene accurately and efficiently has been a long standing problem in the robotics community for a multitude of applications like mobile navigation [53] and manipulation [142]. Most of them require an efficient probabilistic representation of free, occupied and unmapped regions. Moreover, the volume needs to be indexed so that robotic tasks can be performed efficiently. For example, the autonomous system needs to decide its path based on the free space on the fly. Recently, a framework called 'Octomap' was presented in [88] which enabled such a mapping of the volume. In their framework, the entire 3D volume is discretised into voxels and indexed through an octree. For each 3D depth estimate, probabilistic updates are performed to denote the occupied and the free space. As the tree represents the entire scene, a sub-grouping of the volume can be performed easily, as each sub-tree in the octree denotes a contiguous sub-volume in the 3D world. This kind of sub-grouping is additionally attractive to us, as it enables us to impose multi-level object class/occupancy constraints on the voxels and on the sub-volumes.

In the image labelling paradigm, there has been an increasing thrust to solve the object class segmentation problem using a multi-resolution approach, with pixel level information being augmented with image-segment constraints. These image segments are generated via

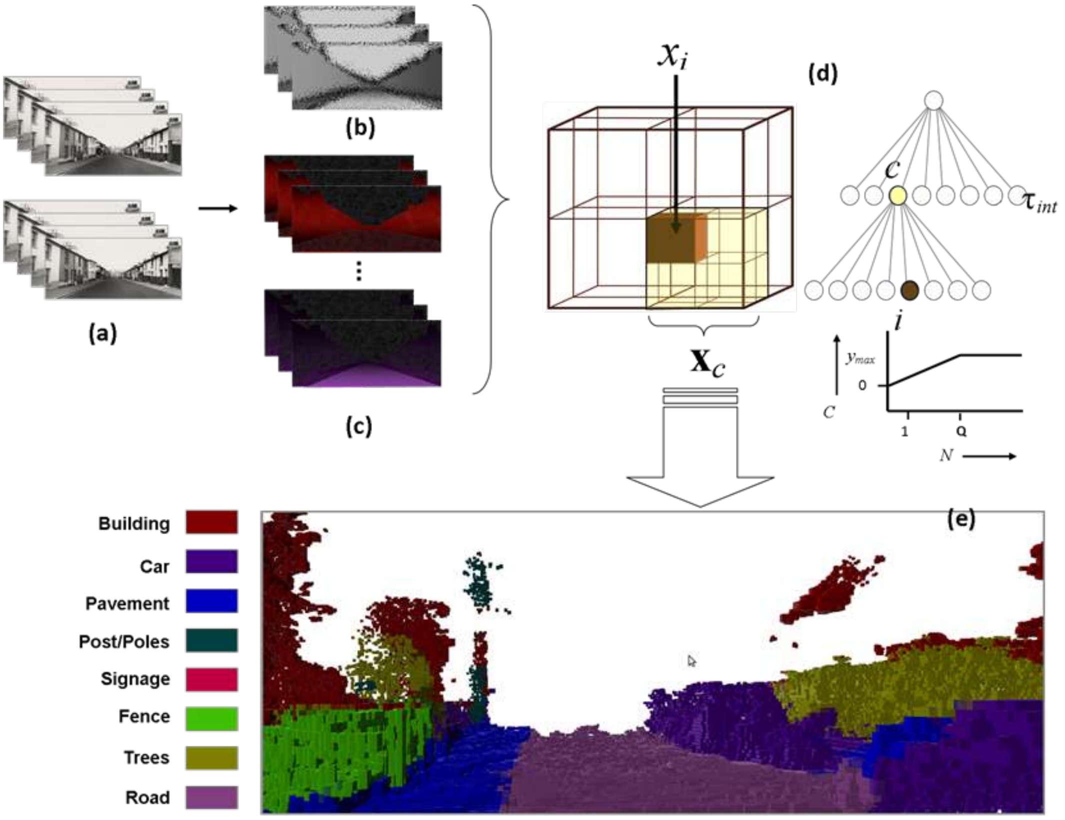

Figure 6.1: *System Overview:* (a) Shows the input to our system which is a sequence of rectified stereo images. (b) depth estimates and the (c) object level cues are computed from the input image sequence. This information is used to generate the object class and occupancy hypothesis for every voxel in the octree. (d) Shows the octree (τ) composed of two levels. The root node (at level 0) represents the entire volume and the leaf node represents the smallest voxel. The leaf nodes of the tree are denoted as τ_{leaf} and the internal nodes as τ_{int}. Each internal node is subdivided into eight child nodes and the corresponding volume (in the left) gets subdivided accordingly. x_i is the voxel (brown) corresponding to the node i in the leaf level of the tree and $\mathbf{x}_c$ is the volume (yellow) corresponding to the internal node c. The CRF is defined over the voxels in the tree. A hierarchical Robust P^N constraint is enforced on the internal nodes in the tree. This penalises the inconsistency in the dominant label in the grouping, where the cost (C) is proportional to the number of variables (N) not taking the dominant label. (e) The CRF energy is minimised to infer the object labels and the 3D occupancy giving us the semantic octree. (*Best viewed in colour*)

multiple unsupervised segmentation algorithms [26, 34, 86]. These segments also known as superpixels, are generated based on the low level image properties like colour, gradients and edges. The reason behind the improved usage is often linked to the fact that pixels belonging to same image segments often share similar object semantics. This has also been extended using multiple overlapping segments, where pixel, region, object and scene level information [101, 108, 110] are used simultaneously to reason about the scene.

In this chapter, we look at introducing a multi-resolution 3D labelling approach towards the aim of unified scene understanding. Our proposed method performs object class labelling and infers the 3D occupancy map for each voxel in a joint fashion. Towards this aim, we propose a hierarchical robust P^N Markov Random Field, defined on the voxels and sub-volumes (group of voxels) in the 3D space determined through an octree. Our work is inspired from the various multi-resolution approaches in the image domain, particularly [101,108], which we naturally extend from pixel based image lattice representation to voxel based 3D volume. The octree based space discretisation provides us a natural grouping of the 3D volumes. For the octree formulation, we use the octomap framework [88]. Our approach works as follows (see Fig. 6.1): The input to the system is a sequence of stereo images (a). The image level information is used to generate point cloud depth estimates (Fig. 6.1b) and the object level cues (Fig. 6.1c). This information is used to generate the object class and occupancy hypothesis for every voxel in the octree. An example octree (τ), composed of two levels, is shown in Fig. 6.1(d). The root node (at level 0) represents the entire volume and the leaf node represents the smallest voxel. Each internal node is subdivided into eight child nodes and the corresponding volume (in the left) gets subdivided accordingly. x_i is the voxel (brown) corresponding to the node i in the leaf level of the tree and $\mathbf{x}_c$ is the volume (yellow) corresponding to the internal node c. The CRF is defined over the voxels in the tree. A hierarchical Robust P^N constraint is enforced on the internal nodes in the tree. This penalises the inconsistency in the dominant label in the grouping, where the cost (C) is proportional to the number of variables (N) not taking the dominant label. (e) The CRF energy is minimised giving us the final object labels and the occupancy for each voxel.

6.1.1 Chapter Outline

In the following §6.2 we give an overview of the related literature. We describe our method to generate a semantic octree representation of the world in §6.3. The qualitative and the quantitative results of our method are shown in section §6.4. Summary and discussion are provided in section §6.5.

6.2 Related Work

Multi-resolution approaches in computer vision for solving tasks such as image classification, object detection and semantic image segmentation have shown much promise over the last decade [70, 81, 108, 110, 158]. Most of these approaches work in a bottom up fashion [27] where information from image pixel, region, object and scene level information are used simultaneously to infer the objects present in the scene. These methods try to encode the information present at various levels of quantisation. The higher level quantisation (superpixels in the image) is obtained through multiple unsupervised segmentation of the image. In [158], dense per-pixel texton features over an image patch centred around the pixel are learnt using boosting. The final solution is obtained using an inference using a conditional random field, inducing smoothness by adding pair-wise potential terms in the graph. In [101], robust higher order P^N potentials were introduced as a constraint over image segments, which penalised label inconsistency among the pixels belonging to the segment through a robust cost function. This work was extended in [108], where an appropriate image quantization was selected from the pixels and multiple overlapping superpixels, to aid the semantic image segmentation task. All these methods demonstrated a common behaviour: consistency and improvement in the solution via a multi resolution approach.

Representing a scene through a 3D model has been a subject of research over the last couple of decades in the robotics community, where multiple representations such as volumetric/voxel based, point clouds, surface maps, height/elevations maps and meshes have been proposed. The aim has been to capture the surface accurately by mapping the 'occupied', 'free' and 'unknown' space and index the volume for efficient occupancy querying. Modelling the environment through an array of voxels (cubic volumes) has been attempted as early as [146]. The cubes were marked as 'empty' or 'occupied' based on the depth estimated from dense range sensors. This is in contrast to popular SLAM systems [33, 160] where discrete 3D point cloud estimates are used to describe the scene. As a result, the notion of the free space is lost in the representation. Similarly, other modalities like 2.5D representation or height/elevation maps such as stixel world [137], provide a single height measurement for every location on a 2D ground plane point, thus inaccurately capturing the surface of the scene. Similarly, the mesh based surface representation [186] have the the limitation of not being able to distinguish between free, occupied and unknown space.

In contrast to these approaches, octree based volumetric representation provides the following advantages: it can map effectively the free, occupied and unknown space and can be updated on-line as the new depth measurements are observed. Additionally the voxels being indexed through the tree, enable efficient querying for determining occupancy. Finally, the tree provides a multi-resolution view of the volume, with the root node denoting the entire space and the leaf node the smallest sized 3D volume (see Fig. 6.2). Various octree based

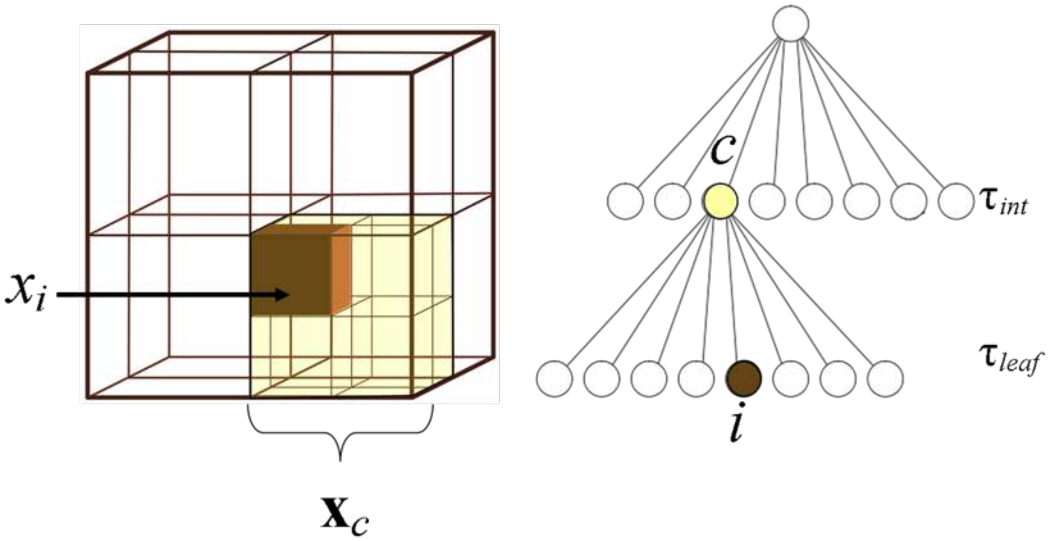

Figure 6.2: An example of an octree (τ) composed of two levels. The root node (at level 0) represents the entire volume and the leaf node (at level 2) represents the smallest voxel. The leaf nodes of the tree are denoted as τ_{leaf} and the internal nodes as τ_{int}. Each internal node is subdivided into eight child nodes and the corresponding volume (in the left) gets subdivided accordingly. x_i is the voxel (brown) corresponding to the node i in the leaf level of the tree and $\mathbf{x}_c$ is the volume (yellow) corresponding to the internal node c.

representation have been reported in in the literature [53, 88, 135] and for a more detailed discussion, we refer the reader to [88]. Recently, a voxel based indoor scene classification system was proposed independently in [161], but they do not consider hierarchical grouping of voxel volumes. In this chapter, we follow the probabilistic octree implementation of [88] and take inspiration from multi-resolution approaches in image labelling, particularly [101, 108] and propose a method to annotate the scene with object class labels or mark them as free.

