Recognition, Reorganisation, Reconstruction and Reinteraction for Scene Understanding

Vibhav Vineet

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

Doctor of Philosophy

Oxford Brookes University

2014

Abstract

Perceiving 3D structure and recognizing objects and their properties around us is central to our understanding of the world. For example, when we drive a car from our home to the workplace, we constantly perceive 3D structure, recognise objects and their properties, and understand their functional attributes so as to interact with the environment. Such capabilities permit free and accurate movement in unknown environments and may seem like an easy task for humans. However, for computer systems using artificial vision, it is not. Thus, researchers from philosophy to neuroscience, from mathematics to computer science, have devoted ample time to understand the underlying principles for developing a vision system which would be able to see as well as we do. Such understanding of (sequences of) images is commonly known as *Scene Understanding*. It consists of solving three classical computer vision problems: recognition, reorganisation and reconstruction. In this dissertation, I focus on some of these problems and propose methods for solving them. The work can be divided into three main parts.

In the first part, I show how the problem of recognition and reorganisation can be improved by incorporating some prior information such as context. Specifically I propose novel algorithms to incorporate higher order information, such as context and label consistency over large regions efficiently in the MRF model with only unary and/or pairwise terms. Inference in a MRF is performed using a filter-based mean-field approach. I demonstrate this techniques on joint object and stereo labelling problems, as well as on object class segmentation, showing in addition for joint object-stereo labelling how the method provides an efficient approach for inference in product label spaces.

In the second part I propose methods that encapsulate the benefits of reconstruction, recognition and reorganisation so as to solve scene understanding problems. First I propose robust real-time systems that reconstruct dense 3D models of environments on-the-fly and associates them with object labels. This approach works for both indoor and outdoor scenes and scale to any size environments. Next I propose an algorithm to solve the problems of recovering intrinsic scene properties such as shape, reflectance and illumination from a single image, along with estimating the object and attribute segmentation separately. I formulate this joint estimation problem in an energy minimization framework which is able to capture the correlations between intrinsic properties (reflectance, shape, illumination), objects (table, tv-monitor), and materials (wooden, plastic) in a given scene. Finally I design an efficient filter-based mean-field algorithm that jointly estimates human segmentation, pose and depth given a pair of stereo images so as to capture the relationships between these three problems.

In the third part I show how human interaction can help in improving the

visual recognition task. I propose an interactive 3D labelling and segmentation system that aims to make acquiring segmented 3D models fast, simple, and user-friendly. Carrying a body-worn depth camera, the environment is reconstructed using standard techniques. The user is able to reach out and touch surfaces in the world, and provide object category labels through voice commands. These user provided data are used to learn random forest based object models on-the-fly. Now when the user encounters a previously unobserved and unlabelled region of space, the forest predicts object labels for each voxel, and the volumetric mean-field based inference smooths the final output. I demonstrate compelling results on several sequences that generalizes to unseen regions of the world.

to ma, pappa, bhaiya

Acknowledgements

This thesis would not have been possible without continuous support of many people. First I would like to thank my supervisor Phil who invited me to come to England to do research in his lab. His vision, passion and craziness towards research is beyond imagination, and I would like to thank him from bottom of my heart for imparting his knowledge to me and helping me to evolve into not only a better researcher but also into a better human being. Then I would like to thank Jonathan who helped me to start with my research very early in my PhD. They helped me to not only think differently, but also in improving my writing and presentation styles. I would also like to thank Prof. Antonio Torralba for examining this thesis.

I was very fortunate to work with many great researchers during my PhD. I would like to thank Shahram Izadi, Jamie Shotton, Pushmeet Kohli and Carsten Rother who introduced me to many new areas of computer vision research. I would also like to thank Fred Nicolls who took a great effort to read through my thesis and provided many comments which helped to improve the quality of this thesis. I would also like to thank Andrew Zisserman, Fabio Cuzzolin, Richard Hartley, Andrea Vedaldi, Fredrick Kahl, Niloy Mitra and Nigel Crook who shared their knowledge with me.

I would now like to thank my friends and co-authors from my lab who beyond doubt are some of the smartest people I met during my research career. We spent many nights discussing various issues that I faced while solving some problems and understanding many fundamental concepts. For this, I would like to thank Paul, Sunando and Ondra. I would also like to thank other members of my lab: Mike, Sam, Ziming, Glenn, Julien, Lubor, Morten, Ming Ming, and Kyle. They helped me to enjoy life in this great city of Oxford. We took time out of our work to go to several restaurants and pubs in Oxford. I had also great pleasure in working with researchers at Microsoft and Stanford, Matthias, David, Christopher, Christoph, Michael. I would also like to thank some people from the Visual Geometry Group at the University of Oxford. They are Omkar, Varun, Relja, Minh, Yusuf, and Karen.

Finally I would like to thank my parents and my brother for their continuous support, encouragement and belief in me. I can not imagine the pain and hardships that they went through in order to raise me and to help me to reach to this level. This thesis will always remind me how hard is it for the poorest of the families in India to raise their children. I would also like to thank Neetu, whose smile always encouraged me to work harder.

Contents

1 Introduction **1**

1.1 Scene Understanding Problems **4**

1.1.1 Object and Attribute Segmentation 4

1.1.2 3D Scene Reconstruction and Recognition 4

1.1.3 Intrinsic Scene Decomposition 7

1.1.4 Human Pose Estimation 9

1.2 Why Scene Understanding Is Hard **9**

1.3 Thesis Approach **12**

1.3.1 World is Structured 12

1.3.2 Reconstruction, Recognition and Reorganisation 12

1.3.3 Human Interaction 14

1.3.4 Probabilistic Framework 14

1.4 Thesis Contributions **16**

1.5 Outline of the Thesis **17**

1.6 Publications **19**

2 Labelling Problems and Probabilistic Models **22**

2.1 Labelling Problems **23**

2.2 Random Field Models **24**

2.2.1 Markov Random Field 25

2.2.2 Conditional Random Field 26

2.2.3 Pairwise Model 27

2.2.4 Higher Order Models 29

3 Mean-field Inference **33**

3.1 Mean-field Inference **34**

3.1.1 Distance Function 35

3.1.2 Naive mean-field 37

3.1.3 Minimizing the distance 38

3.2 Improving naive mean-field approximation **42**

3.2.1 Fully connected pairwise model 43

3.2.2 Structured mean-field 46

3.2.3 Mixture approximations 48

3.3 Properties of naive mean-field approach 49
3.3.1 Mean-field and belief propagation algorithms . . . 50
3.3.2 Non-convexity of naive mean-field . . . 52
3.3.3 Estimating MAP solution . . . 53

4 Filter-based Mean-field Inference for Random Fields with Higher Order Terms and Product Label-spaces 56
4.1 Introduction 57
4.2 Filter-based Inference in Dense Pairwise CRFs 59
4.3 Inference in Models with Higher-order Terms 61
4.3.1 Pattern-based Potentials . . . 62
4.3.2 Co-occurrence Potentials . . . 64
4.4 Inference in Models with Product Label Spaces 67
4.4.1 Joint Formulation for Object and Stereo Labelling . . . 68
4.4.2 Mean-field Updates . . . 68
4.4.3 Cost Volume Filtering . . . 69
4.5 Experiments 70
4.5.1 Implementation Details . . . 70
4.5.2 Joint Object and Stereo Labelling . . . 71
4.5.3 Object Class Segmentation . . . 73
4.6 Mean-field Analysis 75
4.6.1 Mean-Field Vs. Graph-cuts Inference . . . 76
4.6.2 Sensitivity to Initialization . . . 79
4.6.3 General Gaussian mixture pairwise terms . . . 80
4.6.4 Convergence Analysis . . . 85
4.7 Discussion 85

5 Higher Order Priors for Joint Intrinsic Image, Objects, and Attributes Estimation 88
5.1 Introduction 89
5.2 Problem Formulation 91
5.2.1 SIRFS model for a single, given object mask . . . 91
5.2.2 Multilabel Object and Attribute Model . . . 92
5.3 Joint Model for Intrinsic Images, Objects and Attributes 93
5.3.1 SIRFS model for a scene . . . 94
5.3.2 Reflectance, Objects term . . . 94
5.3.3 Reflectance, Attributes term . . . 95

5.4 Inference and Learning **96**
5.5 Experiments **101**
5.5.1 aNYU 2 dataset 101
5.5.2 aPascal dataset 102
5.6 Discussion **103**

6 SemanticPaint: Personalized 3D Recognition at your Fingertips **110**
6.1 Introduction **111**
6.2 Related Work **113**
6.3 System Pipeline **116**
6.3.1 Smoothly Segmenting the Volume 118
6.3.2 Learning to Segment the Dynamic World 125
6.4 Experiments **132**
6.4.1 Qualitative results 133
6.4.2 Quantitative results 134
6.5 Discussion **136**
6.6 Conclusions **137**

7 Dense Semantic Stereo Fusion for Large Scale Semantic Scene Reconstruction **150**
7.1 Introduction **151**
7.2 Large Scale Semantic Reconstruction **153**
7.2.1 Depth Estimation 153
7.2.2 Surface Reconstruction 154
7.2.3 Semantic Fusion and TSDF space 156
7.2.4 Volumetric Mean-field 156
7.3 Experiments **160**
7.3.1 Conclusion 162

8 Applications **173**
8.1 Introduction **174**
8.2 Overview of Dense Random Field Formulation **176**
8.2.1 Joint Formulation 177
8.2.2 Joint energy function 178
8.3 Inference in the Joint Model **180**
8.3.1 Efficient Inference 182

8.4 Learning **183**
8.5 Experiments **184**
8.5.1 H2View dataset 184
8.5.2 Buffy dataset 186
8.6 Discussion **187**

9 Discussions **189**
9.1 Contribution of the Thesis **190**
9.2 Limitations **192**
9.3 Future Work **193**

Bibliography **196**

List of Figures

1.1 Scene understanding problems. 2
1.2 3D kitchen hidden in the classified. 3
1.3 Object and attributes labelling problems. 5
1.4 Scene reconstruction and recognition. 5
1.5 Understanding the functional properties of the objects. 6
1.6 Assistive technologies. 7
1.7 Autonomous driving. 7
1.8 Intrinsic scene decomposition. 8
1.9 Shadows removal for outdoor scene understanding. 8
1.10 Human pose estimation. 9
1.11 Best performing algorithm on Pascal dataset. 10
1.12 Adelson and Pentland's workshop metaphor 11
1.13 Higher order information. 13
1.14 Recognition, reorganisation and reconstruction. 14
1.15 Human interaction for object recognition. 15
1.16 Markov random field and mean-field method. 15
2.1 Labelling problems in computer vision. 24
2.2 Grid Vs Dense MRFs. 26
2.3 Comparison between unary and pairwise MRF on tsukuba image. 28
2.4 Oversmoothing due to grid CRF. 29
2.5 Dense pairwise CRF output. 30
2.6 Qualitative results on PascalVOC-10 dataset. 30
2.7 Pattern based potential for texture restoration. 32
3.1 Mean-field method. 34
3.2 Naive mean-field approximation. 37
3.3 A simple 4-node MRF. 40
3.4 Q distribution values across different iterations of the mean-field. 42
3.5 Output of filter-based mean-field on MSRC dataset. 46
3.6 Convergence analysis of filter-based mean-field approach. 47
3.7 Affect of increasing the density of graph on accuracy 47
3.8 Side-effects of using long-range interaction on object labelling problems. 48
3.9 Structured mean-field approach. 48
3.10 Comparison between mean-field and belief propagation free energies. 52
3.11 Marginal, local and mean-field polytope. 53
3.12 Mean-field and graph-cuts energies. 55
4.1 Qualitative results on Leuven dataset. 72

4.2 Qualitative results on PascalVOC-10 dataset. 75
4.3 Comparison of inference algorithms on PascalVOC-10 using matched energies 77
4.4 Qualitative improvement in α-expansion output. 79
4.5 Qualitative results on PascalVOC-10 before and after better initialization. 79
4.6 Qualitative results on PascalVOC-10 and CamVid datasets. . . 84
4.7 Convergence analysis of higher order mean-field inference. . . . 86
4.8 Q distribution values across different iterations of the mean-field. 86
5.1 Joint estimation of the intrinsic properties, object and attribute labels. 91
5.2 Qualitative results of our algorithm in estimating better intrinsic properties. 105
5.3 Qualitative results on aNYU [163]. 105
5.4 Qualitatively comparison of our algorithm in estimating better the intrinsic properties on aNYU dataset. 107
5.5 More qualitatively results for the intrinsic properties on aNYU dataset. 108
5.6 More qualitatively results for the intrinsic properties on aNYU dataset. 109
6.1 Visualisation of our system. 114
6.2 Overview of the recognition pipeline. 117
6.3 User interaction. 120
6.4 A toy example of splitting a reservoir. 128
6.5 Illustration of Voxel-Oriented Patches (VOPs). 131
6.6 Label propagation. 138
6.7 Forest predictions and final mean-field inference results for two scenes. 139
6.8 Quantitative results for the proposed VOP and baseline features on different scenes. One can observe that the proposed VOP feature leads to qualitatively better results than all the baseline approaches. 140
6.9 More forest evaluation. 141
6.10 Intern Area Sequence. 142
6.11 Intern Area Sequence Meshes. 143
6.12 Kitchen Area Sequence. 144
6.13 Kitchen Area Sequence Meshes. 145
6.14 Seating Area Sequence. 146
6.15 Seating Area Sequence Meshes. 147
6.16 Living Area Sequence. 148

6.17 Living Area Sequence Meshes. 149
7.1 Pipeline of our approach for outdoor scene reconstruction and recognition. 154
7.2 Qualtitative results on the KITTI dataset. 162
7.3 Qualtitative results on the KITTI dataset. 163
7.4 Semantic fusion approach for reconstruction in environments with many moving objects. 163
7.5 Recovering reconstruction and labelling of fine and thin objects such as trees. 164
7.6 Semantic fusion and RSLAM based approach to recover the good 3D reconstruction. 164
7.7 Reconstruction of an urban scene. 165
7.8 Rendered RGB views. 166
7.9 Semantic 3D reconstruction. 167
7.10 More results for semantic 3D reconstruction. 168
7.11 Reconstruction of an urban scene. 169
7.12 Reconstruction of an urban scene. 170
7.13 Reconstruction of an urban scene. 171
7.14 Reconstruction of an urban scene. 172
8.1 Our goal to estimate human segmentation and pose estimation. 175
8.2 PoseField to estimate human segmentation and pose. 178
8.3 Qualitative results on two sets of images from H2View dataset. 185
8.4 Qualitative results on Buffy dataset. 187

List of Tables

3.1 MRF probabilities. 41
4.1 Quantitative comparison on Leuven dataset. 73
4.2 Quantitative comparison on Leuven dataset. 74
4.3 Quantitative results on PascalVOC-10. 76
4.4 Comparison of inference algorithms on PascalVOC-10 using matched energies 78
4.5 Quantitative results on PascalVOC-10 before and after better initialization. 79
4.6 Quantitative results on PascalVOC-10. 85
5.1 Attribute labels for NYU and Pascal dataset. 102
5.2 Object Accuracy: Quantitative results on aNYU 2 dataset. . . . 103
5.3 Attribute Accuracy: Quantitative results on aNYU 2 dataset [163]. 104
5.4 Attribute labels for Pascal dataset. 104
5.5 Quantitative results on aPascal dataset for object class segmentation. 106
5.6 Quantitative results on aPascal dataset. 106
6.1 Approximate system timings. Despite small fluctuations we observed consistently good, interactive frame rates. 134
6.2 Quantitative evaluation of different components. 135
6.3 Quantitative evaluation of features. 136
7.1 Quantitative results on the KITTI dataset. 162
8.1 Quantitative results on H2View dataset for human segmentation. 186
8.2 Quantitative results on H2View dataset for human pose estimation. 186
8.3 Quantitative results on Buffy dataset for human segmentation problem. 187
8.4 Quantitative results on Buffy dataset for human pose estimation problem. 188

Chapter 1

Introduction

What does it mean to see? This question has captured the imagination of many philosophers and scientists in the ancient past as well as researchers of modern science. I would like to use the definition from the most eminent theoretical neuroscientist of the modern era, Prof. David Marr. He defined *vision* as the *"process of dicovering from images what is present in the world, and where it is"* [114]. For example, given a simple image shown in Fig. 1.1(a), we can say many interesting things about the scene. We generally segment the scene into consistent objects (Fig. 1.1(b)) and associate meaningful names (Fig. 1.1(c)). We can make higher level inference about the scene (Fig. 1.1(d, e)), i.e. it is a nice sunny day, people are walking in the park, and a person is feeding ducks in the park! We construct a three dimensional representation of the scene (Fig. 1.1(f)), so as to predict other important higher order inference such as that the person is sitting under the shadow of the tree. But, at this point I would like

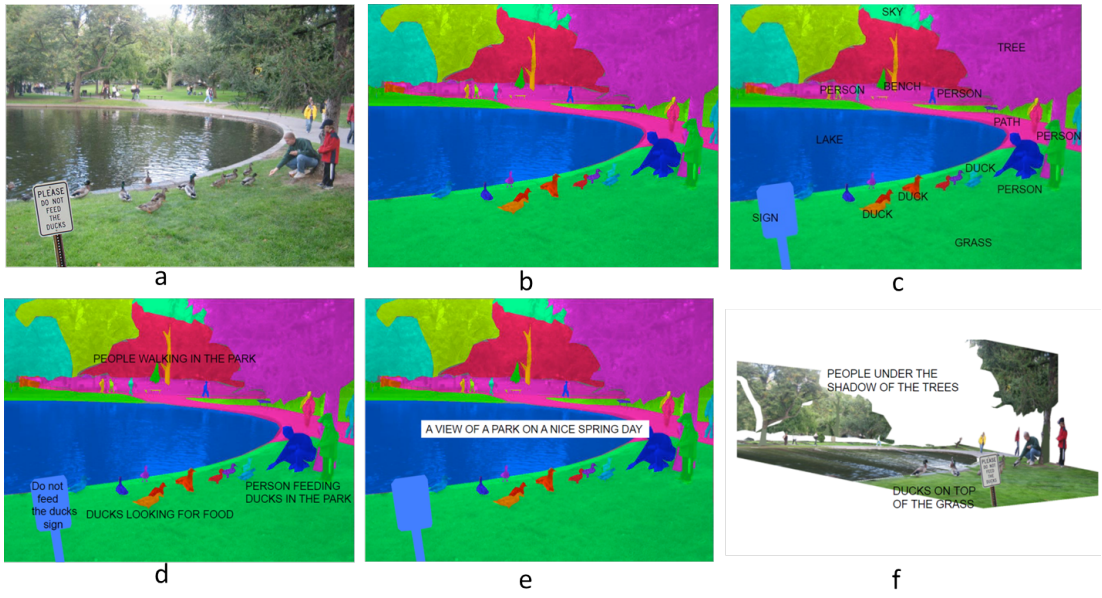


Figure 1.1: *A simple example to highlight important aspects of human vision which tries to capture the richness of the visual world. Human have remarkable ability to understand that this is a park scene in spring. A person is feeding the ducks in the park; possibly he is illiterate. We build a 3D representation of the scene and make the prediction that the person is sitting under shadow. Image courtesy: Antonio Torralba.*

to take another interesting example from a recent newspaper advert (shown in Fig. 1.2). This clever advert for Corona's kitchens by Colombia-based designer Felipe Salazar plays with the geometry of classified ads to give an impression of a three dimensional kitchen, complete with gas hoods, hidden in the text of the newspaper. Such is the remarkable ability of human vision, yet a computer still

sees just pixels. So has the human brain been deceived into seeing something which is not real? On the contrary, humans have evolved over several thousands of years to solve far greater problem — how to survive in a hostile environment — leading to the development of an advanced vision system.

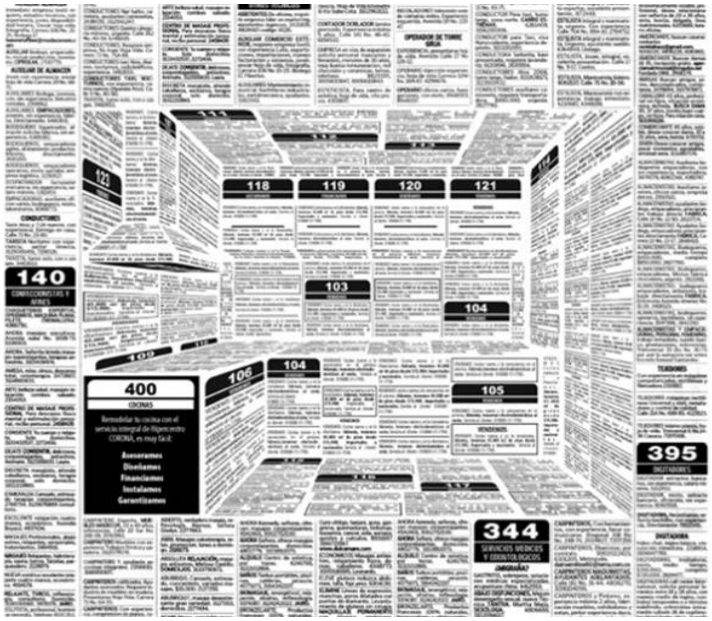


Figure 1.2: *This clever newspaper advert gives an impression of a 3D kitchen hidden in the classifieds. Human vision is able to do amazing work in seeing the 3D structure from 2D images, however a computer still just sees pixels. Such is the remarkable ability of the human brain.*

Thus, researchers from philosophy to neuroscience, from mathematics to computer science, have devoted ample time to understand the underlying principles for developing a vision system which would be able to see as well as we do. Such understanding of (sequences of) images is commonly known as *Scene Understanding*. It consists of solving three classical computer vision problems: *recognition, reorganisation* and *reconstruction*. Recognition refers to assigning meaningful names to each pixel in the image (Fig. 1.1(c)), and reorganisation means segmenting an image into its consistent regions (Fig. 1.1(b)). The task of reconstruction involves figuring out the properties of the physical world that gave rise to the particular image. Classically it means recovering the depth at each pixel (Fig. 1.1(f)). However I take reconstruction as the task of recovering the reflectance properties of the objects and the illumination in the environments, along with generating the depth and structure of the objects in the scene (shown later in the chapter).

In this dissertation, I focus on some of these problems and propose methods for solving them. The work can be divided into three main parts. In the first part, I show how the problem of recognition and reorganisation can be improved

by incorporating some prior information such as context. In the second part I propose methods to capture the interaction between recognition, reorganisation and reconstruction so as to solve scene understanding problems. In the third part I show how human interaction can help in improving the visual recognition task.

I begin by describing the scene understanding problems that have been tackled in this thesis, along with providing motivations and challenges in solving them.

1.1 Scene Understanding Problems

In this thesis I am interested in solving several aspects associated with scene understanding problems. Specifically I talk about four different problems: object and attribute segmentation (Sec. 1.1.1), 3D reconstruction and recognition (Sec. 1.1.2), intrinsic scene decomposition (Sec. 1.1.3) and human pose estimation (Sec. 1.1.4).

1.1.1 Object and Attribute Segmentation

Given an image, the problem of object segmentation involves assigning a meaningful name to each pixel or region in the image. For example, in Fig. 1.3 we would like to associate each pixel with object names such as *wall, floor, picture* or *bed*. Further, we would like to develop vision algorithms that not only associate a meaningful name to a segment/region but to also describe its characteristic properties (or visual attributes). The concepts of objects and attributes are both important for describing images. For instance, it is more natural and descriptive to say a *painted wall* or a *cotton bed* than just saying *wall* or *bed* (shown in Fig. 1.3 (c)). These visual attributes can be categorised as material (wood, plastic), structural (rectangular, cylindrical) or surface (shiny, glossy) properties.

1.1.2 3D Scene Reconstruction and Recognition

The emergence of consumer depth cameras has generated a lot of interest in real-time 3D scanning of physical environments. Despite dramatic progress in real-time 3D reconstruction, many applications could benefit from recognition in the scene. Thus the next important scene understanding problem that we deal with involves jointly reconstructing the 3D environment along with the object label assignment given a sequence of depth images as shown in Fig. 1.4. Our focus is on reconstruction and recognition for both indoor and outdoor environments.

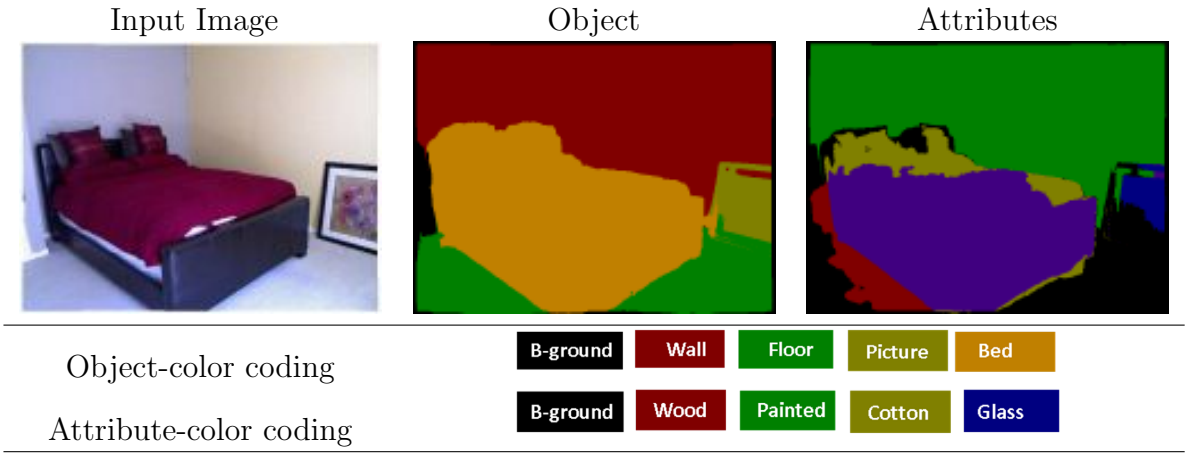


Figure 1.3: *Given an image (a), object and attribute segmentation problems involve assigning an object label (b) and a set of attribute labels (c) to each pixel respectively.*

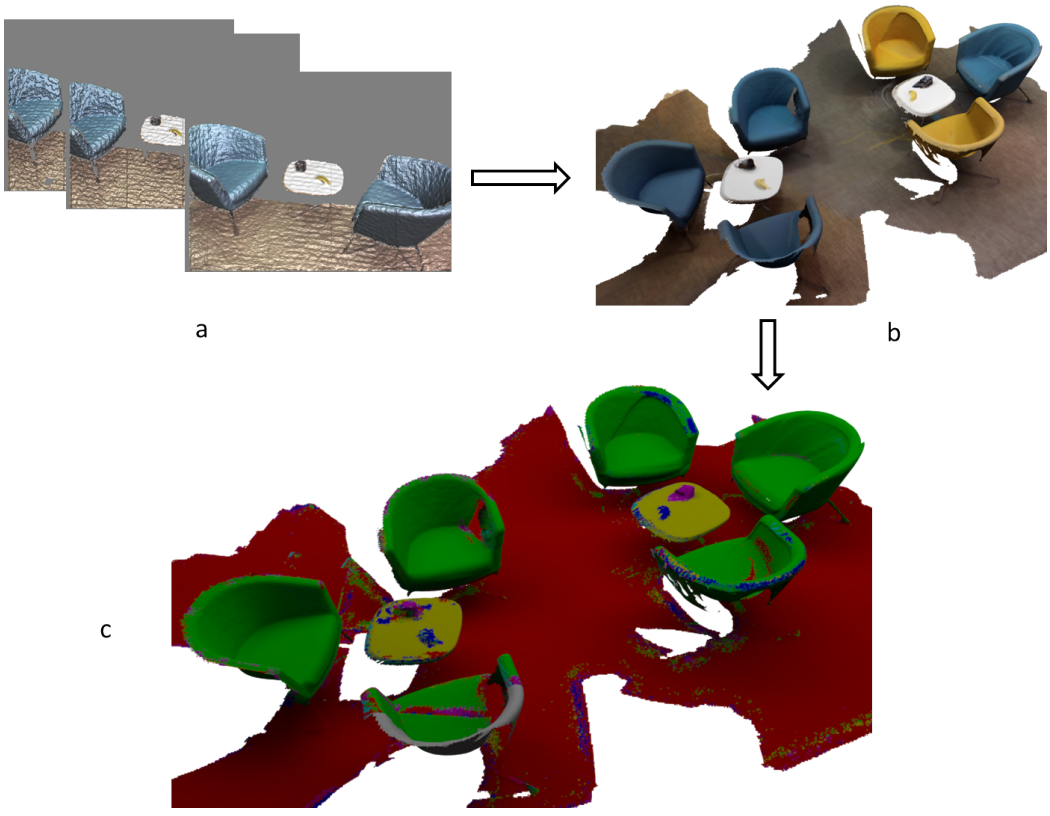


Figure 1.4: *The problem of 3D scene reconstruction and recognition involves building a 3D representation of the environment along with assigning an object label to each voxel given a set of RGB and depth images. This figure shows a 3D RGB model (a), a 3D model (b), and a labelled model (c). Here labels correspond to chair, floor, banana, cup, and table-top.*

Motivation: These two problems are very important in both computer vision and robotics.

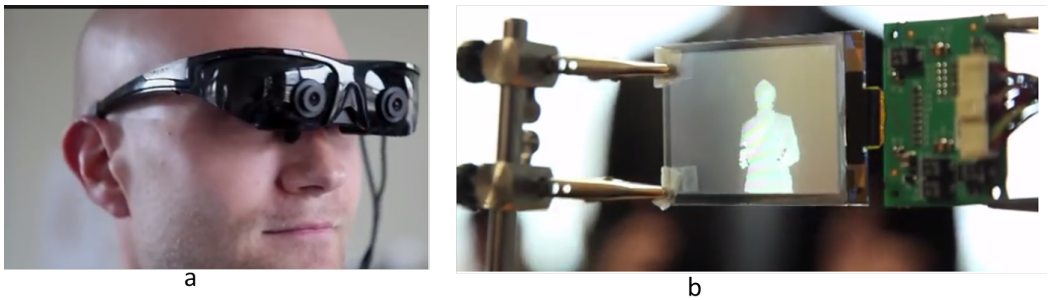
- Robotics: In days to come, robots will form an important part of our lives. For them to work efficiently and reliably, we would like our robots to know what is where. How should it move to find out more? How can a robot read? In order to accomplish these tasks robots need to understand the concepts of objects and their physical properties. Based on this knowledge, they can understand the functional properties of the objects they are interacting with. For example, a robot needs to understand that a *chair* can be used for sitting on but a *coffee machine* can be used for making coffee (shown in Fig. 1.5).



Figure 1.5: *In order for robots to properly interact with the environment, they need to understand objects, their attributes and their functions. For example, a chair can be used for sitting on but not a coffee machine.*

- Assistive technology: Assistive technology is an umbrella term that includes assistive, adaptive, and rehabilitative devices for people with disabilities. There are around 280 million partially sighted people in the world and many devices are developed to benefit them in their day-to-day life. Building 3D environments and recognising objects along with their attribute properties will help in every aspect of their lives, for example while drinking tea, catching a bus, or lying on a bed, etc. A prototype of such an assistive technology is shown in Fig. 1.6.

- Autonomous driving: According to the UN data, many deaths are caused by road accidents, and autonomous driving is one way to enhance road safety. To this end, the vehicle needs to understand obstacles, recognize humans, and perform other common activities. It also needs to understand road signs and identify good driving areas (concrete roads, but not muddy, grassy roads etc.) (Fig. 1.7). To furnish an autonomous agent with the ability to interpret and exploit this information to increase situational awareness, building a model of the 3D environment and understanding the objects and



a b

Figure 1.6: *There are around 280 million people suffering from partial blindness. 3D reconstruction and recognition can play an important role in helping them in their day-to-day life. A prototype of glasses (a), and a human pose recognised by an assistive technology system (b) is shown here.*

their attribute properties are important. Further, such information will help in path planning for going from one place to another place (e.g., *Oxford* to *London*).



a b

Figure 1.7: *In order to reliably accomplish autonomous driving, the vehicle needs to understand the obstacles, pedestrians and other activities. A car is fitted with various input devices such as cameras, lidar (a), and the system needs to find all the important concepts such as signs, pedestrians, other vehicles (b).*

1.1.3 Intrinsic Scene Decomposition

Most previous efforts solve reconstruction in classical terms where we recover the depth at each pixel. However, in general the task of reconstruction involves figuring out the properties of the physical world that gave rise to the particular data. Thus the next important problem involves recovering the reflectance properties of the objects and the illumination in the environments, along with generating the depth and structure of the scene (shown in Fig. 1.8).



Figure 1.8: *Given an image (a), the problem of intrinsic scene decomposition involves recovering the properties of the physical world that gave rise to this image in terms of reflectance properties (b) and illumination dependent components (c).*

Motivation:

- Understanding human vision: Theory suggests that the human brain separates illumination from reflectance and shape [5]. Thus, such intrinsic scene decomposition may take us one step closer to understanding human vision.
- Robotics: In outdoor environments shadows are common and are considered a nuisance for current vision and robotics systems. Intrinsic scene decomposition will recover an illumination independent image which helps in improving object recognition and other scene understanding tasks. As we can see in Fig. 1.9, the road is properly segmented out (Fig. 1.9(d)) when segmentation is done on a shadow invariant image (Fig. 1.9(c)).

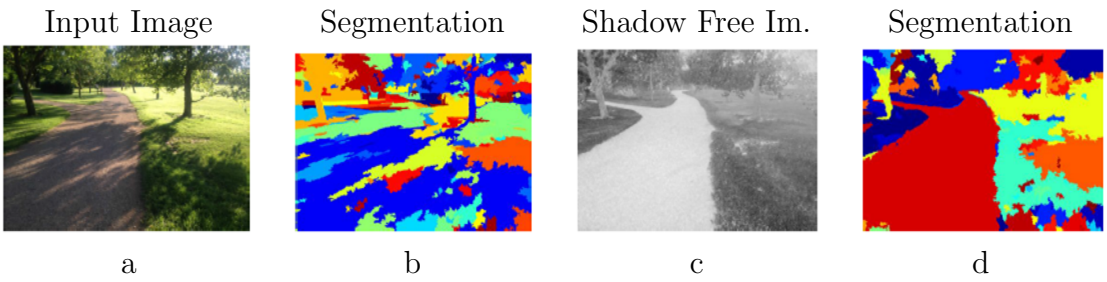


Figure 1.9: *Shadows create very deep effects in images (a). The shadow invariant image eliminates the effect of lighting (c). Images (b) and (d) show the output of colour image segmentation applied to the original and the shadow invariant images respectively. Image courtesy: Paul Newman.*

1.1.4 Human Pose Estimation

Another important problem that we deal with is to estimate the pose of a human body. This may involve intermediate steps of first generating the body joints and parts. For example, a human body may be represented by ten body parts and we consider assigning each pixel to one of these parts (shown in Fig. 1.10).

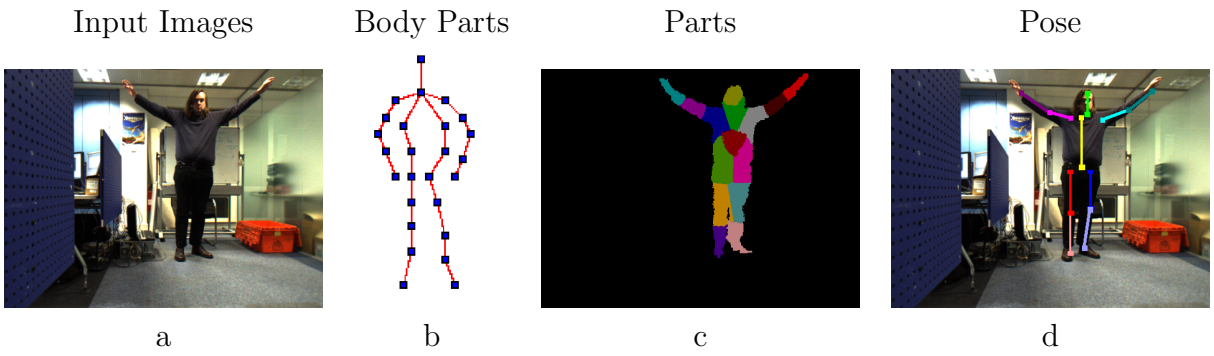


Figure 1.10: *Given an image (a), the pose estimation problem estimates the human pose (d). One possible solution involves the intermediate problem of per-pixel body part estimation (c).*

Motivation: Robust and efficient human pose estimation and tracking has found application in gaming, human-computer interaction, surveillance, telepresence and health-care.

1.2 Why Scene Understanding Is Hard

The tasks described in the previous section seem very simple to humans and we do them flawlessly day after day. However, computer vision systems and algorithms do not achieve such a level of accuracy. For example, for object recognition on the Pascal dataset (a commonly used benchmark dataset for object labelling), the best performing computer vision algorithms achieve below 50% accuracy on many classes such as *sheep, cow, sofa, dog* and *bottle*, and on average they achieve only 20% (shown in Fig. 1.11). This raises a natural question of why is the problem of scene understanding from images so hard? The core of this thesis addresses these issues and presents approaches taken to solve them. In much that follows, we show how the interplay between recognition, reorganisation and reconstruction helps to improve the overall goal of scene-understanding.

The foremost problem that a vision system has to deal with, and in developing an algorithm that can predict the world and conditions that led to the generation

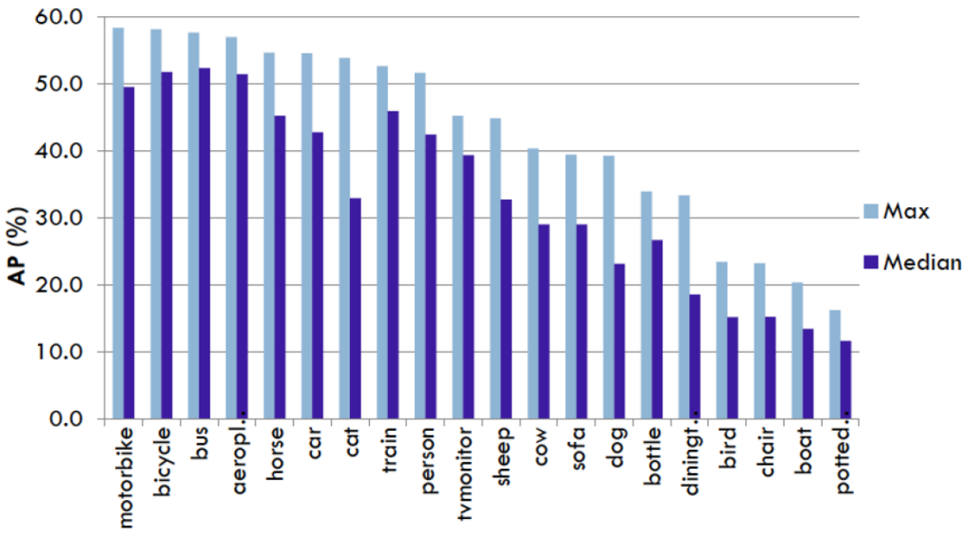


Figure 1.11: *This figure shows the accuracy achieved by a current state-of-the-art algorithm on a benchmark dataset for recognition tasks. On many classes the algorithm achieves only 10%-20%.*

of a scene, is *ambiguity*. A scene could be generated by infinitely many different combinations and arrangements of properties, shapes of objects and illuminations in the environment.

Ambiguity: To demonstrate I highlight the example image taken from Adelson and Pentland's workshop metaphor [5]. In this example, an image is shown to different people and they are asked to build a scene that will look the same. Essentially this is what the human visual system is trying to do: given an image Fig. 1.12(a), try to figure out how it could have come about (Fig. 1.12(b)) [133]. The first solution proposed by the painter (Fig. 1.12(c)) is fully accounted for by the change in the reflectance values; the surface is assumed to be uniformly illuminated. The second solution (Fig. 1.12(d)) is proposed by the metal-sheet worker. The scene is illuminated by a single distant light source; all the image information is accounted for by the change in the shading caused by the variation in surface normals when viewed from a certain position. The final solution (Fig. 1.12(e)) is proposed by the lighting designer. The scene is assumed flat with constant reflectance; the only variation is in the local illumination. Thus in order to recover the physical world out there, we have to deal with a profound amount of uncertainty.

Other issues: Beyond ambiguity, vision systems have to deal with several other issues both in natural and man-made environments. Some of the pertinent issues relate to the variations in the objects' shape and appearances, variations is the

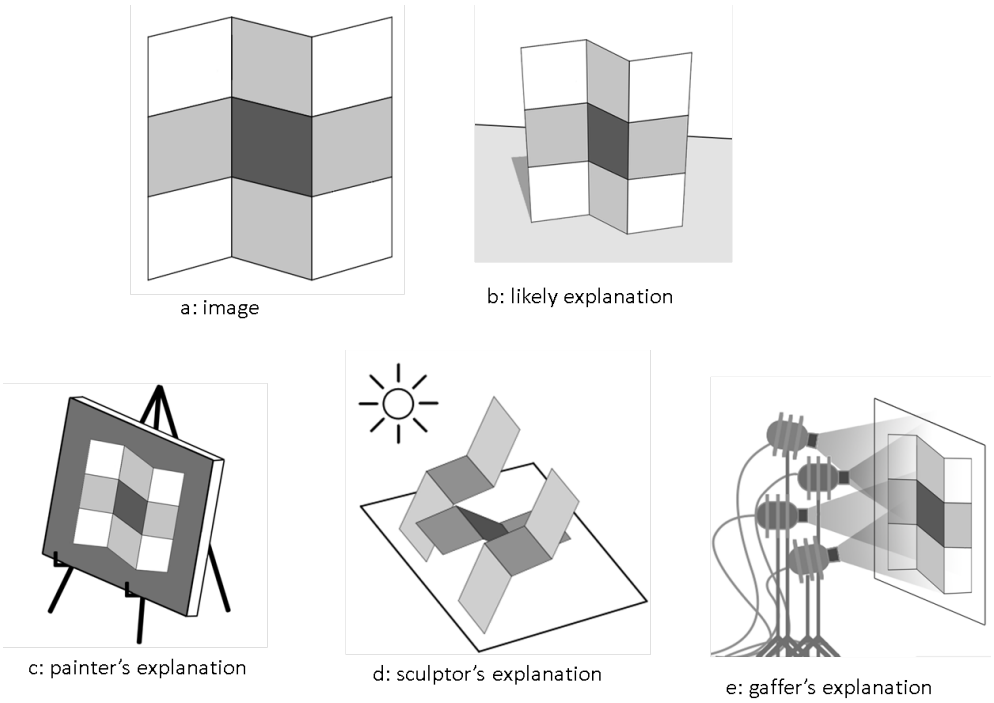


Figure 1.12: *To demonstrate the ambiguity that we have to deal with, I highlight the example image taken from Adelson and Pentland's workshop metaphor. The image (1.12(a)) is the outcome of the reality shown in (1.12(b)) but there are many other possible interpretations as well.*

objects' transparency and translucency, and changes in view point, illumination and scale.

Machine Learning. A standard recognition pipeline consists of learning the model of an object using a collection of datasets [117, 156]. The dataset forms an integral part of current state-of-the-art computer vision systems [51]. Availability of large amount of data in terms of images and videos has contributed remarkably to achieving high accuracy on many vision problems [51, 52, 128]. For example, to a large extent availability of large datasets helped to solve face recognition tasks [141]. Though datasets are generally collected with a focus on properly sampling the variety and richness of natural and man-made environments, there is a downside to the way they are generated. For instance, we are always faced with the issue of dataset bias, where a learning algorithm performs well on unseen data only if they are sampled around the training data [72, 101, 173]. This poses a big challenge in solving real-life problems. For instance, we are not able to train an object model (for example a *chair*) using images taken from Flickr, and then use these models in our environment (to recognise chairs for example in our bedroom). Such dependence on the dataset poses a big challenge when taking a

system into real-life environments.

1.3 Thesis Approach

The fact that vision systems have to deal with large amounts of uncertainty and many other related issues, raise another question as to how then we go about performing vision-related tasks. I now describe the approaches and technical parts of the thesis that attempt to address common problems. In order to find the best possible solution to the understanding of images or scenes, we propose four approaches. Specifically the algorithms proposed in the thesis rely on the ideas of the structured world (Sec. 1.3.1), interplay between reconstruction and recognition (Sec. 1.3.2), and human interaction (Sec. 1.3.3). Finally in Sec. 1.3.4, I show how I use a probabilistic framework to solve the labelling problems discussed earlier.

1.3.1 World is Structured

At the core of our approaches lies the fundamental principle that the physical world is very structured. We observe structure at the object level, the parts level and the scene level. One source of structure comes from incorporating various forms of higher order information. For example, it is observed that pixels belonging to a region often have the same label; for instance they may belong to same object (Fig. 1.13) or may have the same orientation [80]. Another commonly used structure is that of context (or co-occurrence) information which encapsulates the rich information about how the objects are related to each other in a natural scene, i.e. it is highly likely that a table, a monitor, and a keyboard co-exist (Fig. 1.13) but the co-existence of a table and an elephant is not likely [96, 174]. Similarly, we can also incorporate information that the reflectance of an object generally changes at the boundary but the illumination varies smoothly across it [100].

1.3.2 Reconstruction, Recognition and Reorganisation

We sense depth from even a single image of a complex natural environment. Such vivid impression is the result of structure present in the world: mutually consistent information pertinent to the spatial relationships of objects in the scene, including occlusion, perspective, gradients, shading, shadows and other

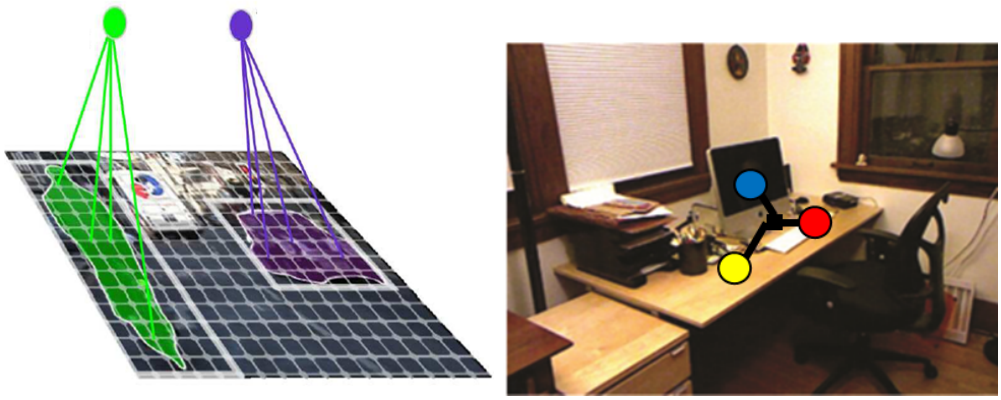


Figure 1.13: *Here I show the concepts of label consistency (left image) and context (right image). Label consistency enforces the prior that a contiguous set of pixels receive the same label (shown by green and blue dots). Context captures the co-occurrence of objects in a scene (here monitor, table and keyboard co-exist).*

effects [61]. This suggests that the problem of scene understanding requires a holistic view of reconstruction, recognition and reorganisation. Thus, we capture the interaction between these three tasks. For example, recognition of objects could benefit from a preliminary reconstruction of 3D structure, instead of just treating it as a 2D pattern classification problem. Similarly, object labels can provide geometric cues about which surface orientations are more likely to appear at a certain location in space, which will improve reconstruction (such as those shown in [56, 57]). Recently, Prof. Jitendra Malik has succinctly put forth the concept of unification of these three tasks as the way forward for further vision research [113] (shown in Fig. 1.14). In the past, several works have been proposed to integrate parts of these three concepts. For example, Barron and Malik [13] use the information provided by the reorganisation step to improve the reconstruction step. Ladicky et al. [94] use the the knowledge from the reorganisation to improve the recognition task. Though these works leverage such unifications of these tasks, most of them have used information from one step to improve the quality of the other steps, or they are restricted to small scale (indoor or outdoor) environments. In this thesis, I show how such unifications of these three concepts can be used to improve the large scale indoor and outdoor dense scene reconstruction, reorganisation and recognition tasks simultaneously.

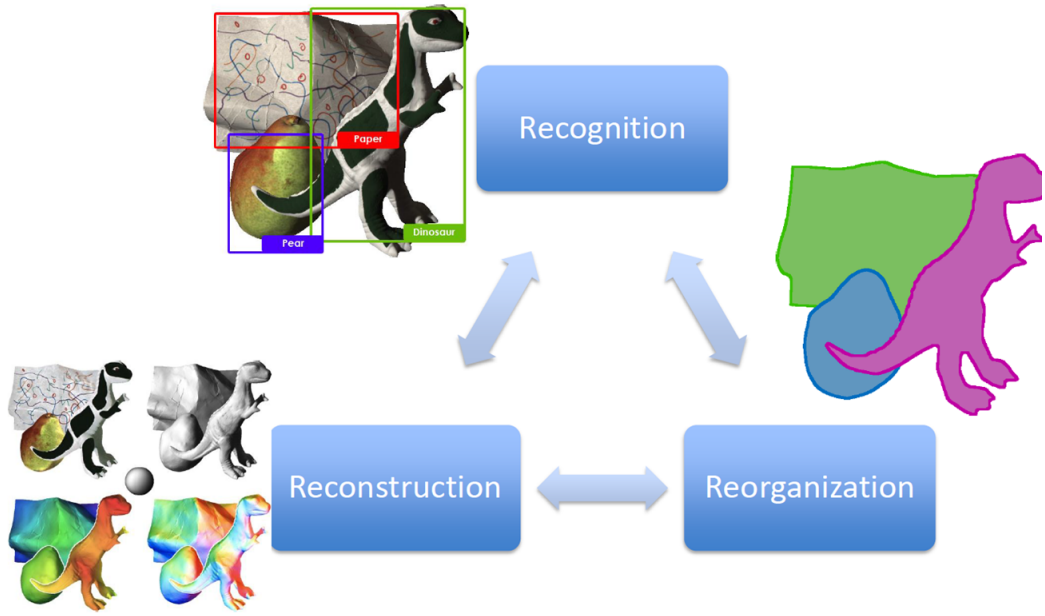


Figure 1.14: *I highlight the importance of capturing the interaction between three classical computer vision tasks: recognition, reorganisation and reconstruction. Image courtesy: Jitendra Malik.*

1.3.3 Human Interaction

Our third strategy revolves around incorporating humans into our framework. In computer vision, a human can act like a *teacher* to assess the results and guide the system to improve the quality of the results [140, 193]. Such human-computer interaction has several advantages. First, we have seen the role of human teachers for visual recognition and search tasks, where human users provide rich supervision to visual learning systems and interactively give feedback to the system so it can learn better models for recognition (such as proposed in [125]). Humans collecting task specific dataset have been at the core of visual recognition research. Second, humans can help in developing better machine learning algorithms as they can select which examples and concepts to present and in what order to present them to the learning system [89, 126, 162]. In this way one can easily guide the training and remarkably increase the speed at which learning can occur. I also use such human interaction to improve the visual recognition tasks (shown in Fig. 1.15).

1.3.4 Probabilistic Framework

I now describe the last concept, where we formulate solving scene understanding problems in a probabilistic framework as one way to deal with the profound



Figure 1.15: *This highlights how users can quickly and interactively label the world around them. The environment is scanned in using a hand-held depth camera, and in real-time a volumetric fusion algorithm reconstructs the scene in 3D (left). At any point the user can reach out and touch objects in the physical world, and provide object class labels through voice commands (middle). This way users can help in improving the quality of output (right image).*

amount of uncertainty.

Random Field Model: The problems that have been discussed earlier are put forth as labelling problems where the task involves associating meaningful names to pixels, voxels, or regions in the data. We represent the problems by a set of random variables $X_i \in \mathcal{X} = \{X_0, X_1, ..., X_{n-1}\}$ and assign a label $l_k \in \mathcal{L} = \{l_0, l_1, ..., l_{h-1}\}$ to each variable X_i. For example, in the object segmentation case each random variable $X_i \in \mathcal{X}$ corresponds to a pixel (in the two dimensional case) or a voxel (in the three dimensional case), and takes an object label l_j such as *car, road, building* or *sky*. Given the random field framework, we

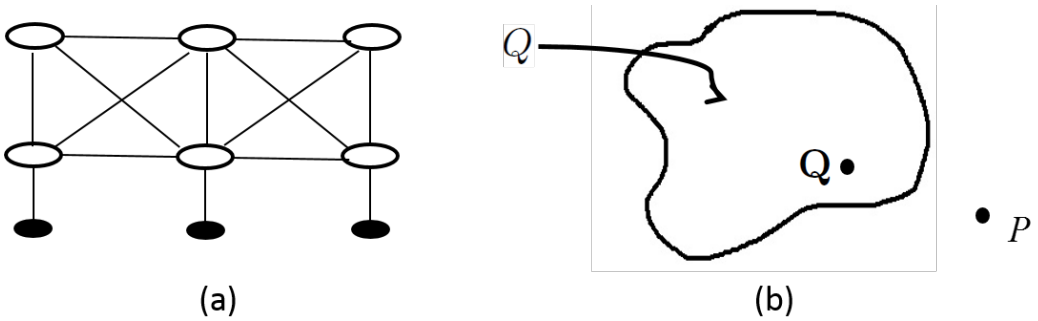


Figure 1.16: *A simple MRF with 3 variables (filled circles) each taking two labels (circles) are shown. Edges shows relations between a pair of variables. The figure in the right highlights the basic idea of a mean-field approach where the true probability distribution Pr is approximated by a simpler family of distributions Q.*

define a probability distribution Pr that captures the correlation between different scene/image elements (or random variables). Probabilistic models such as the

Markov Random Field (MRF) and the Conditional Random Field (CRF) [81, 99] have long formed the basis for solving challenging problems not only in computer vision but in many other areas [79, 189]. One standard formulation of CRF or MRF considers either per-variable terms or terms dependent on pairs of variables. An example is shown in Fig. 1.16(a). I discuss these models in detail in the next chapter. The goal is then to estimate properties of these probability distributions such as the most probable (MAP) solutions and marginal distributions to enable learning and finding best possible solutions using these models.

Mean-field Inference: In general, these two problems of estimating MAP solutions and marginal distributions are considered NP-hard. However many approximate approaches have been developed over the years, including graph-cuts based α-expansion, belief propagation, tree-reweighted message passing, LP-relaxation and decomposition based approaches [20, 70, 82, 84, 90, 195]. In this work we follow a mean-field approach, which belongs to the class of methods known in the Bayesian literature as variational methods [183]. The key idea is to approximate a complex Markov Random Field Pr by a simpler one Q where approximation between these two distributions is measured by KL-divergence. We evaluate the marginals and the MAP solution in the simpler model Q. Such a mean-field method leads to an iterative message passing approach where each variable updates its marginal value by taking the expected mean of the values of its neighbours. However, such updates require $O(n^2)$ complexity for a fully-connected pairwise MRF, making it impractical for many multi-label computer vision problems. However, it has been recently shown that the mean-field updates can be formulated as simple application of Gaussian filters on Q distributions. I use such a filter-based mean-field method for performing fast approximate maximum posterior marginal (MPM) inference in multi-label CRF models with fully connected pairwise terms [87]. This strategy reduces the complexity of each mean-field update to $O(n)$.