6.3 Semantic Octree

In this section we describe our proposed approach of performing the object labelling of the 3D world represented by voxels indexed by an octree. This is visualised in Fig. 6.2. The root node encompasses the entire space. Each internal node corresponds to a volume which is recursively subdivided into eight nodes/sub-volumes. The process of subdivision is continued till the minimum sized voxel is reached corresponding to the nodes in the leaf level in the tree. This is shown in the brown coloured voxel in the Fig. 6.2 and corresponding node is also shown in the octree. Let the total number of nodes in the octree be denoted by the set $\tau = \{\tau_{leaf} \cup \tau_{int}\}$, where τ_{leaf} corresponds to the leaf level nodes and the τ_{int}

denoting the internal nodes in the tree with $\tau_{leaf} \cap \tau_{int} = \emptyset$.

In our case, we are performing a labelling over the entire volumetric space where we intend to classify each voxel as empty or occupied with a certain object class label. Consider a set of random variables $\mathbf{X} = \{X_1, X_2, \ldots, X_N\}$, corresponding to each leaf level voxel $x_i \in \tau_{leaf}$. We intend to assign a label to each of the random variables from the discrete label set $\mathcal{L} = \mathcal{L}_{object} \cup l_{free}$, where $\mathcal{L}_{object} = \{l_1, l_2, ..., l_L, \}$ correspond to the object class labels and l_{free} is the label corresponding to the free space. The class labels corresponding to $\mathcal{L}_{object}$ are road, building, vehicle, pavement, tree, signage, post/pole, wall/fence. A clique c is defined as a set of random variables $\mathbf{X}_c$ which are conditionally dependant on each other. This clique corresponds to the internal nodes in the tree $c \in \tau_{int}$. By traversing the tree downwards from a particular internal node, we can determine the set of leaf voxels and their corresponding random variables forming the set $\mathbf{x}_c = \{x_i, i \in c\}$. Thus the octree internal nodes provide a logical grouping of the 3D space. It is worth noting that the first internal group (from the leaf level) will correspond to the pairwise connection between the leaf voxels in the group. The final energy defined for the volume is given as:

$$E(\mathbf{x}) = \sum_{i \in \tau_{leaf}} \psi_i(x_i) + \sum_{c \in \tau_{int}} \psi_c(\mathbf{x}_c). \tag{6.3.1}$$

The energy minimisation is solved using a graph-cut based Alpha Expansion algorithm [17]. We now discuss the potentials in details.

6.3.1 Unary Potentials

The unary potential ψ_i describe the cost of the voxel x_i taking a particular object class label or the free label. The appearance based object class hypothesis is derived from the stereo image pairs of the scene. A set of point clouds is computed from the stereo depth maps given the estimated camera pose. The point clouds are fused to compute the occupancy probability for each voxels. The image appearance based object class score and the occupancy are combined to determine the unary potential for each voxels. We first describe how the depth measurement from the point clouds are fused to determine occupancy probability and then show how they are combined with the appearance cues.

Octree occupancy update Initially, all the voxels in the tree are uninitialized and correspond to 'unknown' with an occupancy probability $P(x_i)$ set to 0.5. A log-odds value is determined for each voxel as

$$Lg(x_i) = log\left[\frac{P(x_i)}{1 - P(x_i)}\right]. \tag{6.3.2}$$

The depth estimates generated from the point clouds, are then fused into the octree in a ray-casting manner, determining all the voxels along the ray from the camera to the surface point. This set of voxels is now initialised, and belong to either ’free’ or the ’occupied’ category. The ray-casting update is shown in Fig. 6.3. The end point of the ray corresponds to the surface of the object. This is the voxel where the ray gets reflected and is updated as occupied (shown in red). All the voxels between the sensor and the endpoint are updated as free (shown in green). The log-odds for the free and occupied voxels, given a single depth estimate d_t, corresponding to a $3D$ point Z_t is denoted as:

$$Lg(x_i \mid d_t) = \begin{cases} \mathcal{O}_{free} & \text{if the ray passes through} \\ \mathcal{O}_{occ} & \text{if the ray is reflected.} \end{cases} \tag{6.3.3}$$

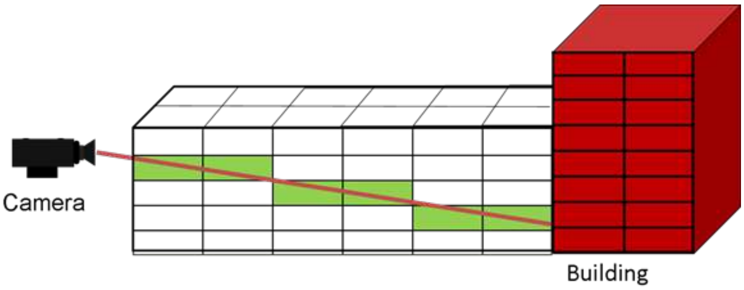

Figure 6.3: A ray is casted from the camera to the world point. All the voxels in between the camera and the point marked in green are updated with $\mathcal{O}_{free}$ and the end point, building in this example shown in red, is updated with $\mathcal{O}_{occ}$. Best viewed in colour.

The log-odds values are set as $\mathcal{O}_{occ} = 0.85$ and $\mathcal{O}_{free} = -0.4$ corresponding approximately to the probability values of .7 for occupied and .4 for free voxel. These values are observed to be sufficient for our experiments. Let a set of depth estimates obtained from point clouds from a stereo pair be given as $d_{1:t-1}$. Then the accumulated log-odds values of a voxel is given as:

$$Lg(x_i \mid d_{1:t}) = Lg(x_i \mid d_{1:t-1}) + Lg(x_i \mid d_t), \tag{6.3.4}$$

where $Lg(x_i \mid d_{1:t-1})$ is the accumulated log-odds corresponding to the set of previous depth estimates $d_{1:t-1}$. As we are accumulating the log-odds, the values are clamped to l_{min} and l_{max}. After merging all the depth estimates $d_{1:T}$ from every point cloud set for each stereo pair, we arrive at the final occupancy probability for each voxel as:

$$P(x_i) = \frac{e^{Lg(x_i|d_{1:T})}}{1 + e^{Lg(x_i|d_{1:T})}}. \tag{6.3.5}$$

We use this probability measure for each voxel towards calculating the unary cost for them. For more details on the log-odds update we refer the reader to [88].

Unary scores Each voxel x_i is associated with a set of point clouds $\mathbf{Z}^i = \{Z_1^i, Z_2^i, ...Z_K^i\}$. This set is an empty set for the free voxels. Given the camera pose, these point clouds are mapped to the set of image pixels $\mathbf{T}^i = \{T_1^i, T_2^i, ...T_K^i\}$. For each pixel T_k^i , a classifier score is obtained using the multi-feature variant [108] of the TextonBoost algorithm [158]. Let these classifier scores be denoted as $\Theta(T_k^i), k = 1, .., K$. In order to obtain the unary cost for the voxel, we combine the pixels scores of $\mathbf{T}^i$ as $\mathcal{H}(x_i) = f(\{\Theta(T_k^i)\}_{k=1,..,K})$. This function f can take the form of class-wise average or maximum of the individual pixels classifier scores. It is important to note that in the image domain, the number of class labels is L_{object}, while in the case of octree voxels, we have an additional label l_{free} indicating the occupancy. We assign a large negative value $-\theta_{max}$ as the score for the free label l_{free}. Thus the unary score for the voxels, given the initial occupancy estimate $P(x_i)$, is given as:

$$\phi_i(x_i)|_l = \begin{cases} \mathcal{H}(x_i) & P(x_i) \geq \mathcal{O}_{thresh} \wedge l \in L_{object} \\ \frac{1}{1-P(x_i)} & P(x_i) < \mathcal{O}_{thresh} \wedge l \in L_{object} \\ -\theta_{max} & P(x_i) \geq \mathcal{O}_{thresh} \wedge l = l_{free} \\ \frac{1}{P(x_i)} & P(x_i) < \mathcal{O}_{thresh} \wedge l = l_{free} \end{cases}, \tag{6.3.6}$$

where $\mathcal{O}_{thresh} = 0.5$ is the occupancy threshold. This ensures that the for every occupied voxel, the score associated with label l_{free} is lower than any of the labels in $\mathcal{L}_{object}$. All the voxels that are uninitialised are marked as 'unknown' and assigned with ∞ cost, which are excluded in the optimisation phase. Finally the unary potential for each voxel is obtained by normalising the class-wise scores and taking a negative logarithm as

$$\psi_i(x_i)|_l = -log\left[\frac{e^{\phi(x_i)|_l}}{\sum_{l \in \mathcal{L}} e^{\phi(x_i)|_l}}\right]. \tag{6.3.7}$$

6.3.2 Hierarchical tree based robust potential

To define the consistency based higher order potential we follow the image based robust P^N formulation of [101]. In our case, the internal nodes in the octree provide a natural grouping voxels in 3D space, corresponding to a clique: $\mathbf{x}_c = \{x_i, i \in c\}$. The hierarchy of the voxels can be visualised in Fig. 6.2, where the node marked yellow in the tree is shown with the corresponding grouping of voxels in the 3D volume. Determining this super-voxel set enables us to enforce a higher order consistency potential given as:

$$\psi_c(\mathbf{x}_c) = \min_{l \in \mathcal{L}}(\gamma_c^{max}, \gamma_c^l + k_c^l N_c^l(\mathbf{x}_c)), \tag{6.3.8}$$

where $\gamma_c^l < \gamma_c^{max}, l \forall \mathcal{L}$ and the function $N_c^l(\mathbf{x}_c) = \sum_{x_i \in \mathbf{x}_c} \delta(x_i \neq l)$ gives the number of inconsistent voxels with the label l. The potential takes cost γ_c^l if all the voxels in the super-voxel take the label l. Each inconsistent voxel is penalised by the cost k_c^l and the maximum cost of the potential is truncated to γ_c^{max}. Setting γ_c^l to zero has the effect of penalising inconsistent grouping of voxels and thus enforcing label consistency.

6.4 Experiments

We evaluate our semantic octree on the stereo image sequences used in §4.5 and §5.4. The image sequence has urban, residential and highway locations captured using a stereo rig mounted on a moving vehicle [62]. We use the manual annotation of object class labels introduced in § 4.5.1. To generate the octree model of the scene, we use the Octomap library [88]. We first describe the octree generation and then move on to object class labelling.

Voxel occupancy estimation To determine the initial estimate of voxel occupancy, we compute depth maps from stereo image sequence. A set of point clouds is generated from the depth maps using the estimated camera pose which are used to update the occupancy of the voxels (see §6.3.1). The depth estimate is computed from the image level disparity as $z_i = \frac{B \times f}{d_i}$, where z_i and d_i are the depth and the disparity corresponding to the i^{th} pixel respectively. The terms B and f are the camera baseline and the focal length respectively, which are computed from the camera matrix. The image disparity is computed using semi-global block matching stereo algorithm [85]. To estimate the camera matrix, we follow the approach presented in [63].The camera internal parameters are assumed to be fixed. The features are tracked using frames from the stereo and egomotion. The camera egomotion is estimated by minimising the reprojection errors with velocity estimates being refined using a Kalman filter with a constant acceleration model. We define the octree of height 16 with a minimum resolution of $(8 \times 8 \times 8) cm^3$ (size of the voxels at the leaf level). This corresponds to $2^{16} \times 8cm \approx 5.24km$ in each dimension which was sufficient for our experiments for an outdoor scene. By extending the level of the tree we can accommodate further volume without losing the granularity of the smallest sized voxels.

The initial estimate of the occupancy can be visualized in Fig. 6.4. Only the occupied voxels are shown. In this example, the occupancy is estimated from 20 stereo pairs. The stereo pairs are captured as the vehicle moves along the road. We can observe the sparsity of occupied voxels along the direction of the motion of the vehicle as shown in red.