1.4 Thesis Contributions

There are five key technical contributions of this work.

Object Labelling: I propose a novel mean-field algorithm to incorporate higher order information into CRFs with only unary and/or pairwise terms which provides an order of magnitude of speed-up compared to the state-of-the-art inference algorithm while also maintaining or improving accuracy.

Reconstruction and Recognition: In this part of the work, I propose robust approaches for real-time dense 3D reconstruction of both indoor and outdoor environments along with associating them with object labels. Our approach scales to large environment. To the best of my knowledge, this is the first work that provides an end-to-end system that performs real-time reconstruction and recognition of large scale indoor and outdoor environments.

Joint Object-Attribute-Intrinsic Scene Decomposition: I develop an algorithm that jointly estimates the intrinsic properties such as reflectance, shading and depth, along with the estimation of the per-pixel object and attribute labels. Though these two problems of intrinsic decomposition and object recognition have been studies extensively in the past, this is the first work, that jointly explores the synergy effects existing between these two tasks.

Human Segmentation and Pose Estimation: I design an efficient filter-based mean-field algorithm that jointly estimates human segmentation, pose and depth given a pair of stereo images achieving a factor of speed-up as well as achieving better accuracy. Previously such joint estimation problems have been solved using slow dual-decomposition approaches, so the work proposed in this thesis provides an really efficient and effective alternate to them.

Reconstruction, Recognition and Interaction: I propose an interactive 3D labelling and segmentation system that aims to make acquiring segmented 3D models fast, simple, and user-friendly. Carrying a body-worn depth camera, the environment is reconstructed using standard techniques. The user is able to reach out and touch surfaces in the world, and provide object category labels through voice commands. This is one of the first works in computer vision where users have physically reached to the world, and so helped the algorithms to better learn the object models. Such physical interaction between human, system and environments will provide a great avenue for collecting a lot of training data which is at the heart of the machine learning algorithms.

1.5 Outline of the Thesis

Chapters 2, 3: In Chapter 2 we review the basic concepts of probabilistic frameworks for solving labelling problems occurring in computer vision. Specifically, we discuss the Markov Random Field (MRF) and Conditional Random Field (CRF) models. We delve into mean-field methods for estimating the properties of these probability distributions, such as marginal distribution and max-

imum a posteriori (MAP) solutions, in Chapter 3. We do an extensive study of the properties of the mean-field method highlighting the mean-field derivation, polytope and convergence properties.

Chapter 4: In Chapter 4 we show how higher order information, such as context information and label consistency over large regions can be efficiently incorporated in the MRF model with only unary and/or pairwise terms. Inference in the MRF is performed using a filter-based mean-field approach. We demonstrate our techniques on joint stereo and object labelling problems, as well as on object class segmentation, showing in addition for joint object-stereo labelling how our method provides an efficient approach to inference in product label spaces.

Chapter 5: In Chapter 5 we propose an algorithm to solve the problems of recovering intrinsic scene properties such as shape, reflectance and illumination from a single image, along with estimating the object and attributes segmentation separately. We formulate this joint estimating problem in an energy minimization framework which is able to capture the correlations between intrinsic properties (reflectance, shape, illumination), objects (table, tv-monitor), and materials (wooden, plastic) in a given scene. For example, our model is able to enforce the condition that if a set of pixels take the same object label, e.g. table, then most likely those pixels will receive similar reflectance values. We demonstrate the qualitative and quantitative improvement in the overall accuracy on the NYU and Pascal datasets.

Chapter 6: In Chapter 6 we develop a system that encapsulates the benefits of reconstruction, recognition and reorganisation. Further, we highlight how human interaction helps in improving recognition tasks. In this chapter we describe an interactive 3D labelling and segmentation system that aims to make acquiring segmented 3D models fast, simple, and user-friendly. Carrying a body-worn depth camera, the environment is reconstructed using standard techniques. The user is able to reach out and touch surfaces in the world, and provide object category labels through voice commands. We propose a GPU-based volumetric mean-field inference algorithm that can efficiently propagate these user labels through the volume and provide a smooth segmentation that follows object boundaries. We also describe a new streaming decision forest approach that employs implicit surface-based volumetric features to learn from the propagated user labels. When the user encounters a previously unobserved and unlabelled region of space, the forest predicts object labels for each voxel, and the same mean-field inference smooths the final output. We demonstrate compelling results on several sequences, highlighting the smooth propagation of user labellings and the ability

to learn and generalize to unseen regions of the world.

Chapter 7: In Chapter 7 we propose a robust approach for dense 3D reconstruction of outdoor environments along with associating them with object labels given stereo pairs of images. At the core of our algorithm is a hash based fusion approach for reconstruction and a mean-field approach for object labelling. In the process we capture the synergy between the reconstruction and recognition tasks. Our system can handle and process a large-scale environment in real time. We demonstrate the effectiveness of our approach on the KITTI dataset [48] and show high quality dense reconstruction and labelling of the scenes.

Chapter 8: In Chapter 8 we show as an application how solving product label space problems using the filter-based mean-field framework can be applied to the human pose estimation problem. We do this by designing an energy function that estimates the human pose along with estimating the human foreground/background segmentation given a pair of rectified stereo images. In all these cases, we provide comparison with the state-of-the-art methods and achieve significant speed-ups.

1.6 Publications

Part of the work described here has resulted in publication of the following papers.

- Chapter 4
  - V. Vineet, J. Warrell and P. H. S. Torr. *Filter-based Mean-Field Inference for Random Fields with Higher Order Terms and Product Label-Spaces.* In European Conference on Computer Vision (ECCV) 2012.
  - V. Vineet, J. Warrell and P. H. S. Torr. *Filter-based Mean-Field Inference for Random Fields with Higher Order Terms and Product Label-Spaces.* In International Journal of Computer Vision (IJCV) 2014.
  - V. Vineet, J. Warrell, P. Sturgess and P. H.S. Torr. *Improved Initialization and Gaussian Mixture Pairwise Terms for Dense Random Fields with Mean-field Inference.* In British Machine Vision Conference (BMVC) 2012.
- Chapter 5

  - V. Vineet, C. Rother and P. H. S. Torr. *Higher Order Priors for Joint Intrinsic Image, Objects, and Attributes Estimation.* In Neural Information Processing System (NIPS) 2013.

  - S. Zheng, M. Cheng, J. Warrell, P. Sturgess, V. Vineet, C. Rother and P. H. S. Torr. *Dense Semantic Image Segmentation with Objects and Attributes.* In IEEE International Conference on Computer Vision and Pattern Recognition (IEEE CVPR), 2014.

  - M. Cheng, S. Zheng, W. Y. Lin, V. Vineet, P. Sturgess, N. Crook, N. Mitra and P. H. S. Torr. *ImageSpirit: Verbal Guided Image Parsing.* In ACM Transactions on Graphics (ACM TOG), 2014.

- Chapter 6

  - J. Valentin, V. Vineet, M. Cheng, D. Kim, S. Izadi, J. Shotton, P. Kohli, M. Neissner and P. H. S. Torr. *Personalized 3D Recognition at your Fingertips.* In ACM Transactions on Graphics (ACM TOG), 2014 (first three authors contributed equally).

- Chapter 7

  - V. Vineet, O. Miksik, M. Lidegaard, S. Izadi and P. H. S. Torr. *Dense Semantic Stereo Fusion for Large Scale Semantic Scene Reconstruction.* In IEEE International Conference on Robotics and Automation (ICRA), 2015 (Under Submission).

  - O. Miksik, V. Vineet, M. Lidegaard, S. Golodetz, V. A. Prisacariu, S. Hicks, P. Perez, S. Izadi and P. H. S. Torr. *The Semantic Paintbrush: Interactive 3D Mapping and Recognition in Large Outdoor Spaces.* In ACM Conference on Human-Computer Interaction (SIG-CHI), 2015 (Under Submission, first two authors contributed equally).

- Chapter 8

  - V. Vineet, G. Sheasby, J. Warrell and P. H.S. Torr. *PoseField: An Efficient Mean-field based Method for Joint Estimation of Human Pose, Segmentation and Depth.* In Energy Minimization Methods in Computer Vision and Machine Learning (EMMCVPR) 2013.

- Other publications

- V. Vineet, J. Warrell and P. H. S. Torr. *Tiered Move Making Algorithm for General Pairwise MRFs.* In IEEE International Conference on Computer Vision and Pattern Recognition (IEEE CVPR), 2012.

- V. Vineet, J. Warrell and P. H. S. Torr. *Tiered Move Making Algorithm for General Pairwise MRFs.* In CoRR, 2014.

- V. Vineet, J. Warrell and P. H. S. Torr. *Learning and Inference for General Non-submodular Pairwise Energies.* In Rank Prize Symposium, 2012.
- M. Cheng, J. Warrell, W. Y. Lin, S. Cheng, V. Vineet and N. Crook . *Efficient Salient Region Detection with Soft Image Abstraction .* In IEEE International Conference on Computer Vision (IEEE ICCV), 2013.

- V. Vineet, J. Warrell, L. Ladicky and P. H. S. Torr. *Human Instance Segmentation from Video using Detector-based Conditional Random Fields.* In British Machine Vision Conference (BMVC), 2011.

- O. Miksik, V. Vineet, P. Perez, and P. H. S. Torr. *Distributed ADMM-based Inference in Large-scale Random Fields.* In British Machine Vision Conference (BMVC), 2014.

- S. Hare, A. Saffari, S. Golodetz, V. Vineet, M.M. Cheng, and P. H. S. Torr. *Struck: Structured Output Tracking with Kernels.* In IEEE Transaction on Pattern Analysis and Machine Intelligence (PAMI), 2014 (Under Submission).

Chapter 2

Labelling Problems and Probabilistic Models

2.1 Labelling Problems

Many problems in computer vision can be considered as labelling problems. The aim of these problems is to assign a label to each pixel or scene element based on some observation about these elements. Examples include image segmentation, disparity estimation, image denoising [20, 43, 170] (shown in Fig. 2.1). In order solve them, a common strategy is to represent the whole problem by a set of random variables $X_i \in \mathcal{X} = \{X_0, X_1, ..., X_{n-1}\}$ associated with a lattice $\mathcal{V} = \{1, 2, ..., n\}$ where each random variable $X_i \in \mathcal{X}$ takes a label from the label set $\mathcal{L} = \{l_1, l_2, ..., l_k\}$ based on the observation $\mathbf{D}$ about the lattice points. Typically each lattice point corresponds to a pixel or a region in the image and the label set is defined according to the problem. For example, in an image segmentation problem, the label set corresponds to the foreground or background object label. Any possible assignment of labels to the variables is called a *labelling* or *configuration*, which is denoted by $\mathbf{x}$ and takes a value from all possible configurations $\mathcal{L}^n$. The goal is then to find the best possible labelling $\mathbf{x}^* \in \mathcal{L}^n$.

In order to estimate $\mathbf{x}^*$, we first define a probability distributions over the unobserved variables $\mathbf{x}$ and the observed variables $\mathbf{D}$. We can either define a joint distribution $\Pr(\mathbf{x}, \mathbf{D})$ or a conditional distribution $\Pr(\mathbf{x}|\mathbf{D})$ over these variables. Once we have defined our distribution we can use it to estimate properties such as the maximum a posterior (MAP) solution

$$\mathbf{x}^* = \arg\max_{\mathbf{x}} \Pr(\mathbf{x}|\mathbf{D}), \tag{2.1.1}$$

where $\mathbf{x}^*$ is the best possible labelling for the random field. We may also like to estimate the marginal distributions

$$\Pr(x_i) = \sum_{\mathbf{x}/x_i} \Pr(\mathbf{x}|\mathbf{D}), \quad \forall i \in \mathcal{V} \quad \text{(marginals)} \tag{2.1.2}$$

where $\mathbf{x}/x_i$ means all possible configurations of all variables except i^{th} variable. The marginal distributions help in generating the max-marginal. These two estimation problems are generally NP-hard. For example, for a simple binary problem with N variables, there are 2^N possible configurations. Thus looking for an exact MAP solution or marginal distributions scales exponentially with the number of variables, making it generally infeasible to get their optimal values. Over the years many approximate algorithms have been developed that try to find solutions which are very close to optimal [20, 183, 196]. They rely on modelling

the problem in Markov or conditional random field frameworks. Next we discuss some instances of these two models in details.

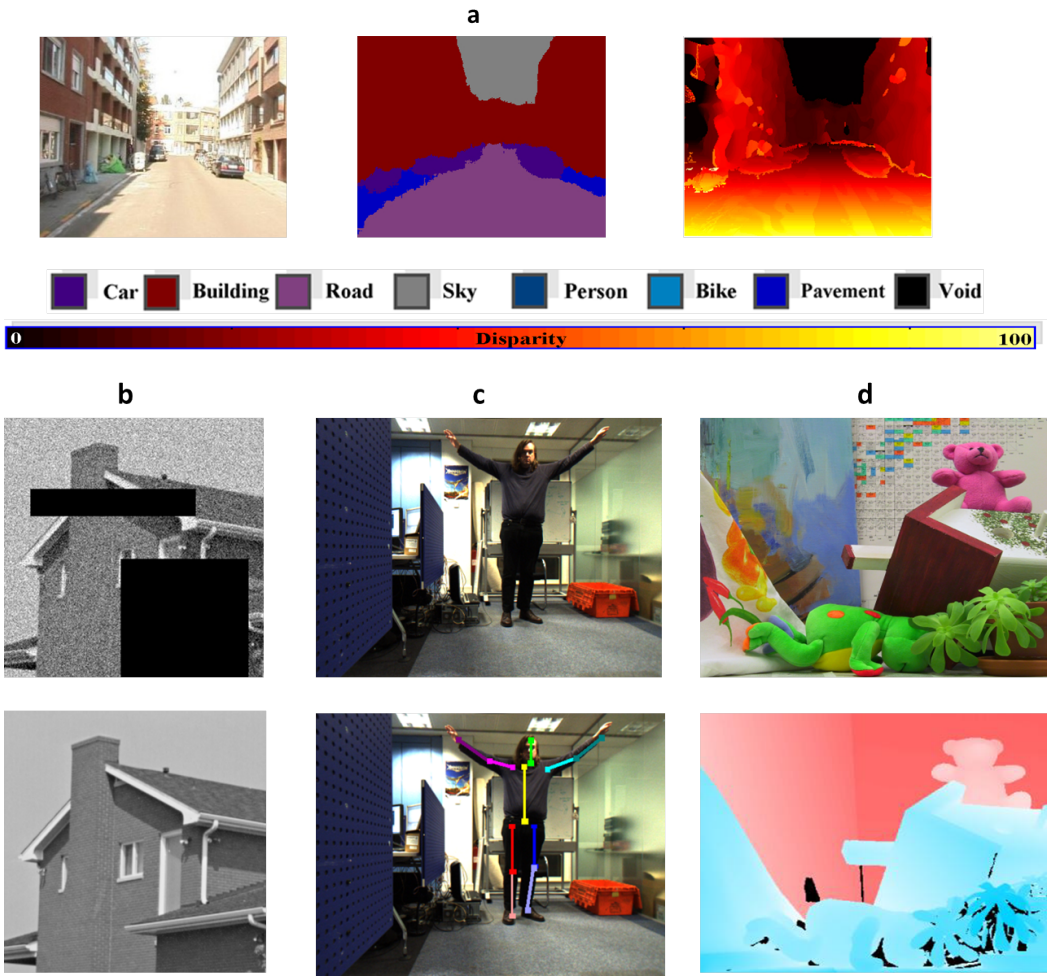


Figure 2.1: *We show some labelling problems that we encounter in computer vision. a) Object segmentation and disparity estimation: the goal is to estimate per-pixel the object labels (car, building, road) and disparity labels given a pair of rectified images. b) Image denoising: in this problem, we are given a noisy image, and the task is to output a denoised image. Each pixel can receive any label from 256 possible intensity values. c) Human pose estimation: the goal is to estimate the pose of a human body. We can represent whole body by ten body parts and we consider assigning each pixel to one of these parts. d) Optical flow: This is a correspondence problem which measures the apparent motion of brightness pattern between a pair of images.*

2.2 Random Field Models

Random field models such as the Markov Random Field (MRF) and the Conditional Random Field (CRF) have long formed the basis for solving challenging problems not only in computer vision but in many other areas [33, 78, 112]. In order to set up the problem we also define a neighbourhood system $\mathcal{N} = \{N_1, N_2, ..., N_n\}$ of the random field, where N_i denotes the set of all neighbours of the variable X_i. We first briefly describe the MRF before going into details of the CRF.

2.2.1 Markov Random Field

Given the data $\mathbf{D}$, a MRF models the joint probability distribution $\Pr(\mathbf{x}, \mathbf{D})$ of the random field configuration $\mathbf{x}$ and the data $\mathbf{D}$. The joint probability $\Pr(\mathbf{x}, \mathbf{D})$ can be written as

$$\Pr(\mathbf{x}, \mathbf{D}) = \Pr(\mathbf{D}|\mathbf{x})\Pr(\mathbf{x}), \tag{2.2.1}$$

where $\Pr(\mathbf{D}|\mathbf{x})$ and $\Pr(\mathbf{x})$ are the data likelihood and prior probabilities on the label space respectively. The probability distribution $\Pr(\mathbf{x}, \mathbf{D})$ is a MRF if and only if it satisfies following properties:

$$\begin{aligned} \Pr(\mathbf{x}, \mathbf{D}) > 0, \qquad & \forall \mathbf{x} \in \mathcal{L}^n \text{ (Positivity)}, \\ \Pr(x_v|\{x_u : u \in \mathcal{V} - \{v\}) = \Pr(x_v|\{x_u : u \in N_v\}), \qquad & \forall v \in \mathcal{V} \text{ (Markovian)}. \end{aligned} \tag{2.2.2}$$

The Markovian property says that a variable X_i is conditionally independent of all the variable given its neighbours X_{N_i}. The Markovian property leads us to define a clique set $c \in \mathcal{C}$ where each clique is a set of random variables $\mathbf{X}_c$ which are conditionally dependent on each other. We also associate a potential function $\psi_c(\mathbf{x}_c, \mathbf{D}_c)$ $\forall c \in \mathcal{C}$ where $\mathbf{D}_c$ corresponds to the observed variables in the clique c.

Given the clique structure and their potential functions, we can write the joint probability distribution $\Pr(.)$ as a product of potential functions following the Hammersley and Clifford theorem [16] as follows:

$$\Pr(\mathbf{x}, \mathbf{D}) = \frac{1}{Z}\prod_{c \in \mathcal{C}} \exp(-\psi_c(\mathbf{x}_c, \mathbf{D}_c)) = \frac{1}{Z}\exp(-\sum_{c \in \mathcal{C}} \psi_c(\mathbf{x}_c, \mathbf{D}_c)), \tag{2.2.3}$$

where Z is the called *partition function* and is a normalizing constant which

ensures that the probabilities sum to one, i.e.

$$Z = \sum_{\mathbf{x}} \sum_{\mathbf{D}} \prod_{c \in \mathcal{C}} \exp(-\psi_c(\mathbf{x}_c, \mathbf{D}_c)) = \sum_{\mathbf{x}} \sum_{\mathbf{D}} \exp(-\sum_{c \in \mathcal{C}} \psi_c(\mathbf{x}_c, \mathbf{D}_c)). \quad (2.2.4)$$

The MRF can be represented on a graph where each random variable corresponds to a node and relationships between variables are represented by edges [18, 81]. For example, in Fig. 2.2(a) we show a simple MRF which consists of two different kinds of variables: D_a represents the observed variable shown by filled circles and x_a represents the hidden (or unobserved) variables. These two types of variables induce different types of clique structures. For example, each link between x_a and D_a induces a clique of size two, and the clique size for each x_a depends on N_a, i.e. x_1 belongs to clique of size two as shown in Fig. 2.2(a).

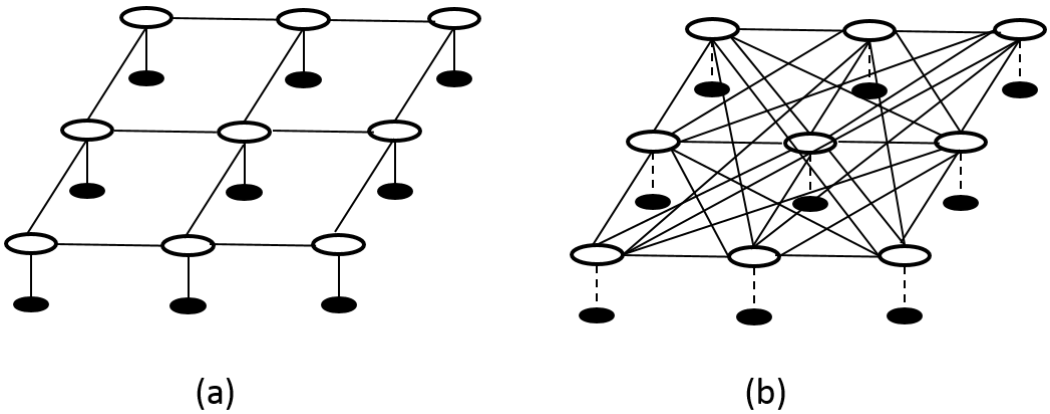


Figure 2.2: *Here we show a grid MRF and a dense MRF. In a grid MRF each variable is connected to its immediate neighbours. This model is ubiqutous in computer vision problems, where each variable X_i is a pixel in the image that is related to its immediate neighbours (shown in (a)). In a dense pairwise MRF each variable is densely connected to every other variable. Such a MRF is useful for capturing long range interaction between variables (shown in (b)).*

2.2.2 Conditional Random Field

Modelling the joint probability $\Pr(\mathbf{x}, \mathbf{D})$ over both the observed and hidden variables requires enumerating over all possible observed variables. As an alternative to this model, the Conditional Random Field directly finds the conditional probability $\Pr(\mathbf{x}|\mathbf{D})$ of the hidden variables given the observed data, thus alleviating the need to model the observed data [99].

The conditional distribution $\Pr(\mathbf{x}|\mathbf{D})$ also satisfies the Markovian property:

$$\Pr(x_v|\{x_v : u \in \mathcal{V} - \{v\}) = \Pr(x_v|\{x_u : u \in N_v\}), \qquad \forall v \in \mathcal{V}. \tag{2.2.5}$$

Similar to the MRF, given the clique structure (defined earlier), the conditional distribution is a *Gibbs* distribution and can also be written as a product of potential functions defined over the cliques following the Hammersley and Clifford theorem:

$$\Pr(\mathbf{x}|\mathbf{D}) = \frac{1}{Z(\mathbf{D})} \prod_{c \in \mathcal{C}} \exp(-\psi_c(\mathbf{x}_c)) = \frac{1}{Z(\mathbf{D})} \exp(-\sum_{c \in \mathcal{C}} \psi_c(\mathbf{x}_c)), \tag{2.2.6}$$

where $Z(\mathbf{D}) = \sum_{\mathbf{x}} \prod_{c \in \mathcal{C}} \exp(-\psi_c(\mathbf{x}_c))$ is a normalizing constant. It should be noted that now Z is a function of given data and summation is over the possible label configurations only.

2.2.3 Pairwise Model

We will now focus on a *pairwise model* where the maximal clique size is two. We describe the forms of the potential functions for CRF models; the MRF follows similar strategies. For the pairwise model, the CRF takes following form:

$$\Pr(\mathbf{x}|\mathbf{D}) = \frac{1}{Z(\mathbf{D})} \exp(-\sum_{i \in \mathcal{V}} \psi_i(x_i) - \sum_{j \in N_i} \psi_{ij}(x_i, x_j)), \tag{2.2.7}$$

where $\psi_i(x_i)$ and $\psi_{ij}(x_i, x_j)$ are unary and pairwise potential functions respectively. The unary term $\psi_i(x_i)$ captures the correlation between an unobserved x_i variable and observed D_i data, measuring the data likelihood $\Pr(\mathbf{D}|\mathbf{x})$. The term $\psi_{ij}(x_i, x_j)$ encodes the correlation between a pair of unobserved variables x_i and x_j, and corresponds to the prior distribution $\Pr(\mathbf{x})$. The corresponding Gibbs energy function is the negative log-likelihood of the conditional probability:

$$E(\mathbf{x}|\mathbf{D}) = \sum_i \psi_i(x_i) + \sum_{ij} \psi_{ij}(x_i, x_j). \tag{2.2.8}$$

For many problems in computer vision, pairwise terms generally enforce smoothness constraints by encouraging a pair of neighbouring variables to take the same label. An instance of enforcing pairwise constraints using a MRF model is shown in Fig. 2.3 where we observe large improvement in qualitative accuracies for stereo labelling problem. Such constraints can be incorporated by taking data dependent contrast-sensitive Potts, (truncated) linear, or (truncated) quadratic models

defined over pairs of variables [20]. These models penalise different label assignments to a pair of variables. For example, the contrast-sensitive Potts model takes following form:

$$\psi_{ij}(x_i, x_j) \quad = \quad \left\{ \begin{array}{ll} 0 & x_i = x_j \\ \kappa(D_i, D_j) & \text{otherwise} \end{array} \right\}, \tag{2.2.9}$$

which does not add any cost for two variables X_i and X_j taking the same label but adds a high data-dependent cost $\kappa(D_i, D_j)$ for any other assignments. One common approach is to take $\kappa(D_i, D_j)$ as a mixture of Gaussian kernels based on the intensities I_i, I_j and positions p_i, p_j of the ith and j^{th} pixels, i.e.

$$\kappa(D_i, D_j) = w_1 \exp\left(-\frac{|p_i - p_j|^2}{2\theta_\alpha^2} - \frac{|I_i - I_j|^2}{2\theta_\beta^2}\right) + w_2 \exp\left(-\frac{|p_i - p_j|^2}{2\theta_\gamma^2}\right), \tag{2.2.10}$$

where w_1 and w_2 are weights of the kernel components. The first kernel is an *appearance kernel* to ensure that the nearby pixels in feature space are more likely to get the same label. The nearness and similarity are measured by the parameters θ_α and θ_β. The second kernel enforces smoothness by removing some isolated regions of labels. The parameters of the CRF are learned by maximizing the likelihood of observing the labels given the data [99].

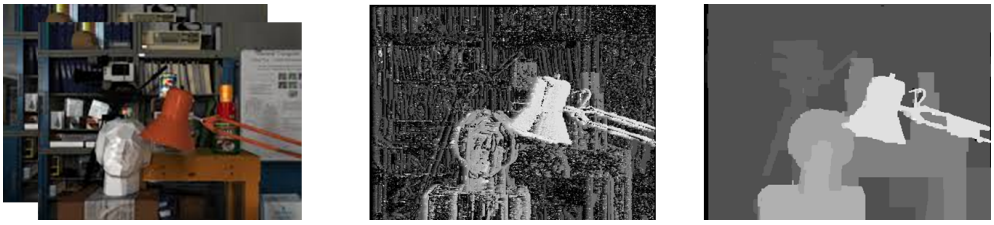


Figure 2.3: *We show importance of pairwise MRF for stereo on the tsukuba image with 16 disparity labels. Given a pair of rectified stereo images (left), output of the pairwise MRF (right) is much better than that of unary potential output (middle).*

In pairwise models we often assume that each variable is connected to or dependent on only four or eight neighbouring variables. Such grid pairwise models have formed the de-facto basis for solving computer vision problems. Nevertheless these models have limited expressive ability, i.e. for many applications it is desirable to get very fine object boundaries but such models fail in this regard. Additionally such models fail to capture the contextual information between two variables which may be closer in colour space. Further, they lead to shrinkage bias that shortens the object boundary. These associated issues are highlighted

in Fig. 2.4.

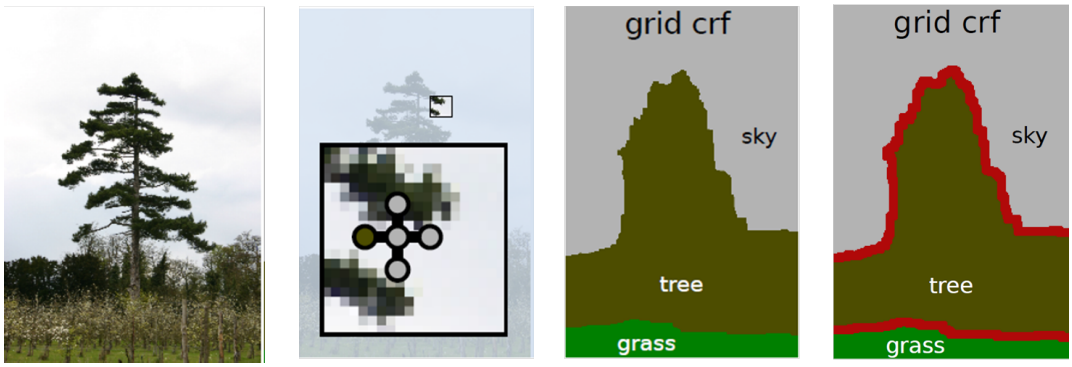


Figure 2.4: *We highlight one issue associated with the grid CRF model. Grid CRF is able to capture only the information about its immediate neighbours, which leads to over-smoothing around the object boundaries. The main reason is that the since each pixel is connected to its immediate neighbours, it does not get the long-range contextual information.*

In this thesis I consider two different kinds of pairwise CRF models. A basic CRF model considers pairwise potentials defined only on neighbouring pixels. However such CRF models have limited expressive power in capturing long range interaction in the image and generally lead to over-smoothing around the object boundaries. An alternative model explores a fully-connected pairwise structure which involves an edge or link between every pair of variables and thus captures long-range interactions in the image. Such a model enables extraction of very fine object boundaries objects. Figure 2.5 demonstrates the benefits of using a fully connected model, which clearly helps to improve the expressive power of the pairwise CRF.

2.2.4 Higher Order Models

Although pairwise models have gathered much popularity due to accuracy and speed achieved on some labelling problems, their representative power is limited. For example, it is observed that pixels belonging to a segment often have the same label; for instance they may belong to the same object or may have the same orientation. An increasing number of algorithms have incorporated such higher order information, and show improved accuracy on many problems [54, 79, 96]. Figure 2.6 shows how higher order potentials based on segment label consistency helped to improve per-pixel object class segmentation problem.

Now let us define higher order potential functions $\psi_c(\mathbf{x}_c); c \in \mathcal{C}$ which depend on a subset of variables $\mathbf{x}_{c\in\mathcal{C}} \subseteq \mathbf{x}$. The corresponding Gibbs energy for the CRF

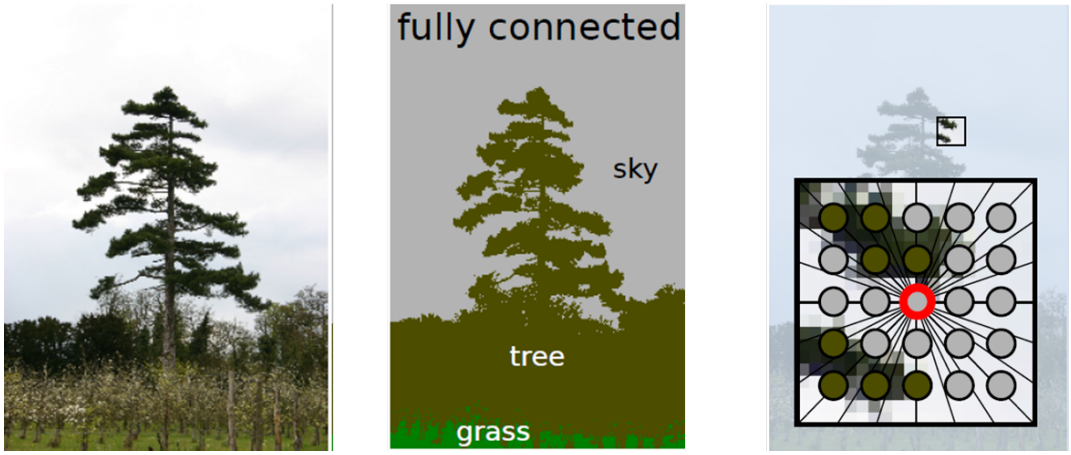


Figure 2.5: *Since dense pairwise CRF is able to capture long range interaction, a sky pixel which is immediately surrounded by tree pixels can also get attached to other sky pixel through long-range interactions. This essentially finally helps these pixels to get the sky label. Thus we are able to recover very fine boundaries which is desirable in many applications.*

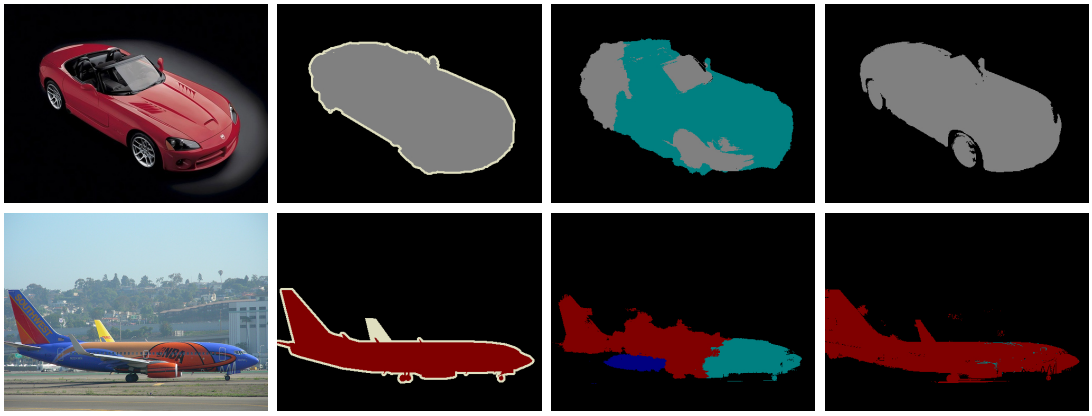


Figure 2.6: *We show how higher order information helps in improving qualitative results on object labelling problems. We demonstrate through experiments conducted on a benchmark PascalVOC-10 dataset. From left to right: input image, ground truth, output from [87] (Dense CRF), output from our dense CRF with Potts and co-occurrence terms.*

representing such a configuration is

$$E(\mathbf{x}|\mathbf{D}) = \sum_{c \in \mathcal{C}} \psi_c(\mathbf{x}_c). \tag{2.2.11}$$

Generally it is not feasible to perform exact inference with each configuration of higher order terms, since for a clique of size $|C|$ there are $2^{|C|}$ possible configura-

tions even in a binary segmentation problem. Many simplified models have been proposed to approximate the exact model, for example by using a model that only contains third order cliques [188]. However we would like our algorithms to handle more complex terms defined over cliques larger than third order. We provide details of two such models. The first assumes consistency over regions such that the pixels or variables within a region take the same label. However, since the assumption of a segment taking the same label is very restrictive, a second model uses *pattern based potentials* (as introduced in [83], [138]).

Label consistency: A common way to enforce label consistency is to first regroup the image into a set of segments, often generated using some standard unsupervised segmentation approach. Generally these segments follow object boundaries; the pixels within a segment are forced to take the same label. Kohli et al. [78] extended the Potts based pairwise model to incorporate higher order terms which take the form of P^n Potts model:

$$\psi_c(\mathbf{x}_c) = \left\{ \begin{array}{cc} 0 & \text{if } x_i = l_k, \forall i \in c \\ \theta_1 |c|^{\theta_\alpha} & \text{otherwise} \end{array} \right\}, \tag{2.2.12}$$

where $|c|$ is the size of the clique, and θ_1 and θ_α are parameters of the Potts potentials. However there is one issue associated with this model: it induces the same cost for a single pixel within the segment taking different label as for any other number of inconsistent pixels. Kohli et al. [79] proposed a random field model which incorporated the degree of inconsistency into the cost, called *robust* higher-order potentials, defined as:

$$\psi_c(\mathbf{x}_c) \quad = \quad \left\{ \begin{array}{cc} N_i(\mathbf{x}_c)\frac{1}{Q}\gamma_{max} & \text{if } N_i(\mathbf{x}_c) \leq Q \\ \gamma_{max} & \text{otherwise} \end{array} \right\}, \tag{2.2.13}$$

where $N_i(\mathbf{x}_c)$ is the number of inconsistent pixels, i.e. the number of pixels in the clique not taking the dominant label, Q is a threshold and $\gamma_{\max}$ is the maximum cost for the constraint violation.

Pattern-based potentials: While using region based consistency terms works very well in problems where contiguous sets of pixel tend to have the same label, as in on object segmentation problem, this is not true in many other cases. For example, a highly textured region may generally require many different labels. We now discuss pattern-based potentials that attempt to model arbitrary labellings. For a clique of size $|c|$ there are a total of $\mathcal{L}^{|c|}$ label configurations. However, it has been observed that only a small fraction of these possible configurations actually appear in natural images [80]. Thus there is a lot of sparsity. We call the small

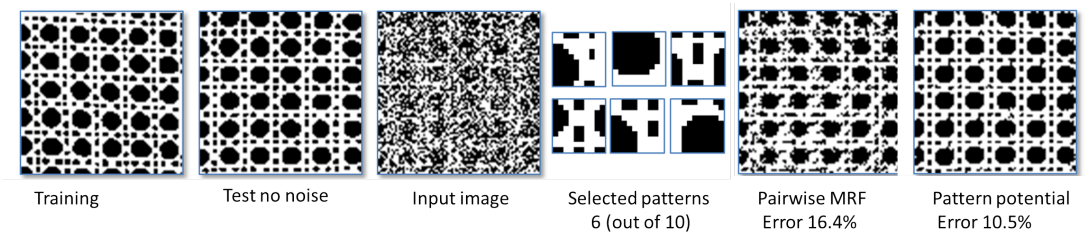


Figure 2.7: *We show the importance of incorporating a pattern based potential on binary texture restoration from Brodatz texture D101. (a) Training image (86×86). (b) Test image. (c) Test image with 60 % noise, used as input. (d) 6 selected patterns of size 10 ×10 pixels. (e) Result of pairwise MRF. (f) Result after enforcing a pattern based potential.*

fraction of possible label configurations $\mathcal{P}_c \subset \mathcal{L}^{|c|}$ the patterns. In Figure 2.7 we show some of the patterns that have been used for texture denoising (refer to [138] for more details). Each pattern $\mathcal{P}_i$ is associated with its corresponding cost, given as γ_i. We design a potential function which gives a small cost to a configuration within the possible patterns, and otherwise assigns a very high cost:

$$\psi_c(\mathbf{x}_c) \quad = \quad \left\{ \begin{array}{cc} \gamma_i & \text{if } \mathbf{x}_c = \mathcal{P}_i \\ \gamma_{max} & \text{otherwise} \end{array} \right\}. \tag{2.2.14}$$

In this chapter I have shown how a range of labelling problems can be formulated in random field models, especially in conditional random field framework. Next I discuss some of these approximate algorithms and describe their properties, especially emphasising a variational mean-field approach [81,183] that transforms two inference problems into an optimization problem.

Chapter 3

Mean-field Inference

3.1 Mean-field Inference

We first come to a class of methods known in the Bayesian literature as variational methods. These have a close relationship to belief propogation and are not to be confused with the variational methods of Chambolle et al. [25]. The key idea is to approximate a complex Conditional Random Field by a simpler one. To make this approximation good one would like to be able to optimize the variational parameters of the simpler network subject to some measure of difference between the two models. These variational parameters are then used to approximate the marginal probabilities. Thus, given a probability distribution $\Pr(\mathbf{x}|\mathbf{D})$ which one cannot solve (in the sense of finding its maximum or mode), the steps of the variational methods are as follows:

- Choose a measure of distance by which one can compare a distribution Pr with similar but simpler distribution Q.
- Specify a class of probability distributions in which one wishes to find a similar distribution Q.
- Find Q from the given class that minimizes the distance to the original distribution Pr.
- Solve the distribution Q.

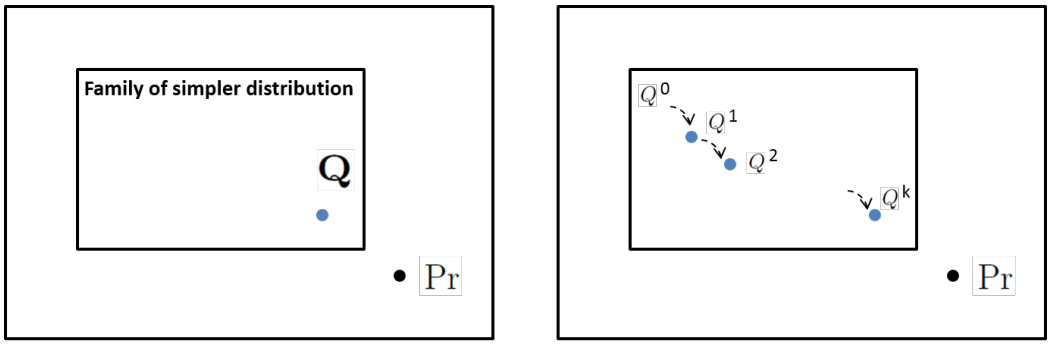


Figure 3.1: *In the mean-field approach we approximate the true distribution Pr by an approximate distribution Q, where Q belongs to a simpler family of distributions. For example, one simple family is where each variable is assumed to be independent. It is an iterative algorithm where each iteration leads to a new Q which is a better approximation to Pr.*

In Fig. 3.1, we visualise the mean-field approach. The steps of finding the closest Q from the chosen class leads to an optimization problem. In order derive the optimization problem, we again consider the probability distribution $\Pr(\mathbf{x}|\mathbf{D})$ for a random field, given by the Gibbs distribution:

$$\Pr(\mathbf{x}|\mathbf{D}) = \frac{1}{Z(\mathbf{D})}\exp(-E(\mathbf{x}|\mathbf{D})) = \frac{1}{Z(\mathbf{D})}\exp(-\sum_{c\in\mathcal{C}}\psi_c(\mathbf{x}_c)). \tag{3.1.1}$$

As usual, $\mathbf{x}_c$ represents the subvector of $\mathbf{x}$ consisting of the values of those variables in the clique c. We will also have cause to use the notation $\mathbf{x}_{\bar{c}}$ to represent the values of those variables not in c. For simplicity, from now on, dependence on $\mathbf{D}$ is being dropped from notations and we write $\Pr(\mathbf{x}|\mathbf{D})$ as $\Pr(\mathbf{x})$, $E(\mathbf{x}|\mathbf{D})$ as $E(\mathbf{x})$, and $Z(\mathbf{D})$ by Z.

3.1.1 Distance Function

A natural measure that has been commonly used is the KL divergence $D_{KL}(Q||\Pr)$, which gives a score for the difference between two probability distributions Q and Pr:

$$\begin{aligned} D_{KL}(Q||\Pr) &= \sum_{\mathbf{x}} Q(\mathbf{x})\log\left(\frac{Q(\mathbf{x})}{\Pr(\mathbf{x})}\right) && (3.1.2)\\ &= -\sum_{\mathbf{x}} Q(\mathbf{x})\log(\Pr(\mathbf{x})) + \sum_{\mathbf{x}} Q(\mathbf{x})\log(Q(\mathbf{x})). && (3.1.3)\end{aligned}$$

Now, plugging in the form of the probability $\Pr(\mathbf{x})$ given in (3.1.3), we obtain

$$\begin{aligned} D_{KL}(Q||\Pr) &= -\sum_{\mathbf{x}} Q(\mathbf{x})\log\left(\frac{1}{Z}\exp(-E(\mathbf{x}))\right) \\ &\quad + \sum_{\mathbf{x}} Q(\mathbf{x})\log(Q(\mathbf{x})) && (3.1.4)\\ &= \sum_{\mathbf{x}} Q(\mathbf{x})E(\mathbf{x}) + \log(Z) + \sum_{\mathbf{x}} Q(\mathbf{x})\log(Q(\mathbf{x})), && (3.1.5)\end{aligned}$$

where Q is a probability distribution so we have $\sum_{\mathbf{x}} Q(\mathbf{x}) = 1$.

In order to get the closest Q to Pr, we minimize the above KL-divergence over Q. Because $\log(Z)$ is constant with respect to the label configuration $\mathbf{x}$, minimizing the KL divergence $D(.)$ between these two distributions is equivalent to minimizing the following energy functional $F(\Pr, Q)$:

$$F(\Pr, Q) = \sum_{\mathbf{x}} Q(\mathbf{x})E(\mathbf{x}) + \sum_{\mathbf{x}} Q(\mathbf{x})\log(Q(\mathbf{x})). \tag{3.1.6}$$

The first term evaluates the expectation of the energy function under the Q distribution. The second term is the negative *entropy* of the probability distribution Q. Overall the functional $F(.)$ is convex in Q since it is a combination of a linear term (first term) and a convex term (second term).

We now expand further the first term of (3.1.6) using the representation of the energy in (3.1.1) as a sum of clique energies. Changing the order of summation results in

$$\sum_{\mathbf{x}} Q(\mathbf{x})E(\mathbf{x}) = \sum_{c\in\mathcal{C}} \sum_{\mathbf{x}} Q(\mathbf{x})\psi_c(\mathbf{x}_c). \tag{3.1.7}$$

For a given clique c the sum over $\mathbf{x}$ can be broken down into separate sum over $\mathbf{x}_c$ and $\mathbf{x}_{\bar{c}}$, those variables in c and those not in c. Then

$$\begin{aligned} \sum_{\mathbf{x}} Q(\mathbf{x})E(\mathbf{x}) &= \sum_{c\in\mathcal{C}} \sum_{\mathbf{x}_c} \sum_{\mathbf{x}_{\bar{c}}} Q(\mathbf{x})\psi_c(\mathbf{x}_c) &(3.1.8)\\ &= \sum_{c\in\mathcal{C}} \sum_{\mathbf{x}_c} \psi_c(\mathbf{x}_c) \sum_{\mathbf{x}_{\bar{c}}} Q(\mathbf{x}), &(3.1.9) \end{aligned}$$

where $\psi_c(\mathbf{x}_c)$ has been moved out of the last summation because it is constant with regard to changing values of $\mathbf{x}_{\bar{c}}$. Finally we observe that $\sum_{\mathbf{x}_{\bar{c}}} Q(\mathbf{x})$ is equal to the marginal probability $Q(\mathbf{x}_c)$, so the final form becomes

$$\sum_{\mathbf{x}} Q(\mathbf{x})E(\mathbf{x}) = \sum_{c\in\mathcal{C}} \sum_{\mathbf{x}_c} \psi_c(\mathbf{x}_c)Q(\mathbf{x}_c). \tag{3.1.10}$$

Thus, the expected value of the energy (under distribution Q) is equal to the sum of the expected clique energies.

Example: For an energy distribution involving unary and pairwise terms, of the form

$$E(\mathbf{x}) = \sum_{i=1}^{n} \psi_i(x_i) + \sum_{i,j} \psi_{ij}(x_i, x_j), \tag{3.1.11}$$

the corresponding summation in Eq. (3.1.6) is

$$\sum_{\mathbf{x}} Q(\mathbf{x})E(\mathbf{x}) = \sum_{i=1}^{n} \sum_{x_i} Q(x_i)\psi_i(x_i) + \sum_{i,j} \sum_{x_i,x_j} Q(x_i, x_j)\psi_{ij}(x_i, x_j), \tag{3.1.12}$$

where as before $Q(x_i)$ and $Q(x_i, x_j)$ are marginal probabilities.

3.1.2 Naive mean-field

Now we discuss the family of distributions for Q. The simplest possible approximation is when we assume all the variables are independent. Under this assumption we can represent the distribution Q as the product of independent marginals, each defined on a single random variable X_i. Thus

$$Q(\mathbf{x}) = \prod_i Q(x_i), \tag{3.1.13}$$

where each $Q(x_i)$ is a probability distribution. A simple example of naive mean-field approximation of a pairwise MRF is shown in Fig. 3.2. At first sight this seems to be so weak an assumption as to not be of much use. However, the assumption leads to a very simple update rule that has turned out to be very useful in practice.

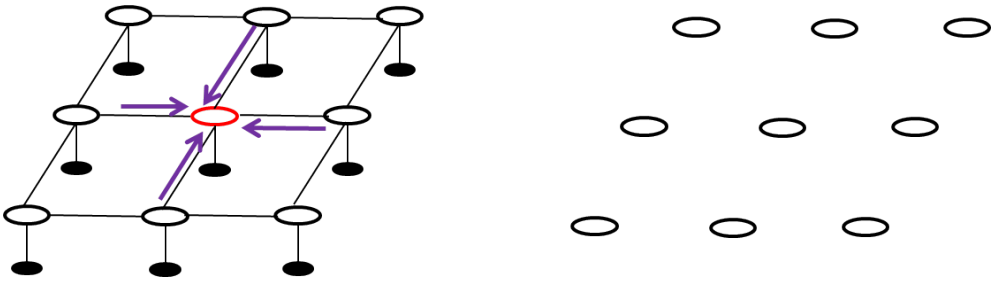


Figure 3.2: *We show a naive mean-field approximation where the true MRF distribution is approximated by a distribution (shown on the right) where each variable is assumed to be independent.*

The advantage of choosing such a simple probability distribution is that the second term of (3.1.6), representing the negative entropy of Q, takes a very simple form.

If $Q(\mathbf{x}) = \prod_{i=1}^{n} Q(x_i)$, each $Q(x_i)$ being a probability distribution, then

$$\sum_{\mathbf{x}} Q(\mathbf{x}) \log(Q(\mathbf{x})) = \sum_i \sum_{x_i} Q(x_i) \log(Q(x_i)). \tag{3.1.14}$$

Thus, the entropy of Q is equal to the sum of entropies of the individual

probability distributions $Q(x_i)$. We prove this for a two variable problem:

$$\begin{aligned}
\sum_{\mathbf{x}} Q(\mathbf{x}) \log(Q(\mathbf{x})) &= \sum_{x_1,x_2} Q(x_1)Q(x_2) \log(Q(x_1)Q(x_2)) \\
&= \sum_{x_1,x_2} Q(x_1)Q(x_2) \log(Q(x_1)) + \\
&\quad \sum_{x_1,x_2} Q(x_1)Q(x_2) \log(Q(x_2)) \\
&= \sum_{x_1} Q(x_1) \log(Q(x_1)) \sum_{x_2} Q(x_2) + \\
&\quad \sum_{x_2} Q(x_2) \log(Q(x_2)) \sum_{x_1} Q(x_1) \\
&= \sum_{x_1} Q(x_1) \log(Q(x_1)) + \sum_{x_2} Q(x_2) \log(Q(x_2)) \\
&= \sum_{i} \sum_{x_i} Q(x_i) \log(Q(x_i)). \qquad (3.1.15)
\end{aligned}$$

As a result the formula (3.1.6) for the label varying terms in the KL divergence becomes

$$F(Q, \Pr) = \sum_{c \in \mathcal{C}} \sum_{\mathbf{x}_c} \left(\psi_c(\mathbf{x}_c) \prod_{i \in c} Q(x_i) \right) + \sum_{i=1}^{n} \sum_{x_i} Q(x_i) \log(Q(x_i)). \quad (3.1.16)$$

In the particular case of energy function with only pairwise terms, the formula becomes

$$\begin{aligned}
F(Q, \Pr) &= \sum_{i} \sum_{x_i} Q(x_i)\psi_i(x_i) + \sum_{i,j} \sum_{x_i,x_j} Q(x_i)Q(x_j)\psi_{ij}(x_i, x_j) \\
&\quad + \sum_{i=1}^{n} \sum_{x_i} Q(x_i) \log(Q(x_i). \qquad (3.1.17)
\end{aligned}$$

3.1.3 Minimizing the distance

The next step in the mean field method is to find the distribution Q from the defined class that most closely approximates Pr. For this we first write the mean-field optimization problem:

$$\begin{aligned}
\min_{Q} \quad & F(\Pr, Q) && (3.1.18) \\
\text{subject to} \quad & \sum_{x_i} Q(x_i) = 1 \ \ \forall i \in \mathcal{V} && (3.1.19) \\
& Q(\mathbf{x}) = \prod_{i} Q(x_i). && (3.1.20)
\end{aligned}$$

This problem can be equivalently written as

$$\min_{Q(\mathbf{x})} \quad \sum_{\mathbf{x}} Q(\mathbf{x}) \left(\sum_{c \in \mathcal{C}} \psi_c(\mathbf{x}_c) \right) + \sum_{\mathbf{x}} Q(\mathbf{x}) \log Q(\mathbf{x}) \tag{3.1.21}$$

$$\text{subject to} \quad \sum_{x_i} Q(x_i) = 1 \;\; \forall i \in \mathcal{V} \tag{3.1.22}$$

$$Q(\mathbf{x}) = \prod_i Q(x_i). \tag{3.1.23}$$

We have now a continuous optimization over a set of probability distributions. In order to optimize the function, we follow a method of Lagrange multipliers for characterizing the fixed point solutions for each marginal $Q(X_i)$ corresponding to the variable X_i given the other marginals $Q(X_1), ..., Q(X_n)$. In general the naive mean-field problem is a non-convex problem (as shown later in the chapter). Now we form the Lagrangian by introducing the multiplier λ corresponding to the constraint on $Q(X_i)$:

$$L_i(Q) = \sum_{\mathbf{x}} Q(\mathbf{x}) \left(\sum_{c \in \mathcal{C}} \psi_c(\mathbf{x}_c) \right) + \sum_{\mathbf{x}} Q(\mathbf{x}) \log Q(\mathbf{x}) + \lambda \left(\sum_{x_i} Q(x_i) - 1 \right). \tag{3.1.24}$$

Taking the derivative of $L_i(Q)$ with respect to $Q(x_i)$ gives

$$\frac{\partial L_i(Q)}{\partial Q(x_i)} = \sum_{c \in \mathcal{C}_i} \sum_{\mathbf{x}_c} \prod_{x_k \neq x_i} Q(x_k) \psi_c(\mathbf{x}_c) + \log Q(\mathbf{x}) + 1 + \lambda, \tag{3.1.25}$$

where we use $\mathcal{C}_i$ to mean the set of all cliques that include the ith variable. Setting the derivative to zero and rearranging terms we get

$$\log Q(x_i) = -1 - \lambda - \sum_{c \in \mathcal{C}_i} \sum_{\mathbf{x}_c} \prod_{x_k \neq x_i} Q(x_k) \psi_c(\mathbf{x}_c). \tag{3.1.26}$$

This is a maximum as $F(P, Q)$ is a sum of the expectation of the Gibbs energy, which is linear in $Q(X_i)$ given all other elements fixed, and the entropy which is a concave function, leading to a concave function in $Q(X_i)$ with a unique global optimum. We can take exponents on both sides and renormalise, and we can also drop λ since it is constant with respect to x_i. We then obtain the following

update equation:

$$Q(x_i) = \frac{1}{Z_i} \exp\left(-\sum_{c \in \mathcal{C}_i} \sum_{\mathbf{x}_c} \prod_{x_k \neq x_i} Q(x_k) \psi_c(\mathbf{x}_c) \right). \tag{3.1.27}$$

Example. The fixed point solution for $Q(x_i)$ for a pairwise MRF takes the following form:

$$Q(x_i = l) = \frac{1}{Z_i} \exp\{-\psi_i(x_i = l) - \sum_{l' \in \mathcal{L}} \sum_{j \neq i} Q(x_j = l') \psi_{ij}(x_i = l, x_j)\}, \tag{3.1.28}$$

where $Z_i = \sum_{x_i = l \in \mathcal{L}} \exp\{-\psi_i(x_i) - \sum_{l' \in \mathcal{L}} \sum_{j \neq i} Q(x_j = l') \psi_{ij}(x_i, x_j)\}$ is a constant which normalizes the marginal at pixel i. If the updates in Eq. 8.3.7 are made in sequence across pixels $i = 1...N$ (updating and normalizing the L values $Q(x_i = l)$, $l = 1...L$ at each step), the KL-divergence is guaranteed to decrease [81]. For a fully connected pairwise model, the marginal update for the variable X_i requires receiving messages from all its neighbours (the second term in Eq. 3.1.28). Thus the time complexity of the naive mean-field algorithm is $O(N^2 L)$.