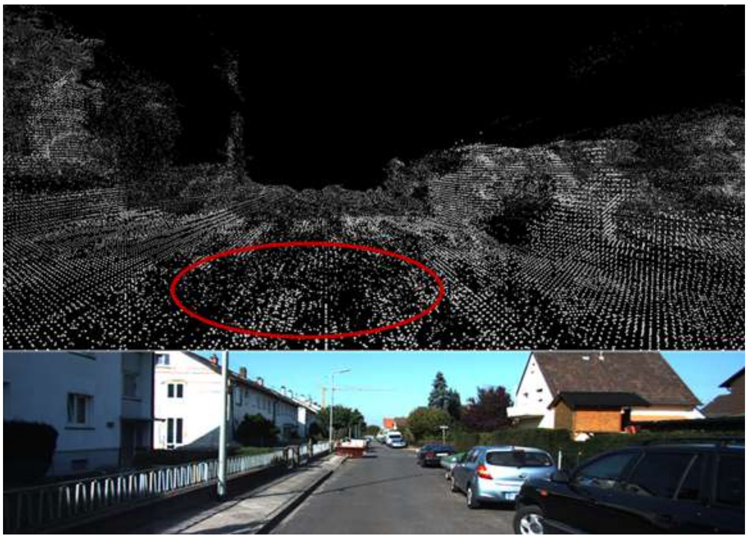

Figure 6.4: Scene after fusion 20 depth maps. Only the occupied voxels are shown. We can observe the sparsity in the number of occupied voxels in the road (highlighted inside the red ellipse).

Semantically labelled octree : Once we have the estimated the occupancy probability for each of the leaf level voxels, we determine the potentials for all the voxels using as described in § 6.3.1 and § 6.3.2. In our experiments, we have considered the super-voxels corresponding to the tree nodes in the level $13 - 15$. Beyond level 13, we observe the groupings of the voxels become large. For example a node at level 12 corresponds to a size of $2^4 \times 8cm = 1.28m$ in each dimension. Having such a large voxel tends to have a detrimental effect on the labelling.

Fig. 6.5 shows the qualitative output of our algorithm corresponding to the test set. The snapshot of the 3D octree model is shown along the corresponding images of the scene. We show the leaf level occupied voxels and the inference and the arrows relate the position in the model and the image. In (a) we can see the pavement and the road, while (b) shows the post and signage. The car is highlighted in (c) while we can see the fence in (d). Overall, we can see how the generated model can capture details of objects of varying scale. A textured scene is shown in Fig. 6.6. To evaluate quantitatively,

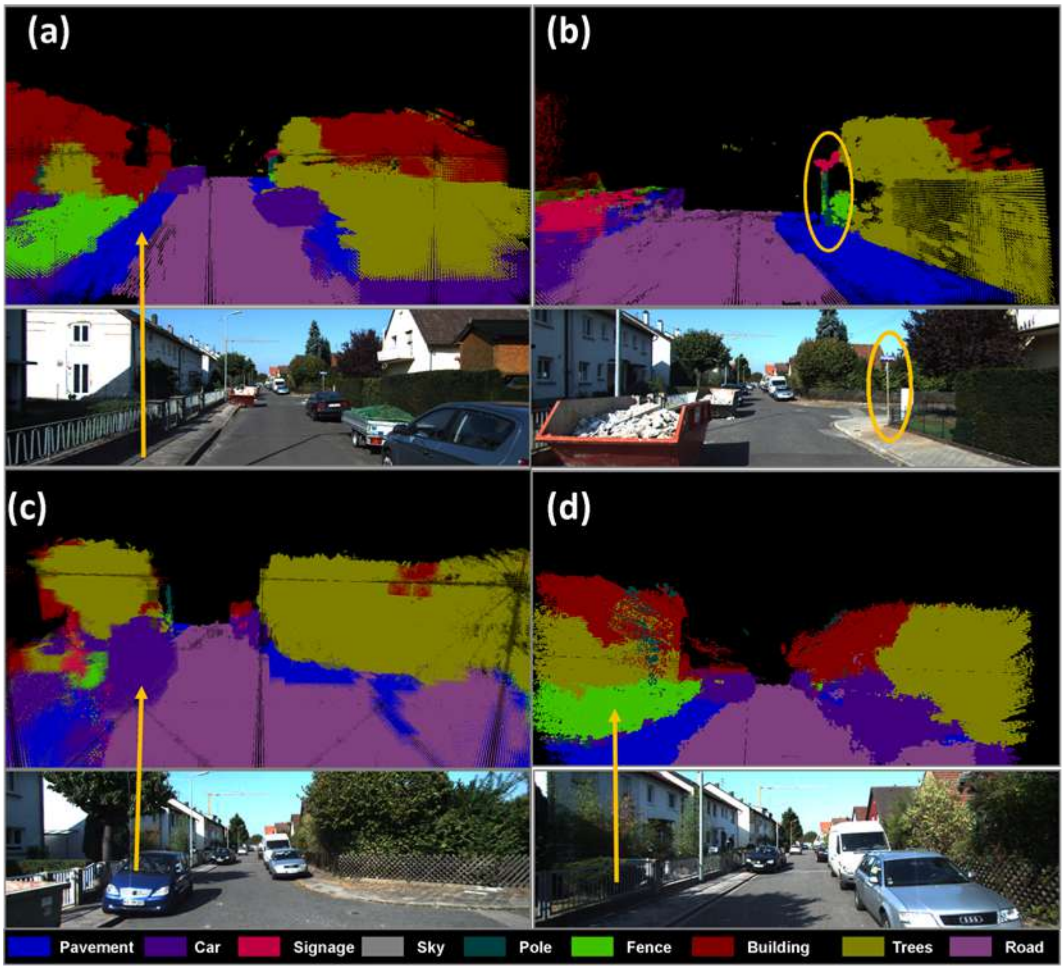

Figure 6.5: Qualitative results for semantic octree along with class labels. The leaf level occupied voxels are shown along with the arrows relating the position in the model and the image. In (a) we can see the pavement and the road, while (b) shows the post and signage. (c) The car is shown in both the image and the model.(d) The arrow highlights the fence in the model and the image. Best viewed in colours.

we project the class labels of the occupied voxels in the model back into the image domain using the estimated camera poses. Voxels in the reconstructed model that are far away from the particular camera ($>$ 20m) are ignored and only the valid pixels are considered for evaluation. Our test set consists of 25 images. We report results with two performance measures, namely $Recall = \frac{\text{True Positive}}{\text{True Positive + False Negative}}$ and $Intersection\ vs\ Union = \frac{\text{True Positive}}{\text{True Positive + False Negative + False Positive}}$. 'Global' refers to the overall percentage of the voxels correctly classified, and 'Average' is the average of the per class measures. The quantitative results are shown in table 6.1. Globally, our method of semantic octree achieves an accuracy of 78.3%, and classwise average score for recall is 61.3% and 48.1% for *Int vs U* measure. We compare the octree based volume labelling method presented in this chapter with the methods presented in the previous chapters 4(§4.5.2) and 5 (§5.4). It is worth noting that the mesh represents the surface much accurately than the voxels indexed through the octree. This explains a general improved abject labelling accuracy for the mesh based surface representations presented in Chapters 4 and 5. Also for small and thin objects, the octree volume grouping results in quantization errors, which explains the low accuracy for small and thin objects like signage and post/poles. However, the octree based method presented here has the advantage of generating the structure and mapping both the free and occupied space, which is essential for robotic applications.

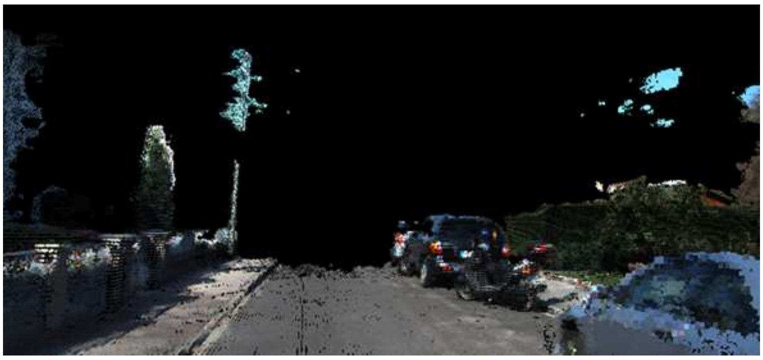

Figure 6.6: Textured model of the scene.

We compare how the higher order robust P^N consistency potentials have an effect on the number of occupied leaf voxels. Figure 6.7 shows qualitatively the effect of higher order constraints. The top left figure shows the semantic octree generated with higher order constraints while the top right shows the same scene with unary. We can see visually that the occupancy is increased with the use of higher order constraints.

We also evaluate the effect of occupancy with increasing size of super-voxels in the higher order robust constraint in plots shown in Figs. 6.8 and 6.9 respectively. We perform this experiment by first having only the leaf level voxels in the labelling problem. Then we include higher order constraints corresponding to the internal nodes in the tree. We add super-voxels corresponding to level 15, progressively going till level 13. At each level, we impose consistency constraints for all the nodes in that level and levels below it. We can observe that the total number of occupied voxels increase (Fig. 6.8) and the corresponding number of free/unoccupied voxels reduce (Fig. 6.9).

6.5 Discussion

In this chapter, we have proposed a volumetric labelling approach, where we annotate the voxels in the 3D volume with object class labels or mark them as free in a joint labelling approach. The representation has the advantage of mapping the free, occupied and unknown

Table 6.1: Semantic Evaluation: percentage accuracy on the test set.

Method	Building	Vegetation	Car	Road	wall/fence	Pavement	Signage	Pots/Pole	**Average**	**Global**
Recall measure										
Semantic Octree	89.1	81.2	72.5	97.0	45.7	73.4	27.5	3.3	61.2	78.3
Chap. 4 backprojected (U+P)	96.4	86.2	89.5	98.0	29.7	86.7	20.9	43.3	68.9	84.9
Chap. 5 backprojected	95.9	87.8	84.8	98.1	31.0	87.4	21.4	44.7	68.95	85.9
Intersection *vs* union										
Semantic Octree	73.8	65.2	55.8	87.8	43.7	49.1	7.2	1.9	48.1	
Chap. 4 backprojected (U+P)	82.6	76.1	68.8	94.8	28.8	65.7	17.2	28.5	57.1	
Chap. 5 backprojected	83.2	77.1	70.3	94.5	29.7	69.4	15.8	20.5	57.6	

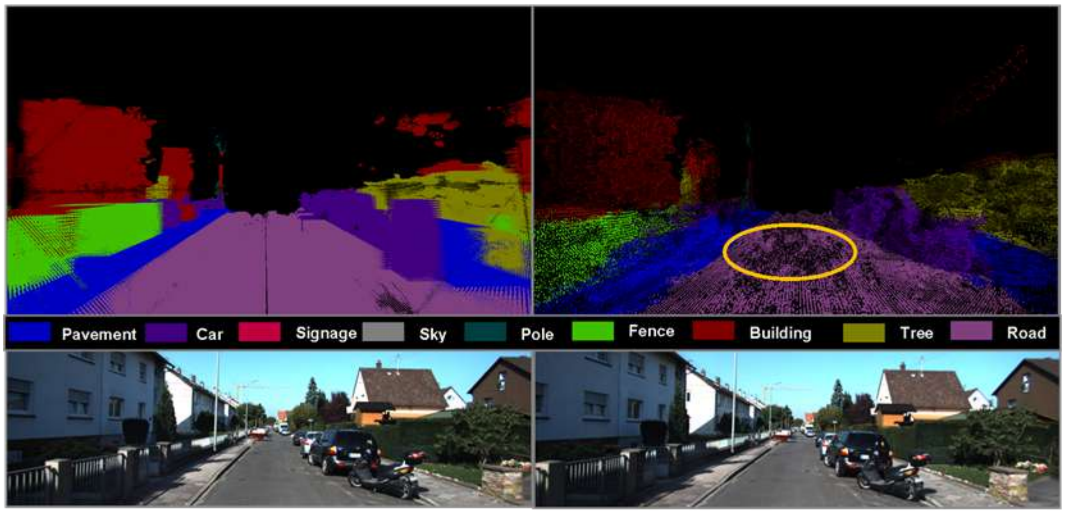

Figure 6.7: Semantic octree: The top left figure shows the semantic octree generates with robust higher order p-n potential. The top right shows the same scene without the higher order potential. We can see visually that the higher order potential has a better occupancy. Best viewed in colour.

space essential for robotic navigation. We define the scene in terms of voxels which are indexed through an octree. We formulate the problem as a hierarchical robust P^n Markov Random Field, defined on voxels and their groupings, allowing us to infer the 3D occupancy map and the object-class labels in a principled manner, through bounded approximate minimisation of a well defined and studied energy functional.