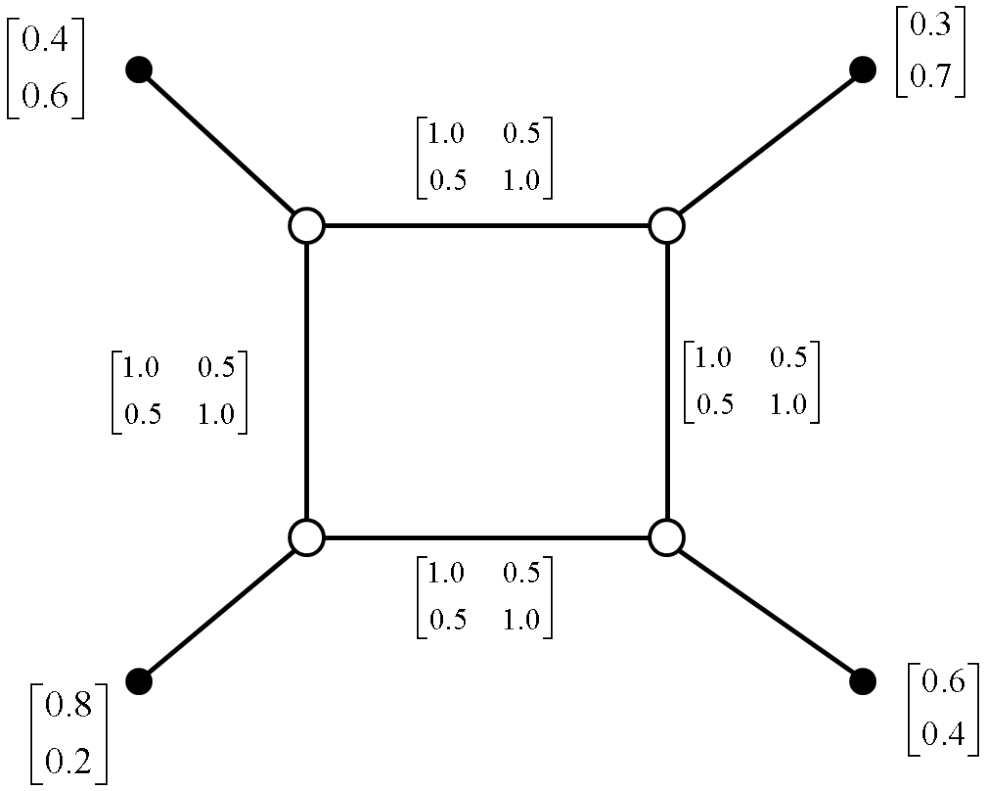


Figure 3.3: *We take a simple MRF consisting of 4 variables where all the unary and pairwise potentials are given respectively. We will show the MAP solution and max-marginals on this example.*

x_1	x_2	x_3	x_4	$\Pr(x_1, x_2, x_3, x_4)$	$\Pr(x_1, x_2, x_3, x_4$	$\mathrm{E}(x_1, x_2, x_3, x_4)$
0	0	0	0	0.0576	0.3870	0.9493
0	0	0	1	0.0216	0.0415	3.1820
0	0	1	0	0.0036	0.0242	3.7214
0	0	1	1	0.0054	0.0104	4.5659
0	1	0	0	0.0096	0.0645	2.7410
0	1	0	1	0.0036	0.0138	4.2830
0	1	1	0	0.0024	0.0161	4.1289
0	1	1	1	0.0036	0.0069	4.9762
1	0	0	0	0.0336	0.1451	1.9303
1	0	0	1	0.0504	0.0622	2.7774
1	0	1	0	0.0021	0.0089	4.7217
1	0	1	1	0.0126	0.0155	4.1669
1	1	0	0	0.0224	0.0967	2.3361
1	1	0	1	0.0336	0.0415	3.1820
1	1	1	0	0.0056	0.0242	3.7214
1	1	1	1	0.0336	0.0415	3.1820

Table 3.1: Un-normalized and normalized probabilities of all 16 possible labellings of the MRF shown in Fig. 3.3. The probabilities and energies are computed using equation 2.2.3. The partition function Z equation 2.2.4. For this CRF parameter Z = 3:153e

Now we show using a toy example that the naive mean-field marginals only approximate the true marginals. For the toy MRF example shown in Fig. 3.3 the probability of each possible configuration is given in Tab. 3.1, so the MAP solution is $(0, 0, 0, 0)$ with probability as 0.3870. The marginal distribution for this MRF is:

$$\begin{aligned} P(x_1 = 0) = 0.5644 \quad P(x_1 = 1) = 0.4356 \\ P(x_2 = 0) = 0.6948 \quad P(x_2 = 1) = 0.3052 \\ P(x_3 = 0) = 0.8523 \quad P(x_3 = 1) = 0.1477 \\ P(x_4 = 0) = 0.7667 \quad P(x_4 = 1) = 0.2333, \end{aligned} \tag{3.1.29}$$

with the max-marginal as $(0, 0, 0, 0)$. Thus the MAP solution and max-marginal agree with each other. The mean-field marginals in the first iteration are:

$$\begin{aligned} Q(x_1 = 0) = 0.3672 \quad Q(x_1 = 1) = 0.6328 \\ Q(x_2 = 0) = 0.6644 \quad Q(x_2 = 1) = 0.3356 \\ Q(x_3 = 0) = 0.8213 \quad Q(x_3 = 1) = 0.1787 \\ Q(x_4 = 0) = 0.7000 \quad Q(x_4 = 1) = 0.3000. \end{aligned} \tag{3.1.30}$$

At the end of the first iteration, the mean-field marginals are different from the actual marginal, so the max-mean-field marginal $(1, 0, 0, 0)$ is also different from the MAP solution. In general, these marginals improve over iterations. In Fig. 3.4 we show how the mean-field marginals improve across iterations over different labels for an object class segmentation problem.

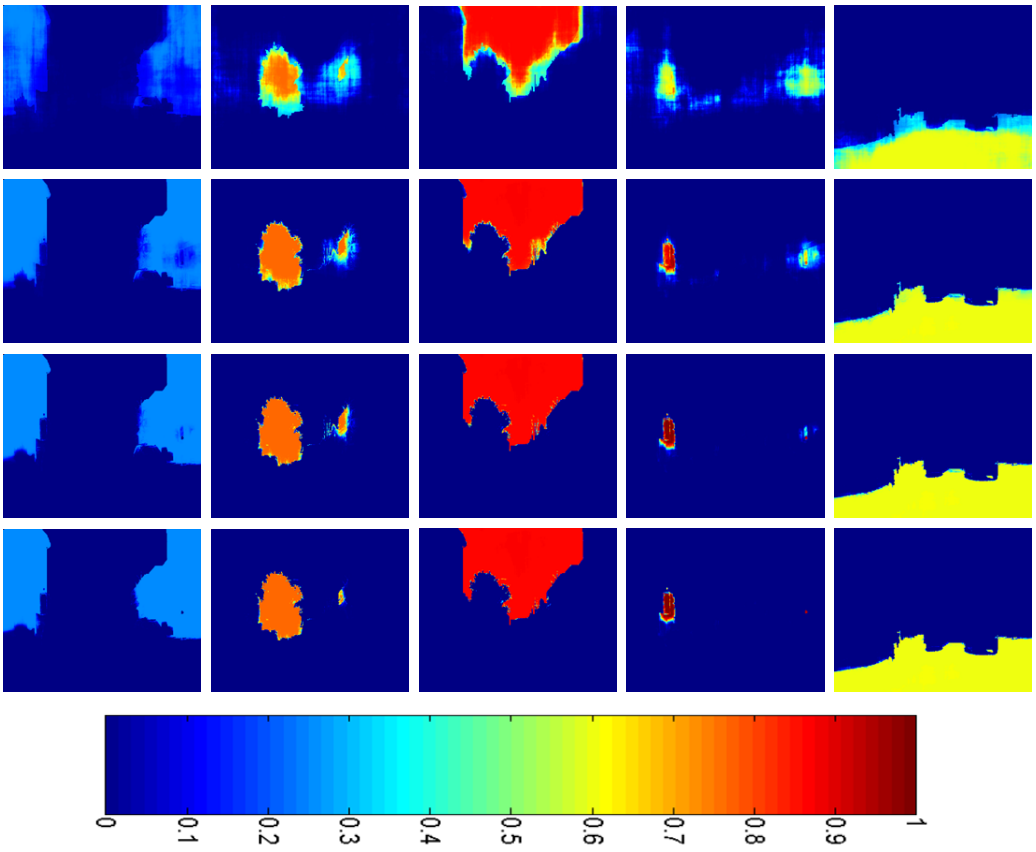


Figure 3.4: *This figure shows the Q distribution values across different iterations of the mean-field method for building, tree, sky, sign, and road classes on CamVID dataset [23] (a benchmark dataset for road scene understanding) involving unary and pairwise potentials. We can observe how the confidence of pixels keeps on increasing after each iteration for each of their respective classes.*

3.2 Improving naive mean-field approximation

The mean-field distribution approximates the true distribution by introducing a very simple model that assumes all the variables independent. This leads to a distribution with which we can do efficient inference, but this comes at the cost of accuracy. There have been many attempts to increase the accuracy of the approximation. Specifically we discuss three ways to improve the naive mean-

field corresponding to a 4-connected grid graph. Firstly we try to improve the richness of the original model, i.e. by having a fully connected pairwise graph where each variable is connected to all other variables. Secondly we look at improving the approximation by introducing a more complicated distribution, or having a mixture distribution.

3.2.1 Fully connected pairwise model

For a fully connected pairwise model, the mean-field update for $Q(X_i)$ given the marginal distributions of all other variables is given by

$$Q(x_i = l) = \frac{1}{Z_i} \exp\{-\psi_i(x_i = l) - \sum_{l' \in \mathcal{L}} \sum_{j \neq i} Q(x_j = l')\psi_{ij}(x_i = l, x_j)\}. \quad (3.2.1)$$

For a densely connected graph a naive mean-field algorithm has a quadratic complexity in the number of variables, because the update for each variable X_i involves summing the messages from all other variables $X_{j \neq i}$. However, recent work [87] has shown how such quadratic complexity can be reduced to linear by transforming the update process into one applying high dimensional filtering in the Q space. We provide a brief overview of the approach below.

Filter-based mean-field: We again take a pairwise CRF where the pairwise terms take the contrast-sensitive Potts model.

$$\psi_{ij}(x_i, x_j) \quad = \quad \left\{ \begin{array}{cc} 0 & x_i = x_j \\ \kappa(D_i, D_j) & \text{otherwise} \end{array} \right\}, \quad (3.2.2)$$

which does not add any cost for two variables X_i and X_j taking the same label but adds a high data-dependent cost $\kappa(D_i, D_j)$ for any other assignments. We also write the pairwise term $\psi_{ij}(x_i, x_j)$ concisely as

$$\psi_{ij}(x_i, x_j) \quad = \quad \mu(x_i, x_j)\kappa(D_i, D_j), \quad (3.2.3)$$

where $\mu(x_i, x_j)$ is a label compatibility term which is 0 if the variables X_i and X_j take the same label and is 1 otherwise. Now assume that $\kappa(D_i, D_j)$ takes the form of a linear combination of Gaussian kernels

$$\kappa(D_i, D_j) = w_1 \exp\left(-\frac{|p_i - p_j|^2}{2\theta_\alpha^2} - \frac{|I_i - I_j|^2}{2\theta_\beta^2}\right) + w_2 \exp\left(-\frac{|p_i - p_j|^2}{2\theta_\gamma^2}\right), \quad (3.2.4)$$

which we simplify to write as

$$\kappa(D_i, D_j) = \sum_m w^m \kappa^m(D_i, D_j). \tag{3.2.5}$$

On substituting the pairwise potential into the mean-field updates we get

$$\begin{aligned} Q_i(x_i = l) &= \frac{1}{Z_i} \exp\{-\psi_i(x_i) - \sum_{l' \in \mathcal{L}} \sum_{j \neq i} Q_j(x_j = l') \left[\mu(l, l') \sum_m w^m \kappa^m(D_i, D_j) \right] \\ &= \frac{1}{Z_i} \exp\{-\psi_i(x_i) - \sum_m w^m \sum_{l' \in \mathcal{L}} \sum_{j \neq i} Q_j(x_j = l') \left[\mu(l, l') \kappa^m(D_i, D_j) \right] \\ &= \frac{1}{Z_i} \exp\{-\psi_i(x_i) - \sum_{l' \in \mathcal{L}} \mu(l, l') \sum_m w^m \sum_{l' \in \mathcal{L}} \underbrace{\sum_{j \neq i} Q_j(x_j = l') \kappa^m(D_i, D_j)}_{\tilde{Q}_i^m(l')} . \end{aligned} \tag{3.2.6}$$

Now we show the expensive step of updating $\tilde{Q}_i^m(l')$ can be written as an application of Gaussian kernels over Q distributions. For this, we transform $\tilde{Q}_i^m(l')$ as

$$\begin{aligned} \tilde{Q}_i^m(l') &= \sum_{j \neq i} Q_j(x_j = l') \kappa^m(D_i, D_j) && (3.2.7) \\ &= \sum_{j \neq i} k^m(D_i, D_j) Q_j(l') \\ &= [G^m \otimes Q(l')](D_i) - Q_i(l') && (3.2.8) \end{aligned}$$

where G^m is a Gaussian kernel corresponding to the mth component of Eq. 3.2.5, and $\otimes$ is the convolution operator. This corresponds to applying a Gaussian convolution over the Q distribution, also sometimes called cross-bilateral filtering. There are many efficient algorithms proposed in literature. In particular one of the most recent popular approaches uses permutohedral lattice based filtering, which reduces the complexity of convolution from $O(N^2)$ to $O(N)$. A naive mean-field update algorithm is given in Alg. 1.

In [87], it is shown that parallel updates for Eq. 3.2.1 can be evaluated by convolution with a high dimensional Gaussian kernel using any efficient bilateral filter, e.g. the permutohedral lattice method of [2] (which introduces a small approximation). Since $\sum_{j \neq i} Q_j(x_j = l') \psi_p(x_i, x_j)$ in Eq. 3.2.1 can be written as $\sum_m w^{(m)} \tilde{Q}_i^{(m)}(l')$, and approximate Gaussian convolution using [2] is $O(N)$, parallel[1] updates using Eq. 3.2.1 can be efficiently approximated in $O(MNL^2)$

[1]Although the updates are conceptually parallel in form, the permutohedral lattice convo-

Algorithm 1: Mean field algorithm

input : Energy function E, and initial Q distribution
converged := 0, ν := 1;
while $converged = 0$ **do**
  $\bar{Q}_i(x_i) \leftarrow \sum_{j \neq i} k(D_i, D_j) Q_j(x_j)$;
  $\bar{Q}_i(x_i) \leftarrow \sum_{l \in \mathcal{L}} \mu^m(x_i, l) \sum_m w^m \bar{Q}_i(l)$;
  $Q_i(x_i) \leftarrow \exp\{-\psi_i(x_i) - \bar{Q}_i(x_i)\}$;
  normalize $Q_i(x_i)$;
end
Return Q;

time (or $O(MNL)$ time for the Potts model), thus avoiding the need for the $O(MN^2L^2)$ calculations which would be required to calculate these updates individually. Since the method requires the updates to be made in parallel rather than in sequence, the convergence guarantees associated with the sequential algorithm are lost [81]. However, [87] observe good convergence properties in practice. The algorithm is run for a fixed number of iterations, and the MPM solution extracted by choosing $x_i \in \text{argmax}_l\, Q_i(x_i = l)$ at the final iteration.

Although [87] use the permutohedral lattice [2] for their filter-based inference, we note that other filtering methods can also be used for the convolutions in Eq. 3.2.7. Particularly, the recently proposed *domain transform* filtering approach [46] has certain advantages over the permutohedral lattice. Domain transform filtering approximates high-dimensional filtering, such as 5-D bilateral filtering in 2-D spatial and 3-D RGB range space, by alternating horizontal and vertical 1-D filtering operations on transformed 1-D signals which are isometric to slices of the original signal. Since it does not sub-sample the original signal its complexity is independent of the filter size, while in [2] the complexity and filter size are inversely related.

Effects of using filtering approach The Gaussian kernels are edge-preserving kernels. Such edge-preserving characteristics are very useful for object class segmentation. For example, it is desirable that two pixels with the same colour should receive the same label and with different colours should get different labels.

We use filtering-based mean-field approach for efficient inference in fully connected graph and show how it helps to recover very fine boundaries. Some of the input images, ground truth results and mean-field results are shown in Fig. 3.5. In Fig. 3.6(a) we show how the KL-divergence values decrease across iterations. Even though the filtering-based approach does not guarantee convergence we consistently see a decrease in the KL-divergence values across iterations. Further, it should be noted that the fully-connected model does not always help to improve

lution is implemented sequentially.

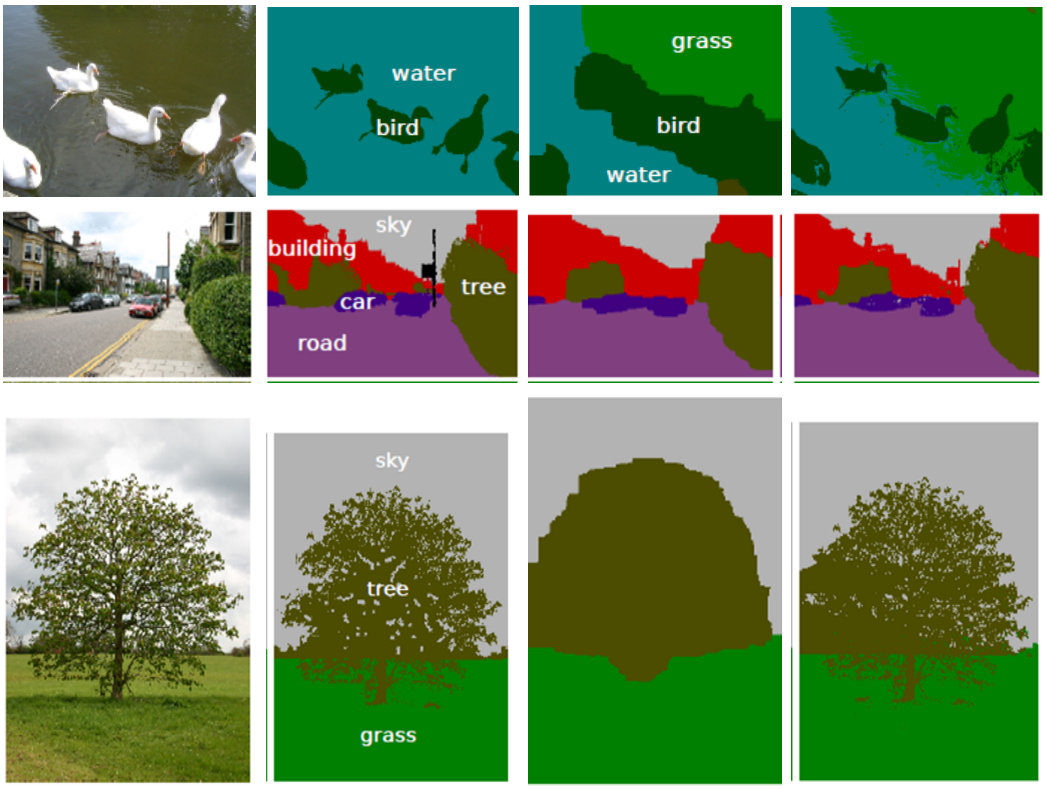


Figure 3.5: *Qualitative results on MSRC dataset. From left to right: input image, ground truth, output from [96] (grid), output from [87] (dense CRF).*

the accuracy. In certain cases it is observed that such long-range interaction in-fact leads to decreasing the overall accuracy (shown in Fig. 3.8). We observe that the accuracy is dependent on the density of graph, and so in Fig. (3.6(b), 3.7) we also show the accuracy versus size of neighbourhood graph to show the optimal sizes of the neighbourhood.

3.2.2 Structured mean-field

The naive mean-field approach approximates the true distribution with a much simpler form that involves complete factorization of the variables. As discussed earlier in Sec. 3.1.3, this simpler approximation leads to poor convergence properties. We now show a class of tractable distributions that captures some structure of the true distribution and thus provides better accuracy at faster convergence. In this framework, we approximate the true model Pr by a set of H tractable substructures $h_i \in \mathcal{H} = \{h_1, ..., h_H\}$. For example in Fig. 3.9 we show such a structured approximation where each tractable component may follow a tree structure. We have shown two such decompositions. In the first we have H chains each corresponding to one row which captures dependencies between variables in

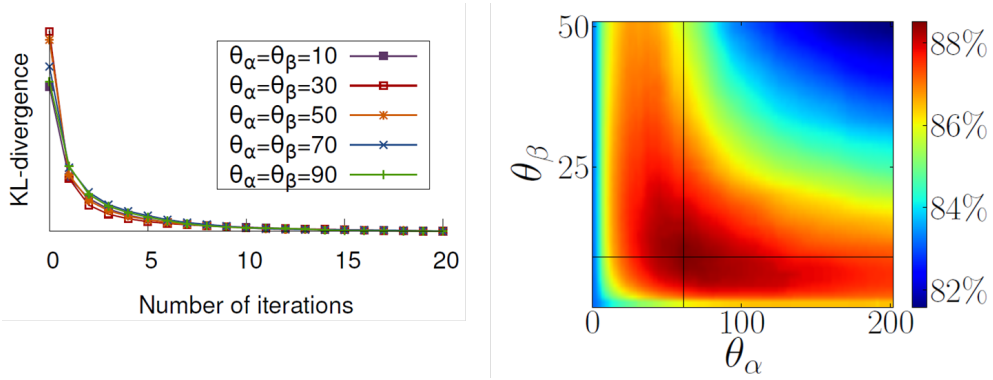


Figure 3.6: *Convergence analysis: these figures show the KL-divergence values of the mean-field approximation after each iteration for a pairwise CRF. Even though the theory does not guarantee convergence, we observe that when we have a pairwise model, the KL-divergence does not oscillate and always decreases. In the second figure we show that the long range interaction does not always lead to increase in accuracy: the curve shows the change in accuracy as we change the spatial and colour standard deviation.*

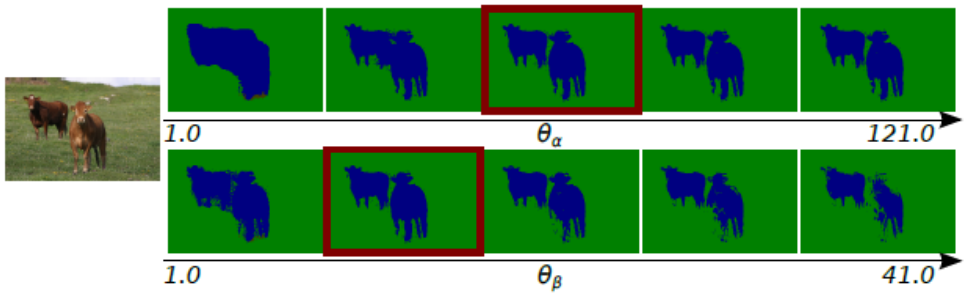


Figure 3.7: *Qualitative analysis: effect of dense or long-range interaction on output. As we increase the colour and spatial standard deviation, we observe increase in accuracy but only up-to a certain point after which we start to see a decrease in accuracy.*

the same row but ignores dependencies over different columns. There are many such tractable decompositions. For example, in the second example we decompose along columns, which captures dependencies between variables in the same column but ignores the dependencies along rows. Further, let us assume that the substructures form disjoint subsets. The important requirement is that we must be able to efficiently calculate probability properties such as MAP or marginals exactly for each sub-structure. A good review of the structured mean-field concept has been discussed in [81, 183].



Figure 3.8: *In this example we show how long-range interaction is not always beneficial. For example, some parts of background which are also black in colour prefer to take cat label as well.*

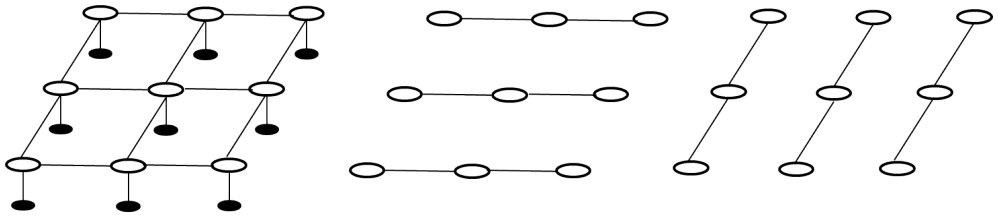


Figure 3.9: *Naive mean-field easily gets stuck in local minima since it assumes that all the variables are independent. In order to improve the quality of the mean-field approximation, many approaches have been proposed. One very interesting approach is that of a structured mean-field method where whole CRF is now represented by a set of tractable subproblems. For example, in the case of grid CRF we can separate the graph along each row, assuming each row is independent and variables within each row are dependent. The true marginals can be calculated on the trees very efficiently.*

3.2.3 Mixture approximations

We next come to another approximate approach to improve the mean-field marginals by introducing a mixture of mean-field distributions to approximate the true distribution [64]. Formally, let us define a mixture distribution as convex combination of the fully factorized models:

$$Q_{mix}(\mathbf{x}) = \sum_{m} \alpha_m Q(\mathbf{x}|m) \tag{3.2.9}$$

where each mixture component $Q(\mathbf{x}|m)$ is a fully factorized distribution and α_m is a component weight such that $\sum_m \alpha_m = 1$ and $\alpha_m > 0, \quad \forall m$.

Update equations: Once we have defined the model, we can derive the update equation by minimizing the KL-divergence with the mixture distribution.

$$
\begin{aligned}
KL(Q_{mix}|\Pr) &= \sum_{\mathbf{x}} Q_{mix}(\mathbf{x}) \log \frac{Q_{mix}}{\Pr} && (3.2.10)\\
&= \sum_{m,\mathbf{x}} \alpha_m \left[Q(\mathbf{x}|m) \log \frac{Q_{mix}}{\Pr} \right] && (3.2.11)\\
&= \sum_{m,\mathbf{x}} \alpha_m \left[Q(\mathbf{x}|m) \log \frac{Q(\mathbf{x}|m)}{\Pr} + Q(\mathbf{x}|m) \log \frac{Q_{mix}}{Q(\mathbf{x}|m)} \right] && (3.2.12)\\
&= \sum_{m} \mathcal{F}(Q(\mathbf{x}|m), \Pr) + \sum_{m,\mathbf{x}} Q(\mathbf{x}|m) \log \frac{Q_{mix}}{Q(\mathbf{x}|m)} && (3.2.13)\\
&= \sum_{m} \mathcal{F}(Q(\mathbf{x}|m), \Pr) + I(m;\mathbf{x}) && (3.2.14)
\end{aligned}
$$

where $I(m;x)$ measures the mutual information between the component and the mixture distribution. The term $I(m;x)$ is always positive which leads to an increase in the KL-divergence values. However, we know that $I(m;x) < \log M$ where M is the number of mixture components. This suggests that the increase in the information gain/KL-divergence is bounded by the number of the components used which helps to improve the naive mean-field approximation. There is however one caveat: since $I(m;x)$ involves summing over all the possible configurations, the computation is in general intractable as there is an exponential number of configurations. It can be generally harder to find the optimum approximating distribution in the mixture distribution than in the factorized distribution, and it may further introduce some more local minima in the KL-divergence.

3.3 Properties of naive mean-field approach

We have discussed various ways to generate approximate marginal distributions using the naive mean-field approach and ways to improve these approximations. We now explain other properties of the mean-field approach and compare it to existing inference methods, especially belief propagation.

3.3.1 Mean-field and belief propagation algorithms

We used the KL-divergence approach to derive mean-field inference. However the mean-field derivation can equivalently be detailed by minimizing a free energy, which will also form the basis for comparing the mean-field and belief propagation based inference approaches.

We saw in the previous discussion that the naive mean-field approach involves minimizing the KL-divergence approximation

$$F(Q,P) = \sum_{ij}\sum_{x_i,x_j} Q_i(x_i)Q_j(x_j)\psi_{ij}(x_i,x_j) + \sum_{i\in\mathcal{V}}\sum_{x_i} Q_i(x_i)\psi_i(x_i) + \sum_{i\in\mathcal{V}}\sum_{x_i} Q_i(x_i)\log Q_i(x_i), \quad (3.3.1)$$

where $F(Q,P)$ is the mean-field free energy and we have assumed that the marginals are independent so that $Q_i(x_i,x_j) = Q(x_i)Q(x_j)$.

In order to compare the mean-field algorithm and belief propagation, we first provide details of the belief propagation (BP) algorithm. BP is another parallel message passing algorithm where each variable sends messages/beliefs to a neighbour based on the messages it received from all other neighbouring variables in the previous iteration. We discuss sum-product based belief propagation, which consists of messages $m_{ij}^{t+1}(x_j)$ from i to j at $t+1$ iteration and has the update rule

$$m_{ij}^{t+1}(x_j) = \sum_{x_i} \exp\{-\psi_{ij}(x_i,x_j) - \psi_i(x_i)\} \prod_{k\neq j} m_{ki}^{t}(x_i). \quad (3.3.2)$$

Using these messages we can generate the unary and pairwise pseudomarginals as:

$$Q_{BP}^{t}(x_i) \propto \exp\{-\psi_i(x_i)\} \prod_{k} m_{kj}^{t}(x_j) \quad (3.3.3)$$

$$Q_{BP}^{t}(x_i,x_j) \propto \exp\{-\psi_{kj}(x_k,x_j) - \psi_k(x_k) - \psi_j(x_j)\}. \quad (3.3.4)$$

It has been shown by Weiss et al. [186] that the belief propagation iteration converges to the fixed point solution of the Bethe free energy, which takes

following form:

$$F_{Bethe}(Q,P) = \sum_{ij}\sum_{x_i,x_j} Q_{ij}(x_i,x_j)\psi_{ij}(x_i,x_j) + \sum_{i\in\mathcal{V}}\sum_{x_i} Q_i(x_i)\psi_i(x_i) + \sum_{i\in\mathcal{V}}\sum_{x_i} Q_i(x_i)\log Q_i(x_i). \quad (3.3.5)$$

Thus BP consists of following minimization problem:

$$\begin{aligned} &\min_{Q_{BP}} \quad F_{Bethe}(Q_{BP},\Pr) \\ &\text{subject to} \quad \sum_i Q_{BP}(x_i) = 1, \\ &\sum_{x_i} Q_{BP}(x_i,x_j) = Q_{BP}(x_j) \end{aligned} \quad (3.3.6)$$

The constraint $\sum_{x_i} Q_{BP}(x_i,x_j) = Q_{BP}(x_j)$ only ensures local consistency between the marginal distribution (approximates the actual marginal $\Pr(x_i) = \sum_{\mathbf{x}/x_i}\Pr(\mathbf{x})$). We show the local (constraint) polytope of the belief propagation algorithm for the feasible parameter set in Fig. 3.11.

There are many interesting relations between the mean-field free energy and the Bethe free energy, which lead to interesting comparisons between the mean-field and belief propagation based inference approaches:

- In general, the belief propagation algorithm does not decrease the energy value. This is in contrast to the mean-field inference approach, where each iteration leads to decrease in mean-field free energy. Only at convergence of belief propagation are local consistency constraints satisfied.
- We see that if the pairwise marginals factorize as $Q_{ij}(x_i,x_j) = Q_i(x_i)Q_j(x_j)$, then two free energies are equivalent and both algorithms attain the same solution.
- If the graph is singly connected, then we will achieve a global optimum of the Bethe free energy but not of the mean-field free energy.
- In general, the mean-field free energy has multiple local minima compared to the Bethe free energy. Thus, though both involve solving non-convex problems, belief propagation in general achieves better optima.
- If the true posterior has one peak, then both these approximate free energies become exact.
- For densely connected graphs the mean-field approach efficiently achieves better marginal distributions. There is no known approach to perform belief

propagation in fully connected models efficiently.

However, there is an interesting benefit of using the mean-field over the Bethe approximation: the mean-field energy can provide a bound on the partition function and is therefore useful for the model selection problem. It should also be noted that both the mean-field and Bethe energies are non-convex. We will cover next the non-convexity of naive mean-field free energy in detail.

In Fig. 3.10, we compare the mean-field free energy and the belief propagation energy for a four-connected pairwise model on an object class segmentation problem. As we can see in this case, the solution based on the mean-field free energy is worse than for the Bethe free energy. For further comparative study between mean-field and belief propagation, refer to [186, 197].

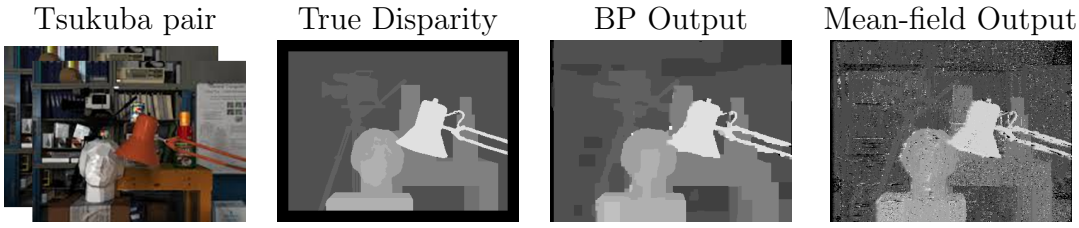


Figure 3.10: *We compare outputs of mean-field and belief propagation methods on stereo disparity estimation problem for a 4-connected pairwise MRF. We observe that belief propagation achieves better qualitative results.*

3.3.2 Non-convexity of naive mean-field

We show now that the naive mean-field problem is a non-convex optimization problem.

For a general optimization problem of the form

$$\begin{aligned} \text{minimize} \quad & f_0(x) && (3.3.7)\\ \text{subject to} \quad & f_i(x) \leq 0, \quad i = 1, ..., m && (3.3.8)\\ & a_i^T x = b_i, \quad i = 1, ..., p && (3.3.9) \end{aligned}$$

to be convex, the functions $f_0, f_1, ..., f_m$ have to be convex and the equality con-

straints must be affine. If we again take the naive mean-field problem:

$$\min_{Q} \quad F(\Pr, Q) \tag{3.3.10}$$

$$\text{subject to} \quad \sum_{x_i} Q(x_i) = 1 \;\; \forall i \in \mathcal{V} \tag{3.3.11}$$

$$Q(\mathbf{x}) = \prod_i Q(x_i), \tag{3.3.12}$$

which corresponds to

$$\min_{Q(\mathbf{x})} \quad \sum_{\mathbf{x}} Q(\mathbf{x}) \left(\sum_{c \in \mathcal{C}} \psi_c(\mathbf{x}_c) \right) + \sum_{\mathbf{x}} Q(\mathbf{x}) \log Q(\mathbf{x}) \tag{3.3.13}$$

$$\text{subject to} \quad \sum_{x_i} Q(x_i) = 1 \;\; \forall i \in \mathcal{V} \tag{3.3.14}$$

$$Q(\mathbf{x}) = \prod_i Q(x_i). \tag{3.3.15}$$

The corresponding mean-field polytope is shown in Fig. 3.11 and the naive mean-field optimization problem is not convex. This leads many significant computational challenges, such as that there are multiple local minima. It also suggests that the mean-field algorithm depends on initialisation.

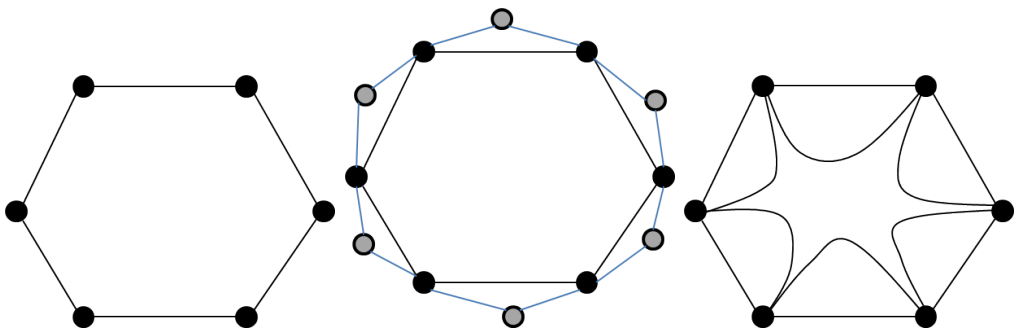


Figure 3.11: *We illustrate here the marginal polytope, local polytope and mean-field polytope (from left to right).*

3.3.3 Estimating MAP solution

We have basically discussed ways to estimate marginal probability using mean-field theory. We now describe ways to estimate the most probable (MAP) state of the probability distribution $\mathbf{x}^* = \arg\max_{\mathbf{x}} \Pr(\mathbf{x})$. We introduce a temperature parameter T, and so generate a family of probability distribution $P(x; T)$. The

corresponding Gibbs distribution is $P(x;T) = \frac{1}{Z(T)} \exp\{-E(x)/T\}$; on setting $T = 1$ we recover the original probability distribution $\Pr(\mathbf{x})$. It can be seen that at lower temperature ($T \to 0$), the distribution becomes peaked about state $x^* = \arg\min_x E(x)$ and at very high temperature ($T \to \infty$) the distribution becomes uniform as all states are equally likely.

Corresponding to this family of probability distributions, the mean-field free energy also involves a temperature parameter as:

$$F_{mf}() = \sum_x Q \log P + T \sum Q \log Q. \tag{3.3.16}$$

As the temperature increases the function becomes more convex, as it is dominated by the convex entropy term, but for small T the remaining terms dominate. Thus for higher temperature it is easier to evaluate the optimal function values, which is not the case when the temperature values are low. One good approach is to calculate the marginal at high temperature quickly and use it to initialise the solution of solving the problem at lower temperature. More discussion can be found in the work of Yuille [197].

In our recent studies we found out that the energy achieved for the max-marginal solutions recovered from fully-connected mean-field is higher than the energy corresponding to the (approximate) MAP solution generated by graph-cuts approach. However, the mean-field achieves higher accuracy than the graph-cuts solution. We show some results on the images taken from MSRC dataset. In both the cases we used the fully-connected pairwise energy functions with same parameters. These results are shown in the Fig. 3.12.

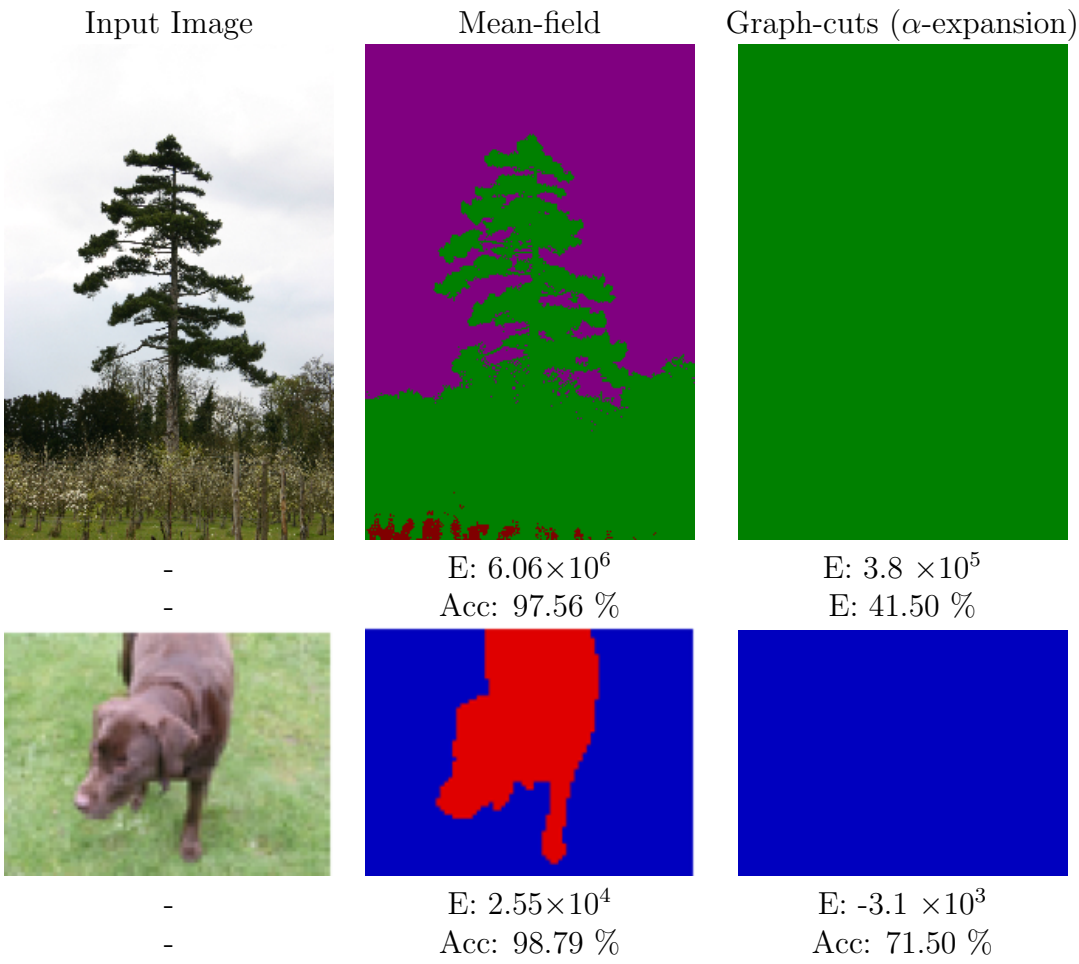


Figure 3.12: *We illustrate here the energies and accuracies achieved by the fully-connected mean-field and graph-cuts approaches. In both the cases, we used the same fully-connected pairwise energy functions. Here E refers to final energy, and Acc. measures the percentage of the correctly labelled pixels.*

Chapter 4

Filter-based Mean-field Inference for Random Fields with Higher Order Terms and Product Label-spaces

Probabilistic models such as Markov Random Field (MRF) and Conditional Random Field (CRF) have long formed a basis for solving challenging assignment problems that are encountered while understanding images and scenes. Computational concerns had limited these models to encode only unary and/or pairwise terms. Recently, a number of cross bilateral filtering methods have been proposed for solving multi-label problems in computer vision, such as stereo, optical flow and object class segmentation that show an order of magnitude improvement in speed over previous methods. These methods have achieved good results despite using models with only unary and/or pairwise terms. However, previous work has shown the value of using models with higher-order terms e.g. to represent label consistency over large regions, or global co-occurrence relations. We show how these higher-order terms can be formulated such that filter-based inference remains possible. We demonstrate our techniques on joint stereo and object labelling problems, as well as object class segmentation, showing in addition for joint object-stereo labelling how our method provides an efficient approach to inference in product label-spaces. We show that we are able to speed up inference in these models around 10-30 times with respect to competing graph-cut/move-making methods, as well as maintaining or improving accuracy in all cases. We show results on PascalVOC-10 for object class segmentation, and Leuven for joint object-stereo labelling.

4.1 Introduction

Many computer vision problems, such as object class segmentation, stereo and optical flow, can be formulated as multi-labelling problems within a Markov Random Field (MRF) or Conditional Random Field (CRF) framework. Although exact inference in such models is in general intractable, much attention has been paid to developing fast approximation algorithms, including variants of belief propagation, dual decomposition methods, and move-making approaches [20, 82, 84]. Recently, a number of cross bilateral Gaussian filter-based methods have been proposed for problems such as object class segmentation [87], denoising [86], stereo and optical flow [136], which permit substantially faster inference in these problems, as well as offering performance gains over competing methods. Our approach builds on such filter-based approaches and shows them to outperform or perform equally well to the previously dominant graph-cut/move-making approaches on all problems considered. This strongly suggests that mean-field message-passing enhanced with recent filtering techniques [87] should be considered as a general state-of-the-art inference method for a large number of computer vision problems currently of interest.

A problem with filter-based methods as currently formulated is that they can only be applied to models with limited types of structure. In [136], dependencies between output labels are abandoned, and the filtering step is used to generate unary costs which are treated independently. In [87], filtering is used to perform inference in MRF models with dense pairwise dependencies taking the form of a weighted mixture of Gaussian kernels. Although allowing fully connected pairwise models increases expressivity over typical 4 or 8-connected MRF models, the inability to handle higher-order terms is a disadvantage.

The importance of higher-order information has been demonstrated in all of the labelling problems mentioned. For object class segmentation, the importance of enforcing label consistency over homogeneous regions has been demonstrated using P^n-Potts models [78], and co-occurrence relations between classes at the image level have also been shown to provide important priors for segmentation [96]. For stereo and optical flow, second-order priors have proved to be effective [188], as have higher-order image priors for denoising [131].

In this chapter, we propose a number of methods by which higher-order information can be incorporated into MRF models for multi-label problems so that, under certain model assumptions, using efficient bilateral filter-based methods for inference remains possible. Specifically, we show how to encode (a) a broad class of local *pattern-based* potentials (as introduced in [83], [138]), which include P^n-Potts models and second-order smoothness priors, and (b) global potentials representing co-occurrence relationships between labels as in [53, 96]. We assume a base-layer MRF with full connectivity and weighted Gaussian edge potentials as in [87]. Our approach allows us to apply bilateral filter-based inference to a wide range of models with complex higher-order structure. We demonstrate the approach on two such models, first a model for joint stereo and object class labelling as in [98], and second a model for object class segmentation with co-occurrence priors as in [96]. In the case of joint stereo and object labelling, in addition to demonstrating fast inference with higher-order terms, we show how cost-volume filtering can be applied in the product label-space to generate informative disparity potentials, and more generally how our method provides an efficient approach to inference in such product label-spaces. Further, we demonstrate the benefits for object-stereo labelling of applying recent *domain transform filtering* techniques [46] in our framework. In both joint stereo-object labelling and object class segmentation, we am able to achieve substantial speed-ups with respect to graph-cut based inference techniques and improvements in accuracy with respect to the baseline methods. In summary, our contributions are:

- A set of efficient techniques for including higher-order terms in random fields with dense connectivity, allowing for mean-field filter-based inference,
- An adaptation of our approach to product label-space models for joint

object-stereo labelling, again permitting efficient inference,

- An investigation of the advantages/disadvantages of alternative filtering methods recently proposed [2, 46, 86] within our framework.

In Sec. 4.2 we review the method of [87]. Sec. 4.3 and Sec. 4.4 provide details on how we encode higher-order terms and product label spaces respectively, Sec. 4.5 gives experimentation on joint stereo and object labelling, and object class segmentation. Finally Sec. 4.6 analyses the mean-field method and Sec. 4.7 concludes with a discussion.

4.2 Filter-based Inference in Dense Pairwise CRFs

We begin by reviewing the approach of [87], which provides a filter-based method for performing fast approximate maximum posterior marginal (MPM) inference[1] in multi-label CRF models with fully connected pairwise terms, where the pairwise terms have the form of a weighted mixture of Gaussian kernels. We define a random field over random variables $\mathcal{X} = \{X_1, ...X_N\}$ conditioned on an image $\mathbf{I}$. We assume there is a random variable associated with each pixel in the image $\mathcal{N} = \{1...N\}$, and the random variables take values from a label set $\mathcal{L} = \{l_1, ..., l_L\}$. We can then express the fully connected pairwise CRF as:

$$P(\mathbf{X}|\mathbf{I}) = \frac{1}{Z(\mathbf{I})} \exp(-E(\mathbf{X}|\mathbf{I})) \tag{4.2.1}$$

$$E(\mathbf{X}|\mathbf{I}) = \sum_{i \in \mathcal{N}} \psi_u(x_i) + \sum_{i<j \in \mathcal{N}} \psi_p(x_i, x_j) \tag{4.2.2}$$

where $E(\mathbf{X}|\mathbf{I})$ is the energy associated with a configuration $\mathbf{X}$ conditioned on $\mathbf{I}$, $Z(\mathbf{I}) = \sum_{\mathbf{X}'} \exp(-E(\mathbf{X}'|\mathbf{I}))$ is the (image dependent) partition function, and $\psi_u(.)$ and $\psi_p(.,.)$ are unary and pairwise potential functions respectively, both implicitly conditioned on the image $\mathbf{I}$. The unary potentials can take arbitrary form, while [87] restrict the pairwise potentials to take the form of a weighted mixture of Gaussian kernels:

$$\psi_p(x_i, x_j) = \mu(x_i, x_j) \sum_{m=1}^{M} w^{(m)} k^{(m)}(\mathbf{f}_i, \mathbf{f}_j) \tag{4.2.3}$$

[1] For exact MPM inference, the solution satisfies $x_i^{MPM} \in \arg\max_l \sum_{\{\mathbf{x}|x_i=l\}} P(\mathbf{x}|I)$.

where $\mu(.,.)$ is an arbitrary *label compatibility function*, while the functions $k^{(m)}(.,.)$, $m = 1...M$ are Gaussian kernels defined on feature vectors $\mathbf{f}_i$, $\mathbf{f}_j$ derived from the image data at locations i and j (where [87] form $\mathbf{f}_i$ by concatenating the intensity values at pixel i with the horizontal and vertical positions of pixel i in the image), and $w^{(m)}$, $m = 1...M$ are used to weight the kernels.

Given this form of CRF, [87] show how fast approximate MPM inference can be performed using cross bilateral filtering techniques within a mean-field approximation framework. The mean-field approximation introduces an alternative distribution over the random variables of the CRF, $Q(\mathbf{X})$, where the marginals are forced to be independent, e.g. $Q(\mathbf{X}) = \prod_i Q_i(x_i)$. The mean-field approximation then attempts to minimize the KL-divergence $\mathbf{D}(Q||P)$ between Q and the true distribution P. By considering the fixed-point equations that must hold at the stationary points of $\mathbf{D}(Q||P)$, the following update may be derived for $Q_i(x_i = l)$ given the settings of $Q_j(x_j)$ for all $j \neq i$ (see [81] for a derivation):

$$Q_i(x_i = l) = \frac{1}{Z_i} \quad \exp\{-\psi_u(x_i) - \sum_{l' \in \mathcal{L}} \sum_{j \neq i} Q_j(x_j = l')\psi_p(x_i, x_j)\} \tag{4.2.4}$$

where $Z_i = \sum_{x_i = l \in \mathcal{L}} \exp\{-\psi_u(x_i) - \sum_{l' \in \mathcal{L}} \sum_{j \neq i} Q_j(x_j = l')\psi_p(x_i, x_j)\}$ is a constant which normalizes the marginal at pixel i. If the updates in Eq. 4.2.4 are made in sequence across pixels $i = 1...N$ (updating and normalizing the L values $Q_i(x_i = l)$, $l = 1...L$ at each step), the KL-divergence is guaranteed to decrease [81]. In [87], it is shown that parallel updates for Eq. 4.2.4 can be evaluated by convolution with a high dimensional Gaussian kernel using any efficient bilateral filter, e.g. the permutohedral lattice method of [2] (which introduces a small approximation). This is achieved by the following transformation:

$$\begin{aligned} \tilde{Q}_i^{(m)}(l) &= \sum_{j \neq i} k^{(m)}(\mathbf{f}_i, \mathbf{f}_j) Q_j(l) \\ &= [G_m \otimes Q(l)](\mathbf{f}_i) - Q_i(l) \end{aligned} \tag{4.2.5}$$

where G_m is a Gaussian kernel corresponding to the m'th component of Eq. 4.2.3, and $\otimes$ is the convolution operator. Since $\sum_{j \neq i} Q_j(x_j = l')\psi_p(x_i, x_j)$ in Eq. 4.2.4 can be written as $\sum_m w^{(m)} \tilde{Q}_i^{(m)}(l')$, and approximate Gaussian convolution using [2] is $O(N)$, parallel[2] updates using Eq. 4.2.4 can be efficiently approximated in $O(MNL^2)$ time (or $O(MNL)$ time for the Potts model), thus avoiding the need for the $O(MN^2L^2)$ calculations which would be required to calculate

[2]Although the updates are conceptually parallel in form, the permutohedral lattice convolution is implemented sequentially.

these updates individually. Since the method requires the updates to be made in parallel rather than in sequence, the convergence guarantees associated with the sequential algorithm are lost [81]. However, [87] observe good convergence properties in practice. The algorithm is run for a fixed number of iterations, and the MPM solution extracted by choosing $x_i \in \arg\max_l Q_i(x_i = l)$ at the final iteration.

Although [87] use the permutohedral lattice [2] for their filter-based inference, we note that other filtering methods can also be used for the convolutions in Eq. 4.2.5. Particularly, the recently proposed *domain transform* filtering approach [46] has certain advantages over the permutohedral lattice. Domain transform filtering approximates high-dimensional filtering, such as 5-D bilateral filtering in 2-D spatial and 3-D RGB range space, by alternating horizontal and vertical 1-D filtering operations on transformed 1-D signals which are isometric to slices of the original signal. Since it does not sub-sample the original signal, its complexity is independent of the filter size, while in [2] the complexity and filter size are inversely related. In Sec. 4.5, we show that for the filter sizes needed for accurate object/stereo labelling, the domain transform approach can allow us to achieve even faster inference times than using [2].

4.3 Inference in Models with Higher-order Terms

We now describe how a number of types of higher-order potential may be incorporated in fully connected models of the kind described in Sec. 4.2, while continuing to permit efficient mean-field updates. The introduction of such higher-order terms not only greatly expands the expressive power of such densely connected models, but also makes efficient filter-based inference possible in a range of models where other techniques are currently used. We show in our experimentation that filter-based inference generally outperforms the best alternative methods in terms of speed and accuracy.