The current method has a disadvantage in comparison to mesh based labelling, as the mesh is able to capture the surface effectively. It is worth noting that the proposed approach can be improved by partitioning the space in a non-uniform fashion such as kd-trees [156] where the object boundary will overlap with the grid boundary correctly. Such partitions should be performed in a class wise fashion mimicking the physical properties of the objects.

Acknowledgements

I would like to thank Paul Sturgess for having long, detailed discussions during the formulation presented in this chapter.

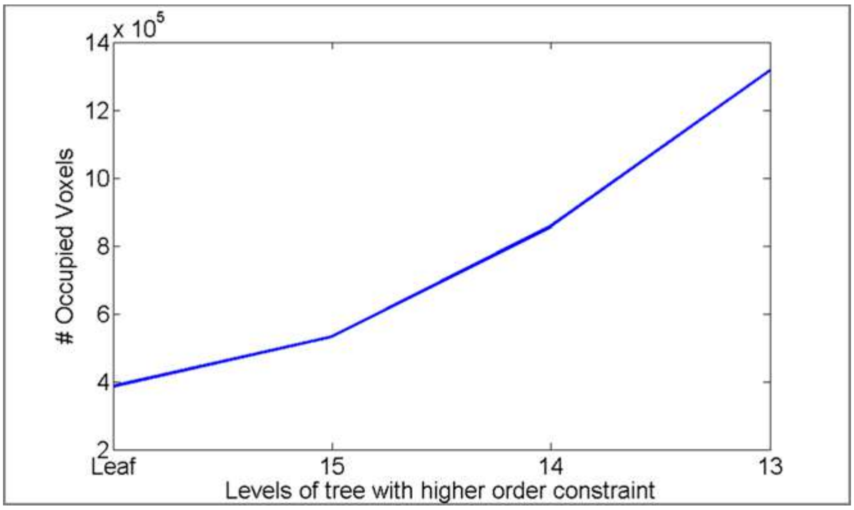

Figure 6.8: The number of voxels occupied increases as more levels in the tree participate in the inference. The leaf level indicates no higher order super-voxels. At each subsequent levels, higher order constraints are imposed on the super-voxels corresponding to the internal nodes of that level and the nodes below in the tree. As expected, we observe an increase in the number of occupied voxels with consistency constraints on the internal nodes.

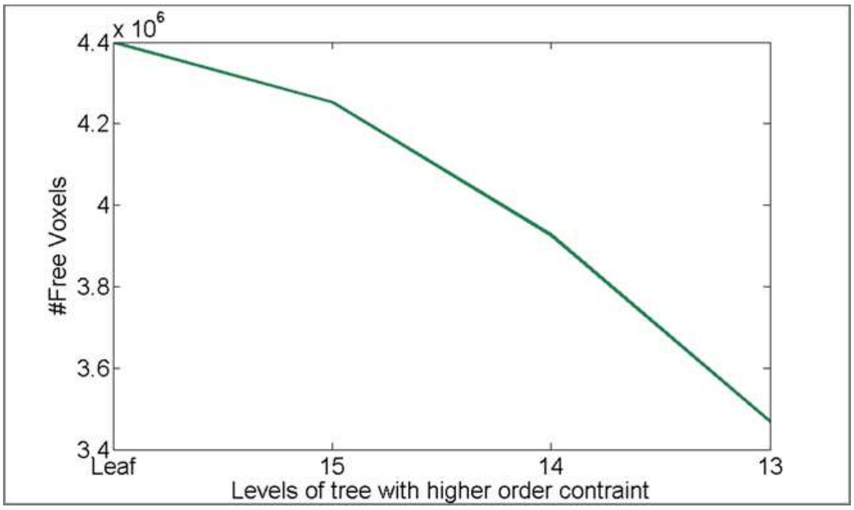

Figure 6.9: The total number of free voxels reduces as we impose higher order constraints on the internal nodes of the tree. The leaf level indicates no higher order super-voxels. At each subsequent levels, higher order constraints are imposed on the super-voxels corresponding to the internal nodes of that level and the nodes below in the tree.

Chapter 7

Discussion

7.1 Summary

In this thesis, we have explored the problems of semantic mapping and reconstruction of road scenes. Initially we have shown how a sequence of images captured from a moving vehicle can be used to generate a bird's eye view of an urban region annotated with semantic labels. We have then considered how the sequence of images can be used for generating a dense 3D reconstruction of large scale urban scenes and proposed methods to perform semantic annotation of the reconstructed model. The entire thesis is summarised as follows:

In chapter 3, we have proposed a technique to aggregate semantic information present in the street level images into a CRF formulation to generate a semantic overhead view of an urban region. The problem is formulated using two conditional random fields. The first CRF is defined over the street images captured from a moving vehicle. The object information present in the local street level images are fed into the second CRF which generates the global semantic map. This kind of two stage approach enables us to map any region irrespective of its extent with details, as the street level imagery provides detailed information of the objects present in the scene. To illustrate the efficacy of the method, we perform experiments with ten's of kilometres of street level imagery. A new dataset is presented which is comprised of ˜15 kilometres of UK roadways along with the corresponding overhead satellite views and the object class labels. We have released the dataset to the community.

In chapter 4, we have extended our work from a ground plane semantic mapping to a large scale dense 3D semantic reconstruction. We performed a volumetric TSDF reconstruction based on the inputs from a stereo camera mounted on a vehicle, capturing images at regular intervals. To deal with a large scale road scene, we incrementally fuse the depth estimates from stereo images in an online fashion, allowing us to reconstruct densely several kilometres of the image sequence. The iso-surface in the TSDF space corresponding to the scene model was then augmented with semantic labels. The object class labels were generated from the street level images exploiting the stereo information and fused to annotate the model. We demonstrated our method qualitatively and quantitatively on a large urban sequence from the KITTI dataset [62]. We have labelled the object class ground truth dataset for training and testing purposes and have released them to the community for further research.

In chapter 5, we have proposed a principled way to generate object labelling in 3D. Our method builds a triangulated meshed representation of the scene from multiple depth estimates enabling us to capture the spatial layout of the objects present in the scene. A CRF is defined over the mesh and inference if performed over the scene, instead of the images. This results in label consistency, fast inference and improved accuracy of object labelling. Our method by virtue of working on meshes, is *highly efficient* ($25\times$) in the inference stage. We have demonstrated the robustness of our framework by labelling large scale outdoor scenes and further show some examples where the labelled meshes can be used to perform active manipulation in the scene.

In chapter 6, we have presented a volumetric labelling approach, where we annotate the voxels in the 3D volume with object class labels or mark them as free in a joint labelling approach. The representation has the advantage of mapping the free, occupied and unknown space which is essential for robotic navigation. The scene is defined in terms of voxels which are indexed through an octree. The problem is formulated as a hierarchical robust P^N Markov Random Field, defined on voxels and their groupings. This allows us to infer the 3D occupancy map and the object-class labels in a principled manner, through bounded approximate minimisation of a well defined and studied energy functional.

7.2 Limitations and Future Work

In this thesis, we have tried to address the greater problem of scene understanding by semantic mapping and reconstruction. This problem is still an active area of research and various new methods have been proposed towards joint structure and object labelling [78,109]. The problem becomes particularly challenging in case of city level understanding due to the scale and the variability present in the data. This calls for faster computation with efficient handling of data, fusion of various sources of information and more importantly adding higher level object information to make the output more meaningful.

In chapter 3, it is shown how a semantic overhead map can be generated from street level images. In this work, the assumption of a flat world is used to link the ground plane and the image plane. This has the potential to create shadow artefacts in the final output map (see Fig. 3.13). This is aggravated when sufficient views are not available to map the object in consideration. Such effects can be reduced, if we introduce higher level shape constraints and replace objects with synthetically synthesized blocks [19,75] to obtain the extent of the objects in the map. The shadow effects can also be reduced if we use the top view (satellite views) obtained from mapping data in conjunction to the street views for determining extent of classes like ‘roads’, ‘buildings’ and ‘vegetation’. However, integrating the satellite view is challenging as they are not captured in the same time, which might result in inconsistency of objects such as ‘vehicles’, ‘signage’ or ‘pedestrians’ present in the scene. Learning selectively from different modalities of data can improve the quality and accuracy of the generated map.

Similarly, a semantic map can be made more meaningful by adding higher level object information like bus-stand, shops, commercial/residential area, etc. Currently the mapping services like Google maps are void of semantics, but have large information in terms of these high level static objects, that can be used in our map building system effectively. Recently there have been such attempts to fuse information from OpenStreet Maps [133] to perform visual localisation [24]. Similarly, an outdoor text recognition system demonstrated in [128] can be used for localisation and identifying objects in the map domain. Finally, adding semantics through maps can increase the efficacy of the emerging wearable technologies like Google Glass [93] which often need to interact with mapping services.

Chapter 4 demonstrates a large scale dense 3D semantic reconstruction of an outdoor scene. In this method, inputs are taken from a stereo camera mounted on a moving vehicle capturing images at regular interval. This assumption of availability of stereo imagery is limiting in nature. Ideally, a system should be able to generate a dense reconstruction from monocular image streams. Recent developments in optical flow methodology allows to perform a detailed dense facial reconstruction from single video [166]. Such techniques are worth considering for outdoor road scenes, which contain large variations in terms of the number of objects, scale and lightings, making it a challenging task.

Another area where our work from chapter 4 can be carried forward is to generate a real-time semantic reconstruction of the scene. Today, better algorithms incorporating sophisticated models coupled with improved computing resources have enabled the field of computing to breach the barrier of time and perform real-time implementation of the vision algorithms. Significant progress has been made to perform a dense structure recovery from a stream of depth images from RGB-D sensors using GPU optimisations. However real-time semantics is still in its infancy and the challenges associated with semantic scene understanding in real-time are plenty. In this context, various GPU implementation of efficient feature computations available through libraries like OpenCV, PCL [141, 149] can be used to speed up the process. Various stages in the semantic scene recovery, such as classifier evaluation and inference needs to be investigated for efficiency and speed-up.

In chapter 5, we have exploited the structure and object layout in the scene to aid labelling and performed inference over a meshed scene. The appearance costs for every mesh location is computed from the images while the mesh connectivity information is used during the inference. The mesh can be further exploited to compute the geometric features for learning object class labels as explored in [98, 178]. The proposed method also requires to have an estimate of the scene beforehand. A joint reconstruction and class segmentation can be performed to overcome the necessity of meshed scene, where a classifier can be learned to generate a pixel wise depth and object class hypothesis [109]. Similarly, we can improve the reasoning of object class accuracy via incorporating higher level constraints from object detectors, and infer about the number of instances and spatial extent of individual objects in the scene [110].

In chapter 5, it is also demonstrated that performing inference in the structure speeds up the process in comparison to the images. This is due to the successive stream of images from a stereo camera contain redundant information of the scene. Redundant data can be further ignored from processing by intelligent selection of image pixels comprising the dynamic regions in the image and reusing past information for unchanged static parts. This is similar to human vision and senses, where our brain tends to focus more towards the dynamic regions in the scene. Recent dynamic vision systems [95] has been proposed which, unlike the conventional vision sensors, send the local pixel-level changes caused by movement in a scene in a synchronised fashion. Such sensors can help us achieve the necessary speed-up and deal with dynamic scenes efficiently, which is the case of an urban environment.

We have described in chapter 6 how 3D volume can be effectively subdivided into hierarchical sub-volumes and perform object recognition. Dividing the space into hierarchical regular grids helps us to reason about objects at different scales. However, it can be seen that many objects do not align with the grid boundaries and overlap with multiple grids. This issue can be addressed by considering non-uniform grids with object class sensitive grid partition while dividing the space using kd-trees [156]. Such partitions should be performed in a class wise fashion mimicking the physical properties of the objects. Finally the proposed method, whilst operating on the voxels, can be extended to represent dynamic scenes, through probabilistic occupancy updates [88]. Object categories provide valuable cues for dynamic scene processing, e.g. a building will not move but a vehicle is more likely to move. In such cases, the different classes should be handled separately from the rest of the static scene, which can enhance the capability of the current system.

Finally to conclude, adding more object level information by increasing the number of classes will take the process closer to total scene understanding [165]. Though this needs a large amount of labelled training data, as our training model needs evidence for each object in the learning stage. Currently, there are multiple training data available through various computer vision datasets [23, 51, 111] and are used for problems like object class segmentation, detection, stereo and classification. Ideally we should aim towards a system that is capable to learn about the scene from multiple sources via transfer learning approach [134], and use them effectively as required on any unseen data.

Chapter 8

Appendix: Datasets

8.1 Yotta Pembrokeshire Dataset

8.1.1 Yotta Pembrokeshire Street Image dataset

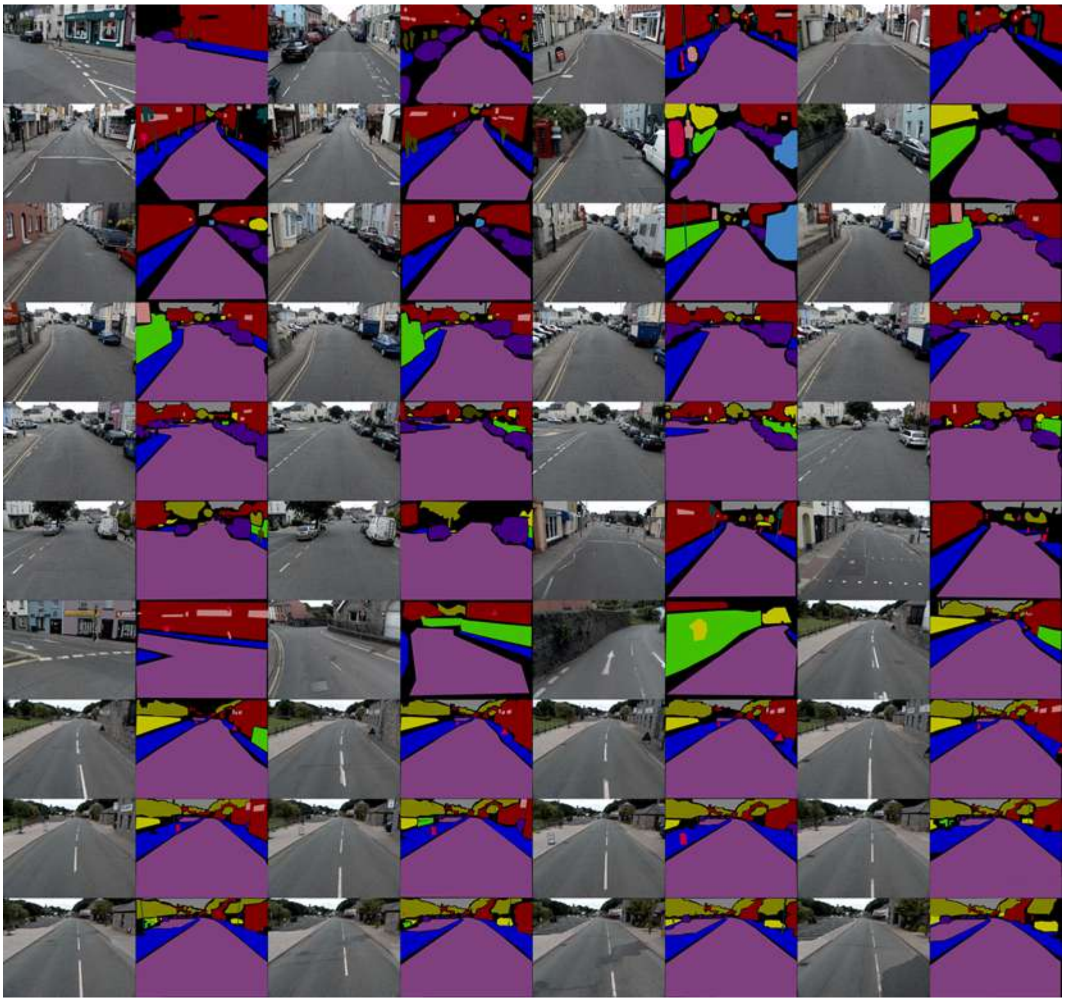

Figure 8.1: Yotta Pembrokeshire Street Image dataset [1]. (Best viewed in colour).

8.1.2 Yotta Pembrokeshire Aerial Image dataset

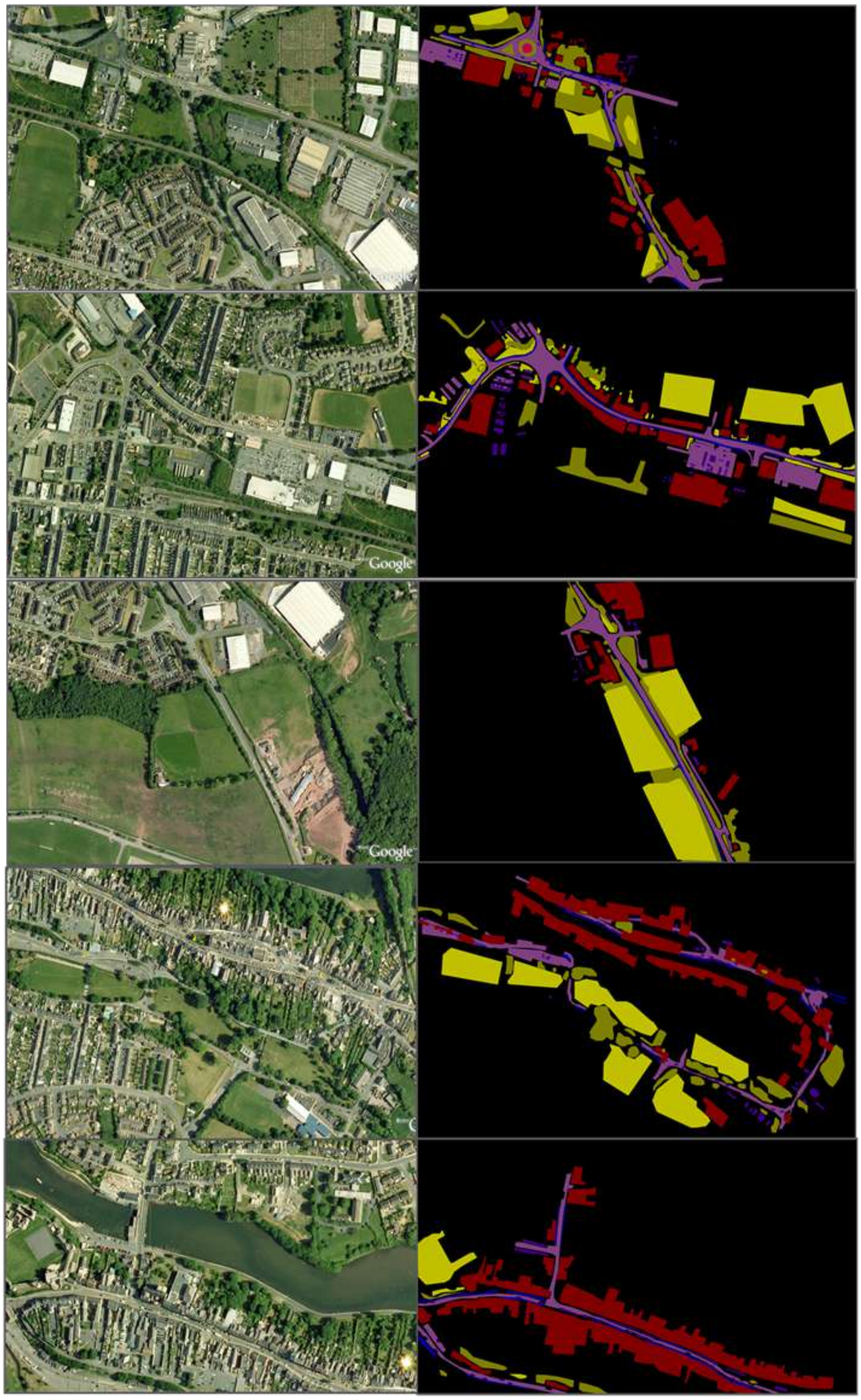

Figure 8.2: Yotta Pembrokeshire Aerial Image dataset [1]. (Best viewed in colour).

8.2 KITTI Object Label dataset

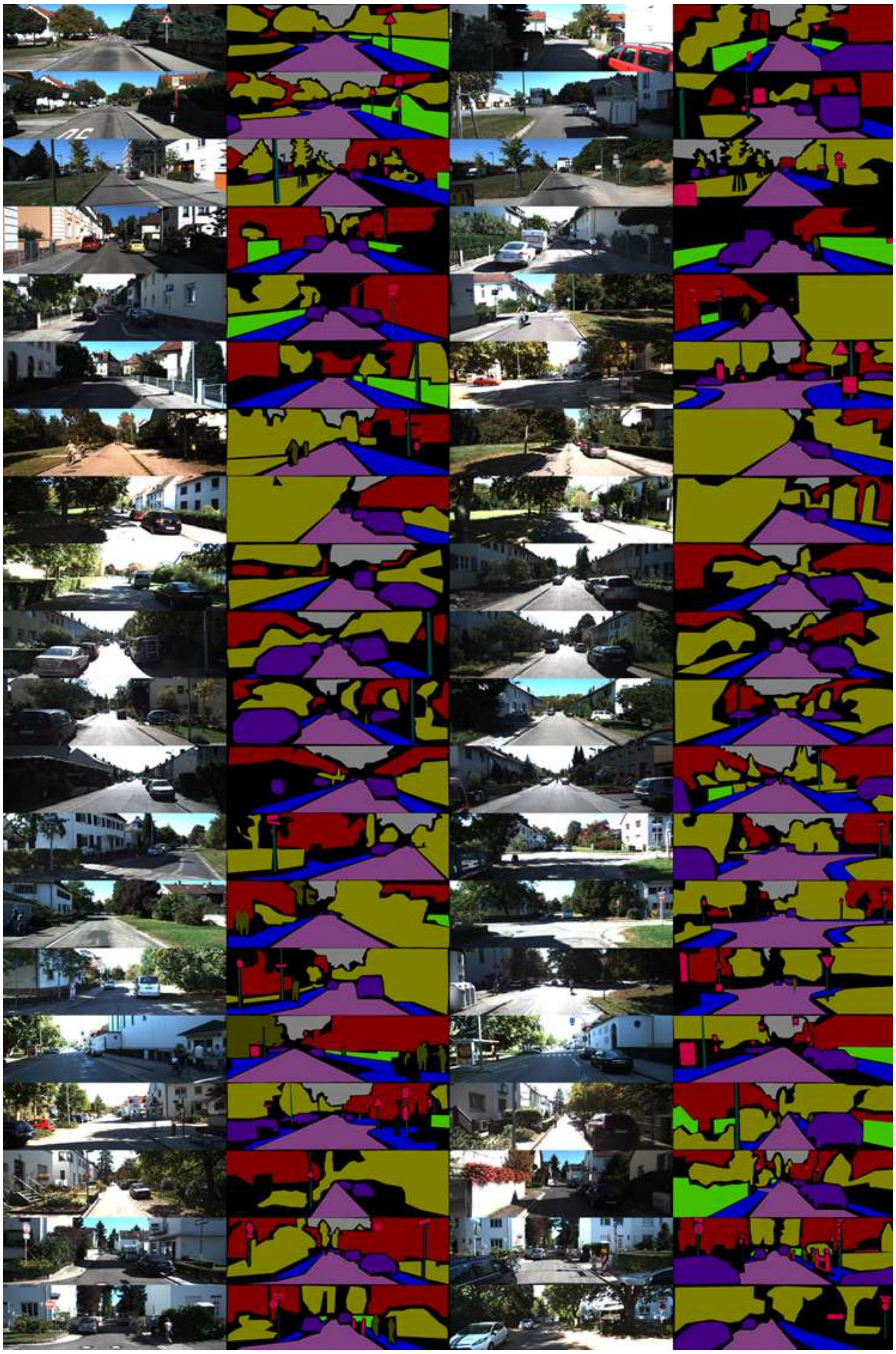

Figure 8.3: Kitti Object label dataset [1]. (Best viewed in colour).

Bibliography

[1] Sameer Agarwal, Yasutaka Furukawa, Noah Snavely, Ian Simon, Brian Curless, Steven M. Seitz, and Richard Szeliski. Building rome in a day. *Commun. ACM*, 54(10):105–112, October 2011.

[2] Sameer Agarwal, Noah Snavely, Steven M. Seitz, and Richard Szeliski. Bundle adjustment in the large. In *Proceedings of the European Conference on Computer Vision*, pages 29–42, 2010.