We first give a general form of the models we will be dealing with. In place of Eq. 4.2.2, we consider the general energy:

$$E(\mathbf{V}|\mathbf{I}) = \sum_{c \in \mathcal{C}} \psi_c(\mathbf{v}_c|\mathbf{I}) \tag{4.3.1}$$

where $\mathbf{V}$ is a joint assignment of the random variables $\mathcal{V} = \{V_1, ..., V_{N_V}\}$, $\mathcal{C}$ is a set of cliques each consisting of a subset of random variables $c \subseteq \mathcal{V}$, and associated with a potential function ψ_c over settings of the random variables in c, $\mathbf{v}_c$. In

Sec. 4.2 we have that $\mathcal{V} = \mathcal{X}$, that each X_i takes values in the set $\mathcal{L}$ of object labels, and that $\mathcal{C}$ contains unary and pairwise cliques of the types discussed. In general, in the models discussed below we will have that $\mathcal{X} \subseteq \mathcal{V}$, so that $\mathcal{V}$ may also include other random variables (e.g. latent variables) which may take values in different label sets, and $\mathcal{C}$ may also include higher-order cliques. The general form of the mean-field update equations (see [81]) is:

$$Q_i(v_i = \nu) \quad = \quad \frac{1}{Z_i} \exp\{-\sum_{c \in \mathcal{C}} \sum_{\{\mathbf{v}_c | v_i = \nu\}} Q_{c-i}(\mathbf{v}_{c-i}) \cdot \psi_c(\mathbf{v}_c)\} \tag{4.3.2}$$

where ν is a value in the domain of random variable v_i, $\mathbf{v}_c$ denotes an assignment of all variables in clique c, $\mathbf{v}_{c-i}$ an assignment of all variables apart from V_i, and Q_{c-i} denotes the marginal distribution of all variables in c apart from V_i derived from the joint distribution Q. $Z_i = \sum_\nu \exp\{-\sum_{c \in \mathcal{C}} \sum_{\{\mathbf{v}_c | v_i = \nu\}} Q_{c-i}(\mathbf{v}_{c-i}) \cdot \psi_c(\mathbf{v}_c)\}$ is a normalizing constant for random variable v_i. We note that the summations $\sum_{\{\mathbf{v}_c | v_i = \nu\}} Q_{c-i}(\mathbf{v}_{c-i}) \cdot \psi_c(\mathbf{v}_c)$ in Eq. 4.3.2 evaluate the expected value of ψ_c over Q given that V_i takes the value ν. For a general higher order graph of size $|C|$, the expectation involves summing over $O(L^{|C|})$ possible configurations of the cliques. For example, in Fig. we show all possible configurations for a 4-node clique. The updates for the densely connected pairwise model in Eq. 4.2.4 are derived by evaluating Eq. 4.3.2 across the unary and pairwise potentials defined in Sec. 4.2 for $v_i = x_{1...N}$ and $\nu = 1...L$. We describe below how similar updates can be efficiently calculated for each of the higher-order potentials we consider.

4.3.1 Pattern-based Potentials

In [83], a *pattern-based* potential[3] is defined as:

$$\psi_c^{pat}(\mathbf{x}_c) = \begin{cases} \gamma_{\mathbf{x}_c} & \text{if } \mathbf{x}_c \in \mathcal{P}_c \\ \gamma_{\max} & \text{otherwise} \end{cases} \tag{4.3.3}$$

where $\mathcal{P}_c \subset \mathcal{L}^{|c|}$ is a set of recognized *patterns* (i.e. label configurations for the clique) each associated with an individual cost $\gamma_{\mathbf{x}_c}$, while a common cost $\gamma_{\max}$ is applied to all other patterns. We assume $|\mathcal{P}_c| << L^{|c|}$, since when $|\mathcal{P}_c| \approx L^{|c|}$ the representation approaches an exhaustive parametrization of $\psi_c(\mathbf{x}_c)$.

Given higher-order potentials $\psi_c^{pat}(\mathbf{x}_c)$ of this form, the required expectation

[3]The class of such sparse higher-order potentials is also considered in [138].

for the mean-field updates (Eq. 4.3.2) can be calculated:

$$\begin{aligned}\sum_{\{\mathbf{x}_c|x_i=l\}} Q_{c-i}(\mathbf{x}_{c-i}) \cdot \psi_c^{pat}(\mathbf{x}_c) &= \sum_{p\in\mathcal{P}_{c|i=l}} \Big(\prod_{j\in c, j\neq i} Q_j(x_j = p_j)\Big)\gamma_p \\ &+ \Big(1 - \Big(\sum_{p\in\mathcal{P}_{c|i=l}} \Big(\prod_{j\in c, j\neq i} Q_j(x_j = p_j)\Big)\Big)\Big)\gamma_{\max}\end{aligned} \tag{4.3.4}$$

where we write $\mathcal{P}_{c|i=l}$ for the subset of patterns in $\mathcal{P}_c$ for which $x_i = l$. Since the expectation in Eq. 4.3.4 can be calculated in $O(|\mathcal{P}_c||c|)$ time, such terms contribute $O(\max_c(|\mathcal{P}_c||c|)|\mathcal{C}^{pat}|)$ to each parallel update, where $\mathcal{C}^{pat}$ is the set of pattern-based clique potentials.[4] If we assume each pixel belongs to at most M^{pat} cliques, and each clique has at most $P^{\max}$ patterns, this complexity reduces to $O(M^{pat}NP^{\max})$.

A particular case of the pattern-based potential is the P^n-Potts model [78]:

$$\psi_c^{potts}(\mathbf{x}_c) = \begin{cases}\gamma_l & \text{if } \forall i \in c,\, x_i = l \\ \gamma_{\max} & \text{otherwise}\end{cases} \tag{4.3.5}$$

where implicitly we have set $\mathcal{P}$ to be the L configurations with constant labellings. The required expectations here can be expressed as:

$$\begin{aligned}\sum_{\{\mathbf{x}_c|x_i=l\}} Q_{c-i}(\mathbf{x}_{c-i}) \cdot \psi_c^{potts}(\mathbf{x}_c) &= \Big(\prod_{j\in c, j\neq i} Q_j(x_j = l)\Big)\gamma_l \\ &+ \Big(1 - \Big(\prod_{j\in c, j\neq i} Q_j(x_j = l)\Big)\Big)\gamma_{\max}\end{aligned} \tag{4.3.6}$$

which contribute $O(L\max_c(|c|)|\mathcal{C}^{potts}|)$ to each parallel update. Assuming each pixel belongs to at most M^{pat} cliques, we can reexpress this as $O(M^{pat}NL)$, which effectively preserves the $O(MNL^2)$ complexity of the dense pairwise updates of Sec. 4.2 (assuming $M^{pat} \approx M$), and further preserves the $O(MNL)$ complexity when the pairwise terms also use Potts models. Further potentials which can be cast as pattern-based potentials are discussed in [83], including second-order smoothness priors for stereo, as in [188].

[4] Eq. 4.3.4 requires evaluation of the joint probability of $c-1$ variable assignments for each of the $|\mathcal{P}_c|$ patterns, leading to the complexity $O(|\mathcal{P}_c||c|)$ for a single evaluation. If Q is prevented from taking the values 0 and 1, the joint pattern probabilities $\prod_{j\in c} Q_j(x_j = p_j)$ can be calculated once for each clique, and the conditional forms $\prod_{j\in c, j\neq i} Q_j(x_j = p_j)$ needed for parallel updates can then be derived by dividing by $Q_i(x_i = p_i)$, leading to the overall $O(\max_c(|\mathcal{P}_c||c|)|\mathcal{C}^{pat}|)$ complexity.

4.3.2 Co-occurrence Potentials

Co-occurrence relations capture global information about which classes tend to appear together in an image and which do not, for instance that busses tend to co-occur with cars, but tables do not co-occur with aeroplanes. A recent formulation [96] which has been proposed attempts to capture such information in a global *co-occurrence potential* defined over the entire image clique c_I (generalization to arbitrary cliques is also possible) as:

$$\psi_{c_I}^{cooc}(\mathbf{X}) = C(\Lambda(\mathbf{X})) \tag{4.3.7}$$

Here, $\Lambda(\mathbf{X}) \subseteq \mathcal{L}$ returns the subset of labels present in configuration $\mathbf{X}$, and $C(.) : 2^{\mathcal{L}} \rightarrow \mathbb{R}$ associates a cost with each possible subset. In [96] the restriction is placed on $C(.)$ that it should be non-decreasing with respect to the inclusion relation on $2^{\mathcal{L}}$, i.e. $\Lambda_1, \Lambda_2 \subseteq \mathcal{L}$ and $\Lambda_1 \subseteq \Lambda_2$ implies that $C(\Lambda_1) \leq C(\Lambda_2)$. We will place the further restriction that $C(.)$ can be represented in the form:

$$C(\Lambda) = \sum_{l \in \mathcal{L}} C_l \cdot \Lambda^l + \sum_{l_1, l_2 \in \mathcal{L}} C_{l_1,l_2} \cdot \Lambda^{l_1} \cdot \Lambda^{l_2} \tag{4.3.8}$$

where we write Λ^l for the indicator $[l \in \Lambda]$, where $[.]$ is 1 for a true condition and 0 otherwise. Equivalently, Λ^l is the l'th entry of a binary vector of length $|\mathcal{L}|$ which represents Λ by its set-indicator function, and $C(\Lambda)$ is a second degree polynomial over these vectors. Eq. 4.3.8 is the form of $C(.)$ investigated experimentally in [96], and is shown perform well there on object class segmentation.

We consider below two approximations to Eq. 4.3.7 which give rise to efficient mean-field updates when incorporated in fully connected CRFs as discussed in Sec. 4.2. Both approximations make use of a set of new latent binary variables $\mathcal{Y} = \{Y_1, ..., Y_L\}$, whose intended semantics are that $Y_l = 1$ will indicate that label l is present in a solution, and $Y_l = 0$ that it is absent. As discussed below though, both approximations enforce this only as a soft constraint.

4.3.2.1 Model 1

In the first, we reformulate Eq. 4.3.7 as:

$$\begin{aligned} \psi_{c_I}^{coocA}(\mathbf{X}, \mathbf{Y}) &= C(\{l | Y_l = 1\}) \\ &+ K \cdot \sum_l [Y_l = 1 \wedge (\sum_i [x_i = l]) = 0] \\ &+ K \cdot \sum_l [Y_l = 0 \wedge (\sum_i [x_i = l]) > 0] \end{aligned} \tag{4.3.9}$$

We consider constructing two CRF distributions $P_1(\mathbf{V}_1|\mathbf{I})$ and $P_2(\mathbf{V}_2|\mathbf{I})$ over the variables sets $\mathcal{V}_1 = \mathcal{X}$ and $\mathcal{V}_2 = \{\mathcal{X}, \mathcal{Y}\}$ respectively, where the clique structure is the same in both distributions, except that a potential $\psi_{c_I}^{cooc}$ in P_1 has been replaced by $\psi_{c_I}^{coocA}$ in P_2. If we set $K = \infty$ in Eq. 4.3.9, the marginals across $\mathbf{X}$ in P_2 will match P_1: $P_1(\mathbf{X}|\mathbf{I}) = \sum_{\mathbf{Y}} P_2(\mathbf{X}, \mathbf{Y}|\mathbf{I})$, since the only joint configurations with non-zero probability in P_2 have identical energies. In general this will not be the case; however, for high K, we can expect that these distributions to approximately match, and hence to be able to perform approximate MPM inference using Eq. 4.3.9 in place of Eq. 4.3.7.

With this approximation, the relevant expectations over the latent variables $Y_1, ..., Y_L$ can be calculated as:

$$\sum_{\{\mathbf{V}|Y_l=b\}} Q_{\mathcal{V}-Y_l}(\mathbf{V} - Y_l) \cdot \psi_{c_I}^{coocA}(\mathbf{V}) = \begin{cases} K \cdot (1 - \prod_i (1 - Q_i(x_i = l))) + \kappa & \text{if } b = 0 \\ C_l + \sum_{l' \neq l} Q_{l'}(Y_{l'} = 1) C_{l,l'} & \\ +K \cdot \prod_i (1 - Q_i(x_i = l)) + \kappa & \text{if } b = 1 \end{cases} \tag{4.3.10}$$

leading to the following mean-field updates for the latent variable distributions:

$$\begin{aligned} Q_l(Y_l = 0) &= \frac{1}{Z_l} \exp\{-K \cdot (1 - \prod_i (1 - Q_i(x_i = l)))\} \\ Q_l(Y_l = 1) &= \frac{1}{Z_l} \exp\{-C_l - \sum_{l' \neq l} Q_{l'}(Y_{l'} = 1) C_{l,l'} \\ &\quad -K \cdot \prod_i (1 - Q_i(x_i = l))\} \end{aligned} \tag{4.3.11}$$

where the expectations can be calculated in $O(N + L)$ time. Further, the expectations for variables X_i can be expressed:

$$\begin{aligned} \sum_{\{\mathbf{V}|X_i=l\}} & Q_{\mathcal{V}-X_i}(\mathbf{V} - X_i) \cdot \psi_{c_I}^{coocA}(\mathbf{V}) \\ &= K \cdot Q_l(Y_l = 0) \\ &+K \cdot \sum_{l' \neq l} Q_{l'}(Y_{l'} = 0)(1 - \prod_{j \neq i} (1 - Q_j(x_j = l'))) \\ &+K \cdot \sum_{l' \neq l} Q_{l'}(Y_{l'} = 1) \prod_{j \neq i} (1 - Q_j(x_j = l')) + \kappa \end{aligned} \tag{4.3.12}$$

which require $O(NL)$ time. This would seem to imply a contribution of $O(NL^2)$ for the cooc-1 terms towards the full parallel update. However, by computing the full products $\prod_i(1 - Q_i(x_i = l))$ once for each l, and then dividing by the relevant terms to calculate the partial products in Eq. 4.3.16 (we must ensure Q does not take the extreme values 0 and 1 during updates to do this) a complexity of $O(NL + L^2)$ is achieved.

4.3.2.2 Model 2

An alternative, looser approximation to Eq. 4.3.7 can be given as:

$$\psi_{c_I}^{coocB}(\mathbf{X}, \mathbf{Y}) = C(\{l|Y_l = 1\}) + K \cdot \sum_{i,l}[Y_l = 0 \wedge x_i = l] \tag{4.3.13}$$

using the same latent binary variables $Y_1, ..., Y_L$ introduced in Eq. 4.3.9. Setting $K = \infty$ in Eq. 4.3.13 does not result in matching marginals in the CRF distributions $P_1(\mathbf{V}_1|\mathbf{I})$ and $P_2(\mathbf{V}_2|\mathbf{I})$ (see above) as it did with Eq. 4.3.9. Since the constraint $Y_l = 1 \Rightarrow \sum_i[x_i = l] > 0$ is not enforced by Eq. 4.3.13, the marginalization for a given $\mathbf{X}$ configuration in P_2 will be across all settings of $\mathbf{Y}$ that include $\Lambda(\mathbf{X})$. Since there are more of these for configurations when $|\Lambda(\mathbf{X})|$ is small than when it is large, this will tend to make configurations with smaller label sets more probable, and those with larger label sets less so, thus accentuating the minimum description length (MDL) regularization implicit in the original cost function, $C(\Lambda(\mathbf{X}))$ (see [96]). For large K (i.e. $K \neq \infty$), we can thus expect similar distortions. Thus, for the latent variables Y_l the required expectations are:

$$\sum_{\{\mathbf{V}|Y_l=b\}} Q_{\mathcal{V}-Y_l}(\mathbf{V} - Y_l) \cdot \psi_{c_I}^{coocB}(\mathbf{V}) = \begin{cases} K \cdot \sum_i Q_i(x_i = l) + \kappa & \text{if } b = 0 \\ C_l + \sum_{l' \neq l} Q_{l'}(Y_{l'} = 1) C_{l,l'} + \kappa & \text{if } b = 1 \end{cases} \tag{4.3.14}$$

where we write $\mathbf{V} - Y_l$ for a setting of all random variables $\mathcal{V}$ apart from Y_l (i.e. $\{\mathbf{X}, \mathbf{Y}_{l' \neq l}\}$), $Q_{\mathcal{V}-Y_l}$ for the marginalization of Q across these same variables, $b \in \{0, 1\}$ is a boolean value, and κ is a constant which can be ignored in the mean-field updates since it is common to both settings of Y_l.

Substituting these into Eq. 4.3.2, we have the following latent variable up-

dates:

$$\begin{aligned} Q_l(Y_l = 0) &= \frac{1}{Z_l} \exp\{-K \cdot \sum_i Q_i(x_i = l)\} \\ Q_l(Y_l = 1) &= \frac{1}{Z_l} \exp\{-C_l - \sum_{l' \neq l} Q_{l'}(Y_{l'} = 1) C_{l,l'}\} \end{aligned} \tag{4.3.15}$$

For the variables X_i, we have the expectations:

$$\sum_{\{\mathbf{V}|X_i=l\}} Q_{\mathcal{V}-X_i}(\mathbf{V} - X_i) \cdot \psi_{c_I}^{coocB}(\mathbf{V}) = K \cdot Q_l(Y_l = 0) + \kappa \tag{4.3.16}$$

where κ is again a common constant. Here $\mathcal{V} - X_i$ is assignment of all variables apart from i^{th} variable. Evaluation of each expectation in Eq. 4.3.15 requires $O(N + L)$ time, while each expectation in Eq. 4.3.16 is $O(1)$. The overall contribution to the complexity of parallel updates for $\psi_{c_I}^{coocB}$ is thus $O(NL+L^2)$, as can also be shown for $\psi_{c_I}^{coocA}$. This does not increase on the complexity of $O(MNL^2)$ for fully connected pairwise updates as in Sec. 4.2.

4.4 Inference in Models with Product Label Spaces

Now we discuss how we provide an efficient inference method for jointly estimating per-pixel object class and disparity labels. Before going into the details of the joint inference, we briefly describe the specific forms of the energy functions we use, which are based on the model of Ladicky et.al. [98] for joint object and stereo labelling.

For object class segmentation, we define a CRF defined over a set of random variables $\mathcal{X} = \{X_1...X_N\}$ ranging over pixels $i = 1...N$ in image $\mathbf{I}_1$, where X_i takes values in $\mathcal{L} = \{1...L\}$ representing the object present at each pixel. The energy function for the object variables includes the unary, pairwise and higher order terms as described in Sec. 4.3 as follows:

$$E^O(\mathbf{x}) = \sum_i \psi_u^O(x_i) + \sum_{ij} \psi_p^O(x_i, x_j) + \sum_c \psi_c^O(\mathbf{x}_c|\mathbf{I}) \tag{4.4.1}$$

Similarly, we express the stereo CRF by a set of variables $\mathcal{U} = \{U_1...U_N\}$ ranging over pixels $i = 1...N$ in the image $\mathbf{I}_1$ and each random variable U_i takes a label in $\mathcal{D} = \{1...D\}$ representing the disparity between pixel i in $\mathbf{I}_1$ at a fixed resolution,

and a proposed match in $\mathbf{I}_2$. We define a multiclass CRF framework for disparity labels using the unary and pairwise energy function as:

$$E^D(\mathbf{u}) = \sum_i \psi_u^D(u_i) + \sum_{ij} \psi_p^D(u_i, u_j) \tag{4.4.2}$$

4.4.1 Joint Formulation for Object and Stereo Labelling

Now we describe our model for jointly estimating per-pixel object and stereo labels. In this model, we define a CRF over two sets of variables $\mathcal{V} = \{\mathcal{X}, \mathcal{U}\}$ conditioned on the images, $P(\mathbf{V}|\mathbf{I}_1, \mathbf{I}_2)$. Each random variable $V_i = [X_i, U_i]$ takes a label $v_i = [x_i, u_i]$ from the product label space of object and stereo labels $\mathcal{L} \times \mathcal{D}$ corresponding to the variable V_i taking object label x_i and disparity label u_i. In this framework, we define our joint energy function as:

$$E^J(\mathbf{v}) = \sum_i \psi_u^J(v_i) + \sum_{ij} \psi_p^J(v_i, v_j) + \sum_c \psi_c^J(\mathbf{v}_c|\mathbf{I}) \tag{4.4.3}$$

where ψ_u^J and ψ_p^J are the joint unary and pairwise terms. We represent the joint unary potential as sum of the object and disparity unary terms, and a connecting pairwise term as:

$$\psi_u^J(v_i) = \psi_u^O(x_i) + \psi_u^D(u_i) + \psi_p(x_i = l, u_i = d) \tag{4.4.4}$$

As discussed, for our mean-field model we replace the 8-connected pairwise structure on $\mathcal{X}$ and $\mathcal{U}$ with dense connectivity. We disregard the joint pairwise term over the product space $\psi_p(x_i = l_1, u_i = d_1, x_j = l_2, u_j = d_2)$ proposed in [98]. Further, we define a set of P^n-Potts higher order potentials over $\mathcal{X}$, as described in Sec. 4.3.

4.4.2 Mean-field Updates

Within this model, the mean-field updates for the object variables, $Q_i^O(x_i = l)$ are calculated as in Eq. 4.2.4, with additional terms for the P^n-Potts model expectation (Eq. 4.3.6) and pairwise expectations for the joint potentials $\psi_p(x_i, u_i)$

as follows:

$$\begin{aligned} Q_i^O(x_i = l) &= \frac{1}{Z_i} \exp\{-\psi_u^O(x_i) - \sum_{l' \in \mathcal{L}} \sum_{j \neq i} Q_j^O(x_j = l')\psi_p^O(x_i, x_j) \\ &\quad - \sum_{\mathbf{x}_c | x_i = l} Q_{c-i}^O(\mathbf{x}_{c-i}).\psi_c^{potts}(\mathbf{x}_c) - \sum_{d' \in \mathcal{D}} Q_i^D(u_i = d').\psi_p(x_i, u_i)\} \end{aligned} \tag{4.4.5}$$

The updates for $Q_i^D(u_i = d)$ are similar, but without higher-order terms, take following form:

$$\begin{aligned} Q_i^D(u_i = d) &= \frac{1}{Z_i} \exp\{-\psi_u^D(u_i) - \sum_{d' \in \mathcal{D}} \sum_{j \neq i}^{D} Q_j(u_j = d')\psi_p^D(u_i, u_j) \\ &\quad - \sum_{l' \in \mathcal{L}} Q_i^O(x_i = l').\psi_p(u_i, x_i)\} \end{aligned} \tag{4.4.6}$$

4.4.3 Cost Volume Filtering

In addition to the model as described above, we also investigate an approach to updating the unary potentials for the disparity variables based on the cost-volume filtering framework of [136]. This approach involves building a cost-volume of labels, performing edge-preserving filtering in each of the label slices, and then finally estimating the per-pixel labels based on winner-take all label selection strategy. They achieve good speed-ups without loosing much accuracy on challenging problems such as stereo correspondence and optical flow. We leverage cost-volume filtering techniques to improve our stereo unary potentials by extending this work to operate in the product label space $L \times D$. First, we define a CRF at each of the disparity label slices $d \in \mathcal{D} = \{1...D\}$ in the cost volume including variables by $\mathcal{V}^d = \{V_1^d ... V_N^d\}$, where each variable V_i^d takes a disparity label d and object labels in $\mathcal{L} = \{1...L\}$. The energy function at each of the disparity label slice in the cost volume takes following form:

$$\begin{aligned} E^d(v^d) &= \sum_i \psi_u^O(x_i = l) + \sum_i \psi_u^D(u_i = d) \\ &\quad + \sum_i \psi_p(x_i = l, u_i = d) + \sum_{ij} \psi_p^O(x_i, x_j) \end{aligned} \tag{4.4.7}$$

We then introduce mean-field distributions $Q_i^t(l, d)$, which represent the probability of assigning pair of object-disparity combination at pixel i over a series of

update steps $t = 0...T$. These updates take following form:

$$Q_i^{t+1}(l,d) = \frac{1}{Z_i}\exp\{-\psi_u^O(x_i = l) - \psi_u^D(u_i = d) \\ -\psi_p(x_i = l, u_i = d) - \sum_{l' \in \mathcal{L}} \sum_{j \neq i} Q_j^t(l,d) \cdot \psi_p^O(x_i, x_j)\} \quad (4.4.8)$$

Further, we set $Q_i^0(l,d) = 1/L$ for all i, l, d. At each step, we can derive costs $\lambda_i^t(l,d)$ for each object-disparity assignment to the pixel i which takes the form as:

$$\lambda_i^t(l,d) = -\log(Q_i^{t+1}(l,d)) \quad (4.4.9)$$

We update $Q_i^t(l,d)$ and $\lambda_i^t(l,d)$ at each iteration via independent mean-field updates across the D cost-volumes $\lambda(.,d)$, $d = 1...D$, using the same kernel and label compatibility function settings as described above. The final output costs are then given by $\lambda_i^T(l,d)$. We form enhanced disparity unary potentials for the full model by adding the maximum across the output costs to the original potential output: $\psi_u'^D(u_i = d) = \max_l \lambda_i^T(l,d) + \psi_u^D(u_i = d)$.

4.5 Experiments

We demonstrate our approach on two labelling problems including higher-order potentials, joint object-stereo labelling and object class segmentation, adapting models which have been proposed independently. Details of the experimental set-up and results are provided below. In all experiments, timings are based on code run on an Intel(R) Xeon(R) 3.33 GHz processor, and we fix the number of full mean-field update iterations to 5 for all models.

4.5.1 Implementation Details

The parameters of the model are set as follows. As in [98], for the joint object-stereo model we use JointBoost classifier responses to form the object unary potentials $\psi_u^O(x_i = l)$ [175]. A truncated l_2-norm of the intensity differences is used to form the disparity potentials $\psi_u^D(u_i = d)$ (using the interpolation technique described in [20]), while the potentials $\psi_p(x_i = l, u_i = d)$ are set according to the observed distributions of object heights in the training set (see [98] for details). For Pascal VOC-10 dataset, we use the unary potentials provided by [87]. Further, for both of these datasets, we use densely connected pairwise terms where

we use kernels and weightings identical to [87] and an Ising model for the label compatibility function, $\mu(l_1, l_2) = [l_1 \neq l_2]$.

For P^n-Potts higher-order potentials over $\mathcal{X}$ for the joint object-stereo problem, as described in Sec. 4.3, we first run meanshift segmentation [30] over image $\mathbf{I}_1$ at a fixed resolution, and create a clique c from the variables X_i falling within each segment returned by the algorithm. However, on the PascalVOC dataset, we generate a set of 10 layers of segments where each layer corresponds to one application of unsupervised segmentation with different parameters of mean-shift and KMeans algorithms. This way of generating multiple segments have been found to be useful in dealing with the complex object boundaries [95]. Once we have generated these higher order cliques, we train the higher-order potentials in a piecewise manner. We first train a classifier using Jointboost [175] to classify the segments associated with the P^n-Potts cliques, and set the parameters γ_l in Eq. 4.3.5 to be the negative log of the classifier output probabilities, truncated to a fixed value $\gamma_{\max}$ set by cross validation. An additional set of P^n-Potts potentials is also included based on segments returned by grabcut initialized to the bounding boxes returned from detectors trained on each of the L classes (see [96]).

A co-occurrence potential is also included for the PascalVOC dataset, which takes the form of either ψ^{coocA} or ψ^{coocB} as in Sec. 4.3. The parameters of the co-occurrence cost Eq. 4.3.8 are set as in [96], by fitting a second-degree polynomial to the negative logs of the observed frequencies of each subset of labels L occurring in the training data. Finally, individual weights on the potentials are set by cross-validation.

4.5.2 Joint Object and Stereo Labelling

We evaluate the efficiency offered by our mean-field update for joint object-stereo estimation to the Leuven dataset [98]. The dataset consists of stereo images of street scenes, with ground truth labelling for 7 object classes, and manually annotated ground truth stereo labellings quantized into 100 disparity labels. We use identical training and test sets to [98].

We compare results from the following methods. As our baseline, we use the method of [98], whose CRF structure is similar to ours, but without dense connectivity over $\mathcal{X}$, and with a truncated L_1-prior on the disparity labels $\mathcal{U}$. Inference is performed by alternating α-expansion on $\mathcal{X}$ with range moves on $\mathcal{U}$ (forming *projected moves*, see [98]). Since the speed and accuracy are affected by the size of range moves considered, we test 3 settings of the range parameter, corresponding to moves to disparity values $d \pm 1$, $d \pm 2$ and $d \pm 3$, for a fixed d at each iteration (see [92]). We also consider a baseline based on the extended cost-volume filtering

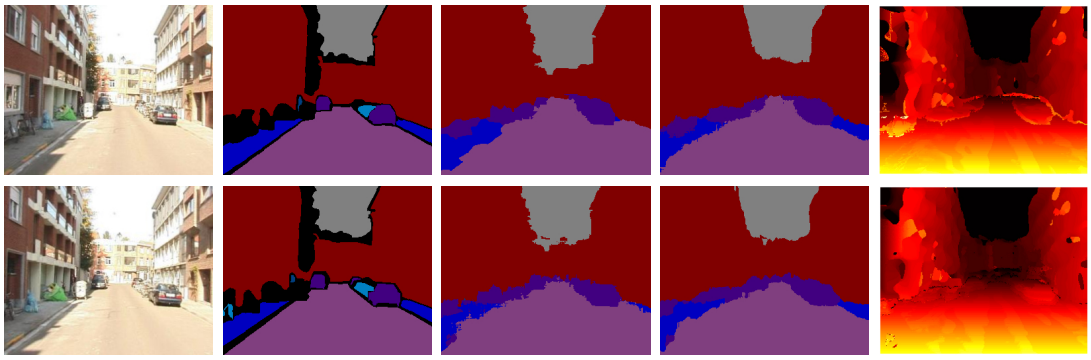


Figure 4.1: *Qualitative results on Leuven dataset. From left to right: input image, ground truth, object labelling from [98] (using graph-cut + range-moves for inference), object labelling and stereo outputs from our dense CRF with higher-order terms and extended cost-volume filtering (see text).*

approach outlined above where we simply select $(x_i, u_i) = \arg\max_{(l,d)} \lambda_i^T(l,d)$ as output. We compare these with our basic higher-order model with full connectivity as described above, and our model combined with extended cost-volume filtered disparity unary terms ψ'_u as described in Sec. 4.4. Further, using our basic model we compare two alternative filtering methods for inference, the first using the permutohedral lattice, as in [2, 87], and the second using the domain transform based filtering method of [46]. We evaluate the average time for the joint inference for object and stereo estimation. Further we evaluate the overall percentage of pixels correctly labelled, the average recall and intersection/union score per class (defined in terms of the true/false positives/negatives for a given class as TP/(TP+FP+FN)) over non *void* pixels. For dense stereo reconstruction, we measure the number of pixels satisfying $\|d_i - d_i^g\| \leq \delta$, where d_i is the disparity label for i^{th} pixel, d_i^g is its corresponding ground truth label and δ is the allowed error. It means a disparity is considered correct if it is within δ pixels of the ground truth.

In Tab. 4.1 we compare the %-correct pixels for object and stereo labelling for different values of the allowed error δ. Further, we also show the average recall and intersection/union (I/U) scores for object labelling in Tab. 4.2. We note that the densely connected CRF with higher-order terms (Dense+HO) achieves comparable accuracies to [98], and that the use of domain transform filtering methods [46] permits an extra speed up, with inference being almost 12 times faster than the least accurate setting of [98], and over 35 times faster than the most accurate. The extended cost-volume filtering baseline described above also performs comparably well, and at a small extra cost in speed, the combined approach (Dense+HO+CostVol) achieves the best overall stereo accuracies. We note that although the improved stereo performance appears to generate a small

Algorithm	Time (sec.)	Object (% corr)	Stereo(1) (% corr)	Stereo(2) (% corr)	Stereo(3) (% corr)
GC+Range(1) [98]	24.6	95.94	43.45	56.67	65.44
GC+Range(2) [98]	49.9	95.94	44.12	56.98	65.84
GC+Range(3) [98]	74.4	95.94	44.14	57.06	65.94
Extended CostVol ([2] filter)	4.2	95.20	43.53	56.44	65.51
Dense+HO ([2] filter)	3.1	95.24	43.58	56.18	65.89
Dense+HO ([46] filter)	2.1	95.06	43.65	56.11	65.47
Dense+HO+CostVol ([46] filter)	6.3	94.98	43.21	56.54	66.07

Table 4.1: *Quantitative comparison on Leuven dataset. The table compares the average time per image and performance (Object and Stereo(δ) labelling accuracy) of joint object and stereo labelling algorithms. δ corresponds to the allowed error such that the disparity for i^{th} pixel is considered correct if it satisfies $\|d_i - d_i^g\| \leq \delta$ where d_i and d_i^g are the disparity label for i^{th} pixel and its corresponding ground truth label respectively. We compare following approaches: graph-cut + range-moves (GC+Range(x), where range moves to disparity values $d \pm x$ are allowed for fixed d at each iteration) [98], an extension of cost-volume filtering (see text), and our dense CRF with higher-order terms and filter-based inference (with and without cost-volume filtered unaries, and using different filtering approaches, see text). Our Dense+HO approach achieves comparable accuracies to [98], and is an order of magnitude faster. The best stereo accuracies occur when our model is combined with cost-volume filtered unaries for disparity. Here '% corr' corresponds to the total proportion of correctly labelled pixels.*

decrease in the object labelling accuracy in our full model, the former remains at an almost saturated level, and the small drop could possibly be recovered through further tuning or weight learning. Some qualitative results are shown in Fig. 4.1.

4.5.3 Object Class Segmentation

We also test our approach on object class segmentation, adapting the Associative Hierarchical CRF (AHCRF) model with a co-occurrence potential proposed in [96]. We compare both the timing and performance of four algorithms. As our two baselines, we take the AHCRF with a co-occurrence potential [96], whose model includes all higher-order terms but is not densely connected and uses α-expansion based inference, and the dense CRF [87], which uses filter-based inference but does not include higher-order terms. We compare these with our approach, which adds first P^n-Potts terms to the dense CRF, and then P^n-Potts and co-occurrence terms. We use the permutohedral lattice for filtering in all

Algorithm	Time (sec)	Oveall (% corr)	Av. Recall	Av. I/U
GC+Range(1) [98]	24.6	95.94	72.79	68.72
GC+Range(2) [98]	49.9	95.94	72.79	68.72
GC+Range(3) [98]	74.4	95.94	72.79	68.72
Extended CostVol ([2] filter)	4.2	95.20	70.43	65.69
Dense+HO ([2] filter)	3.1	95.24	70.83	66.08
Dense+HO ([46] filter)	2.1	95.06	70.62	65.75
Dense+HO+CostVol ([46] filter)	6.3	94.98	70.60	65.63

Table 4.2: *Quantitative comparison on Leuven dataset. The table compares the average time per image and performance in terms of '% correct', average recall and intersection-union scores for object labelling task of our joint object and stereo labelling algorithms, using graph-cut + range-moves (GC+Range(x), where range moves to disparity values $d \pm x$ are allowed for fixed d at each iteration) [98], an extension of cost-volume filtering (see text), and our dense CRF with higher-order terms and filter-based inference (with and without cost-volume filtered unaries, and using different filtering approaches, see text). Our Dense+HO approach achieves comparable accuracies to [98], and is an order of magnitude faster. Here '% correct' measure corresponds to the total proportional of correctly labelled pixels, per class recall measure is defined as $\frac{TP}{TP+FN}$ and intersection vs. union (I/U) measure is defined as $\frac{TP}{TP+FN+FP}$.*

models. We assess the overall percentage of pixels correctly labelled, the average recall and intersection/union score per class (defined in terms of the true/false positives/ negatives for a given class as TP/(TP+FP+FN)).

Qualitative and quantitative results are shown in Fig. 4.2 and Tab. 4.3 respectively. As shown, our approach is able to outperform both of the baseline methods in terms of the class-average metrics, while also reducing the inference time with respect to the AHCRF with a co-occurrence potential almost by a factor of 9. Additional per-class quantitative results for object-class segmentation on Pascal-VOC-10 are given. We compare the performance of the AHCRF model with co-occurrence potentials of [96] with our full model, i.e. a Dense-CRF model with higher-order Potts and co-occurrence potentials, using per-class intersection/union scores. As shown, there is an almost 1.5% improvement in the average score across classes. We do well on some of the difficult classes such as cycle, dinning-table and motor-bike where the relative improvement is almost 6-10% against [96]. We also improve on many classes which had high scores like sheep, train, aeroplane, and see a slight dip in certain classes, e.g. boat, person, TV. Since [96] includes similar higher-order potentials to ours, the improved performance of our model can be attributed to its dense connectivity and/or our use

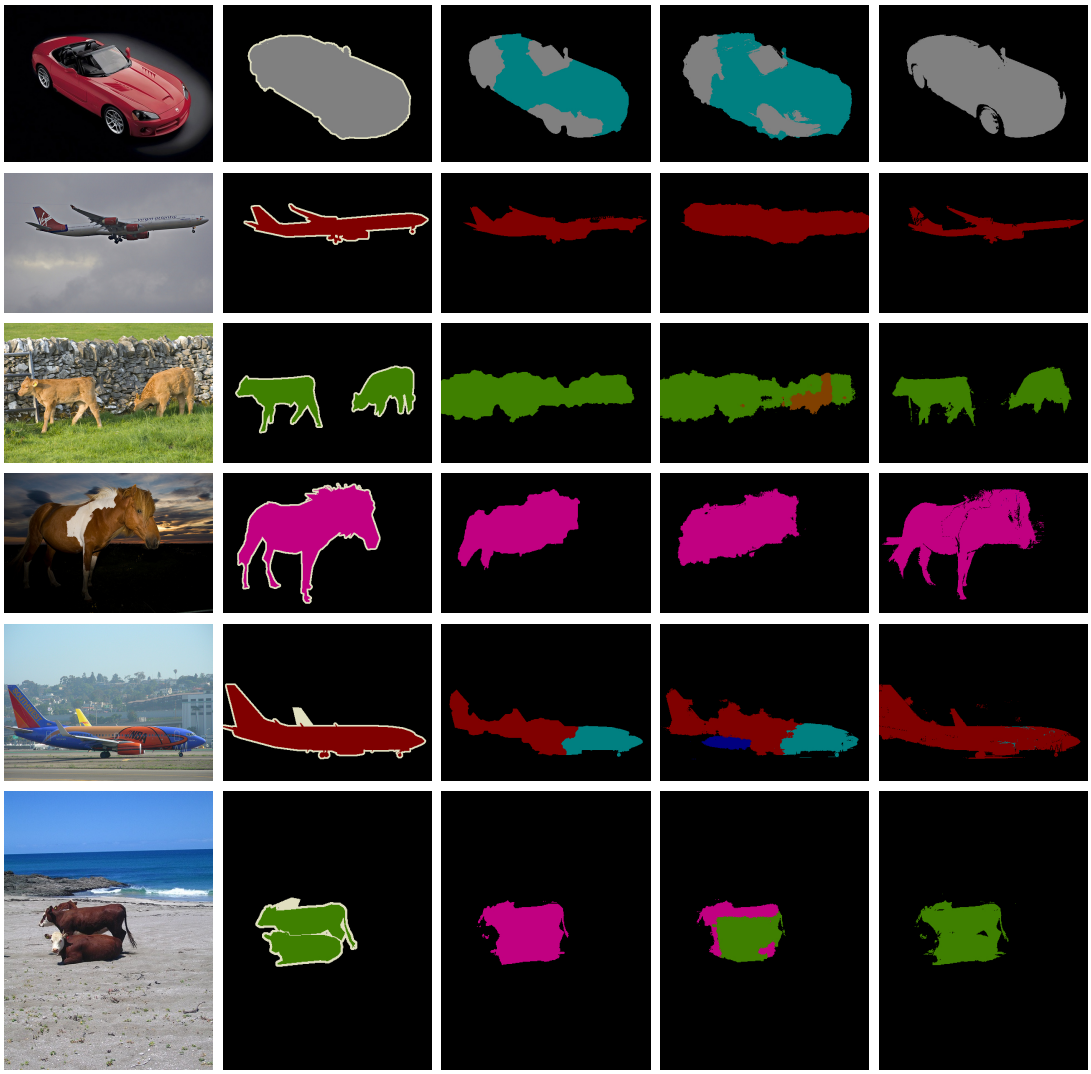


Figure 4.2: *Qualitative results on PascalVOC-10 dataset. From left to right: input image, ground truth, output from [96] (AHCRF+Cooccurrence), output from [87] (Dense CRF), output from our dense CRF with Potts and Co-occurrence terms.*

of mean-field filter-based inference as opposed to graph-cuts (see below Sec. 4.6).

The results shown are only for our approach with the ψ^{coocB} potential, since we found the ψ^{coocA} potential to suffer from poor convergence properties, with performance only marginally better than [87]. We note that our aim here is to assess the relative performance of our approach with respect to our baseline methods, and we expect that our model will need further refinement to compete with the current state-of-the-art on Pascal (our results are $\sim 9\%$ lower for average intersection/union compared to the highest performing method on the 2011 challenge, see [40]). We also note that [87] are able to further improve their average intersection/union score to 30.2% by learning the pairwise label compatibility function, which remains a possibility for our model also.

Algorithm	Time (s)	Overall (%-corr)	Av. Recall	Av. I/U
AHCRF+Cooc [96]	36	81.43	38.01	30.9
DenseCRF [87]	0.67	80.39	35.47	28.44
Dense+Potts	4.35	80.13	40.49	30.27
Dense+Potts+Det	4.35	80.14	44.42	32.66
Dense+Potts+Cooc	4.4	80.52	44.46	33.19

Table 4.3: *Quantitative results on PascalVOC-10. The table compares timing and performance of our approach (final 2 lines) against two baselines. The importance of higher-order information is confirmed by the better performance of all algorithms compared to the basic dense CRF of [87]. Further, our filter-based inference is both able to improve substantially on the inference time and class-average performance of the AHCRF [96], with P^n-Potts and co-occurrence potentials each giving notable gains. Here '% correct' measure corresponds to the total proportional of correctly labelled pixels, per class recall measure is defined as $\frac{TP}{TP+FN}$ and intersection vs. union (I/U) measure is defined as TP/TP+FN+FP.*

4.6 Mean-field Analysis

4.6.1 Mean-Field Vs. Graph-cuts Inference

The results shows that the mean-field methods perform equally well or outperform graph-cut methods on all problems we consider. Since the mean-field methods allow us to perform inference in densely connected CRF models, while we restrict attention to models with 8-connected pairwise terms for graph-cuts (with/without higher-order terms in both cases), the question arises as to whether the performance gains are due to the models used or the optimization technique (or both). To investigate this, we rerun our object-class segmentation experiments on PascalVOC-10 using mean-field and graph-cuts (α-expansion [20]) inference in CRF models with matching forms of pairwise potential based on Gaussian kernels as in [87], using as default standard deviations of 40 and 6 for the spatial and range kernels respectively. Since it is infeasible in terms of time to run α-expansion on a fully connected model, we run it on graphs with gradually increased connectivity, where for a neighbourhood size n, we have that each pixel is connected to all others whose x and y positions differ from it by no more than n (for $n = 1$ this is 8-connectivity). Some qualitative results on increasing the neighbourhood size for α-expansion are shown in Fig. 4.4. For mean-field inference, we use full connectivity throughout. We compare models with pairwise

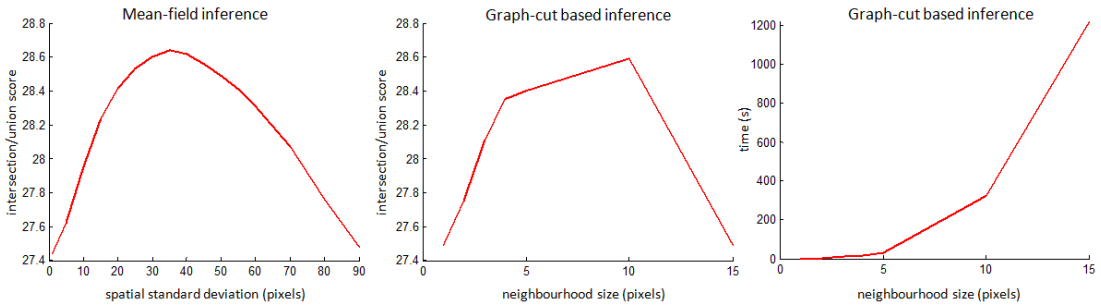


Figure 4.3: *Comparison of inference algorithms on PascalVOC-10 using matched energies with pairwise terms only. The left plot shows the performance of mean-field inference as the spatial standard deviation of the Gaussian pairwise term is varied. The centre plot shows the performance of graph-cut inference (α-expansion) as the pairwise neighbourhood size is varied (maintaining a constant spatial standard deviation of 40 pixels). On the right are shown the inference times per image associated with the centre plot. The inference time for all mean-field settings is $\sim 0.7s$.*

terms only, pairwise with P^n-Potts higher-order potentials, and pairwise with P^n-Potts and co-occurrence terms.[5] For graph-cuts inference, we begin with $n = 1$, and test $n = 1, 2, 3, 4, 5, 10, 15$, stopping when the intersection/union score ceases to increase (/does not increase). We also test mean-field inference on pairwise only models with varying kernel standard deviations, for the spatial kernel setting $\sigma_s = 1, 5, 10, 15...70, 80, 90$ pixels, and the range kernel $\sigma_r = 1, 2, 3, 4...15, 18, 20$.

In Fig. 4.3 we compare performance of the inference methods on the model with pairwise terms only. From the left plot, we see that the best results achieved on the dense model by mean-field occur when the spatial standard deviation is around $\sim$ 40 pixels (and corresponding range standard deviation $\sim$ 6). These are the kernel parameters we use with graph-cuts in all models. The central plot shows that graph-cuts is able to achieve approximately the same performance with a neighbourhood connectivity $n = 10$. This seems to indicate that for pairwise only models, increased connectivity leads to improved performance up to a point, and both graph-cuts and mean-field inference are able to achieve similar results in such models in terms of accuracy. However, as shown on the right plot, substantially longer inference times are needed for graph-cuts at the required connectivity to equal the accuracy of mean-field methods (where the inference time remains around $\sim$ 0.7s for all settings).

[5] In fact we use slightly different co-occurrence potentials with graph-cuts and mean-field, since for graph-cuts we use ψ^{cooc} while for mean-field we use ψ^{coocB}, although we set the costs $C(\Lambda)$ identically. We view the latter as an approximation of the former, and thus view this as a slight handicap for mean-field inference; however, further experiments would be needed to determine if the different forms of this potential lead to better/worse models.

Algorithm	Model	Time (s)	Av. I/U
α-exp (n=10)	Pairwise	326.17	28.59
Mean-field	Pairwise	0.67	28.64
α-exp (n=3)	Pairwise+Potts	56.8	29.6
Mean-field	Pairwise+Potts	4.35	30.11
α-exp (n=1)	Pairwise+Potts+Cooc	103.94	30.45
Mean-field	Pairwise+Potts+Cooc	4.4	32.17

Table 4.4: *Comparison of inference algorithms on PascalVOC-10 using matched energies with pairwise, pairwise and P^n-Potts higher-order potentials, and pairwise, P^n-Potts and co-occurrence potentials. For the α-expansion results, we fix the standard deviation of the Gaussian kernels to the same values as for mean-field (spatial deviation σ_s=40 pixels, range deviation σ_r=6), and optimize over the pairwise neighbourhood size n, where n denotes that each pixel is connected to all others with horizontal/vertial offsets of up to n pixels. Shown are intersection vs. union (I/U) measure defined as $\frac{TP}{TP+FN+FP}$ averaged across all the classes.*

Results in Tab. 4.4 compare the performance of both algorithms on models with higher-order terms, and dense connectivity of various neighbourhood sizes for graph-cut inference, where we quote only the setting at which the optimal accuracy is achieved using the protocol described above. The intersection/union scores quoted here are similar to the one in the Tab. 4.3 for some settings, but with slight differences caused by the fact that we are ensuring that the potentials in all models take matching forms so that the contributions of model and inference method can be separated. As shown, although both mean-field and graph-cuts inference are able to achieve similar accuracies with dense models using pairwise terms only, when higher-order terms are added the α-expansion accuracies are consistently lower than mean-field, even when we allow the former to use models with larger neighbourhood sizes (in fact, for the full model with P^n-Potts and co-occurrence terms, nothing is gained by running graph-cuts with neighbourhood sizes of $n > 1$ as shown). These results imply that, unlike the pairwise only case, when such higher-order terms are included not only is mean-field inference faster than graph-cuts, but it is able to optimize these energies substantially better in terms of accuracy than graph-cuts. We thus claim that the performance gains we observe in the experiments are due to both the densely connectivity of the models we use, and the mean-field techniques we use to optimize these models.

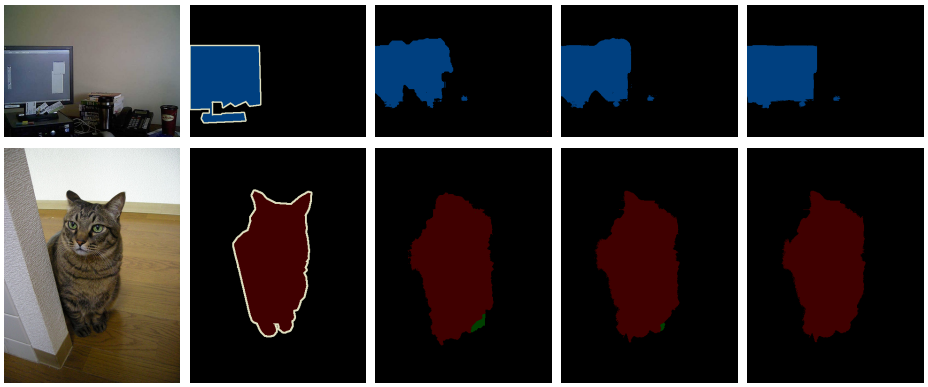


Figure 4.4: *Qualitative improvement in α-expansion output [20] on gradually increasing neighbourhood sizes for each pixel. From left to righ: input image, ground truth, α-expansion output with 8, 24 and 48 neighbours respectively.*

Algorithm	Time (s)	Overall (%-corr)	Av. Recall	Av. U/I
Ours (U+ dense P)	0.67	80.39	35.47	28.44
Ours (U+ dense P+Init)	5.9	79.65	41.84	30.95
Ours (U+ dense P+HO)	4.4	80.52	44.46	33.19
Ours (U+ dense P+HO+Init)	9.7	80.65	44.8	33.9

Table 4.5: *Quantitative results on PascalVOC-10 before and after better initialization. Though the improvement is significant with unary and pairwise terms, we observe slight improvement in accuracy after inclusion of higher order terms and better initialization compared to the model with higher order terms. Here '% corr' measure corresponds to the total proportional of correctly labelled pixels, per class recall measure is defined as $\frac{TP}{TP+FN}$ and intersection vs. union (I/U) measure is defined as $\frac{TP}{TP+FN+FP}$.*

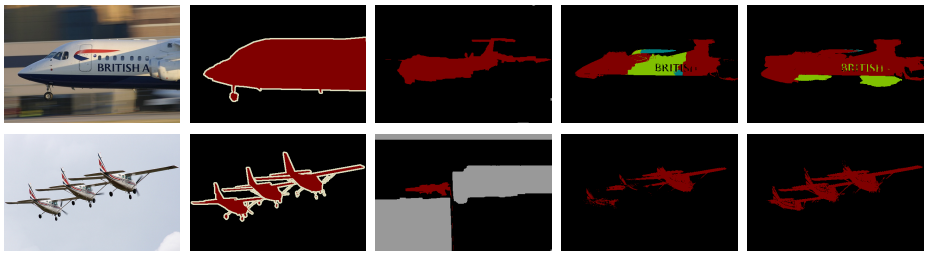


Figure 4.5: *Qualitative results on PascalVOC-10 before and after better initialization. From left to right: input image, ground truth, warped ground truth of the nearest neighbour, output from our dense CRF without better initialisation, and with better initialization.*

4.6.2 Sensitivity to Initialization

It is also worth noting that the mean-field inference methods are sensitive to initialization and can thus get stuck in local minima [186]. Thus, estimating a

good starting point is critical to the mean-field methods. Here, we show how SIFT-flow based label transfer method can be used in providing a good starting point based on the work of Ce Liu et.al. [110], [110]. Suppose we have a large training set of annotated ground truth images with per pixel class labels. Given a test image, we first find the K-nearest neighbour images from the training set using GIST features [123]. In general, we restrict our set to 30 nearest neighbours. We then compute a dense correspondence using the SIFT-flow method from the test image to each of 30 nearest neighbours. We re-rank those nearest neighbours based on the flow values, and pick the best nearest neighbour. Once we have recovered our best candidate, we warp the corresponding ground truth of the candidate image to the current test image. We use these warped labels to initialize the mean-field inference method which acts as a soft constraint on our solutions. We re-weight the unary potential of each pixel based on the label transferred as $\tilde{\psi}_u(x_i) = \lambda * \psi_u(x_i)$, where λ is set through cross-validation. We perform experiments with this initialization method on the PascalVOC dataset, and observe both quantitative and qualitative improvement in the accuracy. Figure 4.5 shows some of query images, their nearest neighbours, and qualitative results before and after SIFT-flow based initialization. Quantitatively, with the better initializations we observe an improvement of almost 2.5% over the baseline methods with unary and pairwise terms, and almost 0.6% over the model with unary, pairwise and higher order terms (see Table 4.5).

4.6.3 General Gaussian mixture pairwise terms

Krahenbuhl and Koltun [87] use the following pairwise energy function:

$$E(\mathrm{X}|\mathrm{I}) = \sum_i \psi(x_i) + \sum_{i<j} \kappa(x_i, x_j) \sum_{v=1}^{V} w^{(v)} k^{(v)}(\mathbf{f}_i, \mathbf{f}_j) \tag{4.6.1}$$

where $\kappa(x_i, x_j)$ is the label compatibility function between pairs of labels, and $k^{(v)}(\mathbf{f}_i, \mathbf{f}_j)$ is the vth Gaussian kernel with zero mean and an arbitrary standard deviation. In this work, we propose a method to alleviate this restrictive assumption by incorporating a more general class of Gaussian mixture function to Eq. 4.6.1. Let our mixture function $\mathcal{G}^{ij}_{(\mathrm{mix})}(\mathrm{I})$ for the ith and j^{th} pair of labels take the following form:

$$\mathcal{G}^{ij}_{(\mathrm{mix})}(\mathrm{I}) = \sum_{\mathrm{m}=1}^{\mathrm{M}} \alpha^{ij}_m \mathcal{G}_{\mathrm{m}}(\mathrm{I}, \mu_{\mathrm{m}}, \Sigma_{\mathrm{m}}) \tag{4.6.2}$$

where α_m^{ij}, μ_{m}, and Σ_{m} are the mixing co-efficients, mean, and co-variance matrix of mth Gaussian mixture component $\mathcal{G}_{\mathrm{m}}$ corresponding to the $(ij)^{th}$ label pair, and I is an image derived feature. Further, we assume the mixing co-efficients α^{ij} to come from a probability distribution. On incorporating this Gaussian mixture function into Eq. 4.6.1, our final more general pairwise energy function takes the following form:

$$E(\mathrm{X}|\mathrm{I}) = \sum_i \psi(x_i) + \sum_{i<j} \kappa(x_i, x_j) \sum_{v=1}^{V} w^{(v)} k^{(v)}(\mathbf{f}_i, \mathbf{f}_j) \\ -\lambda \sum_{i<j} \sum_{m=1}^{M} \alpha_m^{(x_i,x_j)} \mathcal{G}_m(\mathrm{I}, \mu_m, \Sigma_m) \tag{4.6.3}$$

Here λ is a weight which combines the contributions from two different sets of Gaussian kernels, zero-mean kernels $K^{(v)}(.,.)$ and our general learnt kernels $\mathcal{G}^{ij}_{(\mathrm{mix})}$. In principal, we do not need a separate set of zero-mean Gaussian kernels, since they can be absorbed into our general mixture model. However, we found it useful to treat these separately for parameter setting. We now explain our learning method for the mixing co-efficients $\alpha^{(..)}$, the mean μ_m, and the co-variance matrix Σ_m.