[3] M. Alexa, J. Behr, D. Cohen-Or, S. Fleishman, D. Levin, and Claudio T. Silva. Computing and rendering point set surfaces. *Visualization and Computer Graphics, IEEE Transactions on*, 9(1):3–15, 2003.

[4] Jose M. Alvarez, Yann LeCun, Theo Gevers, and Antonio M. Lopez. Semantic road segmentation via multi-scale ensembles of learned features. In *Proceedings of the European Conference on Computer Vision*, pages 586–595, 2012.

[5] Andrew J. Davison. Mobile robot navigation using active vision. *DPhil Thesis, University of Oxford*, 1998.

[6] Minoru Asada, Masahiro Kimura, Yasuhiro Taniguchi, and Yoshiaki Shirai. Dynamic integration of height maps into a 3d world representation from range image sequences. *International Journal of Computer Vision*, 9(1):31–53, October 1992.

[7] T. Bailey and H. Durrant-Whyte. Simultaneous localization and mapping (slam): Part ii. *Robotics & Automation Magazine, IEEE*, 13(3):108–117, 2006.

[8] H. Harlyn Baker and Thomas O. Binford. Depth from edge and intensity based stereo. In *Proceedings of the 7th international joint conference on Artificial intelligence - Volume 2*, IJCAI'81, pages 631–636, San Francisco, CA, USA, 1981. Morgan Kaufmann Publishers Inc.

[9] H.G. Barrow and J.M. Tenenbaum. Recovering intrinsic scene characteristics from images. *Computer Vision Systems*, pages 3–26, 1978.

[10] Matthew Berger, Joshua A. Levine, Luis Gustavo Nonato, Gabriel Taubin, and Claudio T. Silva. A benchmark for surface reconstruction. *ACM Transactions on Graphics*, 32(2):20:1–20:17, April 2013.

[11] Julian Besag. On the Statistical Analysis of Dirty Pictures. *Journal of the Royal Statistical Society. Series B (Methodological)*, 48(3):259–302, 1986.

[12] Stan Birchfield and Carlo Tomasi. A pixel dissimilarity measure that is insensitive to image sampling. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(4):401–406, April 1998.

[13] Stanley T. Birchfield, Braga Natarajan, and Carlo Tomasi. Correspondence as energy-based segmentation. *Image Vision Comput.*, 25(8):1329–1340, August 2007.

[14] J. Biswas and M. Veloso. Depth camera based indoor mobile robot localization and navigation. In *Precooedings of the IEEE International Conference on Robotics and Automation*, pages 1697–1702. Precooedings of the IEEE International Conference on Robotics and Automation, 2012.

[15] Y. Boykov and M. P. Jolly. Interactive graph cuts for optimal boundary and region segmentation of objects in N-D images. In *Proceedings of the International Conference on Computer Vision*, pages 105–112, 2001.

[16] 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.

[17] Yuri Boykov, Olga Veksler, and Ramin Zabih. Fast approximate energy minimization via graph cuts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23:2001, 2001.

[18] G. Bradski. The opencv library. *Dr. Dobb's Journal of Software Tools*, 2000.

[19] M. Brand, P. Cooper, and L. Birnbaum. Seeing physics, or: physics is for prediction [computer vision]. In *Physics-Based Modeling in Computer Vision, 1995., Proceedings of the Workshop on*, pages 144–, June 1995.

[20] Matthieu Bray, Pushmeet Kohli, and P. H.S. Torr. Posecut: Simultaneous segmentation and 3d pose estimation of humans using dynamic graph-cuts. In *European Conference on Computer Vision*, 2006.

[21] Alberto Broggi, Pietro Cerri, Mirko Felisa, Maria Chiara Laghi, Luca Mazzei, and Pier Paolo Porta. The VisLab Intercontinental Autonomous Challenge: an Extensive Test for a Platoon of Intelligent Vehicles. *Intl. Journal of Vehicle Autonomous Systems, special issue for 10th Anniversary*, 10(3), 2012. ISSN: 1471-0226.

[22] RodneyA. Brooks. Steps towards living machines. In Takashi Gomi, editor, *Evolutionary Robotics. From Intelligent Robotics to Artificial Life*, volume 2217 of *Lecture Notes in Computer Science*, pages 72–93. Springer Berlin Heidelberg, 2001.

[23] Gabriel J. Brostow, Jamie Shotton, Julien Fauqueur, and Roberto Cipolla. Segmentation and recognition using structure from motion point clouds. In *Proceedings of the European Conference on Computer Vision*, pages 44–57, 2008.

[24] Marcus A. Brubaker, Andreas Geiger, and Raquel Urtasun. Lost! leveraging the crowd for probabilistic visual self-localization. In *Proceedings of the Computer Vision and Pattern Recognition*, 2013.

[25] Martin Buehler, Karl Iagnemma, and Sanjiv Singh. *The 2005 DARPA Grand Challenge: The Great Robot Race*. Springer Publishing Company, Incorporated, 1st edition, 2007.

[26] J. Carreira and C. Sminchisescu. Cpmc: Automatic object segmentation using constrained parametric min-cuts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(7):1312–1328, 2012.

[27] João Carreira, Fuxin Li, and Cristian Sminchisescu. Object recognition by sequential figure-ground ranking. *International Journal of Computer Vision*, 98(3):243–262, July 2012.

[28] Andrew Chambers, Supreeth Achar, Stephen Nuske, Jorn Rehder, Bernd Kitt, Lyle Chamberlain, Justin Haines, Sebastian Scherer, and Sanjiv Singh. Perception for a river mapping robot. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 227–234, 2011.

[29] Jiawen Chen, S. Paris, Jue Wang, W. Matusik, M. Cohen, and F. Durand. The video mesh: A data structure for image-based three-dimensional video editing. In *Computational Photography (ICCP), 2011 IEEE International Conference on*, pages 1–8, 2011.

[30] Xiaobai Chen, Aleksey Golovinskiy, and Thomas Funkhouser. A benchmark for 3d mesh segmentation. In *ACM SIGGRAPH*, pages 73:1–73:12, New York, NY, USA, 2009. ACM.

[31] Andrea Cherubini and François Chaumette. Visual navigation of a mobile robot with laser-based collision avoidance. *International Journal of Robotics Research*, 32:189–205, 2012.

[32] G.K.M. Cheung, S. Baker, and T. Kanade. Visual hull alignment and refinement across time: a 3d reconstruction algorithm combining shape-from-silhouette with stereo. In *Proceedings of the Computer Vision and Pattern Recognition*, volume 2, pages 375–82, 2003.

[33] D.M. Cole and P.M. Newman. Using laser range data for 3d slam in outdoor environments. In *Precooedings of the IEEE International Conference on Robotics and Automation*, pages 1556–1563, 2006.

[34] D. Comaniciu and P. Meer. Mean shift: A robust approach toward feature space analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2002.

[35] Nico Cornelis, Bastian Leibe, Kurt Cornelis, and Luc Gool. 3d urban scene modeling integrating recognition and reconstruction. *International Journal of Computer Vision*, 78(2-3):121–141, July 2008.

[36] M.M. Crawford and M.R. Ricard. Hierarchical classification of sar data using a markov random field model. In *Image Analysis and Interpretation, 1998 IEEE Southwest Symposium on*, pages 81–86, Apr 1998.

[37] A. Criminisi, A. Blake, C. Rother, J. Shotton, and P. H. Torr. Efficient dense stereo with occlusions for new view-synthesis by four-state dynamic programming. *International Journal of Computer Vision*, 71(1):89–110, January 2007.

[38] Brian Curless and Marc Levoy. A volumetric method for building complex models from range images. In *Proceedings of ACM SIGGRAPH*, pages 303–312. ACM, 1996.

[39] H. Dahlkamp, Gary Bradski, Adrian Kaehler, D. Stavens, and Sebastian Thrun. Self-supervised monocular road detection in desert terrain. In *RSS*, Philadelphia, 2006.

[40] DARPA. The DARPA Urban Challenge, 2007. http://archive.darpa.mil/grandchallenge/.

[41] Andrew J. Davison. Real-time simultaneous localisation and mapping with a single camera. In *Proceedings of the International Conference on Computer Vision*, pages 1403–, Washington, DC, USA, 2003.

[42] Yotta DCL. Yotta dcl case studies, retrived April 2010. http://www.yottadcl.com/surveys/case-studies/.

[43] Ernst D. Dickmanns and Birger D. Mysliwetz. Recursive 3-d road and relative ego-state recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(2):199–213, February 1992.

[44] Dmitri Dolgov, Sebastian Thrun, Michael Montemerlo, and James Diebel. Path planning for autonomous vehicles in unknown semi-structured environments. *International Journal of Robotics Research*, 29(5):485–501, April 2010.

[45] B. Douillard, D. Fox, F. Ramos, and H. Durrant-Whyte. Classification and semantic mapping of urban environments. *International Journal of Robotics Research*, 30(1):5–32, January 2011.

[46] Bertrand Douillard, Dieter Fox, and Fabio Ramos. Laser and vision based outdoor object mapping. In *Robotics: Science and Systems*, 2008.

[47] Ye Duan, Liu Yang, Hong Qin, and Dimitris Samaras. Shape reconstruction from 3d and 2d data using pde-based deformable surfaces. In *Proceedings of the European Conference on Computer Vision*, volume 3023, pages 238–251, 2004.

[48] Chris Engels, Henrik Stewnius, and David Nistr. Bundle adjustment rules. In *In Photogrammetric Computer Vision*, 2006.

[49] Friedrich Erbs, Uwe Franke, and Beate Schwarz. Stixmentation - probabilistic stixel based traffic scene labeling. In *British Machine Vision Conference*, 2012.

[50] Carlos Hernández Esteban and Francis Schmitt. Silhouette and stereo fusion for 3d object modeling. *Comput. Vis. Image Underst.*, 96(3):367–392, December 2004.

[51] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2009 (VOC2009) Results.

[52] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2011 (VOC2011) Results. http://www.pascal-network.org/challenges/VOC/voc2011/workshop/index.html.

[53] Nathaniel Fairfield, George A Kantor, and David Wettergreen. Real-time slam with octree evidence grids for exploration in underwater tunnels. *Journal of Field Robotics*, 2007.

[54] Olivier Faugeras and Renaud Keriven. Variational principles, surface evolution, pde's, level set methods and the stereo problem. *IEEE Transactions on Image Processing*, 7:336–344, 1999.

[55] Georgios Floros and Bastian Leibe. Joint 2d-3d temporally consistent semantic segmentation of street scenes. In *Proceedings of the Computer Vision and Pattern Recognition*, pages 2823–2830, Washington, DC, USA, 2012. IEEE Computer Society.

[56] Thomas Fromherz and Martin Bichsel. Shape from multiple cues: Integrating local brightness information. In *Proceedings of the Fourth International Conference for Young Computer Scientists, ICYCS 95*, pages 855–862, 1995.

[57] P. Fua and Y.G. Leclerc. Object-centered surface reconstruction: Combining multi-image stereo and shading. *International Journal of Computer Vision*, 16(1):35–56, 1995.

[58] Pascal Fua and Peter T. Sander. Segmenting unstructured 3d points into surfaces. In *Proceedings of the European Conference on Computer Vision*, pages 676–680, 1992.

[59] Yasutaka Furukawa and Jean Ponce. Carved visual hulls for image-based modeling. *International Journal of Computer Vision*, 81(1):53–67, January 2009.

[60] Yasutaka Furukawa and Jean Ponce. Accurate, dense, and robust multiview stereopsis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(8):1362–1376, August 2010.

[61] Andrea Fusiello, Emanuele Trucco, and Alessandro Verri. A compact algorithm for rectification of stereo pairs. *Mach. Vision Appl.*, 12(1):16–22, July 2000.

[62] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *Proceedings of the Computer Vision and Pattern Recognition*, June 2012.

[63] Andreas Geiger, Julius Ziegler, and Christoph Stiller. Stereoscan: Dense 3d reconstruction in real-time. In *IEEE Intelligent Vehicles Symposium*, Baden-Baden, Germany, June 2011.

[64] S. Geman and D. Geman. Stochastic relaxation, gibbs distributions, and the bayesian restoration of images. *Transactions on Pattern Analysis and Machine Intelligence*, 1984.

[65] Stuart Geman and D. Geman. Stochastic relaxation, gibbs distributions, and the bayesian restoration of images. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, PAMI-6(6):721–741, 1984.