Learning mixture models: Given this CRF model, we follow the piecewise strategy of [157] for learning the parameters of the CRF. They show how the piecewise method provides an efficient and accurate alternative to joint learning of the parameters. Thus, first we set the parameters of unary $\psi(x_i)$, and first pairwise weights $\sum_{i<j} \kappa(x_i, x_j) \sum_{v=1}^{V} w^{(v)} k^{(v)}(\mathbf{f}_i, \mathbf{f}_j)$ following the works of [157] and [87] respectively. We then set the parameters α, μ, Σ using the method described below, and the value of λ is set through cross validation.

Suppose we have a set of data points $\mathbf{f'_1}, \mathbf{f'_2}, ..., \mathbf{f'_n}$ where each feature vector $\mathbf{f'_i}$ is derived from the image data at two locations $\mathbf{f'_i} = \mathbf{f}_i - \mathbf{f}_j$, and their ground truth labels are l_i and l_j. Let us represent our data by an $N \times D$ matrix $\mathbf{F'}$, where the rows correspond to D dimensional feature points $\mathbf{f'_i}$. Assuming the data points are drawn in an i.i.d. fashion, we define a log-likelihood function as follows:

$$\log P(\mathbf{F'}|\alpha, \mu, \Sigma) = \sum_{\mathbf{i}=1}^{N} \log\{\sum_{m=1}^{M} \alpha_m^{(l_i l_j)} \mathcal{G}_m(\mathbf{f'_i}|\alpha_m, \Sigma_m)\} \tag{4.6.4}$$

Given this setting, we propose an *Expectation Maximization (EM)* method for learning parameters of the mixture models in maximum likelihood framework. During the M step, to satisfy the conditions at the maximum of the function, we first take partial derivatives of the function with respect to each parameter,

Algorithm 2: *EM* based learning Gaussian mixture model

input : Initialize $\alpha^{l_i l_j}$, μ_m, Σ_m

converged := 0, ν := 1;

while $converged = 0$ **do**

E Step: evaluate $\gamma_{\mathbf{i}m} = \frac{\alpha_m^{l_i l_j} \mathcal{G}(\mathbf{f}'_\mathbf{i}|\mu_m, \Sigma_m)}{\sum_{m'} \alpha_{m'}^{l_i l_j} \mathcal{G}(\mathbf{f}'_\mathbf{i}|\mu_{m'}, \Sigma_{m'})}$;

M Step: re-estimate parameters: $\alpha^{l_i l_j}$, μ_m, Σ_m as follows: ;

$\mu_m = \frac{1}{N_m} \sum_{\mathbf{i}=1}^{N} \gamma_{\mathbf{i}m} \mathbf{f}'_\mathbf{i}$, $\Sigma_m = \frac{1}{N_m} \sum_{\mathbf{i}=1}^{N} \gamma_{\mathbf{i}m} (\mathbf{f}'_\mathbf{i} - \mu_m)(\mathbf{f}'_\mathbf{i} - \mu_m)^{\mathrm{T}}$;

$\alpha_m^{l_i l_j} = \frac{N_m^{l_i l_j}}{N^{l_i l_j}}$, $N_m^{l_1 l_2} = \sum_\mathbf{i} \gamma_{\mathbf{i}m}^{l_i l_j} [l_i = l_1 \wedge l_j = l_2]$;

$N^{l_1 l_2} = N^{l_1 l_2} = \sum_\mathbf{i} [l_i = l_1 \wedge l_j = l_2]$, $N_m = \sum_\mathbf{i} \gamma_{\mathbf{i}m}$;

Evaluate the log likelihood $\log P(\mathbf{F}'|\alpha, \mu, \Sigma)$;

end

Return $\alpha^{l_i l_j}, \mu_m, \Sigma_m$;

and we set them to zero. First, we derive the conditions for μ_m by setting the derivative of $\log P(\mathbf{F}'|\alpha, \mu, \Sigma)$ w.r.t. μ_m to zero as follows:

$$\frac{\partial \log P(\mathbf{F}'|\alpha, \mu, \Sigma)}{\partial \mu_m} = -\sum_{\mathbf{i}=1}^{N} \frac{\alpha_m^{l_i l_j} \mathcal{G}(\mathbf{f}'_\mathbf{i}|\mu_m, \Sigma_m)}{\sum_{m'} \alpha_{m'}^{l_i l_j} \mathcal{G}(\mathbf{f}'_\mathbf{i}|\mu_{m'}, \Sigma_{m'})} \sum_m (\mathbf{f}'_\mathbf{i} - \mu_m) \quad (4.6.5)$$

$$= -\sum_{\mathbf{i}=1}^{N} \gamma_{\mathbf{i}m} (\mathbf{f}'_\mathbf{i} - \mu_m) = 0 \quad (4.6.6)$$

On rearranging this, we get the update equation for μ_m as $\mu_m = \frac{1}{N_m} \sum_{\mathbf{i}=1}^{N} \gamma_{\mathbf{i}m} \mathbf{f}'_\mathbf{i}$. Following similar strategy, we get the following update equations for Σ_m, and $\alpha^{l_i l_j}$:

$$\Sigma_m = \frac{1}{N_m} \sum_{\mathbf{i}=1}^{N} \gamma_{\mathbf{i}m} (\mathbf{f}'_\mathbf{i} - \mu_m)(\mathbf{f}'_\mathbf{i} - \mu_m)^{\mathrm{T}}; \alpha_m^{l_i l_j} = \frac{N_m^{l_i l_j}}{N^{l_i l_j}} \quad (4.6.7)$$

where $N_m^{l_1 l_2} = \sum_\mathbf{i} \gamma_{\mathbf{i}m}^{l_i l_j} [l_i = l_1 \wedge l_j = l_2]$, $N^{l_1 l_2} = \sum_\mathbf{i} [l_i = l_1 \wedge l_j = l_2]$, and $N_m = \sum_\mathbf{i} \gamma_{\mathbf{i}m}$. During the M step we assume that the value of $\gamma_{\mathbf{i}m}$ is constant. Then, during E step we evaluate the value of $\gamma_{\mathbf{i}m} = \frac{\alpha_m^{l_i l_j} \mathcal{G}(\mathbf{f}'_\mathbf{i}|\mu_m, \Sigma_m)}{\sum_{m'} \alpha_{m'}^{l_i l_j} \mathcal{G}(\mathbf{f}'_\mathbf{i}|\mu_{m'}, \Sigma_{m'})}$ assuming the Gaussian parameters μ_m, Σ_m and α_m are constant. Details of whole iterative procedure are given in the Algorithm 2.

Our piecewise learning strategy does not guarantee any bound on the solution achieved even though [169] bound the solution achieved in their piecewise learning framework. The first reason is that the parameter λ is not learnt jointly in the CRF. Further, we use a generative model to learn the parameters of the mixture components which given a pair of labels models the distribution of feature vectors,

and use the negative likelihood within the energy. This is in contrast to [157] who maximize the conditional likelihood of the labels given the training data and use the negative conditional log-likelihood as an energy term.

Inference with mixture model: Now, we explain our approach for efficient inference using the mixture model. Each mixture component involves evaluating an extra expensive term: $\sum_{i<j}\sum_{m=1}^{M}\alpha_m^{(x_i x_j)}\mathcal{G}_m(\mathbf{I},\mu_m,\Sigma_m)$. We formulate this expensive step as an efficient Gaussian filtering operation in high dimensional space following the work of Krahenbuhl and Koltun [87]. Thus, our filtering step under a non-zero mean is given by:

$$\tilde{Q}_i^m(l)=\sum_{j\neq i}\mathcal{G}_m(\mathbf{f}_i-\mathbf{f}_j|\mu_m,\Sigma_m)=[\mathcal{G}_m\otimes Q(l)](\mathbf{f}_i-\mu_m)-\mathcal{G}_m(0)Q_{(i)}(l) \quad (4.6.8)$$

We use the permutohedral lattice based filtering method [2] for fast filtering. We first embed the feature points in the high dimensional space translating the points by the means μ_m and project them onto the lattice points. We apply blurring on the mean-shifted feature points.

Experiment results: We demonstrate the accuracy and efficiency offered by our approach on object-class segmentation problems on two challenging datasets: Cambridge-driving Labelled Video Database (CamVid) [23], and PascalVOC-10 segmentation dataset [40]. We evaluate the efficacy of learning the Gaussian mixture components for the pairwise terms in the potts setting for the object-class segmentation problem on PascalVOC dataset. We learn a model with $m = L$ Gaussian components, using data from label pairs $l_i = l_j$ only. This generates $L \times L$ mixing co-efficients $\alpha^{ij} \in [0\ 1]$, thus allowing each $l_i = l_j$ label pair to reweight the L Gaussian components. Further, we note our overall timings do not include the timings for SIFT-flow, and for embedding the feature points in the permutohedral lattice. We assess the overall percentage of pixels correctly labelled, the average recall per class, and the intersection/union (I/U) measure per class.

PascalVOC dataset: We test our model on the PascalVOC-10 training and validation set. We use the same split as used in [87], who randomly partition the available images into 3 groups: 40% training, 15% validation, and 45% test set. Further, we use the unary potentials provided by [87], and an Ising label compatibility function $\mu(l_1, l_2) = [l_1 \neq l_2]$.

Qualitative and quantitative results are shown in Fig. 4.6 and Tab. 4.6 respectively. Our approaches are able to outperform both of the baseline methods in terms of union-intersection (U/I) metrics, demonstrating the importance of the

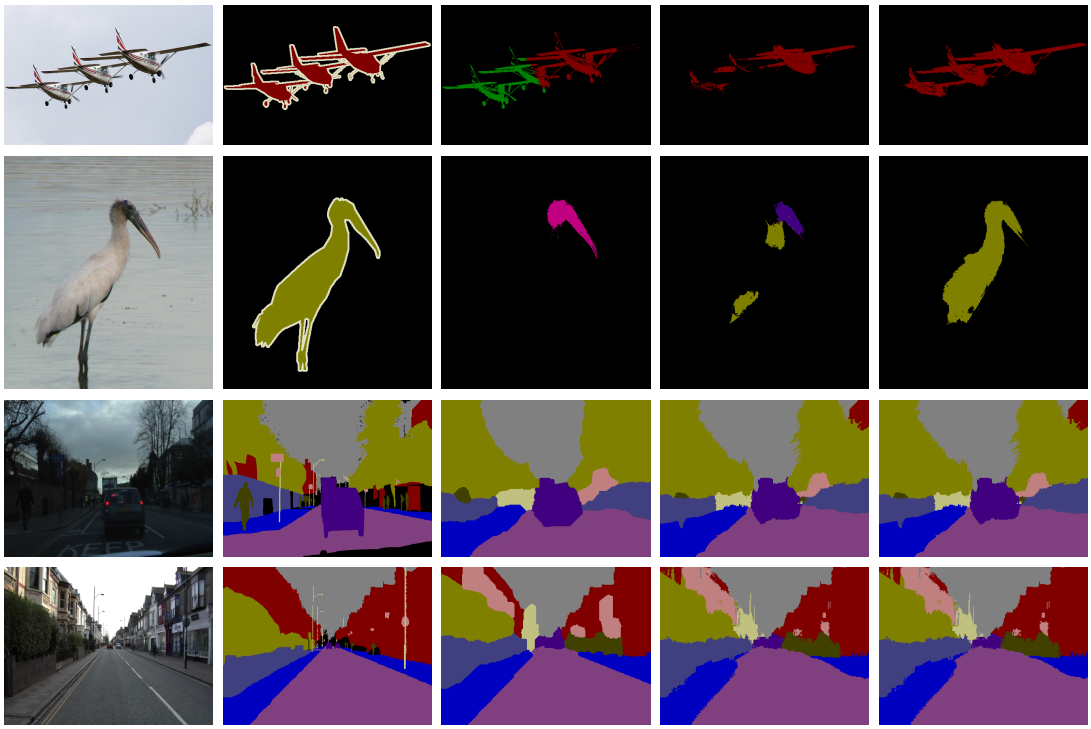


Figure 4.6: *Qualitative results on PascalVOC-10 (first 2 rows) and CamVid (last 2 rows) datasets. From left to right: input image, ground truth, output from [96] (U+P), output from [87] (dense CRF), output from our dense CRF with better initialisation and Gaussian mixture.*

Gaussian mixture components, and hierarchical SIFT-flow based initialisations. As shown, we observe an improvement of almost 1% in union-intersection (U/I) score compared to graph-cuts based α-expansion and 0.3% compared to the dense CRF of [87] on inclusion of the Gaussian mixture components. Further, using our second model with better initialisations, we are able to improve the U/I accuracy by almost 3% and 2.5% compared to the baseline methods. In our final model which includes the mixture components and the better initialisation strategy, we observe an improvement of 3.5% and 3% over the baseline methods. Further, although our final model only includes unary and pairwise terms, we observe 0.5% improvement in U/I score and almost 6% improvement in the average recall scores over the work of [96] who include higher order terms, detector potentials, and object co-occurrence terms along with unary and pairwise potentials. Apart from the improved accuracy offered by our approaches, we also observe an improvement in the inference timing compared to the graph-cuts based baseline methods. Our model with better initialisation achieves a speed up of 3 times compared to the α-expansion method, and a speed up of 40 times compared to the work of [96], although our method with the Gaussian mixture components is slower as we have to evaluate the filtering step separately for each of the mixture components in the

Algorithm	Time (sec)	Overall (%-corr)	Av. Recall	Av. I/U
α-exp (U+P) [20]	3.0	79.52	36.08	27.88
AHCRF (U+P+H) + Cooc [96]	36	81.43	38.01	30.9
dense CRF (U + dense P) [87]	0.67	71.63	34.53	28.4
Ours1 (U + dense P+GM)	26.7	80.23	36.41	28.73
Ours2 (U+ dense P+hierar)	5.90	79.65	41.84	30.95
Ours3 (U+ dense P+hierar+GM)	31.7	78.96	44.05	31.48

Table 4.6: *Quantitative results on PascalVOC-10. The table compares the timing and performance of our approach (last three lines) against three baselines. The importance of better initialisation and Gaussian mixtures is confirmed by the significant improvement achieved compared to the other methods, which use only unary and pairwise connections, and slight improvement in the results compared to the model of [96] which uses segment based higher order terms, detector potentials, and co-occurrence terms as well.*

model. Finally, we note that our aim here is to assess the relative performance of our approach with respect to our baseline methods, and we expect that our model will need further refinement to compete with the current state of the art on Pascal (our results are $\sim 9\%$ lower for average union/intersection compared to the highest performing method on the 2011 challenge, see [40]). We also note that [87] are able to further improve their average union/intersection score to 30.2% by learning the pairwise label compatibility function, which remains a possibility for our model also.

4.6.4 Convergence Analysis

In addition, we also evaluate the convergence of our mean-field algorithm after inclusion of Potts and co-occurrence based higher order terms. In Fig. 4.7 we show the KL-divergence values between Q and P distributions after each iteration of our mean-field update. In practice, we consistently observe that the KL-divergence values always decrease when the energy functions consist of only unary and pairwise terms even though we are using parallel updates. They can oscillate for some iterations when we include the higher order terms, although we still converge to a local minima overall. Further, the Fig. 4.8 visually shows the convergence of our mean-field method with higher order terms across iterations, and how the confidence of car pixels increases after inclusion of higher order terms.

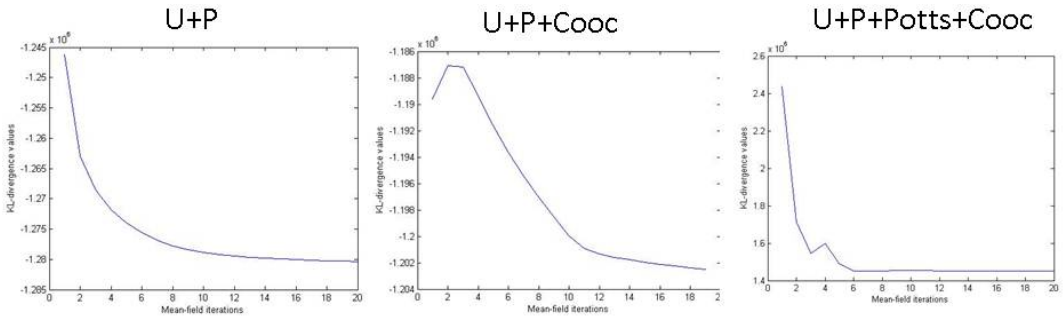


Figure 4.7: *Convergence analysis: these figures show the KL-divergence values of the mean-field approximation after each iteration for three different cases. We observe that in practice the KL-divergence oscillates after inclusion of Potts and co-occurrence potentials, though it does not oscillates with only pairwise terms.*

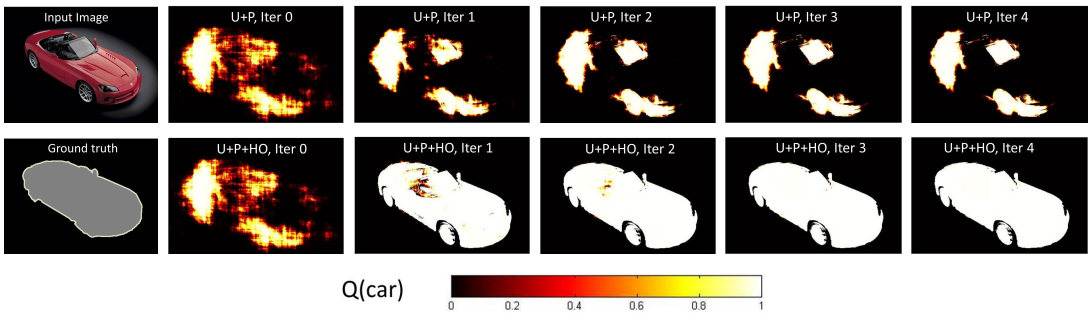


Figure 4.8: *It shows the Q distribution values across different iterations of the mean-field method for car class on PascalVOC-10 dataset before (1st row) and after (2nd row) inclusion of higher order terms. We can observe how the confidence of car pixels increases after inclusion of higher order terms.*

4.7 Discussion

We have introduced a set of techniques for incorporating higher-order terms into densely connected multi-label CRF models. As described, using our techniques, bilateral filter-based methods remain possible for inference in such models, effectively retaining the mean-field update complexity $O(MNL^2)$ as in [87] when higher-order P^n-Potts and co-occurrence models are used. This both increases the expressivity of existing fully connected CRF models, and opens up the possibility of using powerful filter-based inference in a range of models with higher-order terms. We have shown the value of such techniques for both joint object-stereo labelling and object class segmentation. In each case, we have shown substantial improvements in inference speed with respect to graph-cut based methods, particularly by using recent domain transform filtering techniques, while also observing similar or better accuracies. Future directions include investigation of further ways to improve efficiency though parallelization, and learning techniques which can draw on high speed inference for joint parameter optimization in large-scale models.

Chapter 5

Higher Order Priors for Joint Intrinsic Image, Objects, and Attributes Estimation

Many methods have been proposed to solve the problems of recovering intrinsic scene properties such as shape, reflectance and illumination from a single image, and object class segmentation separately. While these two problems are mutually informative, in the past not many papers have addressed this topic. In this chapter we explore such joint estimation of intrinsic scene properties recovered from an image, together with the estimation of the objects and attributes present in the scene. In this way, our unified framework is able to capture the correlations between intrinsic properties (reflectance, shape, illumination), objects (table, tv-monitor), and materials (wooden, plastic) in a given scene. For example, our model is able to enforce the condition that if a set of pixels take same object label, e.g. table, most likely those pixels would receive similar reflectance values. We cast the problem in an energy minimization framework and demonstrate the qualitative and quantitative improvement in the overall accuracy on the NYU and Pascal datasets.

5.1 Introduction

Over the years many theories have been developed to understand the workings of the human visual system. One successful contemporary concept of vision is that the visual system tries to estimate the underlying physical properties of the scene that generated the image [3]. These properties correspond to estimating the illumination, shape, and reflectance of the objects present in the scene. Barrow and Tenenbaum [14] coined the word *intrinsic images* for these physical properties which are related by following elegant equation:

$$C_i = R_i \cdot S_i \tag{5.1.1}$$

where $C_i \in \mathbb{R}^3, R_i \in \mathbb{R}^3$ and $S_i \in \mathbb{R}^3$ are the intensity, reflectance, and illumination induced shading terms for a pixel i respectively. Many models have been developed to solve this under-constrained problem. One of the first theories in the modern time is the Land and McCann's Retinex theory [100] which recovers the reflectance and illumination images. Tappan et.al. [60] developed a machine learning approach for the same problem. Recently, Barron and Malik [10, 11, 12, 13], Gehler et.al. [47] formulate this decomposition problem as an energy minimization problem that captures prior information about the structure of the world.

Further, recognition of objects and their material attributes is central to our interaction with the world. A great deal of works have been devoted to estimate the objects and their attributes in the scene. Shotton et.al. [157] and Ladicky

et.al. [95] propose approaches to estimate the object labels at the pixel level. Separately, Adelson [4], Farhadi et.al. [41], Lazebnik et.al. [172] define and estimate the attributes at the pixel, object and scene levels. Some of these attributes are material properties such as *woollen, metallic, shiny, glossy*, and the structural properties such as *rectangular, spherical, slanted*.

While these methods for estimating the intrinsic images, objects and attributes have separately been successful in generating good results on laboratory and real-world datasets, they fail to capture the strong correlation existing between these properties. Knowledge about the objects and attributes in the image can provide strong prior information about the intrinsic properties. For example, if a set of pixels take same object label, e.g. *table*, most likely those pixels would receive similar reflectance values. Thus the objects and their attributes can help in reducing the ambiguities present in the world and so in better estimation of the reflectance and other intrinsic properties. Additionally such a decomposition might be useful for per-pixel object and attribute segmentation tasks. For example, the estimation of the per-pixel object and attribute label using the illumination invariant reflectance should yield better results [102]. Moreover if a set of pixels get similar reflectance values, they might belong to the same object and attribute class.

Some of the previous works incorporate such high-level prior information by propagating results from one step to the next. Osadchy et.al. [124] use the specular highlights to improve recognition of transparent, shiny objects. Liu et.al. [109] recognize high-level material categories utilizing the correlation between the materials and their reflectance properties (*glass is often translucent*). Weijer et.al. [185] use higher-level information based on the objects present in the scene to better separate the illumination from the reflectance images. However, the problem with these approaches is that the errors in one step can propagate to the next steps. Joint estimation of the intrinsic images, objects and attributes can be used to overcome these issues. For instance, in the context of joint object recognition and depth estimation such positive synergy effects have been shown in e.g. [98].

In this work, our main contribution is to explore such synergy effects existing between the intrinsic properties, objects and material attributes present in a scene (see Fig. 5.1). Given an image, our algorithm jointly estimates the intrinsic properties such as reflectance, shading and depth maps, along with the estimation of the per-pixel object and attribute labels. We formulate it in an energy minimization framework, and thus our model is able to enforce the consistency among these terms. Further, our model also readily permits the unification of reconstruction and recognition with the objects and attributes in a single framework. Finally, we propose a dual decomposition based strategy to efficiently perform inference in the joint model consisting of both the continuous (reflectance, shape and il-

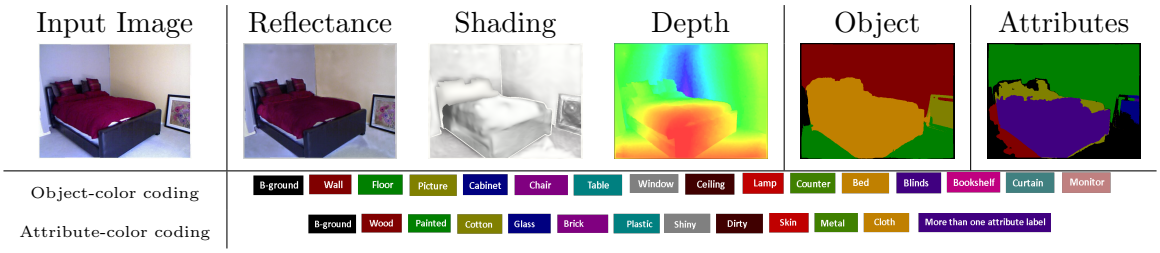


Figure 5.1: *Given an image, our algorithm jointly estimates the intrinsic properties such as reflectance, shading and depth maps, along with the estimation of the per-pixel object and attribute labels.*

lumination) and discrete (objects and attributes) variables. We demonstrate the potential of our approach on the aNYU and aPascal datasets, which are extended versions of the NYU [161] and Pascal [39] datasets with per-pixel attribute labels. We evaluate both the qualitative and quantitative improvements for the object and attribute labelling, and qualitative improvement for the intrinsic images estimation.

We introduce the problem in Sec. 5.2. Sec. 5.3 provides details about our joint model, Sec. 5.4 describes our inference and learning, 5.5 and 5.6 provide experimentation and discussion.

5.2 Problem Formulation

Our goal is to jointly estimate the intrinsic properties of the image, i.e. reflectance, shape and illumination, along with estimating the objects and attributes at the pixel level, given an image array $C = (C_1...C_V)$ where $C_i \in \mathbb{R}^3$ is the associated pixel value in the image $\mathcal{V} = \{1...V\}$. However, before going into the details of the joint formulation, we provide the forms for independently solving these problems. We first briefly describe the SIRFS model [10] for estimating the intrinsic properties for a single given object, and then the CRF model for estimating objects, and attributes [163].

5.2.1 SIRFS model for a single, given object mask

We build on the SIRFS model [10] for estimating the intrinsic properties of an image. They formulate the problem of recovering the shape, illumination and reflectance as an energy minimization problem given an image. Let $R = (R_1...R_V)$, $Z = (Z_1...Z_V)$ be the reflectance, and depth maps respectively, where $R_i \in \mathbb{R}^3$

and $Z_i \in \mathbb{R}^3$, and the illumination L be a 27-dimensional vector of spherical harmonics [10]. Further, let $S(Z, L)$ be a function that generates a shading image given the depth map Z and the illumination L (refer to [10] for details). The SIRFS model then minimizes the energy

$$\begin{aligned} &\underset{R,Z,L}{\text{minimize}}\ E^R(R) + E^Z(Z) + E^L(L) \\ &\text{subject to } C = R \cdot S(Z, L), \end{aligned} \tag{5.2.1}$$

where $E^R(R), E^Z(Z)$ and $E^L(L)$ are the prior costs defined over the reflectance, depth and illumination respectively. The most likely solutions of the energy is then estimated by using a multi-scale LBFGS optimization strategy. The SIRFS model is limited to estimating the intrinsic properties for a single object mask within an image. The recently proposed Scene-SIRFS model [13] proposes an approach to recover the intrinsic properties of whole image by embedding a mixture of shapes models in the scene. In sec. 5.3 we will also extend the per-object SIRFS model to handle multiple objects. The main difference to Scene-SIRFS is that we perform joint optimization over the per-pixel object (and attributes) labelling and intrinsic image properties.

5.2.2 Multilabel Object and Attribute Model

Additionally the problem of estimating the objects and attributes at the pixel level can also be formulated in a CRF framework [163]. Let $O = (O_1...O_V)$ and $A = (A_1...A_V)$ be the object and attribute maps for all V pixels, where each object label O_i takes one out of K discrete labels such as table, monitor, or floor. Each attribute variable A_i takes a label from the power set of the M attribute labels, for example the subset of attribute labels can be $A_i = \{red, shiny, wet\}$. Inference in the model is challenging, so in order to perform efficient inference without losing much of the accuracy, [163] represents each attributes subset A_i as a binary attribute variable $A_{i,m} \in \{0, 1\}$, meaning that $A_{i,m} = 1$ if the ith pixel takes the mth attribute and it is absent when $A_{i,m} = 0$. Under this assumption, the most likely solution for the objects and the attributes correspond to minimizing the following energy function

$$E^{OA}(O, A) = \sum_{i \in \mathcal{V}} \psi_i(O_i, A_i) + \sum_{i<j \in \mathcal{V}} \psi_{i,j}(O_i, A_i, O_j, A_j) \tag{5.2.2}$$

where the joint unary term takes following form

$$\psi_i(O_i, A_i) = \psi_i(O_i) + \sum_m \psi_{i,m}(A_{i,m}) + \sum_{m \neq m'} \psi_{i,m,m'}(A_{i,m}, A_{i,m'}) + \sum_m \psi_{i,m}(O_i, A_{i,m}) \ . \quad (5.2.3)$$

Here $\psi_i(O_i)$ and $\psi_{i,m}(A_{i,m})$ are the object dependent and per-binary attribute unary terms respectively. The terms $\psi_{i,m}(O_i, A_{i,m})$ and $\psi_{i,m,m'}(A_{i,m}, A_{i,m'})$ are the object-attribute and the attribute-attribute correlation terms. Finally the best configuration for the object and attributes are estimated using a mean-field based inference approach. Further details about the form of the unary, pairwise terms and the inference approach are described in the supplementary material [163].

5.3 Joint Model for Intrinsic Images, Objects and Attributes

Now, we provide the details of our formulation for jointly estimating the intrinsic images (R, Z, L) along with the objects (O) and attribute (A) properties given an image C in a probabilistic framework. We define the posterior probability and the corresponding joint energy function E as:

$$P(R, Z, L, O, A|I) = 1/Z(I) \ \exp\{-E(R, Z, L, O, A, I)\} \ \text{with} \quad (5.3.1)$$

$$E(R, Z, L, O, A|I) = E^{SIRFS}(R, Z, L|O, A) + E^{RO}(R, O) + E^{RA}(R, A) + E^{OA}(O, A) \quad (5.3.2)$$

The energy-term $E^{SIRFS}(R, Z, L|O, A)$ enforces for each object and attribute the SIRFS model. The terms $E^{RO}(R, O)$ and $E^{RA}(R, A)$ capture the higher order correlations between the reflectance-objects and the reflectance-attributes properties respectively. To realize this we define three kinds of higher order cliques: object dependent cliques $c \in C^O$, attribute dependent cliques $c \in C^A_m$, and reflectance dependent cliques $c \in C^R$. A clique comprises of all pixels i which are associated to the same discrete label. That is, $C^O = \{c_1...c_L\}$ with $c_l = \{i \in \mathcal{V}|O_i = l\}$, $C^A_m = \{c_1...c_L\}$ with $c_l = \{i \in \mathcal{V}|A_{i,m} = l\}$, and $C^R = \{c_1...c_L\}$ with $c_l = \{i \in \mathcal{V}|R_i = l\}$. For this we discretized the reflectance image R, using the unsupervised segmentation approach of Felzenswalb and Huttenlocker [42].

We now provide details of the forms of these terms. The specific form has two benefits: (i) our model is able to jointly estimate the intrinsic properties, objects

and attributes at the scene level and (ii) efficient inference in the joint model is possible.

5.3.1 SIRFS model for a scene

Given this representation of the scene, we model the scene specific E^{SIRFS} by a mixture of object specific reflectance, and depth terms and an illumination term as:

$$\begin{aligned} \underset{R,Z,L}{\text{minimize}} \quad & E^{SIRFS} = \sum_{c \in C^o} \Big(E^R(R_c) + E^Z(Z_c) \Big) + E^L(L) \\ & \text{subject to} \quad C = R \cdot S(Z, L) \end{aligned} \tag{5.3.3}$$

where $R = \{R_c\}$, $Z = \{Z_c\}$. Here $E^R(R_c)$ and $E^Z(Z_c)$ are the reflectance and depth terms respectively defined over the object clusters $c \in C^o$. In the current formulation, we have assumed that we have a single model of illumination L for whole scene which corresponds to a 27-dimensional vector of spherical harmonics [10].

5.3.2 Reflectance, Objects term

The joint reflectance-object energy term $E^{RO}(R, O)$ captures the relations between the objects present in the scene and their reflectance properties. Essentially this term enforces higher order consistency over the reflectance images based on the cliques defined over the object labels, and vice-versa. For example, this term enforces that pixels inside an object take similar reflectance values and that surface points with similar reflectance values take the same object label. We define

$$E^{RO}(R, O) = \sum_{c \in C^O} \psi(R_c) + \sum_{c \in C^R} \psi(O_c) \tag{5.3.4}$$

where R_c, O_c are the labelling for the subset of pixels c respectively. Hence, the terms $\psi(R_c)$ and $\psi(O_c)$ are higher order terms. The term $\psi(R_c)$ enforces consistency over the reflectance values within object cliques and takes the form $\psi(R_c) = \|c\|^{\theta_\alpha}(\theta_p + \theta_v G^r(c))$ where

$$G^r(c) = \exp\left(-\theta_\beta \frac{\| \sum_{i \in c} (R_i - \mu_c)^2 \|}{\|c\|} \right) . \tag{5.3.5}$$

Here $\|c\|$ is the size of the clique, $\mu_c = \frac{\sum_{i \in c} R_i}{\|c\|}$ and $\theta_\alpha, \theta_p, \theta_v, \theta_\beta$ are constants. Similarly we enforce higher order consistency over the object labelling based on the cliques $c \in C^R$ defined over the reflectance image. The object higher order term takes the form of pattern-based P^N-Potts model [77]:

$$\psi(O_c) = \begin{cases} \gamma_l^o & \text{if } O_i = l, \ \forall i \in c \\ \gamma_{max}^o & \text{otherwise} \end{cases} \tag{5.3.6}$$

where $\gamma_l^o, \gamma_{max}^o$ are constants. To summarize, these two higher order terms enforce the cost of inconsistency within the object and reflectance labels.

5.3.3 Reflectance, Attributes term

Similarly we define the term $E^{RA}(R, A)$ which enforces a higher order consistency between reflectance and attribute variables. For example, our terms enforce that pixels with the same attribute label take similar reflectance values, and that pixels with similar reflectance values are more likely to have the same attribute label. Such higher order consistency over the attributes and reflectance values take the following form:

$$E^{RA}(R, A) = \sum_m \Big(\sum_{c \in C_m^A} \psi(R_c) + \sum_{c \in C^R} \psi(A_{c,m}) \Big) \tag{5.3.7}$$

where $\psi(R_c)$ and $\psi(A_{c,m})$ are the higher order terms defined over the reflectance image and the attribute image corresponding to the mth attribute respectively. Forms of these terms are similar to the one defined for the object-reflectance higher order terms; these terms are further detailed in the supplementary material. The term $\psi(R_c)$ enforces consistency over the reflectance values within attribute cliques and takes the form $\psi(R_c) = \|c\|^{\theta_\alpha}(\theta_p + \theta_v G^r(c))$ where

$$G^r(c) = \exp\left(-\theta_\beta \frac{\| \sum_{i \in c} (R_i - \mu_c)^2 \|}{\|c\|} \right) . \tag{5.3.8}$$

Here $\|c\|$ is the size of the clique, $\mu_c = \frac{\sum_{i \in c} R_i}{\|c\|}$ and $\theta_\alpha, \theta_p, \theta_v, \theta_\beta$ are constants. Similarly we enforce higher order consistency over the object labelling based on the cliques $c \in C^R$ defined over the reflectance image. The attribute higher order

term takes the form of pattern-based P^N-Potts model [8, 9]:

$$\psi(A_{c,m}) = \begin{cases} \gamma_l^a & \text{if } A_{i,m} = 1,\ \forall i \in c \\ \gamma_{max}^a & \text{otherwise} \end{cases} \tag{5.3.9}$$

where $\gamma_l^a, \gamma_{max}^a$ are constants. To summarize, these two higher order terms enforce the cost of inconsistency within the attributes and reflectance labels.

5.4 Inference and Learning

Given the above model, our optimization problem involves solving following joint energy function to get the most likely solution for (R, Z, L, O, A):

$$\begin{aligned} E(R, Z, L, O, A|I) = E^{SIRFS}(R, Z, L) + E^{RO}(R, O) + E^{RA}(R, A) \\ + E^{OA}(O, A) \end{aligned} \tag{5.4.1}$$

However, this problem is very challenging since it consists of both the continuous variables (R, Z, L) and discrete variables (O, A). Thus in order to minimize the function efficiently without loosing accuracy we follow a dual decomposition strategy.

We first introduce a set of duplicate variables for the reflectance (R_1, R_2, R_3), objects (O_1, O_2), and attributes (A_1, A_2) and a set of new equality constraints to enforce the consistency on these duplicate variables. Our optimization problem thus takes the following form:

$$\begin{aligned} \underset{R_1,R_2,R_3,Z,L,O_1,O_2}{\text{minimize}} \quad & E(R_1, Z, L) + E(O_1, A_1) + E(R_2, O_2) + E(R_3, A_2) \\ \text{subject to} \quad & R_1 = R_2 = R_3; \quad O_1 = O_2; \quad A_1 = A_2 \end{aligned} \tag{5.4.2}$$

From now on we have removed the subscripts and superscripts from the energy terms for simplicity of the notations. Now we formulate it as an unconstrained optimization problem by introducing a set of lagrange multipliers $\theta_r^1, \theta_r^2, \theta_o, \theta_a$ and decompose the dual problem into four sub-problems as:

$$\begin{aligned} E(R_1, Z, L) \quad + \quad & E(O_1, A_1) + E(R_2, O_2) + E(R_3, A_2) + \theta_r^1(R_1 - R_2) \\ + \quad & \theta_r^2(R_2 - R_3) + \theta_o(O_1 - O_2) + \theta_a(A_1 - A_2) \end{aligned} \tag{5.4.3}$$

which we express as:

$$g_1(R_1, Z, L) + g_2(O_1, A_1) + g_3(O_2, R_2) + g_4(A_2, R_3), \tag{5.4.4}$$

where

$$\begin{aligned} g_1(R_1, Z, L) &= \text{minimize}_{R_1,Z,L} \quad E(R_1, Z, L) + \theta_r^1 R_1 \\ g_2(O_1, A_1) &= \text{minimize}_{O_1,A_1} \quad E(O_1, A_1) + \theta_o O_1 + \theta_a A_1 \\ g_3(O_2, R_2) &= \text{minimize}_{O_2,R_2} \quad E(O_2, R_2) - \theta_o O_2 - \theta_r^1 R_2 \\ g_4(A_2, R_3) &= \text{minimize}_{A_2,R_3} \quad E(A_2, R_3) - \theta_a A_2 - \theta_r^2 R_3 \end{aligned} \tag{5.4.5}$$

are the slave problems which are optimized separately and efficiently while treating the dual variables $\theta_r^1, \theta_r^2, \theta_o, \theta_a$ constant, and the master problem then optimizes these dual variables to enforce consistency. Next, we solve each of the sub-problems and the master problem.

Solving subproblem $g_1(R_1, Z, L)$: First sub-problem consists of optimizing over the reflectance R_1, depth Z and illumination L dependent terms $E(R_1, Z, L)$ along with the dual variable dependent term $\theta_r^1 R_1$. It takes the following form:

$$g_1(R_1, Z, L) = \underset{R_1,Z,L}{\text{minimize}} \quad E(R_1, Z, L) + \theta_r^1 R_1 \tag{5.4.6}$$

where $E(R_1, Z, L)$ takes the form as described in Sec. 2.2:

$$E(R_1, Z, L) = E^{R_1}(R_1) + E^Z(Z) + E^L(L) \tag{5.4.7}$$

where $E^{R_1}(R_1), E^Z(Z)$ and $E^L(L)$ are the costs of reflectance R_1, depth Z and illumination L respectively. Following the forms of the cost defined over the reflectance (Sec. 2.2), our final optimization problem takes the following form:

$$\begin{aligned} g_1(R_1, Z, L) = \lambda_s E_s^{R_1}(R_1) + \lambda_e E_e^{R_1}(R_1) + \lambda_c E_c^{R_1}(R_1) \\ + \theta_r^1 R_1 + E^Z(Z) + E^L(L) \end{aligned} \tag{5.4.8}$$

To optimize $g_1(R_1, Z, L)$, we follow a multi-scale LBFGS strategy which requires taking gradient of $g_1(R_1, Z, L)$ wrt. R_1, Z, L and then updating their values respectively (refer to [12] for more detail).

Solving subproblem $g_2(O_1, A_1)$: This sub-problem requires solving the following function which consists of only object and attribute dependent discrete

variables (O_1, A_1):

$$\begin{aligned} g_2(O_1, A_1) &= \text{minimize}_{O_1, A_1} \quad E^{OA}(O_1, A_1) + \theta_o O_1 + \theta_a A_1 \\ &= \sum_{i \in \mathcal{V}} \psi_i(O_{1i}, A_{1i}) + \sum_{i<j \in \mathcal{V}} \psi_{ij}^J(O_{1i}, A_{1i}, O_{1j}, A_{1j}) \\ &\quad + \theta_o O_1 + \theta_a A_1 \end{aligned} \tag{5.4.9}$$

The dual variable dependent terms add $\theta_o O_1$ to the object unary potential $\psi_i(O_1)$ and $\theta_a A_1$ to the attribute unary potential $\psi_i(A_1)$. Let $\psi'(O_1)$ and $\psi'(A_1)$ be the updated object and attribute unary potentials. We follow a filter-based mean-field strategy [163, 180] for the optimization. In the mean-field framework, given the true distribution $P = \frac{\exp(-g_2(O_1, A_1))}{Z_1}$, we find an approximate distribution Q, where approximation is measured in terms of the KL-divergence between the P and Q distribution. Here Z_1 is the normalizing constant. Based on the model in Sec. 2.2, Q takes the form as $Q_i(O_i, A_i) = Q_i^{\mathcal{O}}(O_i) \prod_m Q_{i,m}^{\mathcal{A}}(A_{i,m})$, where $Q_i^{\mathcal{O}}$ is a multi-class distribution over the object variable, and $Q_{i,m}^{\mathcal{A}}$ is a binary distribution over {0,1}. With this, the mean-field updates for the object variables take the following form:

$$\begin{aligned} Q_i^{\mathcal{O}}(O_i = l) = \frac{1}{Z_i^O} \exp\{-\psi_i'(O_i) - \sum_{l' \in 1..K} \sum_{j \neq i} Q_j^{\mathcal{O}}(O_j = l')(-g(i,j)) \\ - \sum_{m, b \in \{0,1\}} Q_{jm}^{\mathcal{A}}(A_{jm} = b)\psi_{i,m}(l, b)\}, \end{aligned} \tag{5.4.10}$$

where $g(i, j)$ is a contrast sensitive pairwise costs defined by a mixture of Gaussian kernels [87], and $Z_i^{\mathcal{O}}$ is per-pixel normalization factor. Given this form of the pairwise terms, as in [87], we can efficiently evaluate the pairwise summations in Eq. 16 using $K + M$ Gaussian convolutions. The updates for the attribute variables also take similar form

$$\begin{aligned} Q_{i,m}^{\mathcal{A}}(A_{i,m} = b) &= \frac{1}{Z} \exp\{-\psi_{im}'(b) - \sum_{j \neq i} Q_{jm}^{\mathcal{A}}(A_{jm} = b)(-g(i,j)) \\ &\quad - \sum_{m' \neq m, b' \in \{0,1\}} Q_{im'}^{\mathcal{A}}(A_{im'} = b')\psi_{i,m,m'}^{\mathcal{A}}(b, b') \\ &\quad - \sum_l Q_i^{\mathcal{O}}(O_i = l)\psi_{i,l,m}^{OA}(l, b)\} \end{aligned} \tag{5.4.11}$$

where $Z_{ia}^{\mathcal{A}}$ is the per-pixel normalization factor for the attribute variables, and $b \in \{0, 1\}$.

Solving subproblems $g_3(O_2, R_2)$: We provide the details for the optimization of the joint object and reflectance terms $g_3(O_2, R_2)$ which take the following form:

$$g_3(O_2, R_2) = \underset{O_2, R_2}{\text{minimize}} \sum_{c \in C^O} \psi(R_{2c}) + \sum_{c \in C^R} \psi(O_{2c}) - \theta_o O_2 - \theta_r^1 R_2 \quad (5.4.12)$$

where $\psi(R_{2c})$ and $\psi(O_{2c})$ are the higher order consistency terms defined over the reflectance and object variables, and $(\theta_o O_2), (\theta_r^1 R_2)$ are the dual variable dependent object and reflectance terms. Solving of this problem requires optimization with both the continuous R_2 and discrete O_2 variables. Thus, we use a co-ordinate descent approach where we first optimize the object dependent terms keeping the reflectance variables fixed and then optimize over the reflectance variables keeping the objects fixed. We keep iterating between these two steps till the convergence.

Optimize the object terms: First step involves optimizing with the object variables keeping the reflectance variables fixed. The optimization problem takes the following form:

$$O_2^* = \arg\min_{O_2} \sum_{c \in C^R} \psi(O_{2c}) - \theta_o O_2 \quad (5.4.13)$$

It consists of the higher order term $\psi(O_{2c})$ which takes a robust P^N-Potts form as described in Sec. 3 of the main paper and a dual variable dependent unary term $\theta_o O_2$. We use a mean-field based strategy [5, 6] to optimize it. As mentioned in Eq. 8, the mean-field update requires evaluating the expectation of the higher order cost $\psi(O_{2c})$ under the current Q distribution, which takes the following form:

$$\begin{aligned} \sum_{\{\mathbf{O}_{c \in C^R} | O_i = l\}} Q^O_{c-i}(O_{c-i}) \cdot \psi(O_{2c}) = & \Big(\prod_{j \in c, j \neq i} Q_j^O(O_j = l) \Big) \gamma_l \\ & + \Big(1 - \Big(\prod_{j \in c, j \neq i} Q_j^O(O_j = l) \Big)\Big) \gamma_{\max} \end{aligned} \quad (5.4.14)$$

where $\gamma_{\max}$ is the inconsistency cost. Given the form of the higher order terms, we can efficiently evaluate it the way it has been proposed in [47]. With these terms, the mean-field updates for the object variables with all the cost terms $(E(O))$ take following form:

$$\begin{aligned} Q_i^{\mathcal{O}}(O_i = l) = \frac{1}{Z_i^O} \exp\{ & -\psi_i(O_i) - \sum_{c_r \in C_r} \Big(\prod_{j \in c, j \neq i} Q_j^O(O_j = l) \Big) \gamma_l \\ & + \Big(1 - \Big(\prod_{j \in c_r, j \neq i} Q_j^O(O_j = l) \Big)\Big) \gamma_{\max} \} \end{aligned} \quad (5.4.15)$$

Optimize the reflectance terms: Next we optimize with the reflectance variables keeping the object variables fixed. This part consists of optimizing following function:

$$(R_2^*) = \arg\min_{R_2} \left(g_3^r(R_2) = \sum_{c \in C^O} \psi(R_{2c}) - \theta_r^1 R_2 \right) \tag{5.4.16}$$

We use a gradient descent approach to optimize for the reflectance term R_2 where in each iteration we update the values of R_2 based on the gradient $\nabla_{R_2} g_3^r(R_2)$. Thus the update for the reflectance term takes the following form:

$$R_2^{t+1} = R_2^t - \alpha_r^t \left(\nabla_{R_2} g_3^r(R_2^t) \right) \tag{5.4.17}$$

The updates for the attributes and the reflectance terms take the similar forms.

Solving subproblems $g_4(A_2, R_3)$: These two problems take the following forms:

$$\begin{aligned} g_4(A_2, R_3) = \text{minimize}_{A_2, R_3} \quad & \sum_m \Big(\sum_{c \in C_m^A} \psi(R_{3c}) + \sum_{c \in C^R} \psi(A_{2c,m}) \Big) \\ & - \theta_a A_2 - \theta_r^2 R_3 \end{aligned} \tag{5.4.18}$$

Solving of these two sub-problems requires optimization with both the continuous R and discrete O, A variables respectively. However since these two sub-problems consist of higher order terms and dual variable dependent terms, we follow a simple co-ordinate descent strategy to update the reflectance and the object (and attribute) variables iteratively. The optimization of the object (and attribute) variables are performed in a mean-field framework, and a gradient descent based approach is used for the reflectance variables.

Solving master problem The master problem then updates the dual-variables $\theta_r^1, \theta_r^2, \theta_o, \theta_a$ given the current solution from the slaves. Here we provide the update equations for θ_r^1; the updates for the other dual variables take similar form. The master calculates the gradient of the problem $E(R, Z, L, O, A|I$ wrt. θ_r^1, and then iteratively updates the values of θ_r^1 as:

$$\theta_r^1 = \theta_r^1 + \alpha_r^1 \left(g_1^{\theta_r^1}(R_1, Z, L) + g_3^{\theta_r^1}(O_2, R_2) \right)_+ \tag{5.4.19}$$

where α_r^t is the step size t^{th} iteration and $g_1^{\theta_r^1}, g_3^{\theta_r^1}$ are the gradients w.r.t. to the θ_r^1. Further details on our inference techniques are provided in the supplementary

material.

Learning: In the model described above, there are many parameters joining each of these terms. We use a cross-validation strategy to estimate these parameters in a sequential manner and thus ensuring efficient strategy to estimate a good set of parameters. The unary potentials for the objects and attributes are learnt using modified a TextonBoost model of Ladicky et.al. [95] which uses a colour, histogram of oriented gradient (HOG), and location features.

5.5 Experiments

We demonstrate our joint estimation approach on both the per-pixel object and attribute labelling tasks, and estimation of the intrinsic properties of the images. For the object and attribute labelling task, we conduct experiments on the NYU 2 [161] and Pascal [39] dataset both quantitatively and qualitatively. To evaluate our model with the attribute labels, we use the per-pixel annotation provided by [163] for the NYU 2 dataset and for the Pascal dataset, we generate the per-pixel attribute labels. As a baseline, we compare our joint estimation approach against the mean-field based method [87], the object-attribute estimation approach of [163], and the graph-cuts based α-expansion method [95]. We assess the accuracy in terms of the overall percentage of the pixels correctly labelled, and the intersection/union score per class (defined in terms of the true/false positives/negatives for a given class as TP/(TP+FP+FN)). Additionally we also evaluate our approach in estimating better intrinsic properties of the images though qualitatively only, since it is extremely difficult to generate the ground truths for the intrinsic properties, e.g. reflectance, depth and illumination for any general image. We compare our intrinsic properties results against the model of Barron and Malik [2, 4], Gehler et.al. [47] and the Retinex model [100]. Further, we also show how our approach is able to recover better smooth and de-noised depth maps compared to the raw depth provided by the Kinect [161]. In all these cases, we use the code provided by the authors for the AHCRF [95], mean-field approach [87, 180]. For the intrinsic property tasks we compare against the models of Barron and Malik [2, 4], Gehler et.al. [47] and the Retinex model [100]. Details of all the experiments are provided below.

5.5.1 aNYU 2 dataset

We first conduct experiment on aNYU 2 dataset [163], an extended version of the indoor NYU 2 dataset [161]. The dataset consists of 725 training images, 100

Name	Label categories
Object Labels	Wall, Floor, Picture, Cabinet, Chair, Table, Window, Ceiling, Lamp, Counter, Bed, Blinds, Bookstore, Curtain, Monitor
Attribute Labels	Wooden, Painted, Cotton, Glass, Glossy, Plastic, Shiny, Textured

Table 5.1: *Object and attribute labels for the NYU 2 dataset.*

validation and 624 test images. Further, the dataset consists of per-pixel object and attribute labels (see Fig. 5.1 for per-pixel attribute labels). We select 15 object and 8 attribute classes that have sufficient number of instances to train the unary classifier responses. The object labels corresponds to some indoor object classes as *floor, wall, ..* and attribute labels corresponds to material properties of the objects as *wooden, painted, cotton, glass, glossy, plastic, shiny, and textured* as shown in Tab. 5.1. Further, since this dataset has depth from the Kinect depths, we use them to initialize the depth maps Z for our joint estimation approach.

We show quantitative and qualitative results in Tab. (5.2, 5.3) and Fig. 5.3 respectively. As shown, our joint approach achieves an improvement of almost 6% , and 3% in the overall accuracy and average intersection-union (I/U) score over the model of AHCRF [95], and almost 1.1 % improvement in the average I/U over the model of [163] for the object class segmentation . Similarly we also observe an improvement of almost 6 % and 2 % in the overall accuracy and I/U score over denseCRF model [87], and almost 1.8 % and 0.2 % in the overall accuracy and average I/U over the model of [163] for the per-pixel attribute labelling task. These quantitative improvement suggests that our model is able to improve the object and attribute labelling using the intrinsic properties information. Qualitatively also we observe an improvement in the output of both the object and attribute segmentation tasks as shown in Fig. 5.3.

Further, we show the qualitative improvement in the results of the intrinsic properties in the Fig. (5.2, 5.3, 5.4, 5.5). As shown our joint approach helps to recover better depth map compared to the noisy kinect depth maps; justifying the unification of reconstruction and objects and attributes based recognition tasks. Further, our reflectance and shading images visually look much better than the models of Retinex [100] and Gehler et.al. [47], and similar to the Barron and Malik approach [2,4].

5.5.2 aPascal dataset

We also show experiments on aPascal dataset, our extended Pascal dataset with per-pixel attribute labels. We select a subset of 517 images with the per-pixel object labels from the Pascal dataset and annotate it with 7 material attribute

Algorithm	Av. I/U	Oveall(% corr)
AHCRF [95]	28.88	51.06
DenseCRF [87]	29.66	50.70
DenseCRF+OA [163]	30.21	56.90
Ours (OA+Intr)	31.34	57.23

Table 5.2: *Object Accuracy: Quantitative results on aNYU 2 dataset [163] for the object class segmentation task. The table compares performance of our approach (last line) against three baselines. The importance of our joint estimation for intrinsic images, objects and attributes is confirmed by the better performance of our algorithm compared to the graph-cuts based (AHCRF) method [95], mean-field based (DenseCRF) approach [87] and mean-field based (DenseCRF+OA) approach of [163] for both the tasks. Here intersection vs. union (I/U) measure is defined as* $\frac{TP}{TP+FN+FP}$ *and '% corr' corresponds to the total proportional of correctly labelled pixels.*

labels at the pixel level. These attributes correspond to *wooden, skin, metallic, glass, shiny, glossy, and textured*. These attribute labels are shown in Tab. 5.4.