[66] Michael Goesele, Brian Curless, and Steven M. Seitz. Multi-view stereo revisited. In *Proceedings of the Computer Vision and Pattern Recognition*, pages 2402–2409, Washington, DC, USA, 2006.

[67] Google Earth. Google earth (version 6.0.0.1735 (beta)) [software]. mountain view, ca: Google inc.

[68] Google Inc. Google maps., 2010. maps.google.com.

[69] S. Gould, R. Fulton, and D. Koller. Decomposing a scene into geometric and semantically consistent regions. In *Proceedings of the International Conference on Computer Vision (ICCV)*, 2009.

[70] S. Gould, T. Gao, and D. Koller. Region-based segmentation and object detection. In *Advances in Neural Information Processing Systems*, 2009.

[71] F. Sebastin Grassia. Practical parameterization of rotations using the exponential map. *J. Graph. Tools*, 3(3):29–48, March 1998.

[72] D. M. Greig, B. T. Porteous, and A. H. Seheult. Exact maximum a posteriori estimation for binary images. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 271–279, 1989.

[73] Eric Greveson and Ali Sharokhni. Personal communications, June 2012.

[74] W. E. L. Grimson. A computer implementation of a theory of human stereo vision. *Philosophical Transactions of the Royal Society of London. Series B, Biological Sciences*, 292(1058), 1981.

[75] Abhinav Gupta, Alexei A. Efros, and Martial Hebert. Blocks world revisited: image understanding using qualitative geometry and mechanics. In *Proceedings of the European Conference on Computer Vision*, pages 482–496, Berlin, Heidelberg, 2010. Springer-Verlag.

[76] M. Habbecke and L. Kobbelt. A surface-growing approach to multi-view stereo reconstruction. In *Proceedings of the Computer Vision and Pattern Recognition*, pages 1–8, 2007.

[77] R. Hadsell, P. Sermanet, J. Ben, A. Erkan, J. Han, U. Muller, and Y. LeCun. Online learning for offroad robots: Spatial label propagation to learn long-range traversability. In *Proceedings of Robotics: Science and Systems*, Atlanta, GA, USA, June 2007.

[78] Christian Hane, Nikolay Savinov, and Marc Pollefeys. Class specific 3d object shape priors using surface normals. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2014.

[79] C. J. Harris and M. Stephens. A combined corner and edge detector. In *Proceedings of the 4th Alvey Vision Conferences*, pages 147–151, 1988.

[80] Richard Hartley and Andrew Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2003.

[81] Geremy Heitz and Daphne Koller. Learning spatial context: Using stuff to find things. In *Proceedings of the European Conference on Computer Vision*, pages 30–43, 2008.

[82] D. Held, J. Levinson, and S. Thrun. A probabilistic framework for car detection in images using context and scale. In *Precooedings of the IEEE International Conference on Robotics and Automation*, pages 1628 –1634, may 2012.

[83] Peter Henry, Michael Krainin, Evan Herbst, Xiaofeng Ren, and Dieter Fox. RGB-D mapping: Using Kinect-style depth cameras for dense 3D modeling of indoor environments. *International Journal of Robotics Research*, 31(5):647–663, April 2012.

[84] Stephen L Hicks, Iain Wilson, Louwai Muhammed, John Worsfold, Susan M Downes, and Christopher Kennard. A depth-based head-mounted visual display to aid navigation in partially sighted individuals. *PloS one*, 8(7):e67695, 2013.

[85] Heiko Hirschmuller. Stereo processing by semiglobal matching and mutual information. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(2):328–341, February 2008.

[86] Derek Hoiem, Alexei A. Efros, and Martial Hebert. Geometric context from a single image. In *Proceedings of the International Conference on Computer Vision*, volume 1, pages 654 – 661. IEEE, October 2005.

[87] Berthold K. P. Horn. Height and gradient from shading. *International Journal of Computer Vision*, 5(1):37–75, September 1990.

[88] Armin Hornung, Kai M. Wurm, Maren Bennewitz, Cyrill Stachniss, and Wolfram Burgard. OctoMap: An efficient probabilistic 3D mapping framework based on octrees. *Autonomous Robots*, 2013. Software available at `http://octomap.github.com`.

[89] Cathering Q. Howe and Dale Purves. *Perceiving Geometry - Geometrical Illusions Explained by Natural Scene Statistics*. Springer, 2005.

[90] Qixing Huang, Martin Wicke, Bart Adams, and Leo Guibas. Shape decomposition using modal analysis. *Computer Graphics Forum*, 28(2):407–416, 2009.

[91] D. H. Hubel and T. N. Wiesel. Receptive fields, binocular interaction and functional architecture in the cat's visual cortex. *The Journal of Physiology*, 160(1):106–154, 1962.

[92] Google Inc. Keyhole markup language 2.2 reference document., 2010.

[93] Google Inc. Google glass, 2013. http://www.google.com/glass/start/.

[94] Nokia Inc. Nokia ovi maps. nokia., 2010. maps.ovi.com.

[95] iniLabs Ltd. Dynamic vision sensor (dvs) - asynchronous temporal contrast silicon retina, 2013. http://www.inilabs.com/products/dvs128.

[96] John Isidoro and Stan Sclaroff. Stochastic refinement of the visual hull to satisfy photometric and silhouette consistency constraints. In *Proceedings of the Ninth IEEE International Conference on Computer Vision - Volume 2*, ICCV '03, pages 1335–, Washington, DC, USA, 2003. IEEE Computer Society.

[97] Hailin Jin, Stefano Soatto, and Anthony J. Yezzi. Multi-view stereo reconstruction of dense shape and complex appearance. *International Journal of Computer Vision*, 63(3):175–189, July 2005.

[98] Evangelos Kalogerakis, Aaron Hertzmann, and Karan Singh. Learning 3d mesh segmentation and labeling. In *SIGGRAPH*, 2010.

[99] G. Klein and D. W. Murray. Improving the agility of keyframe-based SLAM. In *Proceedings of the European Conference on Computer Vision*, 2008.

[100] P. Kohli, M.P. Kumar, and P. H S Torr. P3 and beyond: Move making algorithms for solving higher order functions. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 31(9):1645–1656, 2009.

[101] P. Kohli, L. Ladicky, and P. H. S. Torr. Robust higher order potentials for enforcing label consistency. In *Proceedings of the Computer Vision and Pattern Recognition*, 2008.

[102] V. Kolmogorov and R. Zabih. What energy functions can be minimized via graph cuts?. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2004.

[103] N. Komodakis, G. Tziritas, and N. Paragios. Fast, approximately optimal solutions for single and dynamic MRFs. In *Proceedings of the Computer Vision and Pattern Recognition*, 2007.

[104] H. Koppula, A. Anand, T. Joachims, and A. Saxena. Semantic labeling of 3d point clouds for indoor scenes. In *In proceedings of Advances in Neural Information Processing Systems*, 2011.

[105] A. F. Koschan. Perception-based 3d triangle mesh segmentation using fast marching watersheds. In *Proceedings of the Computer Vision and Pattern Recognition*, pages 27–32, 2003.

[106] Swarun Kumar, Lixin Shi, Nabeel Ahmed, Stephanie Gil, Dina Katabi, and Daniela Rus. Carspeak: a content-centric network for autonomous driving. *ACM SIGCOMM Computer Communication Review*, 42(4):259–270, 2012.

[107] K. N. Kutulakos and M. Seitz. A theory of shape by space carving. *International Journal of Computer Vision*, 38(3), 2000.

[108] Lubor Ladicky, Chris Russell, Pushmeet Kohli, and Philip H.S. Torr. Associative hierarchical crfs for object class image segmentation. In *Proceedings of the International Conference on Computer Vision*, 2009.

[109] Lubor Ladicky, Jianbo Shi, and Marc Pollefeys. Pulling things out of perspective. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2014.

[110] L'ubor Ladický, Paul Sturgess, Karteek Alahari, Chris Russell, and Philip H. S. Torr. What, where and how many? combining object detectors and crfs. In *Proceedings of the European Conference on Computer Vision*, pages 424–437, 2010.

[111] Lubor Ladicky, Paul Sturgess, Chris Russell, Sunando Sengupta, Yalin Bastanlar, William Clocksin, and Philip H.S. Torr. Joint optimisation for object class segmentation and dense stereo reconstruction. In *British Machine Vision Conference*, 2010.

[112] X. Lan, S. Roth, D. P. Huttenlocher, and M. J. Black. Efficient belief propagation with learned higher-order markov random fields. In *European Conference on Computer Vision*, pages 269–282, 2006.

[113] Svetlana Lazebnik, Cordelia Schmid, and Jean Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *Proceedings of the Computer Vision and Pattern Recognition*, pages 2169–2178, 2006.

[114] Bastian Leibe, Nico Cornelis, Kurt Cornelis, and Luc J. Van Gool. Dynamic 3d scene analysis from a moving vehicle. In *Proceedings of the Computer Vision and Pattern Recognition*, 2007.

[115] J. Leonard, J. P. How, S. Teller, M. Berger, S. Campbell, G. Fiore, L. Fletcher, E. Frazzoli, A. Huang, S. Karaman, O. Koch, Y. Kuwata, D. Moore, E. Olson, S. Peters, J. Teo, R. Truax, M. Walter, D. Barrett, A. Epstein, K. Maheloni, K. Moyer, T. Jones, R. Buckley, M. Antone, R. Galejs, S. Krishnamurthy, and J. Williams. *The DARPA Urban Challenge: Autonomous Vehicles in City Traffic*, volume 56, chapter A Perception-Driven Autonomous Urban Vehicle. Springer Verlag, 2010.

[116] M. E. Leventon, W. E. L. Grimson, and O. D. Faugeras. Statistical shape influence in geodesic active contours. In *Proceedings of the Computer Vision and Pattern Recognition*, pages 1316–1323, 2000.

[117] J. Levinson, J. Askeland, J. Becker, J. Dolson, D. Held, S. Kammel, J.Z. Kolter, D. Langer, O. Pink, V. Pratt, M. Sokolsky, G. Stanek, D. Stavens, A. Teichman, M. Werling, and S. Thrun. Towards fully autonomous driving: Systems and algorithms. In *IEEE Intelligent Vehicles*, pages 163 –168, june 2011.

[118] S. Z. Li. *Markov Random Field Modeling in Computer Vision*. Springer-Verlag, London, UK, 1995.

[119] Rong Liu and Hao Zhang. Segmentation of 3d meshes through spectral clustering. In *Proceedings of the Computer Graphics and Applications, 12th Pacific Conference*, PG '04, pages 298–305, Washington, DC, USA, 2004. IEEE Computer Society.

[120] William E. Lorensen and Harvey E. Cline. Marching cubes: A high resolution 3d surface construction algorithm. In *Proceedings of ACM SIGGRAPH*, pages 163–169, 1987.

[121] Mistsubishi Heavy Inndustries Ltd. Communication robot wakamuru. http://www.mhi.co.jp/en/products/detail/wakamaru.html.

[122] David Marr. *Vision: A Computational Investigation into the Human Representation and Processing of Visual Information*. Henry Holt and Co., Inc., New York, NY, USA, 1982.

[123] David Marr and T. Poggio. Cooperative computation of stereo disparity. *Science*, 194:283–287, 1976.

[124] Annalisa Milella, Giulio Reina, James Patrick Underwood, and Bertrand Douillard. Combining radar and vision for self-supervised ground segmentation in outdoor environments. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 255–260, 2011.

[125] G. Monteiro, C. Premebida, P. Peixoto, and U. Nunes. Tracking and classification of dynamic obstacles using laser range finder and vision. In *In Workshop on "safe navigation in open and dynamic environments–autonomous systems versus driving assistance systems" at the IEEE/RSJ International Conference on Intelligent Robots and Systems.*, 2006.

[126] Gabriele Moser, Vladimir Krylov, Sebastiano B. Serpico, and Josiane Zerubia. High resolution sar-image classification by markov random fields and finite mixtures. In *Proc. SPIE*, volume 7533, pages 08–12, 2010.

[127] Pushmeet Kohli Nathan Silberman, Derek Hoiem and Rob Fergus. Indoor segmentation and support inference from rgbd images. In *Proceedings of the European Conference on Computer Vision*, 2012.

[128] Lukas Neumann and Jiri Matas. Real-time scene text localization and recognition. In *Proceedings of the Computer Vision and Pattern Recognition*, pages 3538–3545, 2012.