Some quantitative and qualitative results are shown in Tab. (5.5, 5.6) and Fig. 5.3 respectively. As shown, our approach achieves an improvement of almost 1.7 % and 0.6 % in the I/U score for the object and attribute labelling tasks respectively over the model of [163]. We observe qualitative improvement in the accuracy shown in Fig. 5.3.

5.6 Discussion

In this chapter, we have explored the synergy effects between intrinsic properties of an images, and the objects and attributes present in the scene. We cast the problem in a joint energy minimization framework; thus our model is able to encode the strong correlations between intrinsic properties (*reflectance, shape,illumination*), objects (*table, tv-monitor*), and materials (*wooden, plastic*) in a given scene. We have shown that dual-decomposition based techniques can be effectively applied to perform optimization in the joint model. We demonstrated its applicability on the extended versions of the NYU and Pascal datasets. We achieve both the qualitative and quantitative improvements for the object and attribute labelling, and qualitative improvement for the intrinsic images estimation.

Future directions include further exploration of the possibilities of integrating priors based on the structural attributes such as *slanted, cylindrical* to the joint

Algorithm	Av. I/U	Oveall(% corr)
AHCRF [95]	21.9	40.7
DenseCRF [87]	22.02	37.6
DenseCRF+OA [163]	23.95	41.425
Ours (OA+Intr)	24.175	43.25

Table 5.3: *Attribute Accuracy: Quantitative results on aNYU 2 dataset [163] for and attributes segmentation task. The table compares performance of our approach (last line) against three baselines. The importance of our joint estimation for intrinsic images, objects and attributes is confirmed by the better performance of our algorithm compared to the graph-cuts based (AHCRF) method [95], mean-field based (DenseCRF) approach [87] and mean-field based (DenseCRF+OA) approach of [163] for both the tasks. Here intersection vs. union (I/U) measure is defined as* $\frac{TP}{TP+FN+FP}$ *and '% corr' corresponds to the total proportional of correctly labelled pixels.*

Name	Label categories
Object Labels	Person, Bird, Cat, Cow, Dog, Horse, Sheep, Aeroplane, Bicycle, Boat, Bus, Car, Motorbike, Train, Bottle Chair, Dining table, Potted plant, Sofa, TV/Monitor
Attribute Labels	Wooden, Skin, Glass, Glossy, Plastic, Shiny, Textured

Table 5.4: *Object and attribute labels for the Pascal dataset.*

intrinsic properties, objects and attributes model. For instance, knowledge that the object is *slanted* would provide a prior for the depth and distribution of the surface normals. Further, the possibility of incorporating a mixture of illumination models to better model the illumination in a natural scene remains another future direction.

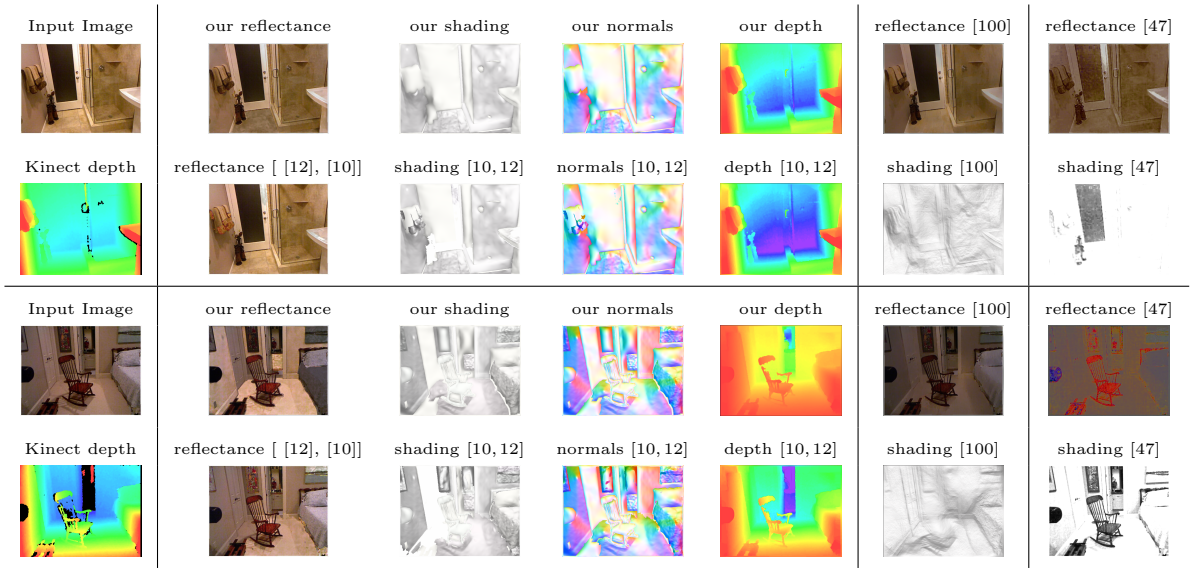


Figure 5.2: *Given an image and its depth image for the aNYU dataset [163], these figures qualitatively compare our algorithm in jointly estimating better the intrinsic properties such as reflectance, shading, normals and depth maps. We compare against the model Barron and Malik [10, 12], the Retinex model [100] (2nd last column) and the Gehler et.al. approach [47] (last column).*

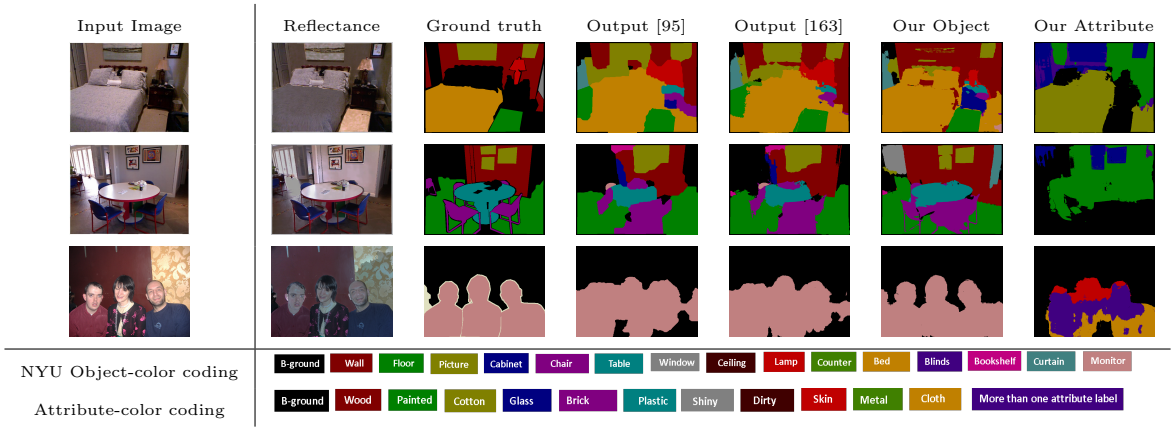


Figure 5.3: *Qualitative results on aNYU [163] (first 2 lines) and aPascal (last line) dataset. From left to right: input image, ground truth, output from [95] (AHCRF), output from [163] (mean-field based (DenseCRF+OA) method), our output for the object class segmentation. Last column shows our attribute segmentation output. (Attributes for NYU dataset: wood, painted, cotton, glass, brick, plastic, shiny, dirty; Attributes for Pascal dataset: skin, metal, plastic, wood, cloth, glass, shiny.)*

Algorithm	Av. I/U	Oveall(% corr)
AHCRF [95]	32.53	82.30
DenseCRF [87]	36.9	79.4
DenseCRF+OA [163]	36.4	79.5
Ours (OA + Intr)	38.1	81.4

Table 5.5: *Quantitative results on aPascal dataset for both the object class segmentation task. The table compares performance of our approach (last line) against three baselines. The importance of our joint estimation for intrinsic images, objects and attributes is confirmed by the better performance of our algorithm compared to the graph-cuts based (AHCRF) method [95], mean-field based (DenseCRF) approach [87] and mean-field (DenseCRF+OA) approach of [163] for both the tasks. Here intersection vs. union (I/U) measure is defined as* $\frac{TP}{TP+FN+FP}$ *and '% corr' corresponds to the total proportional of correctly labelled pixels.*

Algorithm	Av. I/U	Oveall(% corr)
AHCRF [95]	17.4	95.1
DenseCRF [87]	18.28	96.2
DenseCRF + OA [163]	18.42	96.2
Ours (OA+Intr)	18.85	96.7

Table 5.6: *Quantitative results on aPascal dataset for the attributes segmentation task. The table compares performance of our approach (last line) against three baselines. The importance of our joint estimation for intrinsic images, objects and attributes is confirmed by the better performance of our algorithm compared to the graph-cuts based (AHCRF) method [95], mean-field based (DenseCRF) approach [87] and mean-field (DenseCRF+OA) approach of [163] for both the tasks. Here intersection vs. union (I/U) measure is defined as* $\frac{TP}{TP+FN+FP}$ *and '% corr' corresponds to the total proportional of correctly labelled pixels.*

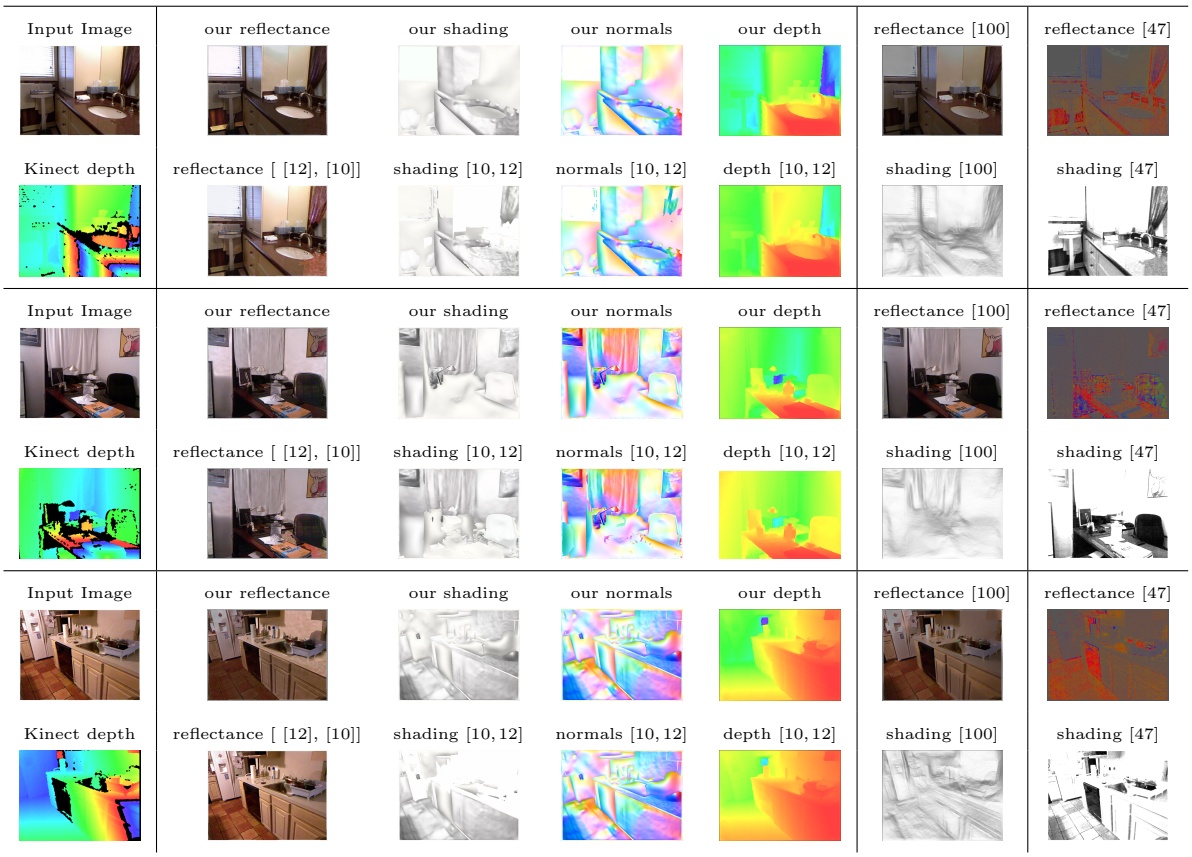


Figure 5.4: *Given an image and its depth image for the NYU dataset, these figures qualitatively compare our algorithm in jointly estimating better the intrinsic properties such as reflectance, shading, normals and depth maps. We compare against the model Barron and Malik [10, 12], the Retinex model [100] (2nd last column) and the Gehler et.al. approach [47] (last column).*

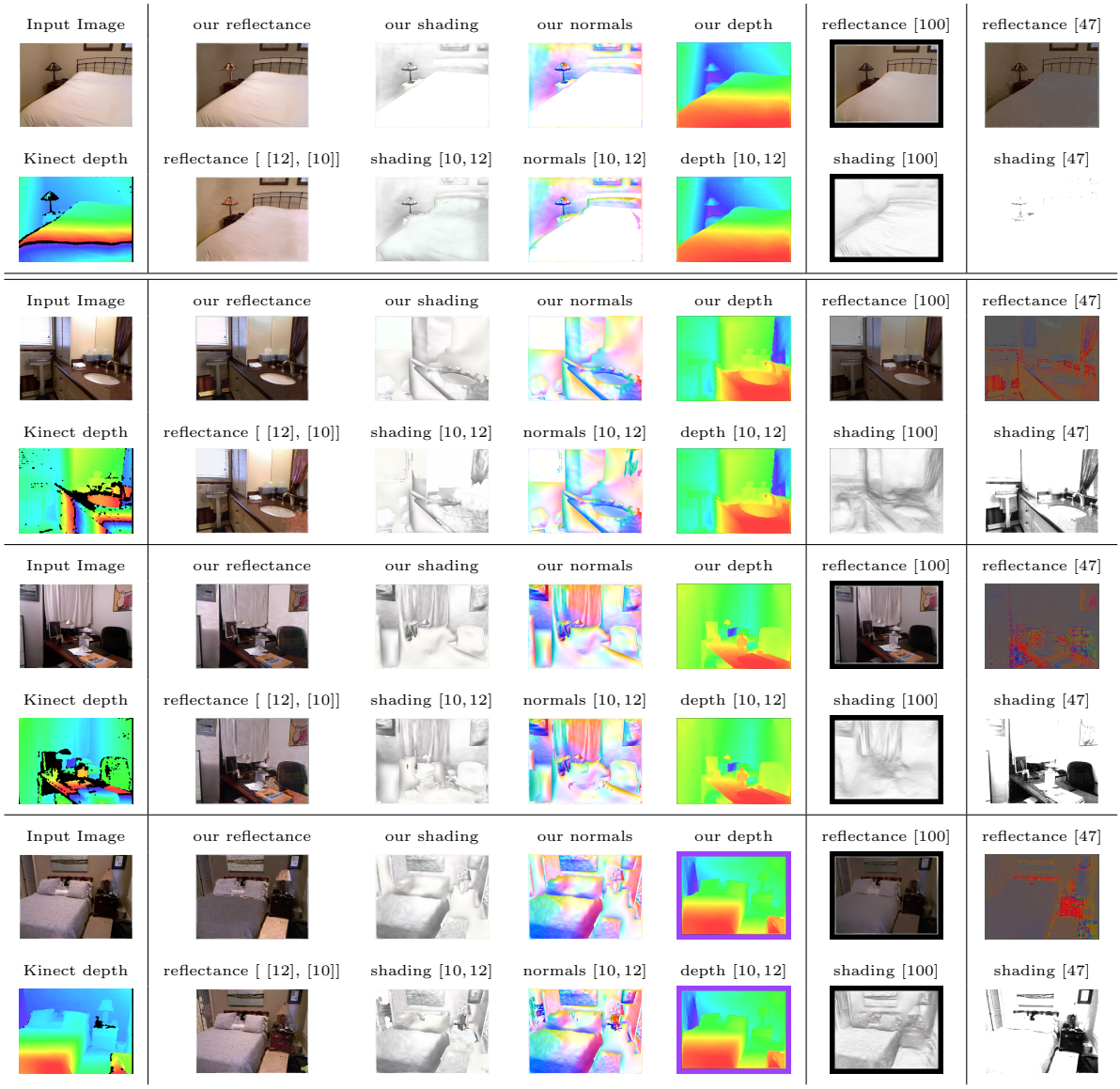


Figure 5.5: *Given an image and its depth image for the NYU dataset, these figures qualitatively compare our algorithm in jointly estimating better the intrinsic properties such as reflectance, shading, normals and depth maps. We compare against the model Barron and Malik [10, 12], the Retinex model [100] (2nd last column) and the Gehler et.al. approach [47] (last column).*

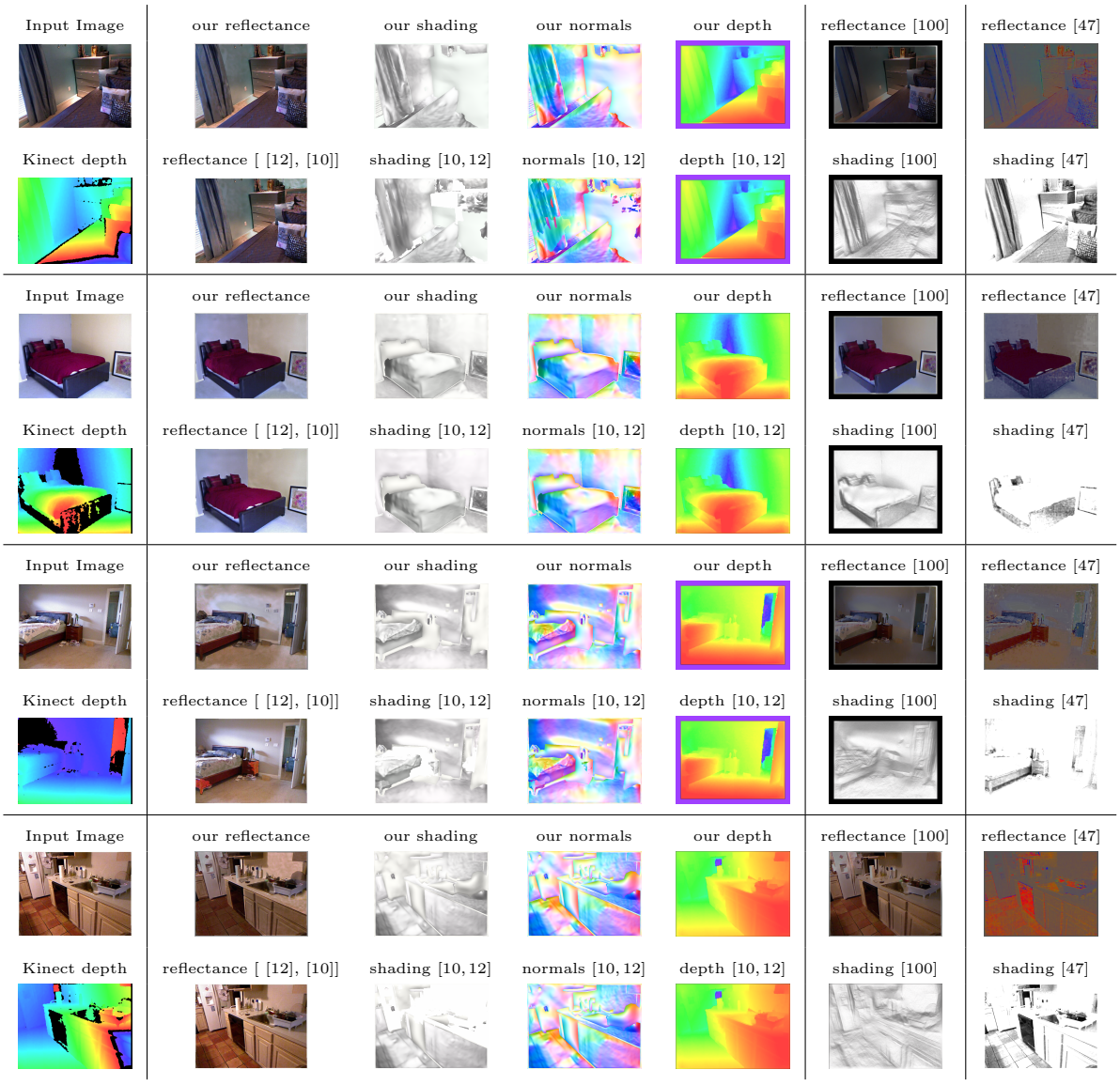


Figure 5.6: *Given an image and its depth image for the NYU dataset, these figures qualitatively compare our algorithm in jointly estimating better the intrinsic properties such as reflectance, shading, normals and depth maps. We compare against the model Barron and Malik [10, 12], the Retinex model [100] (2nd last column) and the Gehler et.al. approach [47] (last column).*

Chapter 6

SemanticPaint: Personalized 3D Recognition at your Fingertips

The emergence of consumer depth cameras has generated a lot of interest in real-time 3D scanning of physical environments. Despite dramatic progress in real-time 3D *reconstruction*, many applications could benefit from *recognition* in the scene. Progress in recognition, and specifically object ('semantic') segmentation in volumetric 3D environments is hindered by the need to collect training data, which can be a laborious and expensive task. In this chapter we describe an interactive 3D labelling and segmentation system that aims to make acquiring segmented 3D models fast, simple, and user-friendly. Carrying a body-worn depth camera, the environment is reconstructed using standard techniques. The user is able to reach out and touch surfaces in the world, and provide object category labels through voice commands. We propose a GPU-based volumetric mean-field inference algorithm that can efficiently propagate these user labels through the volume and provide a smooth segmentation that follows object boundaries. We also describe a new streaming decision forest approach that employs implicit surface-based volumetric features to learn from the propagated user labels. When the user encounters a previously unobserved and unlabeled region of space, the forest predicts object labels for each voxel, and the same mean-field inference smooths the final output. Our approach differs significantly from existing work: all parts of our pipeline, including low-level acquisition, user annotation, label propagation, feature computation, learning, classification and volumetric inference happen interactively on a single desktop machine. We demonstrate compelling results on several sequences, highlighting the smooth propagation of user labellings and the ability to learn and generalize to unseen regions of the world. In contrast to offline systems, the user receives almost instant feedback about the effect of their actions on the segmentation, allowing mistakes to be easily and quickly corrected.

6.1 Introduction

Imagine walking into a room with a body-worn consumer depth camera. As you move within the space, the dense 3D geometry of the room is automatically scanned in real-time. As you begin to physically interact with the room, touching surfaces and objects, the 3D model begins to automatically segment itself into meaningful regions, each belonging to a specific object category such as wall, door, table, book, or cup. As you continue interacting with the world, a learned model of each object category is updated and refined with new examples, allowing the model to handle more variation in object shape and appearance. When you start observing new instances of these learned object categories, perhaps in another part of the room, the 3D model will automatically and densely infer their labels

and segmentations, almost as quickly as the geometry is acquired. If the inferred labels are imperfect, the user can quickly and interactively both correct the labels and improve the learned model. It thus becomes possible to rapidly generate accurate, *labeled* 3D models of large environments.

Although a simple and intuitive vision of user interaction, this type of *scene understanding* scenario has remained challenging since the field of 3D reconstruction was first established over 50 years ago [137]. Approaches typically first perform an *offline acquisition phase* where data is recorded [63, 119] and labeled (often using crowd-sourcing techniques) [161]. The labeled data is then used for offline *batch* training of generative or discriminative models [85, 161]. This is followed by a *test phase* where the learned models are used for labelling semantic parts and even understanding support structures [161]. It is not unusual for existing systems to take hours or even days to train [158], and even the test phase can often take multiple seconds per image [79, 158] due to the use of expensive feature extraction and recognition algorithms. Furthermore, it is hard to know in advance how much labeled training data is required and how varied it should be; existing approaches therefore tend to conservatively capture large corpora. Even then, the learned model is not guaranteed to generalize well, and if so the whole process must begin again with further acquisition.

Our scenario described above requires somewhat of a paradigm shift: rather than offline data collection, labelling, and training, we need systems that can perform *all* parts of the pipeline, from low-level 3D reconstruction, through to semantic segmentation, training, and testing, interactively and in real-time. We present a new system that brings us closer to realizing such a challenging and yet compelling scenario. Our system is fully online, and capable of interactively acquiring semantically labeled 3D models of arbitrary indoor environments in minutes. A dense 3D model of the environment is captured by fusing noisy depth maps into an implicit volumetric surface representation [122]. Our system then allows the user to walk up to any object of interest, simply touch and stroke the physical surface, and vocally call out a new or existing object category name. This is lightweight to perform and places the user 'in the loop' during the learning process, perhaps allowing her to focus on objects that are of personal interest, rather than those predefined by application developers or dataset creators. Furthermore, minimal user labelling is required: if after initial labelling, the learned model does not generalize well to new object instances or 3D viewpoints, the user can quickly relabel parts of the scene as necessary and see the improved results almost instantaneously as the learned models are updated online.

Under the hood, our method first cleanly segments any touched object from its supporting or surrounding surfaces, using a new volumetric inference technique based on an efficient mean-field approximation. In the background, a new

form of streaming classification forest is trained and updated as new labeled object examples become available. The classifier can quickly infer the likelihood that any newly-observed voxel belongs to each object category. The final stage of our pipeline estimates a spatially consistent object labelling and segmentation of the voxel reconstruction by again performing mean-field inference over the voxel space but now using the results from the classifier. All parts of our pipeline (including the feature used during learning and inference) work directly on the volumetric data used for reconstruction, as opposed to requiring lossy and potentially expensive conversion of the data to depth image, point-cloud, or mesh-based representations.

Contributions. Our work builds on the existing body of research on 3D reconstruction, semantic modeling, and scene understanding in the following ways. Firstly, we present a general-purpose 3D recognition pipeline, which efficiently takes live fused 3D scene geometry as input, allows the user to segment and label surfaces interactively, continuously learns models of object categories, and performs classification and inference, all in an online manner. This, to our knowledge, is the first time that *all* parts of a 3D semantic pipeline have been shown to work online and in real-time, alongside reconstruction. Secondly, we present specific technical contributions associated with the following components: (i) an efficient mean-field algorithm for performing inference in a dynamically-changing volumetric random field model; (ii) an online learning algorithm called streaming decision forests that uses reservoir sampling to maintain fixed-length unbiased samples of streaming data; and (iii) robust 3D rotation-invariant appearance and geometric features that are computed directly on the volumetric data.

Our results demonstrate high-quality object segmentations on varied sequences (see Sec. 6.4). The segmentations are achieved with minimal user interaction: seconds suffice to label individual objects, and a reasonably sized room can be scanned in and fully labeled in just a few minutes. Our entire semantic processing pipeline can run at or very near 30 Hz on a commodity desktop PC with a single GPU. This chapter is a part of a bigger project which involved three researchers working on it. I was mainly involved with interactive segmentation, mean-field inference and features.

6.2 Related Work

Acquiring 3D models of the real-world is a long standing problem in computer vision and graphics, dating back over five decades [137]. Since then, offline 3D reconstruction techniques have digitized cultural heritage with remarkable quality [105, 198], and given rise to world-scale, Internet-accessible, 3D maps recon-

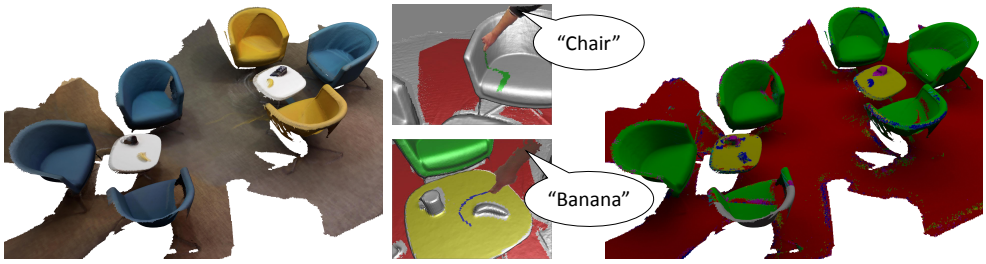


Figure 6.1: *Our system allows users to quickly and interactively label the world around them. The environment is scanned in using a hand-held depth camera, and in real-time a volumetric fusion algorithm reconstructs the scene in 3D (left). At any point the user can reach out and touch objects in the physical world, and provide object class labels through voice commands (middle). Then, an inference engine propagates these user-provided labels through the reconstructed scene, in real-time. In the meantime, in the background, a new, streaming decision forest learns to assign object class labels to voxels in unlabeled regions of the world. Finally, another round of volumetric label propagation produces visually smooth segmentations over the entire scene (right). Class transitions at object boundaries are correctly respected.*

structed using street-side [127], aerial [59] and online photo collections [147, 164]. More recently, *real-time* or *online* 3D scanning have emerged, with the rise of consumer depth cameras and general-purpose graphics hardware algorithms. Methods for real-time dense reconstructions, even over large physical scales, with only a single commodity depth or RGB camera have been demonstrated [27, 63, 119, 121, 122, 132, 142]. This has given rise to compelling new applications for real-time 3D reconstruction, live 3D scanning, physically-plausible augmented reality (AR), and autonomous robot or vehicle guidance, as well as more traditional offline applications including mapping, cultural heritage preservation, and 3D fabrication.

Beyond low-level geometry acquisition and reconstruction, a natural next step is to interpret higher level semantics from captured 3D scans. There is considerable interest and work on scene understanding and semantic modeling dating back as far as the reconstruction algorithm themselves, when primitive 'block world' representations were first devised [137]. Much work has happened in this space since. 2D techniques have been proposed that automatically partition an input RGB image into semantically meaningful regions, each labeled with a specific object class such as ground, sky, building, and so forth (e.g. [158]). Others have focused on geometric reasoning to extract 3D structure from single RGB images (e.g. [55]). We focus below on semantic modeling methods that operate with consumer depth cameras rather than purely 2D input, but refer the reader

to [191] for an excellent review of 2D approaches.

With the advent of affordable depth sensors, there has been growing interest in working with RGB-D input [31, 68, 135, 160, 161], as well as 3D point clouds [7,23,85,165], meshes [178] or voxel representations [57,71,73,144]. Methods have been proposed to break down meshes into semantic parts [28], localize objects in small scenes [1, 19, 71, 97, 107], or operate on larger indoor [75, 118, 144, 148, 178] or outdoor scenes [97, 108, 145]. There has also been a wide body of research on capturing large and compelling datasets which have moved from traditional 2D object images to RGB-D and full 3D scenes [48, 192, 193].

In the computer graphics literature there has been significant work on automatically segmenting 3D meshes into semantic parts [28, 69, 74, 149], including incremental depth camera-based methods [153]. Most of these methods consider only connected noise-free meshes, and geometric properties, ignoring the appearance. Furthermore, these techniques operate only on single objects, and do not operate in real-time. More recently, there has been relevant work on matching scan data to synthetic 3D model databases [75,118,148], with the aim to replace noisy point clouds with detailed CAD models. These approaches are compelling in that they increase final reconstruction fidelity and exploit repetition of objects to minimize the memory footprint. These systems first perform automatic or interactive segmentation of the scene into constituent parts which are then individually matched to the model database. However, these techniques require a model database to be built and learned offline, and the test-time matching techniques can take seconds to minutes to perform.

[144] takes this concept a step further, by building an online SLAM system that can recognize objects and update the model live. However, the model database is still captured and generated offline. Only a single object class (chair) is recognized and it is unclear how the system can support larger surfaces such as floors, walls and ceilings. However, this system demonstrates the power of semantic recognition alongside the reconstruction process, improving relocalization, memory efficiency, and loop closure. This type of semantic information has also been explored in the context of bundle adjustment [45], and extended to sparse map representations [24, 134].

For outdoor scene labelling, much of the work has concentrated on classification of images [23, 97, 130]. [57, 145] generate a dense semantic 3D reconstruction but the object labelling is performed in the image domain, and then projected to the final models. As a result, these methods cannot fully exploit the 3D geometry of the scene in their inference. [108] decomposes outdoor scenes into semantic parts and employs 3D model matching techniques similar to [75,118,148] to create reconstructions from LiDAR data. None of these systems operate in a real-time or in an online manner.

[31, 68, 135, 160, 161] attempt to label indoor scene images captured using RGB-D sensors. Classification or recognition is performed in image-space, along with 3D priors to aid segmentation. Again these systems fail to exploit full 3D geometry and are the counterpart of image-based segmentation but for RGB-D frames. [178] exploits 3D meshes and geometric and appearance features for improved inference in outdoor and indoor scenes. [73] use a voxel-based conditional random field (CRF) for segmentation and occupancy-grid based reconstruction. However, these techniques are not efficient enough to be used in an online system, and operate only on coarse reconstructions.

Our approach differs from these systems in several compelling ways. Firstly the system runs entirely online and in real-time, including data capture, feature computation, labelling, segmentation, learning and inference. Second, our pipeline leads to robust and dense object recognition alongside acquisition. Finally, in our system, the user is 'in the loop' during the labelling and training process, allowing the system to evolve to new object classes in an online fashion, and allowing the user to label a minimal amount and correct any mistakes interactively. This allows the user to rapidly build up models personalized to their spaces and goals.

6.3 System Pipeline

We now describe the overall architecture of our system, as illustrated in Figure 6.2, specifying its key components and how they interact with each other. The detailed operation of the components will be presented in subsequent sections of the paper.

3D model acquisition engine. The first component of our system is the 3D model acquisition engine. This component takes the color and depth image frame stream coming from the RGB-D sensor and fuses it on the GPU to generate a 3D model. We adopt the truncated signed distance function (TSDF) of [142] to fuse depth images into a 3D volume as was done in KinectFusion [63, 119]. To handle large-scale scenes we employ the hashed volumetric representation from [122].

User interaction. Our system allows for two modes of user interaction: touch-based interaction using hands and feet, and voice based commands. We detect touch by looking for large components in the observed depth image that differ from the ray-cast model and come close to the implicit surface [63]. This allows the user to draw strokes on real-world objects and have those strokes appear as labels on screen. We use a standard voice recognition system to input labels such as 'table', 'chair', etc., and to recognize commands such as 'training mode' or

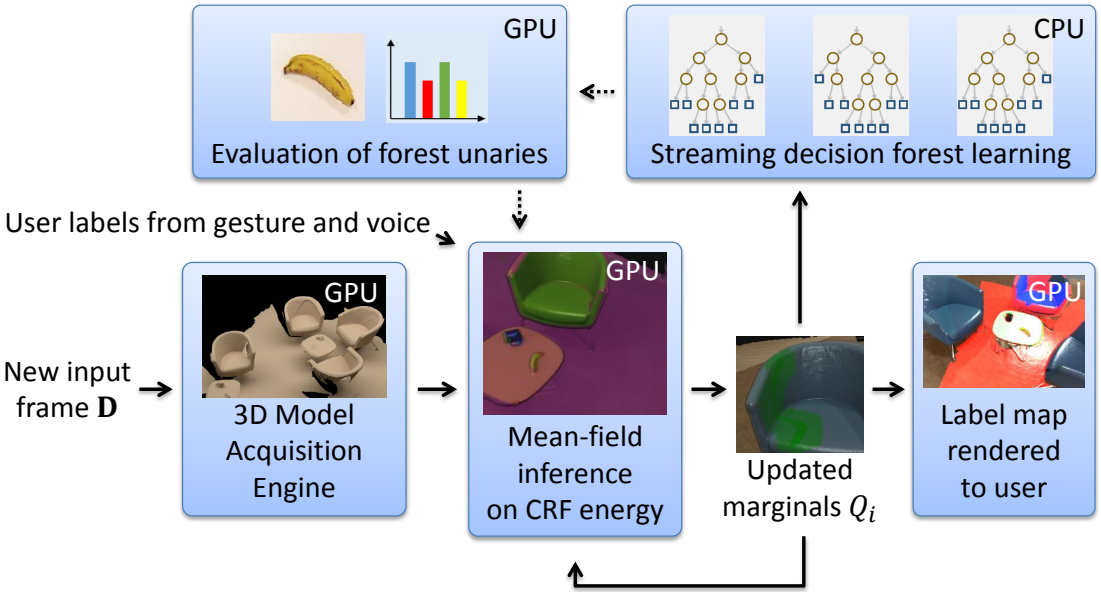


Figure 6.2: Overview of the recognition pipeline. See text for details.

'test mode'.

Mean-field inference on CRF energy. At the heart of our semantic labelling approach is an efficient mean-field inference engine [87] applied to a dynamic conditional random field (CRF) model. The inference algorithm computes a per-voxel approximate posterior distribution Q_i over the set of object labels that the user has provided. These distributions can be rendered as label maps and provided to the user as feedback. Both the user specified interaction hints and the predictions coming from the classification forest (see below) are integrated into the CRF through unary likelihood functions that operate on individual voxels. Pairwise terms in the CRF model ensure a smooth segmentation and allow the user labels to propagate outwards to object boundaries in the scene. In our application, the unary likelihoods and thus the CRF model changes dyamically from one frame to the next as (i) more data is acquired and the 3D model is updated, and (ii) the user continues to interact and specifies further labellings. We show how the mean-field updates can be applied to such dynamic environments, can be implemented on the GPU, and the computational cost can be amortized over multiple video frames to enable an efficient implementation. This ensures that super real-time speeds can be maintained (one update of the messages requires ∼6 milliseconds), and as a by-product, results in a visually pleasing 'label propagation' effect.

Streaming decision forest classifiers. Our system operates in two modes: training and test. The user can switch between these using voice commands. During the training mode, the labels that result from running the mean-field inference given the user-specified labels are fed into a streaming decision forest algorithm

that learns to predict likelihoods for assigning object labels to unlabelled parts of the 3D scene. The streaming decision tree algorithm can continue to adapt in the background given updates to the reconstruction volume and any newly provided user labels. The forest employs a new type of 3D rotation invariant appearance feature that can be efficiently computed directly from the TSDF volume. In test mode, the forest is evaluated in parallel on the GPU for every visible voxel, and the results are used to update the unary likelioods in the CRF. The mean-field inference then in turn produces a smoothed output to display to the user.

In the next sections, we first detail the volumetric mean-field inference and user interaction in Sec. 6.3.1, before describing the online learning in Sec. 6.3.2.

6.3.1 Smoothly Segmenting the Volume

In this section we describe the volumetric segmentation algorithm used to generate the smooth results that are presented to the user. Our results depend both on labels provided interactively by the user, and on inferences made by the learned classifier (described later in Sec. 6.3.2). We formulate the problem of assigning semantic labels to voxels using a pairwise Conditional Random Field (CRF) [20]. While pairwise CRFs have been widely used for image labelling problems such as image segmentation, stereo and optical flow, a key distinguishing feature of our formulation is its *dynamic* nature that requires a special purpose inference routine. This allows us to deal with a continuously changing underlying 3D model (as more depth frames are fused), new user provided labels, and an on-the-fly trained random forest predictor based likelihood function.

6.3.1.1 Dynamic Conditional Random Field Model

Each voxel i in the 3D reconstruction volume (denoted by $\mathcal{V}$) of the scene is represented by a discrete random variable x_i that represents the semantic class (e.g. floor, wall, table, mug) that the voxel belongs to. Note that the choice and number of labels will depend on the interactive user input. The posterior distribution over the labelling of voxels under the pairwise CRF factorizes into likelihood terms ψ_i defined over individual voxels and prior terms ψ_{ij} defined over pairs of random variables. The posterior is formally written as

$$P(\mathbf{x}|\mathbf{D}_t) \propto \prod_{i \in \mathcal{V}} \psi_i(x_i) \prod_{(i,j) \in \mathcal{E}} \psi_{ij}(x_i, x_j) \tag{6.3.1}$$

where $\mathbf{x}$ is the concatenation of the x_i for all $i \in \mathcal{V}$, $\mathbf{D}$ is the volumetric data at time t, and set $\mathcal{E}$ with $i \in \mathcal{V}$ and $j \in \mathcal{V}$ defines the neighbourhood system of the

random field[1]. To encourage smoother results, we employ a large neighbourhood system which densely includes all voxels within a 6cm radius. This can be handled efficiently by our GPU-based volumetric mean-field inference algorithm described below.

An equivalent but perhaps more convenient definition can be reached by taking the negative log of the posterior. This gives the *energy* of the labelling under the CRF as:

$$E_t(\mathbf{x}) = \sum_{i \in \mathcal{V}} \phi_i(x_i) + \sum_{(i,j) \in \mathcal{E}} \phi_{ij}(x_i, x_j) + K \tag{6.3.2}$$

where ϕ_i encode the cost of assigning label x_i at voxel i, ϕ_{ij} are pairwise potentials that encourage neighbouring (i.e. $(i,j) \in \mathcal{E}$) voxels to take the same label, K is a constant, and the conditioning on the data $\mathbf{D}$ is now implicit.

Note that the unary and pairwise terms in (6.3.2) will be constantly changing in our dynamic CRF. This is because (i) the volumetric reconstruction is constantly being updated with new observations, and (ii) the user is interacting with the environment and providing new labels. We next provide more details of the particular forms of these dynamic potentials.

Transition-sensitive smoothness costs. For our application, we employ a standard Potts model for the pairwise potentials, defined as:

$$\phi_{ij}(l, l') = \begin{cases} \lambda_{ij} & \text{if } l \neq l' \\ 0 & \text{otherwise.} \end{cases} \tag{6.3.3}$$

In the 2D segmentation domain, the cost λ_{ij} of assigning different labels to neighbouring pixels is generally chosen such that it preserves image edges [20, 139]. Inspired from these edge-preserving smoothness costs, we make the label discontinuity cost λ_{ij} dependent on a number of appearance and depth features:

$$\lambda_{ij} = \theta_{\mathrm{p}} e^{-\|\mathbf{p}_i - \mathbf{p}_j\|_2} + \theta_{\mathrm{a}} e^{-\|\mathbf{a}_i - \mathbf{a}_j\|_2} + \theta_{\mathrm{n}} e^{-\|\mathbf{n}_i - \mathbf{n}_j\|_2} \tag{6.3.4}$$

where $\mathbf{p}_i$, $\mathbf{a}_i$ and $\mathbf{n}_i$ are respectively the 3D world coordinate position, RGB appearance, and surface normal vector of the reconstructed surface at voxel i, and $\theta_{\mathbf{p}}$ $\theta_{\mathbf{a}}$ and $\theta_{\mathbf{n}}$ are hand-tuned parameters. Observe that as the 3D model is updated from one frame to the next, the appearance and surface normals associated with the voxels change. The energy landscape thus continuously changes over time.

[1]We have not listed the data $\mathbf{D}$ as an argument in the potential functions ψ_i and ψ_{ij} for the sake of a simpler and uncluttered exposition.

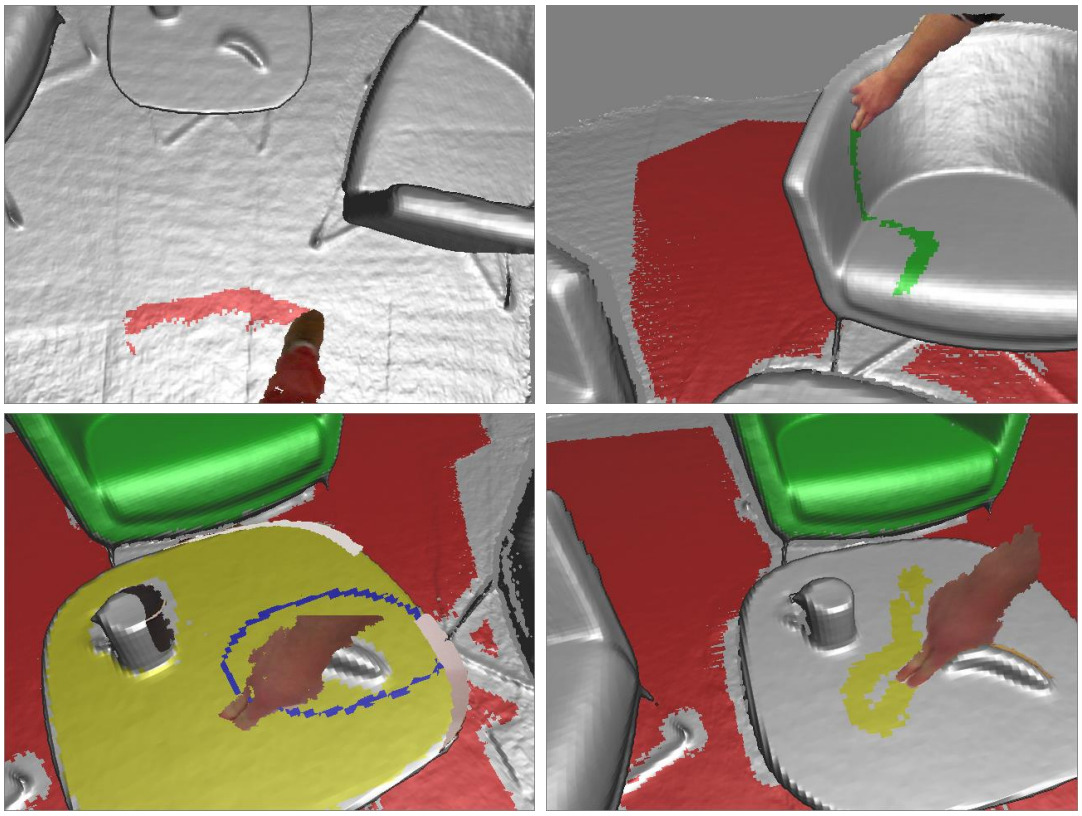


Figure 6.3: Our system allows the user to reach out and touch the world to provide object labels. We support hand and foot interactions, and both 'stroke' and 'enclose' gestures. These labels are then propagated through the volume; see Fig. 6.6.

Initialization of the Unary Costs The unary potentials ϕ_i of the CRF model defined above are *initialized* for all $i \in \mathcal{V}$ by specifying a fixed cost that initially encourages all voxels to take the background label:

$$\phi_i(l) \leftarrow \begin{cases} 0 & \text{if } l \text{ is the background label} \\ \theta_{\text{bg}} & \text{otherwise.} \end{cases} \tag{6.3.5}$$

Note that each unary can be thought of as a *table* of values (one entry for each $l \in \mathcal{L}$). During operation of our system, the entries in these tables are gradually replaced based on user interactions and predictions from the streaming random forest based classifier that encodes the cost for a voxel being assigned a particular label, as described next.

User 'Stroke' Interactions. As illustrated in Fig. 7.1, our system allows two types of user interaction gestures for labelling the world. The first user interaction supported is a 'stroke' gesture. The user reaches out, and touches the surface of

the object they want to label. Note that the user need not precisely label every voxel belonging to the object, and can instead roughly mark a small set of voxels. We denote the set of voxels that the user has stroked as $\mathcal{H}_{\mathrm{S}}$. The user also specifies a semantic label l_{S} through speech recognition, and can specify an existing label or a new one, in which case the label set $\mathcal{L}$ is enlarged. These labellings are used to update the unary potentials to enforce that these voxels take the specified semantic label. The update is applied to all voxels $i \in \mathcal{H}_{\mathrm{S}}$ as

$$\phi_i(l) \leftarrow \begin{cases} 0 & \text{if } l = l_{\mathrm{S}} \\ \infty & \text{otherwise.} \end{cases} \tag{6.3.6}$$

Multiple such labellings with different regions $\mathcal{H}_{\mathrm{S}}$ and labels l_{S} can be provided in sequence. In the case of incorrect propagation due either to wrongly placed user strokes or to problems with the mean field inference, the user can specify another stroke labelling. This will overwrite any existing labelling and thus update the unary and allow the new label to propagate.

User 'Enclose' Interactions. The second form of user interaction supported is the 'enclose' gesture, where the user reaches out and draws a rough enclosing circle around the object they want to describe. This interaction mode is most useful for small objects such as bananas and pens. Again, voice is used to provide the semantic label l. To obtain an accurate labelling, we follow an approach similar to [139]. First, the user annotation is projected into the current frame's input image. On the CPU, a Gaussian mixture model (GMM) is fit to the colors in the foreground and background regions. Foreground is taken as the interior of the convex hull of the user annotations, and background as the rest of the image. Then we transfer the GMMs to the GPU where a shader computes in parallel the foreground probability

$$P_{\mathrm{E}}(\mathrm{fg}|\mathbf{a}_i) = \frac{P(\mathbf{a}_i|\mathrm{fg})}{P(\mathbf{a}_i|\mathrm{fg}) + P(\mathbf{a}_i|\mathrm{bg})} \tag{6.3.7}$$

based on the voxel's foreground and background color likelihoods (and assuming a uniform prior). This is computed for all voxels i in a bounding volume surrounding the user annotations.

Inference is performed within this bounding volume to estimate which of the two labels (foreground-background) is most likely to be assigned to each voxel . This is used to update the unary cost in the energy defined over the full 3D

model as

$$\phi_i(l) \leftarrow \begin{cases} 0 & \text{if } l = \hat{l}_i \\ \infty & \text{otherwise} \end{cases} \tag{6.3.8}$$

Learned Class Predictions There is one final source of updates to the unary costs. We describe below in Section 6.3.2 how a decision forest is able to learn and make predictions about object categories in newly observed regions of the world that are not hand-labeled. The output of the forest is a prediction of the distribution $P_{\mathrm{F}}(x_i = l \mid \mathbf{D})$ over semantic labels $l \in \mathcal{L}$ at voxel i. For voxels i that have not been hand labeled using either of the interactions described above or label-propagation by mean field, this distribution is used to update the unary likelihood costs as:

$$\phi_i(l) \leftarrow -\log P_{\mathrm{F}}(x_i = l \mid \mathbf{D}) \ . \tag{6.3.9}$$

6.3.1.2 Efficient Mean-Field Inference

Given the unary and pairwise terms defined above, the labelling can be propagated through the volume by inferring the optimal labelling $\mathbf{x}$ given the pairwise energy functions. The Maximum a Posteriori (MAP) labelling for pairwise energy functions such as the one defined in equation (6.3.2) could be computed using standard graph cut based move making algorithms like α-expansion and $\alpha\beta$-swap. However, these algorithms are intrinsically sequential, and it is hard to tailor them to high throughput architectures like GPUs without significant engineering overheads [179].

Instead, we propose an online volumetric mean-field inference framework that efficiently infers the approximate maximum posterior marginal (MPM) solution of our energy. While based on [87], we make two key technical contributions. Firstly we show how such a volumetric energy minimization can be implemented efficiently on the inherently parallel architecture of the GPU. Secondly, we exploit the fact that the energy landscape usually changes only gradually from one frame to the next. This allow us to amortize the optimization cost over multiple frames. This not only enables the system to run at a high frame rate, but also results in a pleasing result to the user as user labels appear to gradually propagate out from the initial strokes until they accurately delineate the objects boundaries based on the energy function (6.3.2) of the model.

The mean-field optimization [87] proceeds as follows. We introduce a probability distribution $Q(\mathbf{x})$ that approximates the original distribution $P(\mathbf{x})$ (6.3.1) under the KL-divergence $D(Q||P)$. Further, we choose a factorized distribution $Q(\mathbf{x})$ such that the marginal of each random variable is independent, i.e. $Q(\mathbf{x}) = \prod_i Q_i(x_i)$. Taking the fixed point solution of the KL-divergence [81], we

obtain the following mean-field update:

$$Q_i^t(l) \leftarrow \frac{1}{Z_i} e^{-M_i(l)} \tag{6.3.10}$$

$$M_i(l) = \phi_i(l) + \sum_{l' \in \mathcal{L}} \sum_{j \in \mathcal{N}(i)} Q_j^{t-1}(l') \phi_{ij}(l, l') \tag{6.3.11}$$

$$Z_i = \sum_{l \in \mathcal{L}} e^{-M_i(l)} \tag{6.3.12}$$

where $l \in \mathcal{L}$ is a label taken by random variable x_i, $Q_i^t(l)$ denotes the marginal probability at iteration t of variable x_i taking label l, $\mathcal{N}_i$ denotes the set of neighbours of i (i.e. $j \in \mathcal{N}_i \Leftrightarrow (i, j) \in \mathcal{E}$), and Z_i normalizes the distribution. After iterating the updates (6.3.10) to iteration T, the output MPM estimates can be obtained as

$$x_i^\star = \operatorname*{argmax}_{l \in \mathcal{L}} Q_i^T(l) \ . \tag{6.3.13}$$

Given unlimited computation, one might run multiple update iterations until convergence.[2] However, in our online system, we assume that the next frame's updates to the volume (and thus to the energy function) are not too radical, and so we can make the assumption that the Q_i distributions can be temporally propagated from one frame to the next, rather than re-initialized (e.g. to uniform) at each frame. Thus, running even a *single iteration* of mean-field updates per frame effectively allows us to amortize an otherwise expensive inference operation over multiple frames and maintain real-time interactive speeds. Note that this effectively means that the t variable above becomes the frame number and the mean-field iteration count. Furthermore, this approach naturally results in the almost magical way the inferred labels appear to propagate through the world from one frame to the next.

Integration of Forest Predictions. As described above, the output of the online decision forest (Section 6.3.2) is used to update the unary distributions, which will, over several frames, impact the final segmentation that results from the mean field inference. However, to speed up convergence, we propose an additional step that exploits our temporal propagation of the Q distributions. Rather than simply propagating the Q_i^{t-1}s from the previous frame, we instead provide the next iteration of mean-field updates with a weighted combination of

[2] If applied sequentially on a fixed energy function, the mean-field inference comes with some convergence guarantees [87]. These do not apply to our algorithm as our algorithm is dynamically updated in each frame, but this does not appear to be a problem in practice.

Q_i^{t-1} and the forest prediction $P_{\mathrm{F}}(x_i = l \mid \mathbf{D})$. We thus use

$$\bar{Q}_i^{t-1}(l) = \gamma Q_i^{t-1}(l) + (1-\gamma) P_{\mathrm{F}}(x_i = l \mid \mathbf{D}) \tag{6.3.14}$$

in place of the Q_i^{t-1} in (6.3.11), where γ is a weighting parameter. In practice, this step appears to result in considerably quicker updates to the segmentation result given the forest predictions.

6.3.1.3 Implementation on the GPU.

To achieve real time performance, we run the mean-field inference on the GPU.

Memory Model A voxel in the 3D volume is associated with a number of different quantities. In order to maintain proper synchronization between different types of labels and considering memory requirements on GPUs, for a voxel we store all these different types of labels in a single 32 unsigned bit. First 8 bits are assigned for labels inferred by random forest and mean-field inference, 2nd 8 bits are used by label propagation/obj-field. Further we store the user stroke ids and stroke masks (refer to Fig.). So during label propagation, we write to the second 8-bits and during random-forest-MF prediction stage, we use the first 8 bits. So, at any time, label propagation and RFMF do not write to the same memory location. Further, at a time, only one process happens, i.e. during label propagation stage we switch off the random forest based prediction and during RFMF prediction stage label propagation is switched off. In our current system, the labels predicted during RFMF stage can be modified and so corrected by label propagation stage, but the reverse is not true.

Execution Model Each GPU thread independently computes the $\bar{Q}_i^t$ distribution in parallel for all voxels i based on the previous frame's estimates $\bar{Q}_i^{t-1}$ and the forest probabilities $P_{\mathrm{F}}(x_i \mid \mathbf{D})$. We employ three GPU shaders to achieve this which are executed in sequence. The first shader calculates $\bar{Q}$ according to (7.2.8). The second shader applies the mean-field update (6.3.10) to the class probabilities of each voxel. This entails looking up the unary and adding in the pairwise terms for for all neighbours of the voxel. While previous methods such as [87] have considered fully connected CRFs, a reasonable trade-off between accuracy and speed was achieved by using a radius of 6cm to determine the neighbourhood. The third shader finally evaluates the MPM solution using (7.2.7) for the current time t.

Each thread is processed independently, and the computation required is proportional to the neighbourhood size times the number of classes. We do other

optimizations to improve the speed of mean-field inference on the GPU. These take the form of adaptive scheduling of message updates. First, we do not apply the mean-field update for voxels where the probability for a particular class is very high as this is unlikely to change the final labelling. Second, we do not perform the mean-field updates for voxels that lie away from the surface (i.e. outside the truncation region of the TSDF). Finally, we look at the neighbours only if they lie within the truncation region. All these updates are important to enable inference in real time.

6.3.2 Learning to Segment the Dynamic World

The previous section detailed our efficient mean-field inference that, given a set of unary and pairwise potentials, optimizes an energy function to produce spatially-smooth segmentations. We have seen how the unary potentials in (6.3.2) are initially encouraged to be background (6.3.5), and later are updated based on user interactions (6.3.6, 6.3.8). This section details how we can learn and infer distributions $P_{\mathrm{F}}(x_i = l \mid \mathbf{D})$ that can be used as another form of unary (6.3.9) to allow the CRF inference to automatically predict smooth segmentations for *unlabeled* parts of the world. We apply a new approach called *streaming decision forests* to this task. These forests are extremely fast both to train and test, and are capable of learning online from a stream of labeled training voxels, and being updated to correct for mistakes. Another contribution in this section is a new set of features called voxel-oriented patches (VOPs). VOPs can be efficiently computed from the raw TSDF volume and thus avoid the expense of computing an explicit surface reconstruction. They are also designed to be 3D rotation invariant allowing the inference to generalize well across 3D object rotations in the world. The following sections elaborate on our approach.

6.3.2.1 Decision Forests

Decision forests [22, 32] have proven successful for many applications. Typically trained offline with large datasets, forests can learn to recognize the trained object-classes in new scenes. However, we cannot hope to employ such offline learning approaches to our scenario due to the often long training times (typically hours or even days), and our desire to allow interactive updates and corrections to the classifier. We thus turn to online forest learning [17, 37, 143]. While typically less accurate than an offline-trained forest, online learning supports incremental (and thus more interactive) updates given new or improved training data, and can require less memory to train. Our new streaming decision forests framework extends [143] to use reservoir sampling [182]. This allows us to maintaining a

fixed-size unbiased sample of all training data seen so far and avoid discarding samples observed early on during training, resulting in faster and more accurate classifiers. Our approach also requires considerably less memory to train.

We first briefly review the standard offline (or 'batch') decision forests (see e.g. [32] for more details), before describing [143], and finally our new streaming decision forest algorithm.

Offline Forests. A forest comprises an ensemble of decision trees. Each tree comprises binary split nodes and leaf nodes. At test time, starting at the root, a left/right decision is made for voxel i according to the evaluation of a binary split function $f(i;\theta) \in \{\mathrm{L},\mathrm{R}\}$ with learned parameters θ. As described in Sec. 6.3.2.3, parameters θ are used to specify the particular features used at this node. According to the result of this function, the left or right child branch is followed, and the process is repeated for each split node encountered, until a leaf node is reached. At the leaf node a stored distribution $P_{\mathrm{F}}(x_i = l \mid \mathbf{D})$ is looked up and used to update the voxel's unary as described above.

Each tree is trained independently (offline) on subsets of the training data. A set $\mathcal{S}_n$ of examples is provided to the root node $n = 0$. Example set $\mathcal{S}$ consists of example pairs (i, l) where i represents the training voxel index, and l is the associated class label provided by user interaction. A set Θ_n of candidate binary split function parameters θ is proposed randomly. Each candidate split induces a partitioning of the set of examples into left and right subsets, $\mathcal{S}_n^{\mathrm{L}}(\theta)$ and $\mathcal{S}_n^{\mathrm{R}}(\theta)$. We can compute the *information gain* objective as

$$G(\mathcal{S}, \mathcal{S}^{\mathrm{L}}, \mathcal{S}^{\mathrm{R}}) = H(\mathcal{S}) - \sum_{d \in \{\mathrm{L},\mathrm{R}\}} \frac{|\mathcal{S}^d|}{|\mathcal{S}|} H(\mathcal{S}^d) \ , \tag{6.3.15}$$

where $H(\mathcal{S})$ is the Shannon entropy of the distribution of labels l in $\mathcal{S}$. The (locally) optimal parameters can then be obtained as the maximization

$$\theta_n := \operatorname*{argmax}_{\theta \in \Theta_n} G(\mathcal{S}_n, \mathcal{S}_n^{\mathrm{L}}(\theta), \mathcal{S}_n^{\mathrm{R}}(\theta)) \ . \tag{6.3.16}$$

Having found the best split parameters for node n, the tree growing proceeds greedily for the left and right children until a predefined stopping criterion is reached.

Online Random Forests. We next review the online forest algorithm of [143]. Given a current tree structure (initially this will be a single root node), any incoming observations (i, l) are passed down the tree until they reach a leaf node. At each leaf there now exists a set Θ_n of candidate features, and for each feature

$\theta \in \Theta_n$, a set of left and right class label statistics are stored. These histogram statistics are sufficient to compute the information gain (6.3.15). Based on the output of $f(i;\theta)$, the new observation's label l is used to update the relevant (i.e. left or right) statistics for each θ. If the updated leaf has now seen a large enough number of observations, and if at least one of the candidate features gives a good local optimum (6.3.16), then the leaf will be split, and two child nodes with their statistics created. Hoeffding trees [37] work in a similar way, but use a Hoeffding bound to decide whether or not to split.

Streaming Decision Forests. We propose a new method called the 'streaming decision forest'. This can be seen as an extension of [143] with reservoir sampling. Reservoir sampling [182] is a method of storing an unbiased sample of a fixed maximum size from a stream of data. If the incoming data were I.I.D., then we could of course simply store the first K samples and we would be done. However, in practice our stream of training data is very much not I.I.D., due in part to considerable correlation between one frame and the next. Using a reservoir accumulated over a potentially large temporal window of observations can thus smooth out any imbalance in the distribution of incoming samples and thus improve the quality of the classifier.

A reservoir maintains a list $\mathcal{T}$ of at most K examples, and a count m of the total number of examples observed so far. Given a new labeled observation (i, l), if $m < K$ then the example is simply appended to $\mathcal{T}$. If instead $m \geq K$ and thus the reservoir is full, then a uniform random integer $k \in \{1, \ldots, m\}$ is chosen. If $k \leq K$ then reservoir entry k is replaced by (i, l), otherwise the observation is discarded. Finally, m is incremented, ready for the next observation. Note that the probability of retaining a new sample is $\frac{K}{m}$, and so as m increases we are more likely to discard new observations. However, in the limit of M samples, the probability of any individual sample remaining in the reservoir is exactly uniform at $\frac{K}{M}$.

In [143], at each current leaf node n, it was necessary to store a set of class label statistics at the potential left and right *children* that would result for *every* candidate split function. We propose instead to store a *single* reservoir $\mathcal{R}_n = (\mathcal{T}_n, m_n)$ at each potential *parent* (i.e. current leaf), which saves considerable memory. By storing the count m_n of observations separately, the reservoir allows us to maintain a fixed maximum size set $|\mathcal{T}| \leq K$ of unbiased samples of the incoming observations, where K is a parameter of the method. We show below how the reservoir representation allows us to efficiently compute a good approximation of the information gain (6.3.15) from only a small subset of the observations.

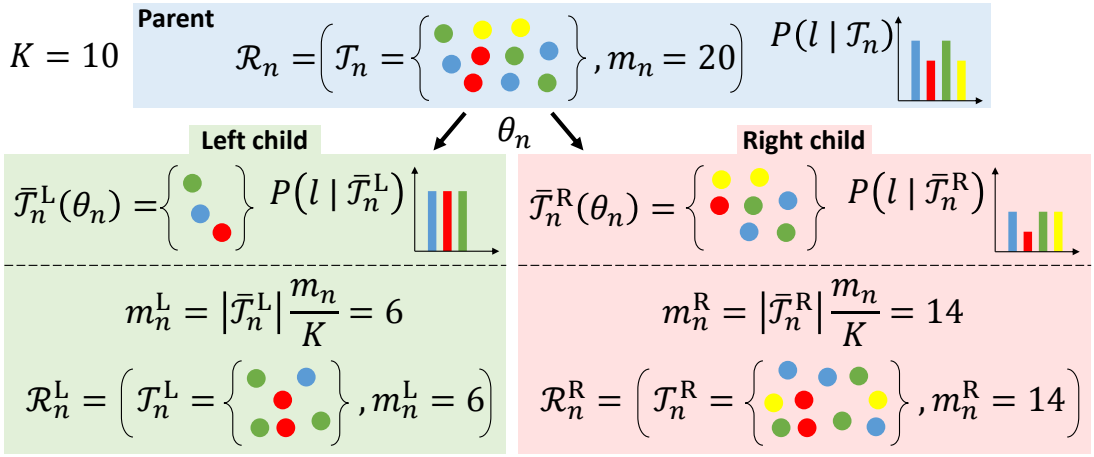


Figure 6.4: A toy example of splitting a reservoir. Top: a reservoir $\mathcal{R}_n$ of capacity $K = 10$ that has observed $m_n = 20$ samples. The dots represent examples i and their colors the labels l. The empirical distribution $P(l|\mathcal{T}_n)$ is computed from the labels (colors) in $\mathcal{T}_n$. Bottom, above dashed line: For any setting of the split parameters θ_n, the list $\mathcal{T}_n$ can be efficiently partitioned into two halves, $\bar{\mathcal{T}}_n^d$ for $d \in \{\mathrm{L}, \mathrm{R}\}$. Given these, the corresponding empirical distributions can be computed as before. Bottom, below dashed line: Once the optimal θ_n has been chosen, we can resample to generate child reservoirs: the child counts m_n^d are computed, and then m_n^d samples from $\bar{\mathcal{T}}_n^d$ are drawn and added to the new reservoir $\mathcal{R}_n^d$.

Optimizing the objective. When it is decided to split node n, we must evaluate the objective (6.3.15) with the example sets $\mathcal{S}$s now replaced by reservoirs $\mathcal{R}$s. This requires a sweep through reservoir $\mathcal{R}_n$ for each $\theta \in \Theta_n$. This can be performed efficiently, as follows.

The procedure is illustrated in Fig. 6.4. At the parent n we simply define $|\mathcal{R}_n| = m_n$, and the distribution required to evaluate the entropy $H(\mathcal{R}_n)$ can be calculated by normalizing the histogram of class labels l in $\mathcal{T}_n$. The remaining terms in the objective are $|\mathcal{R}_n^d|$ and $H(\mathcal{R}_n^d)$ for each child $d \in \{\mathrm{L}, \mathrm{R}\}$. These can be computed *without explicitly computing the sub-reservoirs* $\mathcal{R}_n^d$ (cf. 'Splitting the reservoirs' below, and compare above and below the dashed line in Fig. 6.4). First, the examples $(i, l) \in \mathcal{T}_n$ are partitioned into left and right subsets $\bar{\mathcal{T}}_n^d$ using split function $f(i; \theta)$. We can then efficiently compute the 'effective value' for child count $|\mathcal{R}_n^d|$ as

$$\bar{m}_n^d = |\bar{\mathcal{T}}_n^d| \max(1, \frac{m_n}{K}) \ , \tag{6.3.17}$$

and compute the entropies $H(\mathcal{R}_n^d)$ from $P(l|\bar{\mathcal{T}}_n^d)$, the normalized histogram of labels l in $\bar{\mathcal{T}}_n^d$.

Splitting the reservoirs. Fortunately, the slight extra cost of the above sweep compared to [143] does come with a further benefit beyond reduced memory consumption. Having chosen the optimal θ_n based on (6.3.15), rather than throw away the statistics stored at the node (as done in [143]), we can instead split the reservoir $\mathcal{R}_n$ into sub-reservoirs $\mathcal{R}_n^d$. For each of $d \in \{\mathrm{L}, \mathrm{R}\}$, we start with a new, empty sub-reservoir $\mathcal{R}_n^d$ of capacity K. We first compute the partitions $\bar{\mathcal{T}}_n^d$ as described above. To the nearest integer, the number of examples in $\mathcal{R}_n^d$ should be $m_n^d = \lfloor |\bar{\mathcal{T}}_n^d| \max(1, \frac{m_n}{K}) + 0.5 \rfloor$. We thus draw m_n^d random samples (with replacement) from $\bar{\mathcal{T}}_n^d$, and add each into the new reservoir $\mathcal{R}_n^d$ in the standard fashion. Except for rounding errors, this gives us the reservoirs containing unbiased sets of examples for the left and right children.

Advantages of reservoirs. The ability not to throw the statistics away gives us a considerable head start compared to [143] in which each new node must start with an initially empty set of statistics after being created. This in turn means that the new nodes can themselves be split much sooner with fewer additional observations. Furthermore the reduced memory consumption gives the potential for the algorithm to scale to larger trees. In practice for our scenario, and in experiments on standard classification datasets, we observed the new approach to give a considerable improvement in accuracy.

On average, our CPU implementation of the streaming decision forest training process is able to process the addition of 10000 new samples in 400 ms per tree. In order not to stall the interactive system while the learner is updating its structure, the learning process is running as a background task, and can make use of all the available CPU cores. We believe that a careful GPU implementation [150] could drastically speed-up the learning process.

Sampling Training Voxels. In order for the online classifier to update its structure and the distributions stored in the leaves, training samples must be fed to the classifier. We sample at random an equal number of voxels for each class from regions of the volume that have been hand labeled. Each voxel is assigned its current label as given by the mean-field inference, allowing us to learn both from the explicitly user-labeled regions ($\mathcal{H}_\mathrm{S}$ and $\mathcal{H}_\mathrm{E}$) and also from the labels that have been propagated by the CRF inference.

Given the voxels in the current view frustum, an equal number of random samples of each class are extracted. Samples are extracted in small batches spaced by constant time steps, except for when the user is interacting with the world.

6.3.2.2 Voxel-Oriented Patch Features

The ability of the decision forest to learn to distinguish different object categories depends on the availability of discriminative features. A variety of features have been used with great success in the past. For 2D applications, these have included pixel comparisons [103], texton region integrals [158], and invariant descriptors such as SIFT [111] and HOG [36]. Features that describe 3D point clouds and meshes have also been proposed, including rotation invariant spin images [65], SfM point-cloud derived features [23], and difference of normals [62]. Such features are typically designed with certain invariances in mind, including additive photometric invariance, and 2D or 3D rotation invariance. Such invariances can help the classifier by reducing the amount of training data required (though, too much invariance can lead to loss of discriminative power).

For our real-time application, speed is crucial, and given the dynamic nature of the scene, we do not have the time to first extract a mesh or point-cloud before computing features. As one of our key contributions, we thus propose a new type of feature, the *voxel-oriented patch* (VOP) that can be efficiently computed *directly* from the TSDF volume, and are 3D rotation invariant. Our implementation of these feature can quickly handle the computation of features for millions of voxels.

For a voxel i of interest at position $\mathbf{p}_i$, a VOP V_i is extracted as follows; see also Fig. 6.5. The voxel's normal $\mathbf{n}_i$ is calculated directly from the gradient of the TSDF values, and defines a plane $(\mathbf{p} - \mathbf{p}_i) \cdot \mathbf{n}_i = 0$. Choosing an arbitrary vector on this plane as an initial 'x axis' and third, orthogonal vector as a 'y axis', we form an image patch of size $r \times r$ that contains the color values stored in the TSDF on the plane. To reduce the effect of illumination (especially specularities) we employ the CIELab color space, storing the raw L component as well as $a/(a+b)$ and $b/(a+b)$. Note that the resolution of the patch size (i.e. mm per pixel) can be independent of the resolution of the TSDF reconstruction volume; our current implementation uses $r = 13$ with a resolution of 20 mm per pixel.

To achieve rotation invariance, we compute a histogram of intensity gradients in the image patch, and rotate the patch to align with the strongest gradient orientation, in precisely the same way as SIFT [111]. Note also that the original patch and the rotated patch need never be explicitly computed - the split functions (see below) only ever look at pairs of VOP pixels which can be computed on the fly from the TSDF.

We end up with a VOP V_i that can now store discriminative information about the local appearance around voxel i with full 3D rotation invariance. Examples are given in Figure 6.5. Note that these features exploit the fact that the TSDF volumetric integration process 'spreads' the color information along the camera ray within the truncation window, such that the voxel i does not need to be pre-

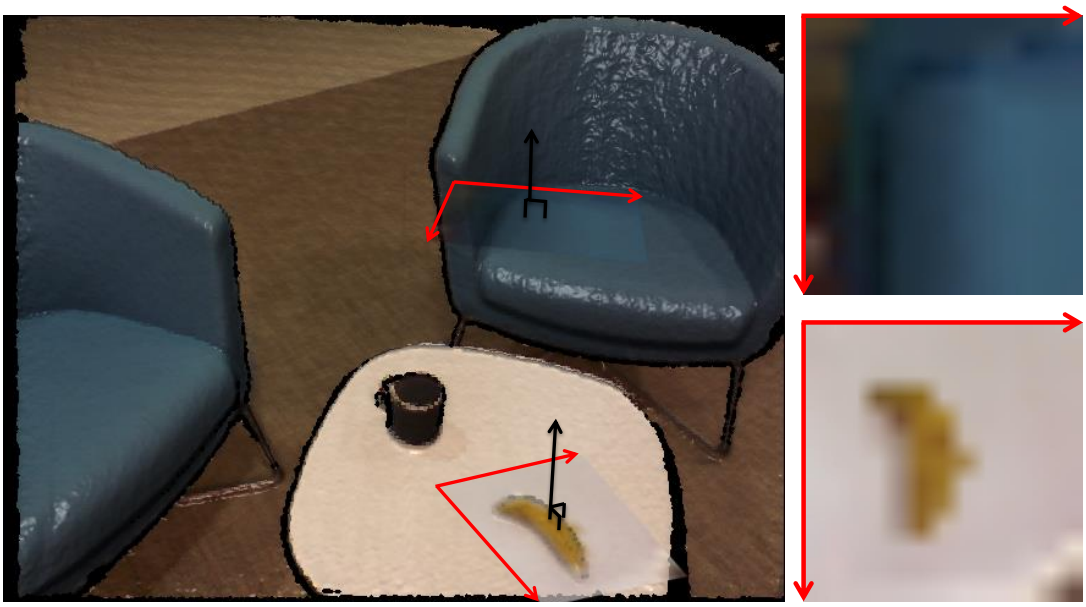


Figure 6.5: Illustration of Voxel-Oriented Patches (VOPs). VOPs encode color information directly from the TSDF without requiring explicit surface extraction. They are also fully 3D rotation invariant.

cisely aligned with the implicit surface to contain interesting and discriminative information. If the patch intersects regions of empty space (those with a TSDF weight of zero), the relevant pixel in the patch is flagged with an 'invalid' color value, which provides for a weak geometric cue.

A single VOP is computed on the GPU at each voxel being processed by the forest. This is done without having to extract the surface explicitly from the TSDF volume. Note that the forest will evaluate split functions (see below) with different parameters on the same VOP as it descends its trees; the VOP needs only be computed once per voxel for a full evaluation of the forest.

Possible variants of VOPs traverses the volume along the normal direction towards the nearest zero-crossing in the TSDF and then either take the color and the distance along the normal as the VOP pixel value. This was found to be too expensive for our real-time requirements and were not explored further. Other potential variants to explore could make use of the raw TSDF value as a way to describe the geometric properties of the object to learn.

6.3.2.3 Split Functions

Our split functions come in three varieties: VOP-based, surface orientation, and world height.

The VOP-based split functions work as follows. Each split node in the forest performs a comparison features by taking the differences or sums between two

VOP pixels' colors

$$f(i;\theta) = V_i(x_1, y_1, c_1) \;\; op \;\; V_i(x_2, y_2, c_2) > \tau \tag{6.3.18}$$

where $op \in \{+, -\}$, (x_1, y_1) and (x_2, y_2) specify particular pixels in the VOP, c_1 and c_2 specify color channels, τ is a threshold, and $\theta = (\text{VOP}, op, x_1, y_1, c_1, x_2, y_2, c_2, \tau)$ define the parameters of this VOP-based split function. The second type of split function applies a threshold to the dihedral angle between the surface normal and the world up vector. These are designed to help separate classes such as wall and table which are often flat and textureless, and thus would be difficult to split using a texture or geometry-based feature such as a VOP. The final split function applies a threshold to the world height, allowing the classifier to efficiently deal with 'easy' classes such as the floor.

The choice between the three types of split function, and the parameters θ thereof, is determined during the learning process. Note that the classifier will make the easiest choice available to it. For example, if world height is discriminative for the current training examples, it will use it. If this results in mistakes (for example, the user moves an object from a table to the floor) then the user can correct the labels and the forest can then be updated to use other features that are more appropriate for this object class.

6.3.2.4 Efficient test time classification

The learned decision forests are extremely efficient to evaluate at test time. Our GPU-based implementation can handle approximately 17 million voxel classifications per section. Which compares well with the roughly 3-10 million voxels visible the view frustum at any one time [122]. To maintain interactive rates, we batch up the visible voxels randomly and only test one batch at each frame. Each voxel is assigned a flag to say whether it has been classified yet or not. Once all voxels have been processed, the process repeats, applying the forest to the volume which may have been updated in the mean-time.

6.4 Experiments

We now demonstrate compelling results on several sequences, highlighting the smooth propagation of user labellings and the ability to learn and generalize to unseen regions of the world. To this end we conduct qualitative and quantitative experiments on four different indoor scenes: *intern area, kitchen area, seating area* and *living area* sequence. All our experiments are performed on an Nvidia

Titan with 6GB of RAM, although the system works happily on lower-end setups for smaller scenes.

6.4.1 Qualitative results

Propagation of User Labels. The user strokes the surface of objects in the physical world. Our system interprets such gesture as a brush stroke, and voice input is used to associate an object class label with the corresponding 'touched' voxels. Then, our mean-field inference engine (see Sec. 6.3.1) propagates these labels through the reconstructed scene, very efficiently. Thanks to the pairwise potentials (6.3.3) the result is a spatially smooth segmentation that adheres to object boundaries. Examples of label propagation are shown in Fig. 6.6.

Discriminative features: The training data given to the Streaming Decision Forest is a transformation of the RGBD values around each training point. That transformation is usually referred to as a feature. The discriminative power of the feature used to describe these voxels directly impacts the quality of the object segmentation. Here we compare the proposed VOP feature against fast and established features in the 2D object segmentation literature as well as one widely used 3D feature. Each feature is used to train a Streaming Decision Forest. The features we compare against are OpenCV's GPU implementation of SURF [15], Depth Probe [154], Difference of mean color of two randomly sampled boxes [157], Color Probe (similar to the Depth Probe, but in the RGB image) and PCL's [106] implementation of SPIN images [66]. Fig. 6.8 illustrates the recognition results obtained by the aforementioned features.

Forest Predictions. Our system learns a streaming decision forest classifier in a background CPU thread given the labels provided by the user. At some point, the user selects 'test mode', and the forest starts classifying all voxels. In Fig. 6.7 we illustrate the resulting intermediate predictions (the 'unaries' in the middle column), and compare them with the final, smoothed result obtained by running the mean-field inference on these unaries (right column). Let us focus on Scene 1 in the figure. In the first row, we see an example result in the region used for training the forest. Here, all the chairs are blue, and as expected, the unaries and mean-field results are of very high quality. The second row shows a failure case (highlighted by the arrows) in a region of the environment that was not used for training. The seat of the yellow chair gets confused with the floor, since only a blue chair was used for learning. At this point the user makes a stroke interaction (not shown) to correct the labels of the chair, and the forest is updated. After correction (row three) the chair can be correctly recognized.

Component	Timing
Reconstruction [122]	20 ms
Forest update	30 ms
Sampling	140 ms
Mean-field update	2-10ms
Forest inference	5 ms

Table 6.1: Approximate system timings. Despite small fluctuations we observed consistently good, interactive frame rates.

Note, the ability of our system to correct mistakes is crucial: no learning system is perfect, and our approach allows the user to see immediately where more training data (and thus user interaction) is required and to provide it in a natural way. This final row illustrates the generalization capabilities of our system to previously unavailable viewpoints. Scene 2 shows slightly noisier predictions from the forest, due in part to more challenging lighting conditions and problems with holes in the reconstruction (no user corrections were made in this sequence). The mean-field inference does a good job of smoothing and improving the final result, though some errors do remain that we expect could be corrected through limited further interaction.

We highlight more qualitative results on all four sequences. In Fig. 6.10, we first show the user-interaction, label propagation and mean-field inference results for the intern area sequence. Final meshes corresponding to the user provided labels and inference are shown in Fig. 6.11. Similar qualitative experiments for rest three sequences are shown in Figs. 6.12, 6.13 for the kitchen area, Figs. 6.12, 6.13 for the seating area Figs. 6.12, 6.13 for the living area sequences. As shown our system consistently achieves very high accuracy on all four dataset across different objects.

Computational Efficiency. We provide approximate system timings in Table 6.1. Although the timings change as a function e.g. of the number of visible voxels, in all tests we performed we observed interactive frame rates. Note that the forest *learning* runs asynchronously in a background thread. This thread continuously samples new labeled training data from the current view frustum and updates itself. This ensures an up-to-date forest is available for classification whenever the user requests it.

6.4.2 Quantitative results

Evaluation of the whole system: In this paragraph, we discuss the results of the main components of our system, namely the user interaction, the Streaming

Component	Sitting area	Bedroom	Kitchen	Intern area	Average
User Interaction	99.35%	97.61%	96.09%	97.73%	97.7%
Forest prediciton	94.57%	88.31%	82.58%	90.29%	88.94%
Final Inference	96.26%	95.19%	90.69%	95.55%	94.42%

Table 6.2: *Evaluation of different components of the system on different scenes. The measure corresponds to the percentage of correctly classified pixels.*

Decision Forest and the mean-field filtering of the forest's predictions on different sequences. For each sequence, key-frames have been hand labeled to cover the whole scene. These ground-truth images have been projected and aggregated onto the underlying TSDF, and then back-projected to all the views of the sequence. Tab. 6.2 shows the percentage of correctly classified pixels for each component. It can be observed that the accuracy of the user interaction is almost perfect and that the mean-field filtering provides for a good improvement on top of the predictions made by the forest.

Streaming Decision Forests: To evaluate the contribution of our new learning algorithm, we compare it against well known online Decision Forest algorithms: Online Random Forest (ORF) [143] and Hoeffding Trees (HT) [37]. As object can be trained in a sequential fashion, our system inherently has to deal with non-i.i.d. data. In order to perform quantitative evaluations, we use the dataset of [67] which contains 300 objects organized into 51 categories. Each off these objects is spun around a turntable at constant speed. One revolution of each object is recorded from 3 different points of view using a RGBD camera. To generate an online and non-i.i.d. setting, each category is added sequentially. For each object and each point of view, a consecutive segment of one third of the images of each object is kept for test purposes and the rest is used for training. All the learning methods have been evaluated over an increasing number of object classes, where the classes present and their order vary for each configuration. Fig. 6.9 show the corresponding results which demonstrate that the proposed learning algorithm outperforms the baseline methods regardless the number of classes used. Note that for these experiments, we used the difference of mean color of two randomly sampled boxes feature [158].

Voxel-Oriented Patch Features: Using the dataset presented in the previous paragraph, we evaluate the discriminative power of the proposed VOP feature against the baseline features listed in Sec. 6.4. For all these baseline features, we set the neighbourhood from which data are sampled to be similar to the neighbourhood from which VOP sample data from. Tab. 6.3 compiles these

Feature	Sitting	Bedroom	Kitchen	Intern	Average
VOP	**94.57%**	**88.31%**	82.58%	**90.29%**	**88.94%**
Diff. of RGB means	80%	71.84%	76.29%	73.42%	75.39%
Depth probe	77.54%	61.79%	**84.9%**	68.9%	73.06%
Color probe	56.39%	65.68%	60.77%	60.74%	60.9%
SURF	43.74%	67.12%	57%	58.13%	56.5%
SPIN	58.77%	43.22%	48.41%	36.1%	46.63%

Table 6.3: *Evaluation of the proposed feature descriptor against the baseline descriptors. The measure used is the percentage of correctly classified pixels. It has to be noted that overall VOP are also much better than the baseline methods under the class average precision and intersection over union measures.*

results. One can observe that the proposed VOP feature provides for a substantial margin of discriminative power compared to the different baseline methods.

6.5 Discussion

We have demonstrated a practical system which allows a user to interactively segment and label the surrounding environment in real-time. The labelling happens both explicitly through user interaction with the physical objects, and implicitly through the decision forest's ability to infer class labels in unlabeled parts of the scene.

Applications. We foresee numerous potential practical applications of our system. These include: quickly gathering large numbers of labeled 3D environments for training large-scale visual recognition systems such as [88,160]; generating personalized environment maps with known object class segmentations to be used for the navigation of robots or partially sighted people; recognizing and segmenting objects for augmented reality games which 'understand' the player's environment; and planning the renovation of a building, automating inventory, and designing interiors [116].

Limitations. Despite very encouraging results, our system is not without limitations. Currently our system uses a voice command to switch between training and test modes. We are planning an extension where both the learning and forest predictions are always turned on. This will require considerable care to avoid 'drift' in the learned category models: the feedback loop would mean that small errors could quickly get amplified. As with all recognition algorithms, the results are not always voxel-perfect. As we have demonstrated, allowing the user to in-

teractively make corrections can help reduce such errors. We believe additional modes of interaction such as voice priors (e.g. 'walls are vertical'), as well as more intelligently sampling the training examples could further improve results. Algorithmic parameters such as the pairwise weights are currently set by hand. Given a small training set (perhaps boot-strapping), more reliable settings could be automatically selected.

6.6 Conclusions

We have presented a system that allows a user to interactively segment and label an environment quickly and easily. A real-time algorithm reconstructs a 3D model of the surrounding scene as the user captures it. The user can interact with the world by touching surfaces and using voice commands to provide object category labels. A new, GPU-enabled mean-field inference algorithm then propagates the user's strokes through a volumetric random field model representing the scene. This results in a spatially smooth segmentation that respects object boundaries. In the background, the propagated labels are used to build a classifier, using our new streaming decision forests training algorithm. Once trained, the forest can predict a distribution over object labels for previously unseen voxels. These predictions are finally incorporated back into the 3D random field, and mean-field inference provides the final 3D semantic segmentation to the user.

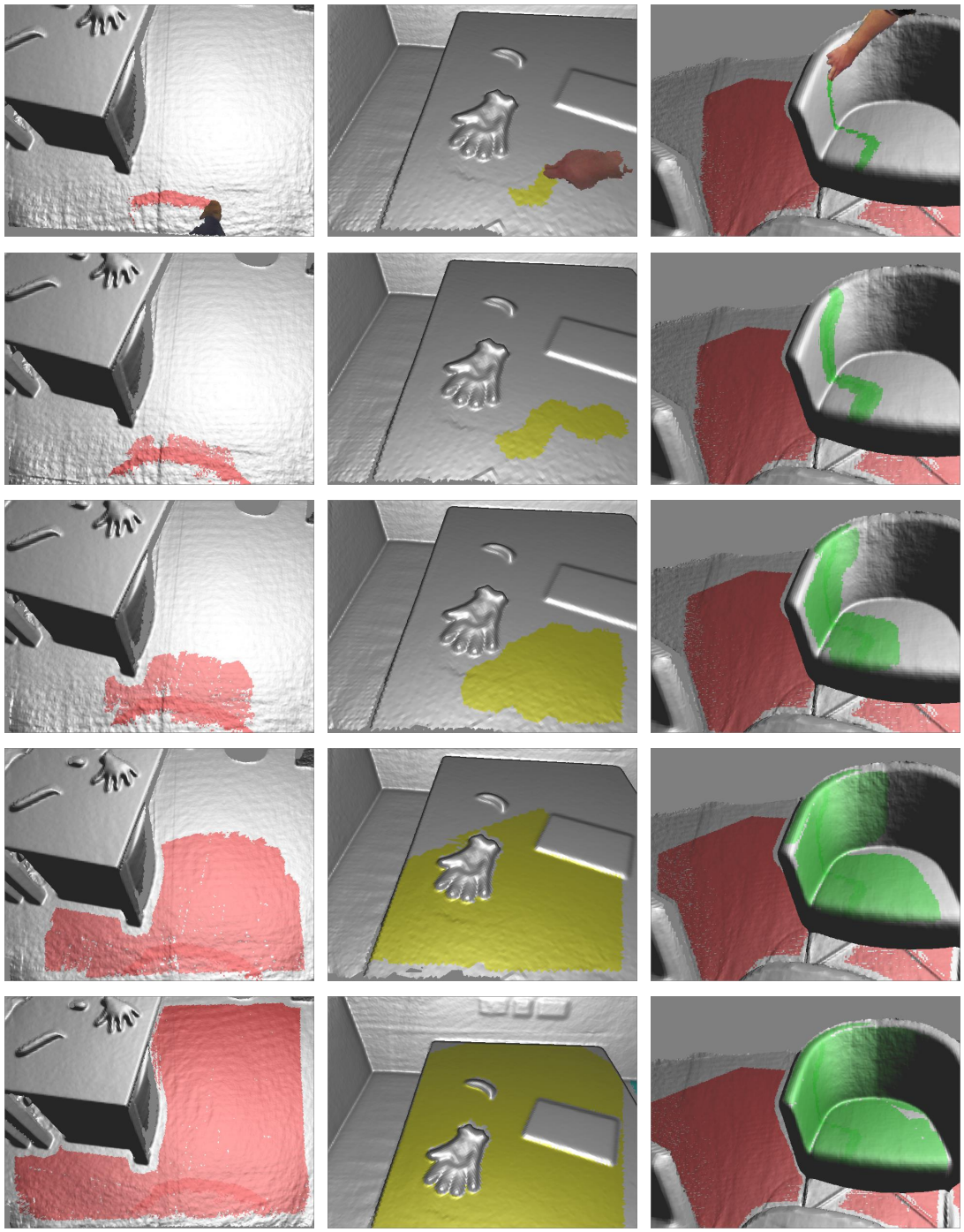


Figure 6.6: Our efficient inference engine smoothly propagates class labels from the voxels touched by the user to the rest of the volume. Here we show examples taken from three environments of the coarse user labels (top row) and three time steps (middle three rows) as the mean-field updates are applied over time. The pairwise terms in our energy encourage a smooth segmentation that respects object boundaries. The last row shows the final label propagation results for all hand-labeled objects.

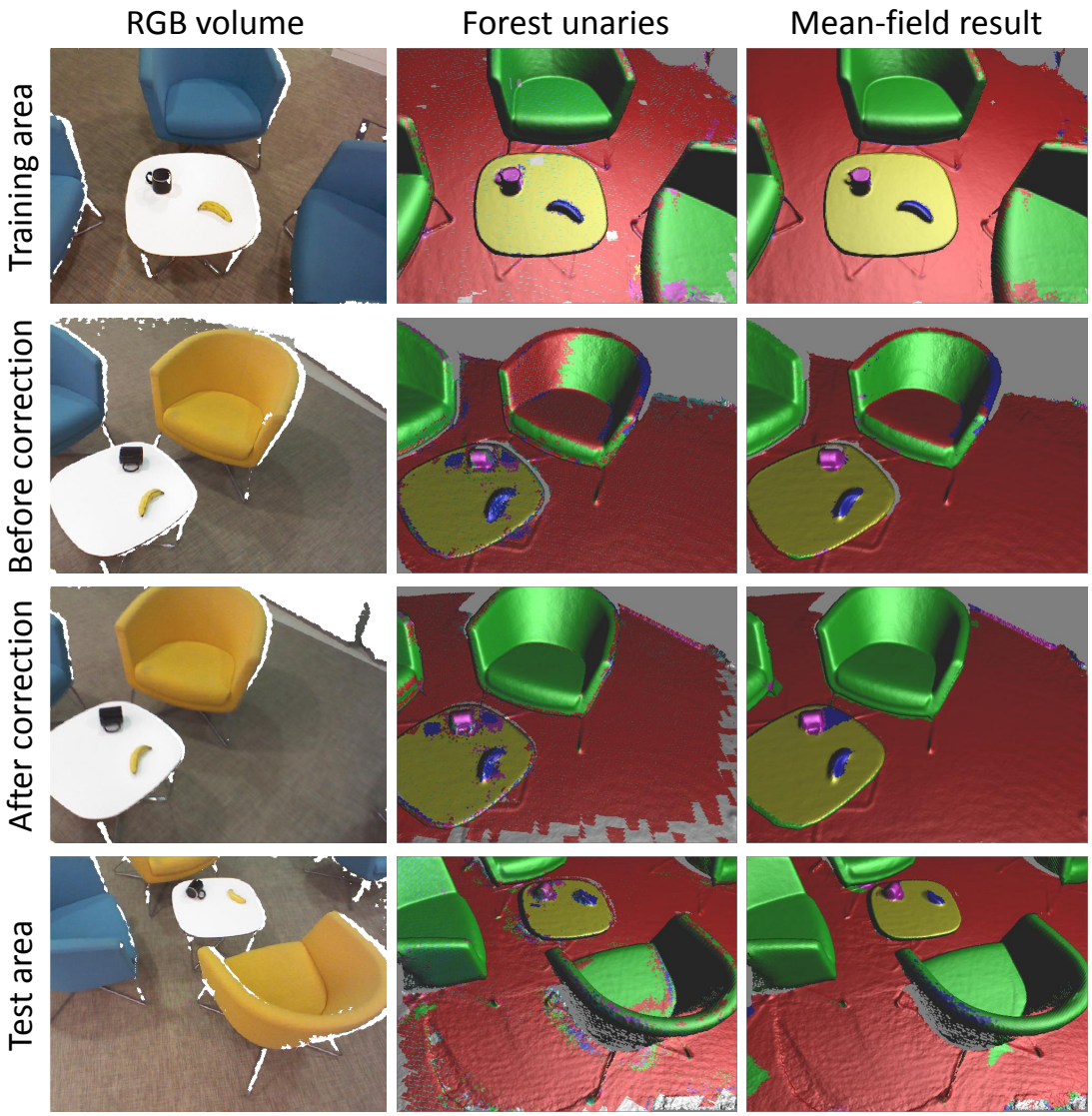


Figure 6.7: Our streaming decision forest is able to learn to make per-voxel predictions about the object classes present in the scene. Each pixel is classified independently, and so the forest predictions can be somewhat noisy. The mean-field inference effectively smooths these predictions to produce a final labelling output to display to the user. The arrows indicate a region that is initially incorrectly labeled (2nd row), but is successfully corrected (3rd row) by updating the forest based on new user interactions.

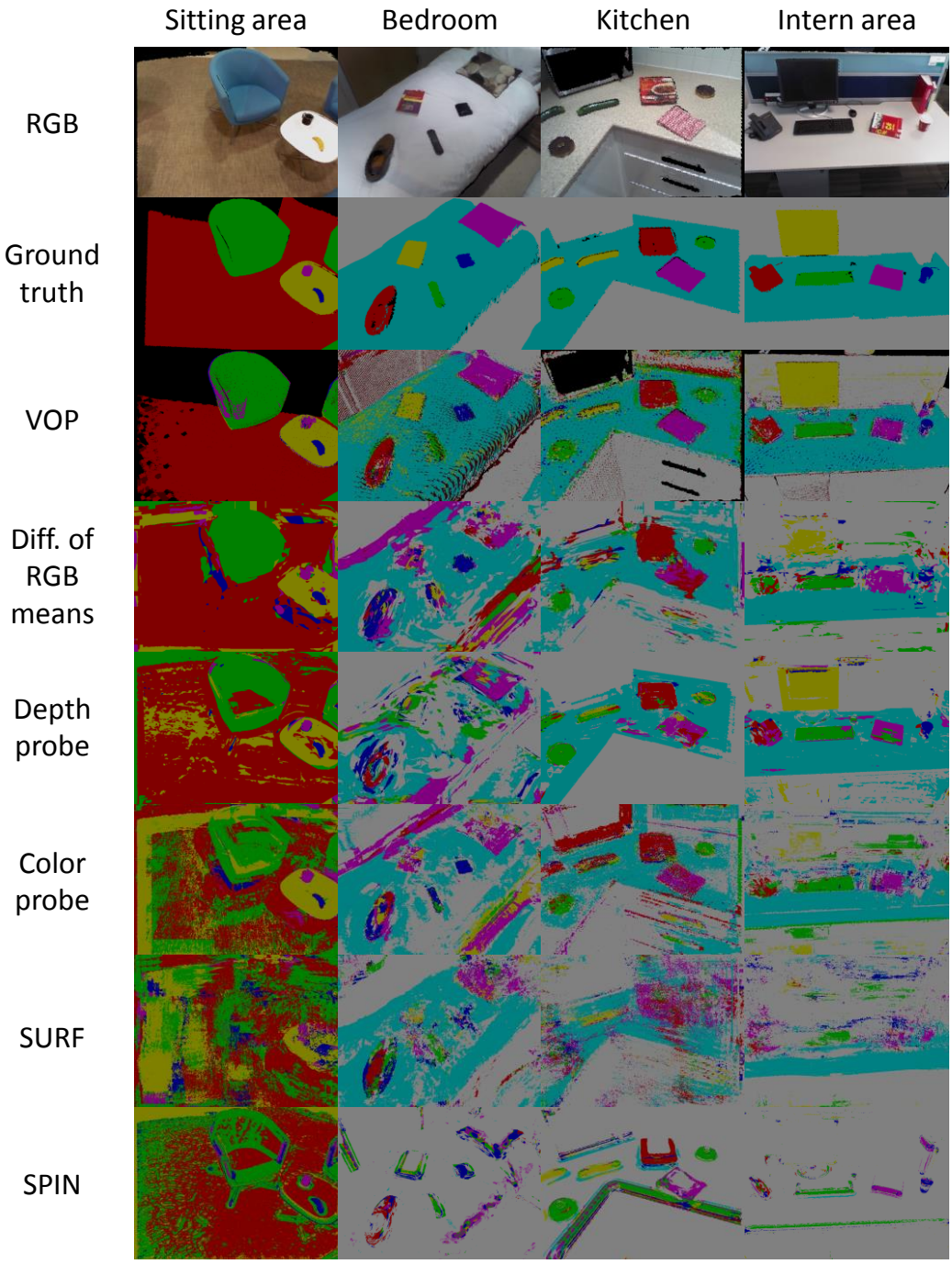


Figure 6.8: Quantitative results for the proposed VOP and baseline features on different scenes. One can observe that the proposed VOP feature leads to qualitatively better results than all the baseline approaches.

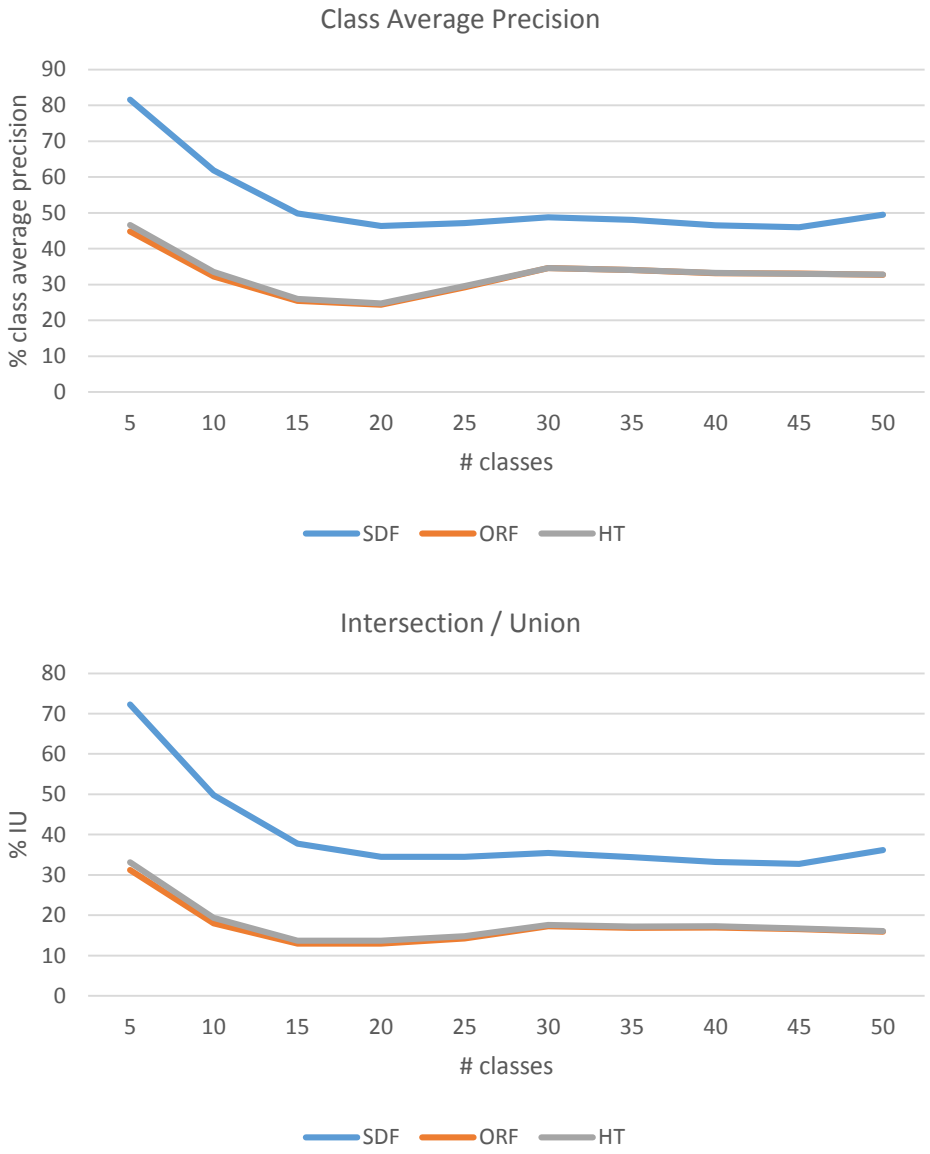


Figure 6.9: *Comparison of the proposed Streaming Decision Forest (SDF) algorithm against Online Random Forest (ORF) and Hoeffding Trees (HT) on the dataset described in 6.4.2. For each configuration of the classes, each learner contains 3 trees and the results have been averaged over 10 folds. These figures demonstrate that the proposed method comfortably outperforms all the baselines.*

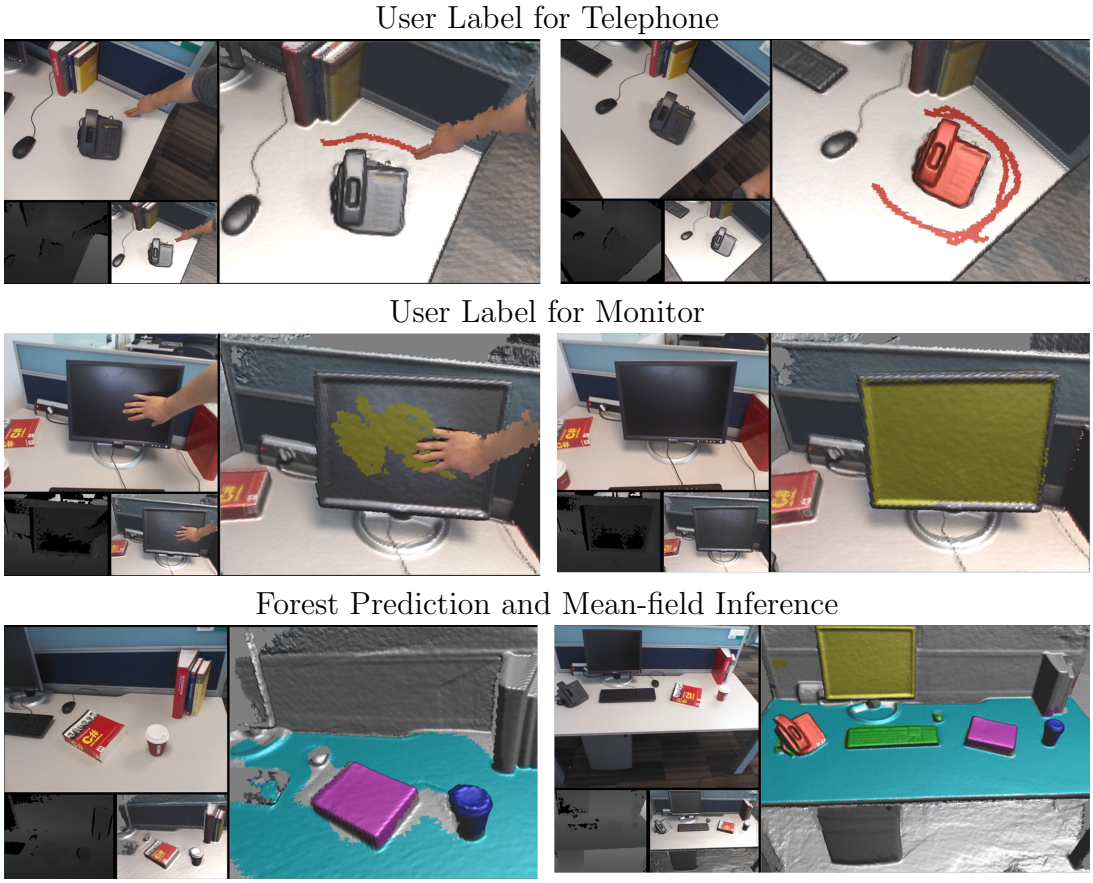


Figure 6.10: *Intern area sequence: user interaction and volumetric grab-cut for telephone (top row), interaction and label propagation for monitor (middle row) and finally mean-field inference results at two different time steps (last rows).*

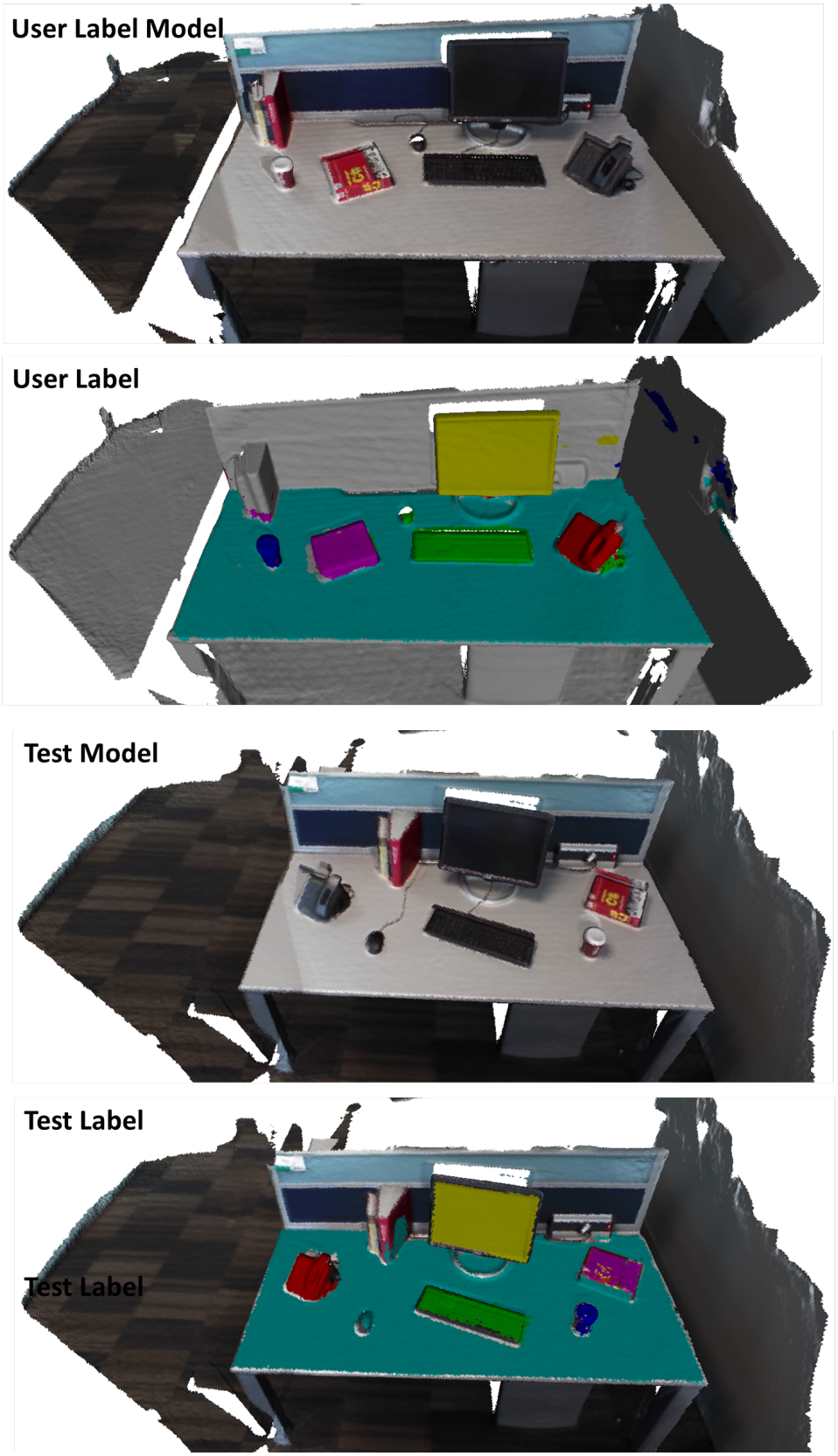


Figure 6.11: *Intern Area Sequence Meshes: here we show final meshes generated. From top to bottom: mesh (first) corresponds to training sequence where label propagation generates the ground truth labels from initial user interaction which are fed into forest. Second mesh shows the final generated user labelled mesh. Next we show the test sequence mesh and last one shows the labelled mesh based on the forest prediction and mean-field method.*