[129] Richard A. Newcombe, Shahram Izadi, Otmar Hilliges, David Molyneaux, David Kim, Andrew J. Davison, Pushmeet Kohli, Jamie Shotton, Steve Hodges, and Andrew Fitzgibbon. Kinectfusion: Real-time dense surface mapping and tracking. In *IEEE ISMAR*, pages 127–136, 2011.

[130] Paul Newman. Robotcar uk, retrived July 2013. http://mrg.robots.ox.ac.uk/robotcar/.

[131] Paul Newman, Gabe Sibley, Mike Smith, Mark Cummins, Alastair Harrison, Chris Mei, Ingmar Posner, Robbie Shade, Derik Schroeter, Liz Murphy, Winston Churchill, Dave Cole, and Ian Reid. Navigating, recognizing and describing urban spaces with vision and lasers. *International Journal of Robotics Research*, 28(11-12):1406–1433, November 2009.

[132] Andreas Nüchter and Joachim Hertzberg. Towards semantic maps for mobile robots. *Robotics and Autonomous Systems*, 56(11):915–926, November 2008.

[133] OpenStreetMap Foundation. Openstreetmaps., 2012.

[134] Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, 22(10):1345–1359, October 2010.

[135] K. Pathak, A. Birk, J. Poppinga, and S. Schwertfeger. 3d forward sensor modeling and application to occupancy grid based sensor fusion. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2059–2064, 2007.

[136] B.A. Payne and A.W. Toga. Surface mapping brain functions on 3d models. In *IEEE Computer Graphics and Applications*, 1990.

[137] D. Pfeiffer and U. Franke. Modeling dynamic 3d environments by means of the stixel world. *Intelligent Transportation Systems Magazine, IEEE*, 3(3):24–36, 2011.

[138] Jean Ponce, Martial Hebert, Cordelia Schmid, and Andrew Zisserman, editors. *Toward Category-Level Object Recognition*, volume 4170 of *Lecture Notes in Computer Science*. Springer, 2006.

[139] Ingmar Posner, Mark Cummins, and Paul M. Newman. A generative framework for fast urban labeling using spatial and temporal context. *Auton. Robots*, 26(2-3):153–170, 2009.

[140] Ingmar Posner, Derik Schroeter, and Paul Newman. Using scene similarity for place labelling. *Springer Tracts in Advanced Robotics*, 39:85–98, 2008.

[141] Kari Pulli, Anatoly Baksheev, Kirill Kornyakov, and Victor Eruhimov. Realtime computer vision with opencv. *ACM Queue*, 10(4):40:40–40:56, April 2012.

[142] Morgan Quigley, Siddharth Batra, Stephen Gould, Ellen Klingbeil, Quoc Le, Ashley Wellman, and Andrew Y. Ng. High-accuracy 3d sensing for mobile manipulation: Improving object detection and door opening. In *Precooedings of the IEEE International Conference on Robotics and Automation*, 2009.

[143] C. Rhemann, A. Hosni, M. Bleyer, C. Rother, and M. Gelautz. Fast cost-volume filtering for visual correspondence and beyond. In *Proceedings of the Computer Vision and Pattern Recognition*, pages 3017–3024, Washington, DC, USA, 2011.

[144] Rethink Robotics. Baxter. http://www.rethinkrobotics.com/products/baxter/.

[145] S. Roth and M. J. Black. Fields of experts: A framework for learning image priors. In *Conference on Computer Vision and Pattern Recognition*, pages 860–867, 2005.

[146] Y. Roth-Tabak and R. Jain. Building an environment model using depth information. *Computer*, 22(6):85–90, 1989.

[147] C. Rother, V. Kolmogorov, and A. Blake. Grabcut: interactive foreground extraction using iterated graph cuts. In *SIGGRAPH*, pages 309–314, 2004.

[148] Sebastien Roy and Ingemar Cox. A maximum-flow formulation of the n-camera stereo correspondence problem. In *Proceedings of the International Conference on Computer Vision*, 1998.

[149] Radu Bogdan Rusu and Steve Cousins. 3D is here: Point Cloud Library (PCL). In *Precooedings of the IEEE International Conference on Robotics and Automation*, Shanghai, China, May 9-13 2011.

[150] D. Scaramuzza and F. Fraundorfer. Visual odometry [tutorial]. *Robotics & Automation Magazine, IEEE*, 18(4):80–92, December 2011.

[151] Daniel Scharstein and Richard Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *International Journal of Computer Vision*, 47(1-3):7–42, April 2002.

[152] S.M. Seitz, B. Curless, J. Diebel, D. Scharstein, and R. Szeliski. A comparison and evaluation of multi-view stereo reconstruction algorithms. In *Proceedings of the Computer Vision and Pattern Recognition*, volume 1, pages 519–528, 2006.

[153] Steven M. Seitz and Charles R. Dyer. Photorealistic scene reconstruction by voxel coloring. In *Proceedings of the Computer Vision and Pattern Recognition*, pages 1067–, 1997.

[154] Sunando Sengupta, Eric Greveson, Ali Shahrokni, and Philip Torr. Urban 3d semantic modelling using stereo vision. In *Precooedings of the IEEE International Conference on Robotics and Automation*, 2013.

[155] Sunando Sengupta, P. Sturgess, L. Ladicky, and P.H.S. Torr. Automatic dense visual semantic mapping from street-level imagery. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, oct 2012.

[156] Maxim Shevtsov, Alexei Soupikov, and Alexander Kapustin. Highly parallel fast kd-tree construction for interactive ray tracing of dynamic scenes. *Computer Graphics Forum*, 26(3), 2007.

[157] J. Shi and J. Malik. Normalized cuts and image segmentation. *Trans. Pattern Anal. Mach. Intell.*, 22(8):888–905, 2000.

[158] Jamie Shotton, John Winn, Carsten Rother, and Antonio Criminisi. Textonboost for image understanding: Multi-class object recognition and segmentation by jointly modeling texture, layout, and context. In *Proceedings of the International Conference on Computer Vision*, 2009.

[159] Nathan Silberman and Rob Fergus. Indoor scene segmentation using a structured light sensor. In *Proceedings of the International Conference on Computer Vision*, 2011.

[160] Noah Snavely, Steven M. Seitz, and Richard Szeliski. Photo tourism: exploring photo collections in 3d. In *ACM SIGGRAPH 2006 Papers*, SIGGRAPH '06, pages 835–846, New York, NY, USA, 2006. ACM.

[161] Byung soo Kim, Pushmeet Kohli, and Silvio Savarese. 3d scene understanding by voxel-crf. In *Proceedings of the International Conference on Computer Vision*, December, 2013.

[162] Ioannis Stamos and Peter K. Allen. 3-d model construction using range and image data. In *Proceedings of the Computer Vision and Pattern Recognition*, pages 531–536, 2000.

[163] J. Stuckler, N. Biresev, and S. Behnke. Semantic mapping using object-class segmentation of rgb-d images. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3005–3010, 2012.

[164] P. Sturgess, K. Alahari, L. Ladicky, , and Philip H. S. Torr. Combining appearance and structure from motion features for road scene understanding. In *British Machine Vision Conference*, 2009.

[165] Paul Sturgess, Lubor Ladick, Nigel Crook, and Philip Torr. Scalable cascade inference for semantic image segmentation. In *Proceedings of the British Machine Vision Conference*, pages 62.1–62.10, 2012.

[166] Supasorn Suwajanakorn, Ira Kemelmacher-Shlizerman, and Steven Seitz. Total moving face reconstructio. In *European Conference on Computer Vision*, September 2014.

[167] R. Szeliski and P. Golland. Stereo matching with transparency and matting. In *Proceedings of the International Conference on Computer Vision*, pages 517–524, 1998.

[168] R. Szeliski, R. Zabih, D. Scharstein, O. Veksler, V. Kolmogorov, A. Agarwala, M. F. Tappen, and C. Rother. A comparative study of energy minimization methods for markov random fields. In *Proceedings of the European Conference on Computer Vision*, 2006.

[169] Richard Szeliski. *Computer Vision: Algorithms and Applications*. Springer-Verlag New York, Inc., New York, NY, USA, 1st edition, 2010.

[170] Camillo J. Taylor. Surface reconstruction from feature based stereo. In *Proceedings of the International Conference on Computer Vision*, pages 184–192, Washington, DC, USA, 2003.

[171] Alex Teichman and Sebastian Thrun. Practical object recognition in autonomous driving and beyond. In *IEEE Workshop on Advanced Robotics and its Social Impacts*, pages 35–38. IEEE, 2011.

[172] Sebastian Thrun. What we're driving at, retrived July 2013. http://googleblog.blogspot.co.uk/2010/10/what-were-driving-at.html.

[173] Sebastian Thrun, Michael Montemerlo, Hendrik Dahlkamp, David Stavens, Andrei Aron, James Diebel, Philip Fong, John Gale, Morgan Halpenny, Gabriel Hoffmann, Kenny Lau, Celia Oakley, Mark Palatucci, Vaughan Pratt, Pascal Stang, Sven Strohband, Cedric Dupont, Lars-Erik Jendrossek, Christian Koelen, Charles Markey, Carlo Rummel, Joe van Niekerk, Eric Jensen, Philippe Alessandrini, Gary Bradski, Bob Davies, Scott Ettinger, Adrian Kaehler, Ara Nefian, and Pamela Mahoney. Stanley: The robot that won the darpa grand challenge. *Journal of Field Robotics*, 23(9):661–692, June 2006.

[174] P. H. S. Torr and A. Zisserman. Mlesac: A new robust estimator with application to estimating image geometry. *Comput. Vis. Image Underst.*, 78(1):138–156, April 2000.

[175] Adrien Treuille, Aaron Hertzmann, and StevenM. Seitz. Example-based stereo with general brdfs. In *Proceedings of the European Conference on Computer Vision*, volume 3022, pages 457–469, 2004.

[176] R. Triebel, K. Kersting, and W. Burgard. Robust 3d scan point classification using associative markov networks. In *Precooedings of the IEEE International Conference on Robotics and Automation*, pages 2603–2608, 2006.

[177] Bill Triggs, Philip McLauchlan, Richard Hartley, and Andrew Fitzgibbon. Bundle adjustment – A modern synthesis. In W. Triggs, A. Zisserman, and R. Szeliski, editors, *Vision Algorithms: Theory and Practice*, LNCS, pages 298–375. Springer Verlag, 2000.

[178] J.P.C. Valentin, S. Sengupta, J. Warrell, A Shahrokni, and P.H.S. Torr. Mesh based semantic modelling for indoor and outdoor scenes. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pages 2067–2074, June 2013.

[179] Vibhav Vineet, Jonathan Warrell, and Philip H. S. Torr. Filter-based mean-field inference for random fields with higher order terms and product label-spaces. In *Proceedings of the European Conference on Computer Vision*, pages 1–10, 2012.

[180] Parma University VisLab. The vislab intercontinental autonomous challenge, 2010. http://viac.vislab.it/.

[181] G. Vogiatzis, P. H S Torr, and R. Cipolla. Multi-view stereo via volumetric graph-cuts. In *Proceedings of the Computer Vision and Pattern Recognition*, volume 2, pages 391–398 vol. 2, 2005.

[182] Susan Wardle and Barbara Gillam. Phantom surfaces in da vinci stereopsis. *Journal of Vision*, 13(2), 2013.

[183] Charles Wheatstone. Contributions to the physiology of vision. part the first. on some remarkable, and hitherto unobserved, phenomena of binocular vision. *Philosophical Transactions of the Royal Society of London*, 128:371–394, January 1838.

[184] Kai M. Wurm, Rainer Kümmerle, Cyrill Stachniss, and Wolfram Burgard. Improving robot navigation in structured outdoor environments by identifying vegetation from laser data. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1217–1222, 2009.

[185] Jianxiong Xiao and Long Quan. Multiple view semantic segmentation for street view images. In *Proceedings of the International Conference on Computer Vision*, pages 686–693, 2009.

[186] Hans-Peter Seidel Yutaka Ohtake, Alexander Belyaev. An integrating approach to meshing scattered point data. In *ACM Symposium on Solid and Physical Modeling*, 2005.

[187] Chenxi Zhang, Liang Wang, and Ruigang Yang. Semantic segmentation of urban scenes using dense depth maps. In *Proceedings of the European Conference on Computer Vision*, pages 708–721, 2010.