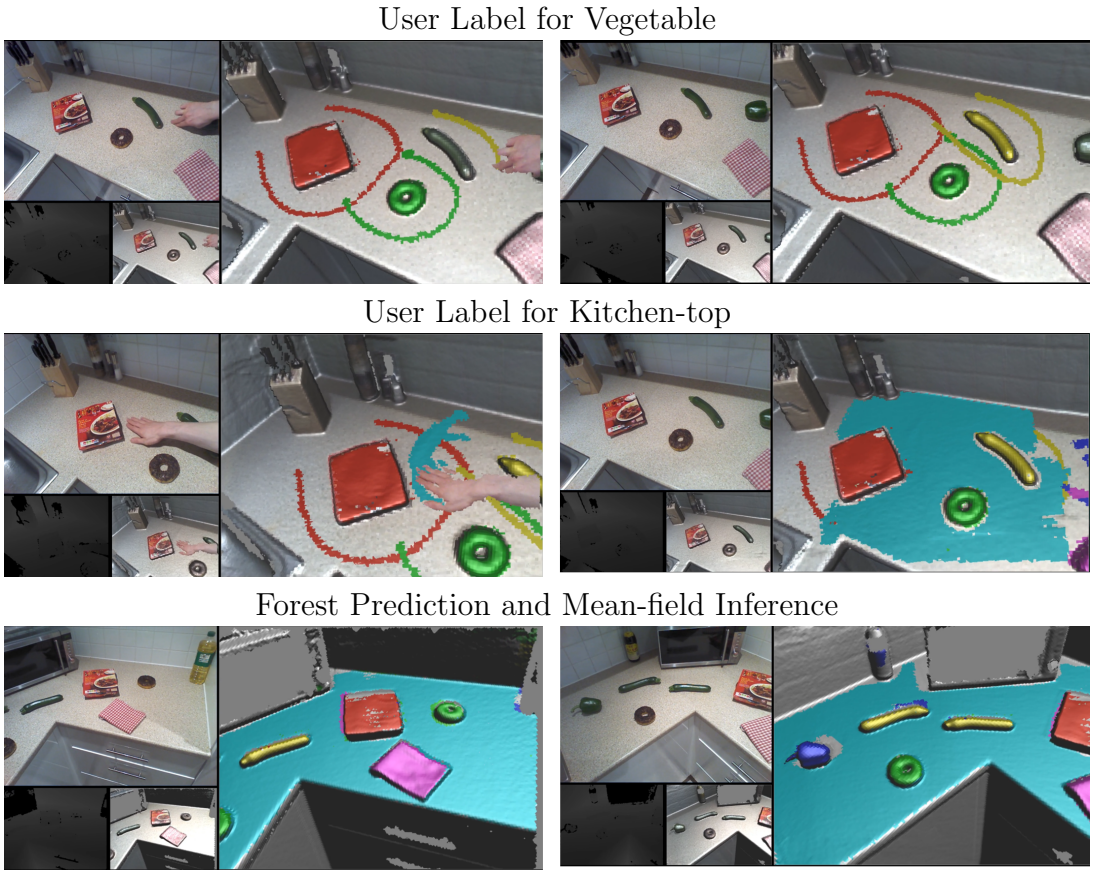


Figure 6.12: *Kitchen area sequence: user interaction and volumetric grab-cut for vegetable (top row), interaction and label propagation for kitchen-top (middle row) and finally mean-field inference results at two different time steps (last rows).*

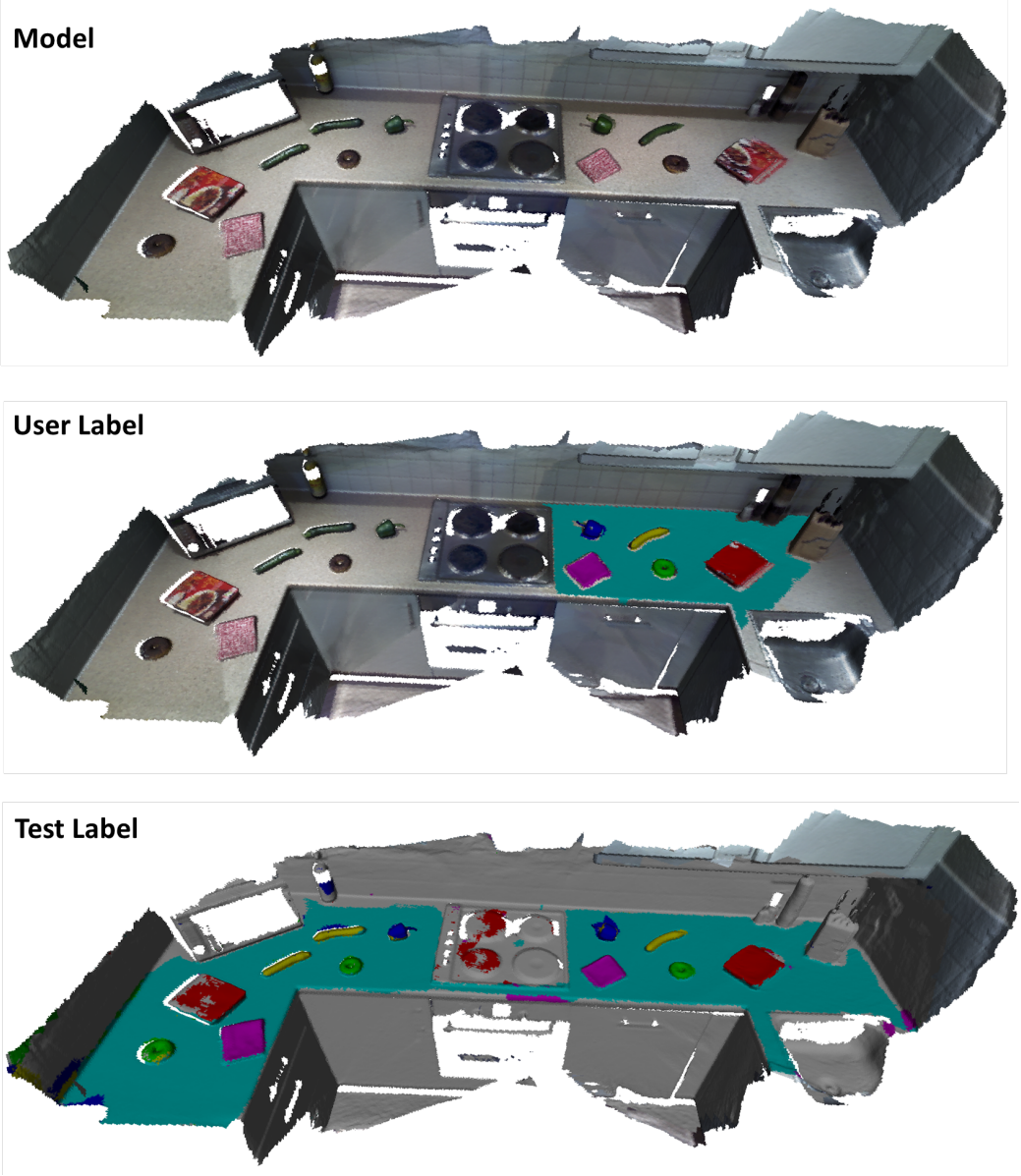


Figure 6.13: *Kitchen Area Sequence Meshes: here we show final meshes generated. From top to bottom: mesh (first) corresponds to whole sequence where in the first part of the sequence label propagation generates the ground truth labels from initial user interaction which are fed into forest. Second mesh shows the final generated user labelled mesh. Next we show the final labelled mesh based on the forest prediction and mean-field method.*

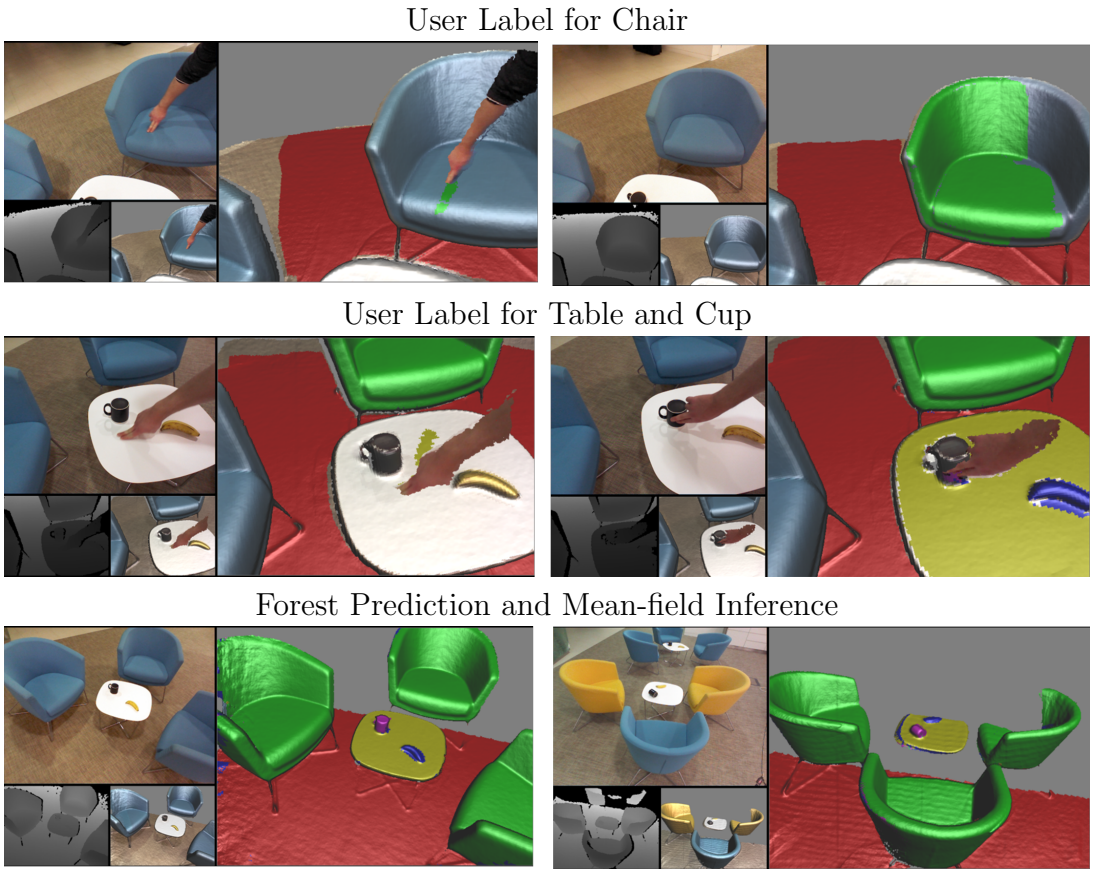


Figure 6.14: *Seating area sequence: user interaction and label propagation for chair (top row), interaction for table-top and cup (middle row) and finally mean-field inference results at two different time steps (last rows).*

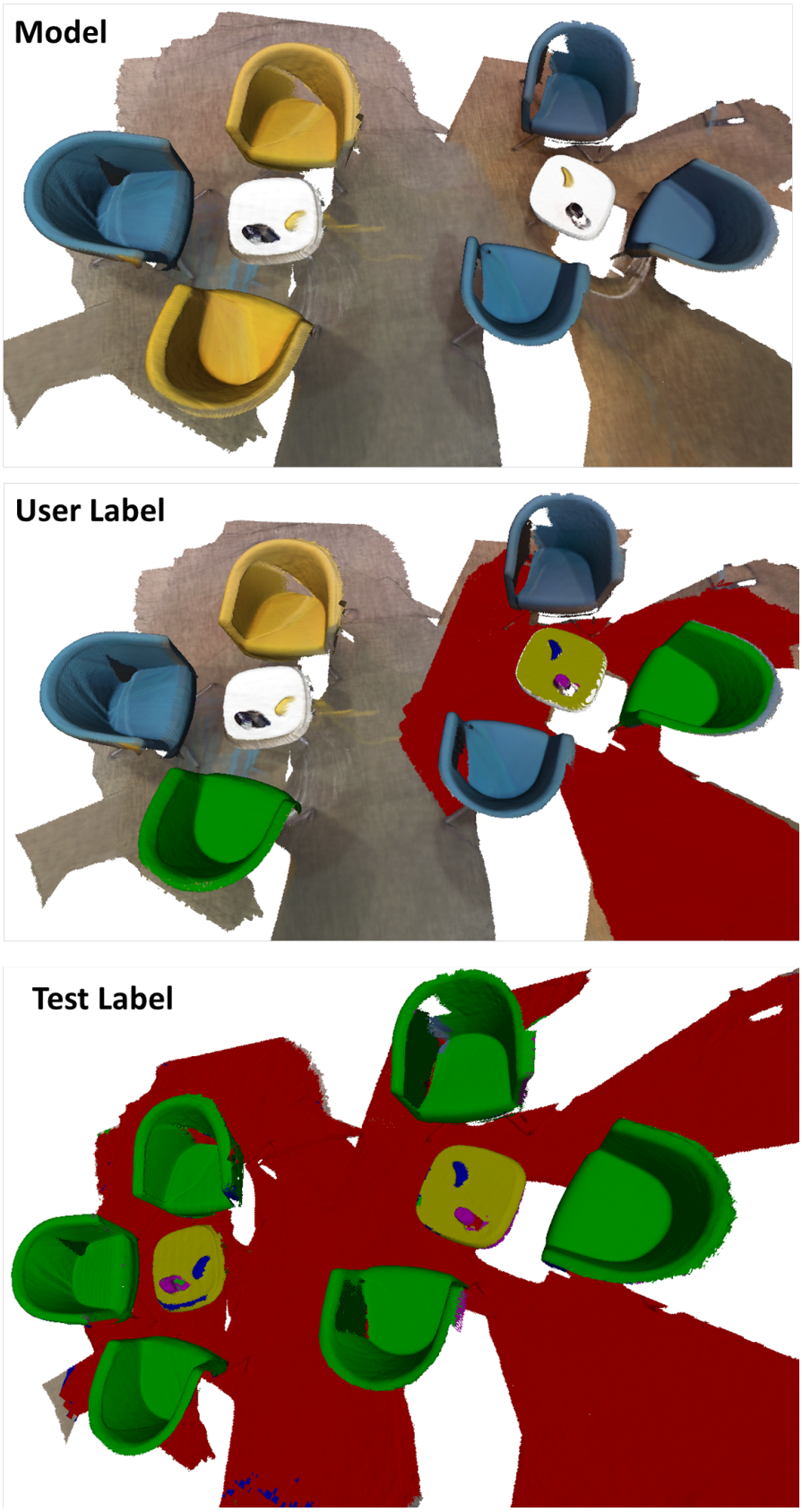


Figure 6.15: *Seating Area Sequence Meshes: here we show final meshes generated. From top to bottom: mesh (first) corresponds to whole sequence where in the first part of the sequence label propagation generates the ground truth labels from initial user interaction which are fed into forest. Second mesh shows the final generated user labelled mesh. Next we show the final labelled mesh based on the forest prediction and mean-field method.*

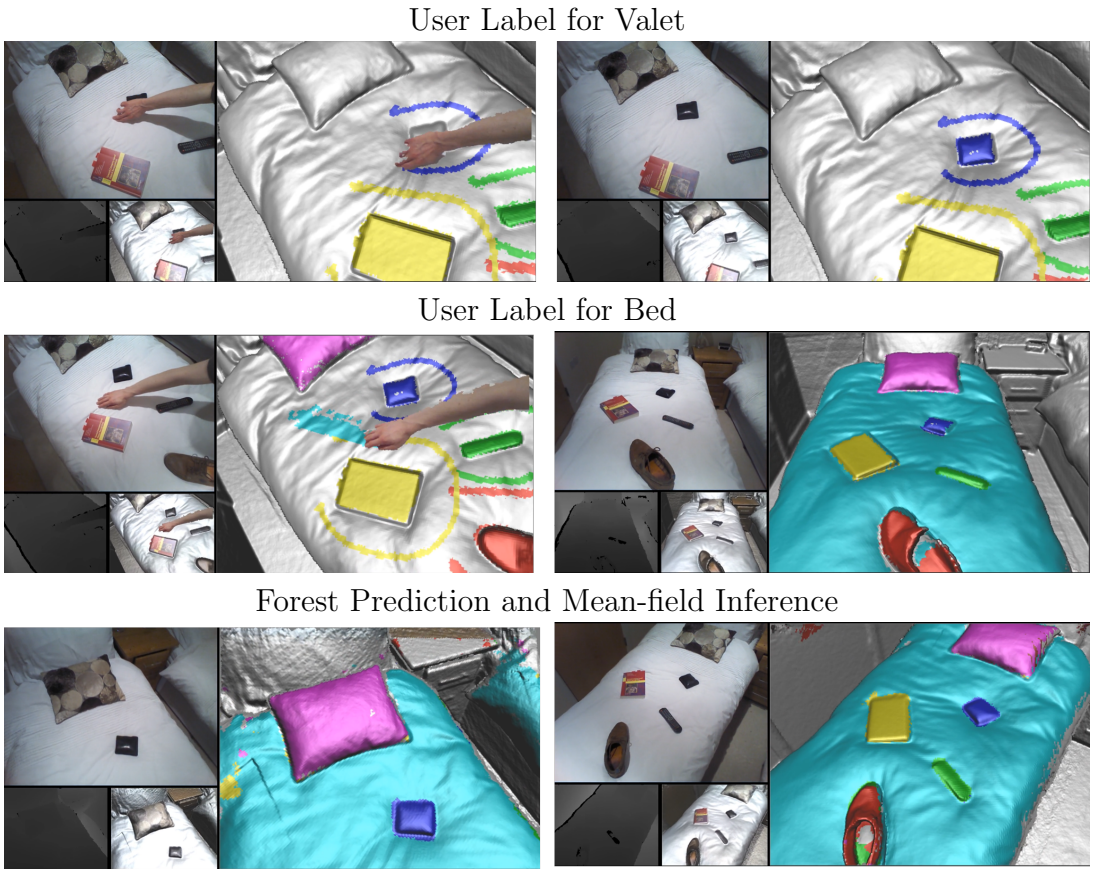


Figure 6.16: *Living area sequence: user interaction and volumetric grab-cut for valet (top row), interaction and label propagation for bed (middle row) and finally mean-field inference results at two different time steps (last rows).*

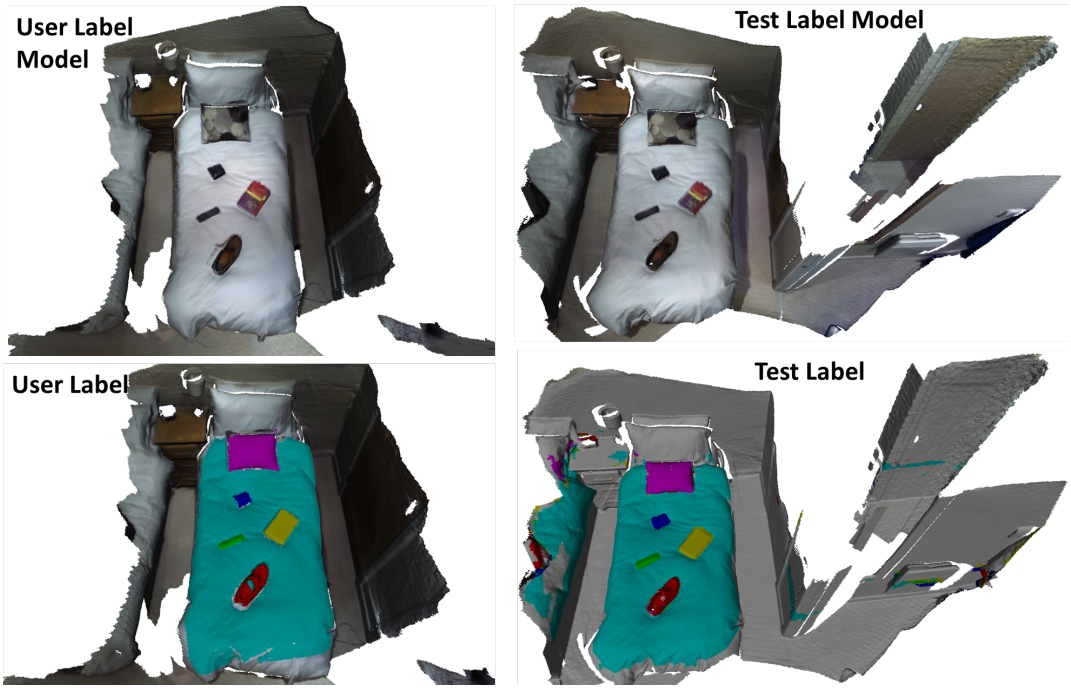


Figure 6.17: *Living Area Sequence Meshes: here we show final meshes generated. From top to bottom: mesh (first) corresponds to training sequence where label propagation generates the ground truth labels from initial user interaction which are fed into forest. Second mesh shows the final generated user labelled mesh. Next we show the test sequence mesh and last one shows the labelled mesh based on the forest prediction and mean-field method.*

Chapter 7

Dense Semantic Stereo Fusion for Large Scale Semantic Scene Reconstruction

Perceiving 3D structure and recognizing objects around us is central to our understanding of the world and may seem like an easy task for humans. However, for computer systems using artificial vision, it is not. In this regard, several approaches have been proposed to solve this problem. Nevertheless, most of these methods are either restricted by the scale they can handle (DTAM system works with small scale like a desk environment) or is restricted to work in indoor environments (e.g. Kinect Fusion system). In this paper we propose a robust approach for dense 3D reconstruction of outdoor environments along with associating this with object labels given stereo pairs of images. At the core of our algorithm is a hash based fusion approach for reconstruction and a mean-field approach for object labelling. In the process we capture the synergy effects between the reconstruction and recognition task. Further, we harness the processing power of GPUs to provide us with the computation capabilities required to run the system at real-time rate. Thus our system can handle and process large-scale environment in real time. We demonstrate the effectiveness of our approach on the KITTI dataset [49] and show high quality dense reconstruction and labelling of the scenes.

7.1 Introduction

When we drive a car from our home to the work place, we constantly perceive the 3D structure and recognize the objects around us. Such capabilities allow us free and accurate movement in unknown environments. Many commercial systems like Microsoft Virtual Map and Google Earth have built the virtual 3D map of cities using aerial and street-level imagery. These systems help us in our everyday lives to navigate in unfamiliar places. Building such a system that could automatically perceive the structure as well as recognize the environment is now of key importance especially in autonomous driving [104], robotics [38], [190], [35], assistive technologies and in understanding physiology of the human brain. In this paper, we target this problem by focusing on reconstructing the 3D structure and recognizing the outdoor scene given stereo image pairs.

In the past several years there have been rapid developments in algorithms and systems for indoor and outdoor scene reconstruction. Some of them build a sparse map of the world such as large-scale structure from motion (SfM) [176] using bundle adjustment in an off-line fashion [6]. Algorithms based on a visual Simultaneous Localization and Mapping (vSLAM) [76] usually works in real-time, however represent the world by a very small number of reconstructed points.

Dense real-time reconstructions, such as approach of Stühmer et.al. [166] or DTAM [120] work only in a very small environment. Another approach is to

replace a difficult estimation of depth by a sensor that measures it using the structured light or time-of-flight principles, such as the Kinect cameras or Velodyne lidar. The depth data from the Kinect camera are generally noisy. A popular research line enjoys the fact that the frequency of measurements is much larger than the perceived scene, hence algorithms have been designed to fuse the noisy depth data generated over time to recover very smooth high quality surfaces [63], [27], [122]. However, these approaches usually rely on pose estimation via the Iterative Closes Point (ICP) [29] that estimates a transformation between the perceived depth data and reconstructed model, i.e. it tracks the pose w.r.t. the reconstructed model in a *depth domain*. Though it works well in an indoor environment, the outdoor reconstruction brings several challenges: 1) the depth range is much larger, 2) the depth is usually estimated from a stereo camera (Kinect does not work, lidars provide rather sparse point clouds) and hence is more noisy, 3) only a few measurements are available for a typical "road-scene sequence" as opposed to the original environment for the KinectFusion, 4) dynamically moving objects are very common. All these issues make the data association in ICP more complicated and hence such algorithms are usually prone to fail.

Some of the other interesting works have focused on solving problems at the city-scale. Examples include the work of Taneja et al. [171], Chen et al. [26] who localize the landmarks at city-scale on mobile devices. Geiger et al. [50] use inputs from stereo camera to build dense 3D reconstruction of scene in real-time.

Additionally recognition of objects is also important for our understanding of the visual world. A great deal of papers have focussed on developing efficient and accurate algorithms to predict object labels at the pixel level. Examples include the models of Ladicky et al. [95], Shotton et al. [157], Gould et al. [54]. Recently many others have focussed on labelling the voxels in 3D and thus are able to leverage the information contained in the volumetric data. Some of them focus on indoor scenes [178], and other on outdoor scenes [145], [146], [129]. Some other recent works have also tried to jointly optimize for both the tasks of reconstruction and recognition, and so incorporate the synergy effects between these two high level vision tasks [57].

Thus the set of desiderata that we would like our method to accomplish are: i) it should build dense semantic 3D representation of any environment (indoor/outdoor) at any scale (even at city scale), ii) it should be amenable to handle dynamic/moving objects, iii) it should do processing at real-time rate iv) should explore more synergy effects between reconstruction and recognition tasks.

In this work we propose a method that fulfils all these desired tasks. At the core of our reconstruction and recognition system is the fusion based approach [122] with pose obtained by visual odometry [50] and the mean-field based approach of [87], [180] given pairs of stereo images. The reconstruction task is

facilitated by replacement of the ICP-based pose estimation by a visual odometry [50] that works in the *RGB space* and hence overcomes issues with pose estimation in noisy depth. Further, our approach leverages the information about objects available during the fusion stage to mitigate the affect of large motion. Additionally we formulate the problem of voxel labelling as a conditional random field (CRF) estimation problem where we allow each voxel to be densely connected to other voxels in the volumetric data. We propose a volumetric approach for efficient on-line mean-field inference for labelling the 3D data. Finally in order to achieve real-time performance, we harness the processing power of the GPUs. We demonstrate the performance of our system on the KITTI dataset [49].

- A fusion based robust approach to generate dense 3D reconstruction of outdoor scene given a pair of stereo images at real-time rate.
- Incorporation of visual odometry for pose estimation.
- Semantic Fusion to handle moving objects.
- An adaptation of densely connected pairwise models to generate dense per-pixel labelling of the volumetric data.

In Sec. 2 we provide the detailed description of each step involved in our approach. Sec. 3 shows our experimentations on publicly available datasets. Finally Sec. 4 concludes with a discussion.

7.2 Large Scale Semantic Reconstruction

We now describe our system that robustly builds the 3D environment and labels the volumetric data with the object labels such as *car, building, road*. In the process, it captures strong correlation between them to improve both the individual tasks. Fig. 1 shows overview of the whole pipeline for generating the dense semantic reconstruction from stereo image pairs.

7.2.1 Depth Estimation

Given the stereo image pairs, we compute the depth data as $z_i = Bf/d_i$ where z_i and d_i are the depth and disparity values for i^{th} pixel. Parameter B is the stereo camera baseline and f the focal length of the camera. We use openCV Semi-Global Block Matching (SGBM) algorithm [58] for disparity calculation.

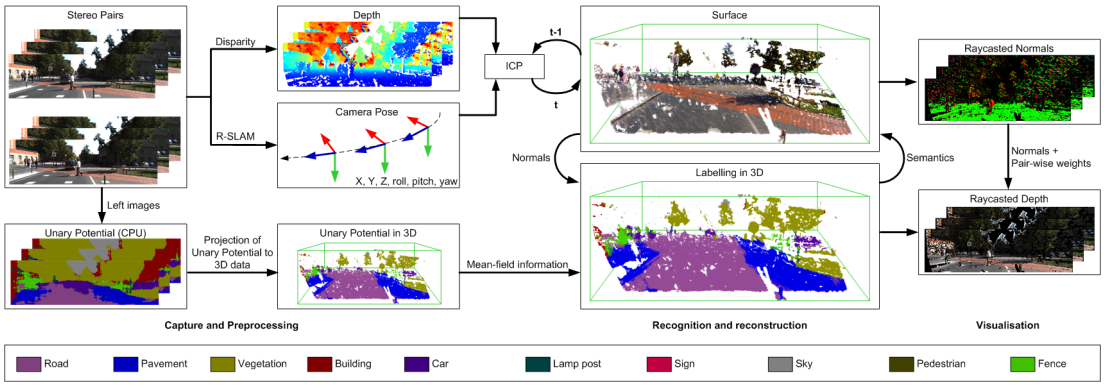


Figure 7.1: We here show the whole pipeline of our approach. Given stereo pairs as input, we generate depth and get camera poses from RSLAM [115]. At this stage, we also train our object classifier and generate unary potential of each pixel taking an object label. The depth data are fused to generate a volumetric 3D representation of the scene, and the unary potentials from the images are projected to the voxel data where inference is performed in 3D to generate high quality labelling. This stage also captures the correlation between these two tasks. Last steps involves raycasting the surfaces, normals and labels for visualisation.

7.2.2 Surface Reconstruction

We now describe our proposed approach for generating dense 3D reconstruction of large scale environment at real-time rate. In order to generate high quality surfaces, we follow the scalable hashing based fusion approach of Neissner et al. [122]. The key property of their approach is that they are able to generate high quality surfaces of large scale indoor scenes. However, there are two main drawbacks of this system: 1) the system is fully dependent on Kinect data, hence it fails to work in an outdoor environment and 2) their method depends on the ICP approach [29] for camera pose estimation. Their pose estimation system generally fails when there are moving objects in the scene and the camera starts drifting while reconstructing large scale scenes. In this work, our first key contribution is to adapt their scalable hashing to work with outdoor scenes given stereo pairs and also to solve the issues associated with camera pose estimation and its drifting.

Camera pose estimation. For camera pose estimation, we use a modified approach of Geiger et al. [50]. The input are stereo pair images from frame at two time-steps t and $t+1$. First, the algorithm extracts a set of sparse local features which generally corresponds to blobs and corners in the scene. Next, we remove all features whose locations correspond to pixels with assigned semantic classes of objects that "might move" (e.g. car, pedestrian, etc.) and the algorithm proceeds without any other modifications. That means, the extracted features are

spatially tiled and matched between the previous and current frames and left and right cameras using the epipolar constraint. To deal with the remaining outliers, the egomotion is estimated with a RANSAC. The estimated pose is refined by a standard Kalman filter with a constant acceleration model. This can be further improved by replacing of the Kalman filter with more advanced SLAM back-end such as iSAM2 [67]. A natural extension is to use a full SLAM system with camera re-localization and loop closures.

Fusion for surface reconstruction: The depth data generated using stereo pairs are generally noisy. However recent work on Kinect Fusion [122] reconstructs high quality surface by fusing noisy depth data measured over time. We follow a similar approach to recover high quality surfaces. Given noisy depth data generated using stereo pairs, we incrementally fuse them into a single 3D volume using a signed distance function within a volumetric data structure. Each voxel is represented by a signed distance and weight where signed distance corresponds to the distance of the voxel to the closest surface interface and weight measures the confidence of that voxel belonging to the surface. In order to further reduce the memory requirements, we use the concept of truncated signed distance function (TSDF) [34] which assumes that the true surface lies only within a truncated distance of the observed values from the depth data. Thus, the points lying outside these regions are not stored which reduces the memory requirements. Such a representation is important since most of the space is free/unoccupied. Given this representation, the depth maps are fused together over time to generate very smooth, high quality surface. In order to efficiently fuse/integrate the depth data, a standard approach consists of first uniformly sweeping through the volume, culling the volume outside the view frustum, projecting the voxel centres onto the depth map, and estimating the rigid six degrees-of-freedom (DOF) based visual odometry approach discussed earlier.

The TSDF representation for the volumetric data is very efficient, but we still need a data structure to store the voxel data and efficiently retrieve any information related to the voxel data at any point of time. For this, we have used the recently proposed hashing based data structure that is very efficient to retrieve and insert any information related a voxel [122]. Such capabilities are very important since the number of voxels in the current view frustum keeps on changing dynamically. Thus we store only the information related to the TSDF values within the view frustum. Even though this representation is very compact and efficient, the volume that can be reconstructed is limited by the available GPU memory. Thus, in order to allow for unbounded volume reconstruction, we allow for streaming of the data from the GPU to the CPU and vice-versa[1]. For

[1]The reconstruction is actually still limited by CPU RAM and HDD, however these are not

more details, please refer to the work of [122].

7.2.3 Semantic Fusion and TSDF space

Let $T_i^t \in \mathcal{T} = \{T_1^t, T_2^t, ..., T_N^t\}$ and $C_i^T \in \mathcal{C} = \{C_1, C_2, ..., C_N\}$ be the TSDF and colour values of ith voxel at time t. The scalable KinectFusion work of [122] fuses the current TSDF and the colour values with incoming values across iterations in online fashion as:

$$T_i^t = (1 - \alpha^t)T_i^t + \alpha^t T_i^{t-1} \quad C_i^t = (1 - \alpha^c)C_i^t + \alpha^c C_i^{t-1} \tag{7.2.1}$$

where the parameter α^t and α^c are the weighting parameters that shows the confidence of the current depth and colour measurement. This fusion step generally fails when there are moving objects in the scene. However we observe that the depth and colour fusion can be improved by utilizing the semantic knowledge of the objects. For example the knowledge about the object could provide information that fusing the depth will result in wrong results since the objects have moved. In this work, we use such observations to improve our fusion results i.e, our confidence of fusing the depth map is now dependent on whether the object can move or move. Examples include car, pedestrian and bicycle which can move through environment but building and road are always static. Thus our semantic fusion results into following TSDF and colour updates:

$$T_{i_o}^t = (1 - \alpha_o^t)T_{i_o}^t + \alpha_o^t T_{i_o}^{t-1} \quad C_{i_o}^t = (1 - \alpha_o^c)C_{i_o}^t + \alpha_o^c C_{i_o}^{t-1} \tag{7.2.2}$$

where α_o^t and α_o^c are now object specific weighting parameters.

7.2.4 Volumetric Mean-field

7.2.4.1 Model

We now discuss our approach for labelling of the volumetric data. We begin by defining a random field over random variables $\mathcal{X} = \{X_1, ..., X_N\}$ where each discrete random variable x_i is associated with a voxel $\mathcal{V} = \{1, ..., V\}$ in the 3D reconstruction volume. Each random variable x_i takes a label in $\mathcal{L} = \{l_1, ..., l_L\}$ where each label l_i corresponds to different object classes such as car, building and road that the voxel belongs to. We then formulate the problem of assign-

expensive nowadays. The same "streaming in and out" strategy used between the GPU and CPU can be used between the CPU RAM and HDD to allow really unbounded reconstruction.

ing an object label to the voxels using a fully connected pairwise Conditional Random Field (CRF). Though the fully connected model has been applied in various applications in computer vision such as image segmentation [87], optical flow [167]. Our formulation is different from theirs since we propose an efficient inference over the volumetric data. Further, since we define CRF over the voxels in the current view frustum, we have to deal with dynamic energy function as our function keeps on changing in each iteration as the volumetric reconstrunction is dynamically changing as new observations are updated. We then express the fully connected CRF over the voxels as:

$$\begin{aligned} P(X|I) &= \frac{1}{Z(I)} \exp(-E(X|I)) \\ E(X|I) &= \sum_{i \in \mathcal{N}} \psi_u(x_i) + \sum_{i<j \in \mathcal{N}} \psi_p(x_i, x_j) \end{aligned} \tag{7.2.3}$$

where $P(X|I)$ and $E(X|I)$ are the posterior distribution and energy function associated with a configuration X conditioned on the volumetric data I, $Z = \sum_{\mathbf{x}'} \exp(-E(\mathbf{x}))$ is the partition function. $\psi_u(x_i)$ are the likelihood or unary potential terms of a voxel taking an object label and $\psi_p(x_i, x_j)$ are prior terms defined over pairs of random variables which enforces consistency or smoothness over pairs of variables.

Unary potential terms: Our unary potentials $\psi_u(x_i)$ correspond to cost of the voxel i taking an object label $l_i \in \mathcal{L}$. Generally these correspond to the classifier responses over the feature generated using appearance and geometric features using 2D image and 3D volumetric data. In this work, we first generate unary potential on images using features defined on the image data. We project and accumulate the unary potentials on to the volumetric data over time. Such data accumulation in practice provides better results since it averages out the noise in the unary potential evaluation on a single image.

Pairwise smoothness terms: Our pairwise potential function enforces consistency over a pair of variables and thus generally leads to a smooth output. In our application, we have used the weighted Potts model which take the form as: $\psi_{ij}(l, l') = \lambda_{ij}(f_i, f_j)[l \neq l']$, where [.] is the Iverson bracket whose value is 1 if the condition in the square bracket is true and 0 otherwise. Here f_i, f_j are features at i^{th} and j^{th} voxels. In the 2D segmentation domain, the cost λ_{ij} of assigning different labels to neighbouring pixels is generally chosen such that it preserves image edges. Inspired from these edge-preserving smoothness costs, we make the label discontinuity cost λ_{ij} as weighted combination of the Gaussian

kernels dependent on appearance and depth features:

$$\lambda_{ij} = \sum_k \theta^k \lambda_{ij}^k = \theta_{\mathrm{p}} e^{-\|\mathbf{p}_i - \mathbf{p}_j\|_2} + \theta_{\mathrm{a}} e^{-\|\mathbf{a}_i - \mathbf{a}_j\|_2} + \theta_{\mathrm{n}} e^{-\|\mathbf{n}_i - \mathbf{n}_j\|_2} \tag{7.2.4}$$

where $\mathbf{p}_i$, $\mathbf{a}_i$ and $\mathbf{n}_i$ are respectively the 3D world coordinate position, RGB appearance, and surface normal vector of the reconstructed surface at voxel i, and $\theta_{\mathbf{p}}$ $\theta_{\mathbf{a}}$ and $\theta_{\mathbf{n}}$ are parameters obtained by cross-validation. Note that the surface normals are calculated using the TSDF values where each voxel looks at its neighbours within the zero crossing region. Observe that as the 3D model is updated from one frame to the next, the appearance and surface normals associated with the voxels change. The energy landscape thus continuously changes over time.

7.2.4.2 Efficient Mean-Field Inference

Given the form of the energy function defined above, we now give details of the inference approach to find the optimal labelling of the volumetric data. One of the most popular approaches for labelling in CRF has been graph-cuts based α-expansion approach [20] that finds the maximum a posteriori (MAP) solution. However, as has been shown, graph-cuts lead to slow inference and is not easily parallelizable. In this work, we follow the mean-field based optimization method; a filter-based variant of which has been shown to be very efficient in 2D image segmentation case [87], [180].

In the mean-field framework, we approximate the true distribution $P(X)$ with an approximate distribution $Q(X)$, where the approximation is measured in terms of the KL-divergence $D(Q||P)$ between the true distribution P and tractable Q distributions. Further, we take a family of Q distribution where the marginal of each random variable is assumed to be independent, i.e. $Q(X) = \prod_i Q_i(x_i)$. Under this assumption, the fixed point solution of the KL-divergence leads to the following mean-field update (please refer to [81] for more detailed description):

$$\begin{aligned} Q_i(x_i = l) &= \frac{1}{Z_i} \exp(-E(X) = \psi_u(l) + \sum_{j \neq i} \sum_{l' \in \mathcal{L}} \psi_p(x_i = l, x_j = l')) \\ Z_i &= \sum_{l \in \mathcal{L}} \exp(-E(X)) \end{aligned} \tag{7.2.5}$$

The complexity for the mean-field update for the volumetric data is $O(N^2)$. Now we propose two technical contributions that lead to efficient updates of the marginals. First, we propose an online volumetric mean-field inference framework that utilizes the benefits of the dynamic nature of the energy function to efficiently infer the approximate maximum posterior marginal (MPM) solution.

In this framework we incrementally refine the marginals of a voxel taking an object label across iterations. This way we only need a small number of iterations of mean-field update each time step. Our second contribution relates to designing a volumetric filter that can be inherently implemented on parallel architecture such as GPUs.

7.2.4.3 Volumetric filtering based mean-field

The most time consuming step in the mean-field inference is updating the pairwise terms which lead to a complexity of $O(N^2)$. Now we will discuss how we reduce the complexity to $O(N)$ for certain kinds of pairwise terms. Our work is motivated by recent work of Krahenbuhl et al. [87] and Vineet et al. [180] who show that fast approximate MPM inference can be achieved by applying cross bilateral filtering techniques when the pairwise terms take a weighted combination of the Gaussian kernels. First we show how the mean-field update in Eq. 6 is a filtering step. For this we take following transformation of Eq. 6:

$$\tilde{Q}_i^{(m)}(l) = \sum_{j \neq i} \lambda^{(m)}(\mathbf{f}_i, \mathbf{f}_j) Q_j(l) = [G_m \otimes Q(l)](\mathbf{f}_i) - Q_i(l) \tag{7.2.6}$$

where G_m is a Gaussian kernel corresponding to the mth component and $\otimes$ is the convolution operator. Since $\sum_{j \neq i} Q_j(x_j = l')\psi_p(x_i, x_j)$ can be written as $\sum_m w^{(m)} \tilde{Q}_i^{(m)}(l')$, and approximate Gaussian convolution using is $O(N)$, parallel[2] updates can be efficiently approximated in $O(MNL)$ time for the Potts model. The algorithm is run for a fixed number of iterations, and the MPM solution extracted by choosing $x_i \in \text{argmax}_l Q_i(x_i = l)$ at the final iteration. Now we describe steps to perform the high-dimensional filtering on the 3D volumetric data. Our 3D filtering is motivated by the work of the permutohedral lattice based filter [9].

There are four steps: generating the feature points, embedding (splatting) the points in the lattice, blurring the lattice points and projecting back (slicing) those points in the original space. After iterating the updates to iteration $\tilde{t}$, the output MPM estimates can be obtained as

$$x_i^\star = \underset{l \in \mathcal{L}}{\text{argmax}}\, Q_i^{\tilde{t}}(l) \ . \tag{7.2.7}$$

[2]Although the updates are conceptually parallel in form, the permutohedral lattice convolution is implemented sequentially.

7.2.4.4 Online mean-field

Given unlimited computation, one might run multiple update iterations until convergence.[3] However, in our online system, we assume that the next frame's updates to the volume (and thus to the energy function) are not too radical, and so we can make the assumption that the Q_i distributions can be temporally propagated from one frame to the next, rather than re-initialized (e.g. to uniform) at each frame. Thus, running even a *single iteration* of mean-field updates per frame effectively allows us to amortize an otherwise expensive inference operation over multiple frames and maintain real-time speeds. Note that this effectively means that the t variable above becomes the frame number and the mean-field iteration count.

As described above, the output of the classifier responses is used to update the unary potentials, which will, over several frames, impact the final segmentation that results from the online mean field inference. However, to speed up convergence, we propose an additional step that exploits our temporal propagation of the Q distributions. Rather than simply propagating the Q_i^{t-1}s from the previous frame, we instead provide the next iteration of mean-field updates with a weighted combination of Q_i^{t-1} and the classifier prediction $P_u(x_i = l \mid \mathbf{I})$. We thus use

$$\bar{Q}_i^{t-1}(l) = \gamma Q_i^{t-1}(l) + (1-\gamma) P_u(x_i = l \mid \mathbf{I}) \tag{7.2.8}$$

in place of the Q_i^{t-1}, where γ is a weighting parameter.

In practice, this step appears to result in considerably quicker updates to the segmentation results given the classifier responses.

7.3 Experiments

We demonstrate the effectiveness of our approach on the publicly available KITTI dataset [49]. The images are 1241x376 pixels, captured using a specialized car for various outdoor environment, i.e. city, road, campus. The dataset is very challenging since it consists of many moving objects such as car, pedestrians, bike, and there are many changes in lighting conditions. We have used the per-pixel annotations of Sengupta et al. [145] which consists of 45 images for training and 25 test images with object class information. The class labels are road, building, vehicle, pedestrian, pavement, tree, sky, signage, post/pole, wall/fence. All the experiments are conducted on a machine with NVidia TITAN GPU and Intel 2.8GHz CPU.

[3] If applied sequentially on a fixed energy function, the mean-field inference comes with some convergence guarantees [87]. These do not apply to our algorithm as our algorithm is dynamically updated in each frame, but this does not appear to be a problem in practice.

We show some qualitative results of our reconstruction and recognition approach. For this purpose, we have selected four sequences from KITTI dataset from the city, road and campuses scenes. These sequences are the most relevant for us since there are many moving objects such as vehicles or pedestrians. In Fig. 7.5 we highlight the importance of using mean-field inference in order to recover very fine tree structure. Further we also observe high quality reconstruction of roads and building. Next, we show the results on the campus sequence. As shown in the figure 7.4, we see many pedestrians walking in the scene. In such a scenario, the ICP approach starts failing, and camera loses tracks. However our semantic stereo fusion along with visual odometry pose estimation helps to recover from this issue and we see a high quality reconstruction. As shown we are able to properly reconstruct road, pavement or trees. Our inference algorithm also generates visually good per-voxel labelling of the scene. Further, we show the importance of incorporating the semantic fusion and visual odometry to handle moving objects. The right part of Fig. 7.6 shows the reconstruction and labelling if only ICP is used for camera tracking, and left part shows the output of the proposed semantic fusion and visual odometry approach. ICP based approach completely fails and leads to a very sparse representation of the scene. But our approach helps to mitigate the effect of motion to a large extend and recovers a very dense representation of the scene. We also like to point out that the dense surface consists of actual 3D points and not just their interpolation, generated by, e.g. a mesh. Inference with this dense representation helps to generate fine object boundaries such as the examples show in Fig. 7.6– green bush in the right bottom corner and a small patch of pavement between the building and vehicle. In Fig. 7.3, we show how our approach recovers a very fine (close to) dense model of moving objects such as car. Thus we believe our approach will be useful (among other applications) for collection of large-scale datasets in the outdoor environments in the future. Finally, we show labelling on the original sequence, raycasted normals and label data (shown in Fig. 7.2). We also show some quantitative comparison between performing inference on image data to when we project the inferred 3D label of the volumetric data to the image. These results are shown in Tab. 7.1.

Beyond these we show more qualitative results. In Fig. 7.14, we show mesh generated for the road scene sequence. We observe that our fusion based approach is able to generate very dense and smooth reconstruction. We show the phong-shaded RGB views in Fig. 7.8 and phong-shaded labelled images in Fig. 7.9 and 7.10.

Algorithm	road	pavement	vegetation	building	car	lamp post	sky
Avg. Precision							
Image Labelling	0.963	0.989	0.903	0.954	0.941	0.905	0.785
Ours	0.982	0.982	0.971	0.983	0.965	0.951	0.786
Avg. Recall							
Image Labelling	0.971	1	0.928	0.968	0.973	0.94	0.848
Ours	0.985	0.998	0.993	0.987	0.997	0.987	0.983

Table 7.1: Quantitative results on the KITTI dataset. We compare the average precision and average recall between two approaches. First *Image Labeling* when the labelling is done directly on the image domain. We do labelling in the 3D, and then project them to image domain for comparison.

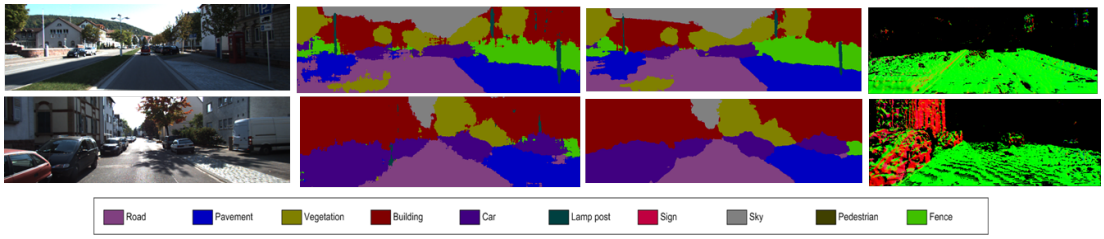


Figure 7.2: Qualtitative results on the KITTI dataset. We show results of image domain labelling (2nd column), our labels project to the images (3rd column), and raycasted normal maps (4th column) given input images (1st column).

7.3.1 Conclusion

Perceiving a 3D structure and recognizing the objects around us is central to our understanding of the world. In this paper, we propose a robust and accurate approach for dense 3D reconstruction of outdoor and indoor environments along with associating them with object labels given stereo images pairs. At the core of our algorithm is a hash based fusion approach for reconstruction and a hash based approach to the mean-field inference for object labelling. In the process, we capture the synergy effects between the reconstruction and recognition tasks. Further, we harness the processing power of GPUs to provide us with the computation capabilities required to run the system at real-time rate. Thus our system scales well into large-scale environments. We demonstrate the effectiveness of our system and both, high quality dense reconstruction and labelling of the scenes on the KITTI dataset.

There are many interesting future works spawning from our paper. First we would like to enforce object specific shape prior on 3D reconstruction. Currently feature generation and learning of the class models have been done in an offline fashion, we would like to integrate the online aspects of these tasks.

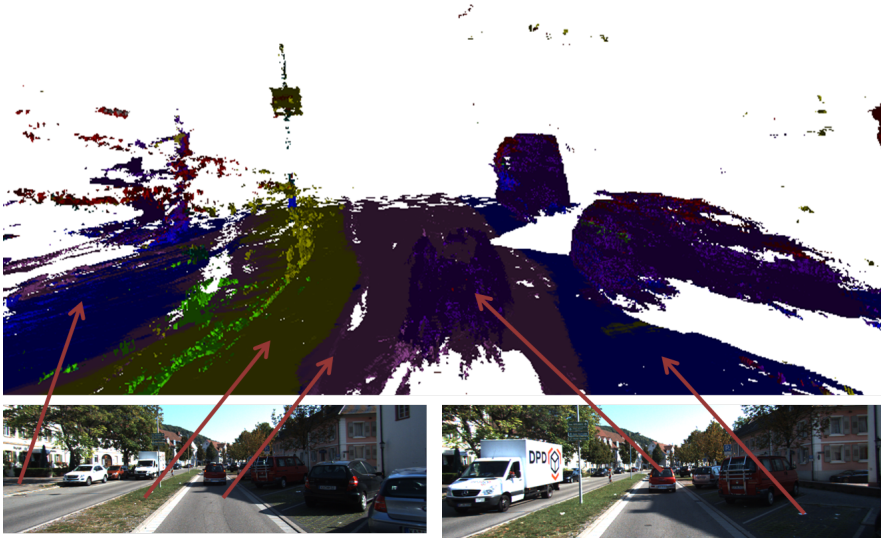


Figure 7.3: Qualtitative results on the KITTI dataset. We show how our approach recovers very fine and close to dense model of moving objects such as car. We also recover good labelling of the scene such as road, pavement, vegetation, lamp post.

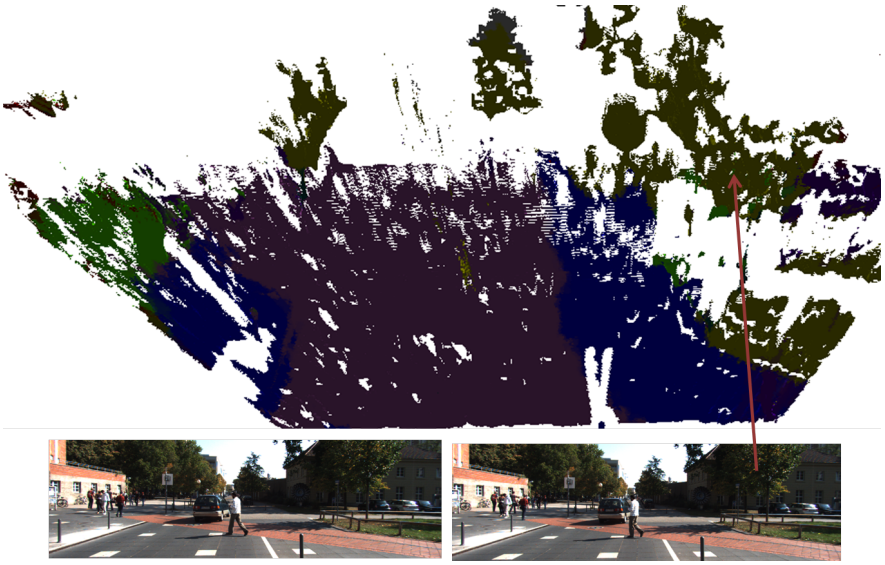


Figure 7.4: We show how our semantic fusion approach produces good reconstruction in environments with many moving objects. In this scene, there are many pedestrians, and our approach build dense and fine reconstruction and labelling of trees, roads, pavements etc.

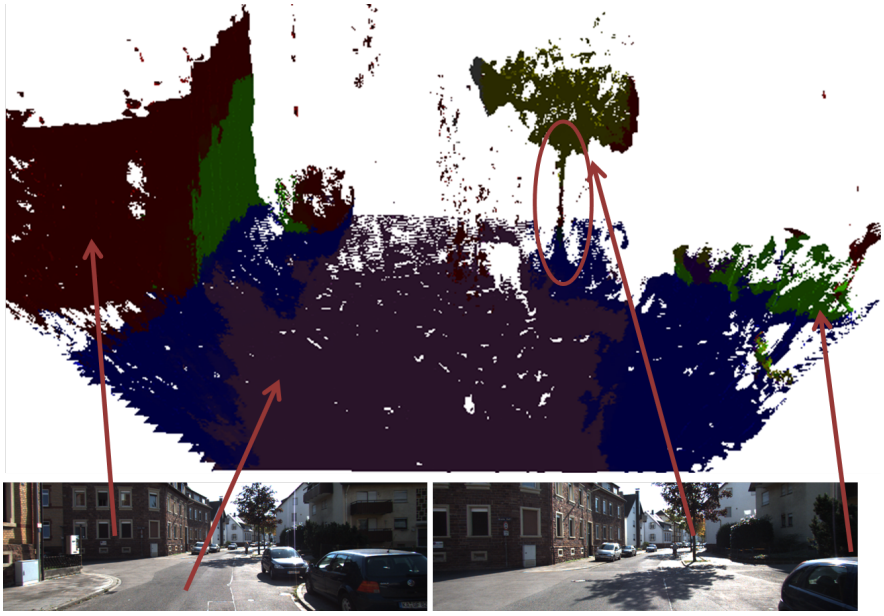


Figure 7.5: In this figure again we show how our approach recovers fine reconstruction and labelling of fine and thin objects such as trees.

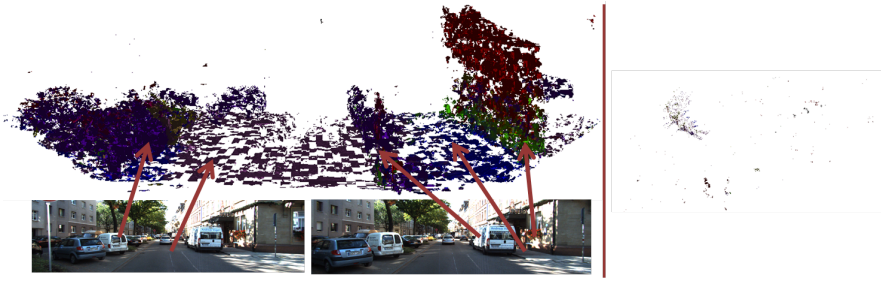


Figure 7.6: We show a case where ICP approach completely fails which our semantic fusion and visual odometry based pose estimation approach helps to recover the good 3D reconstruction. We observe that the fine objects such as green vegetation in bottom right is recovered. Similarly a small patch between the building and vehicle is recovered as well.

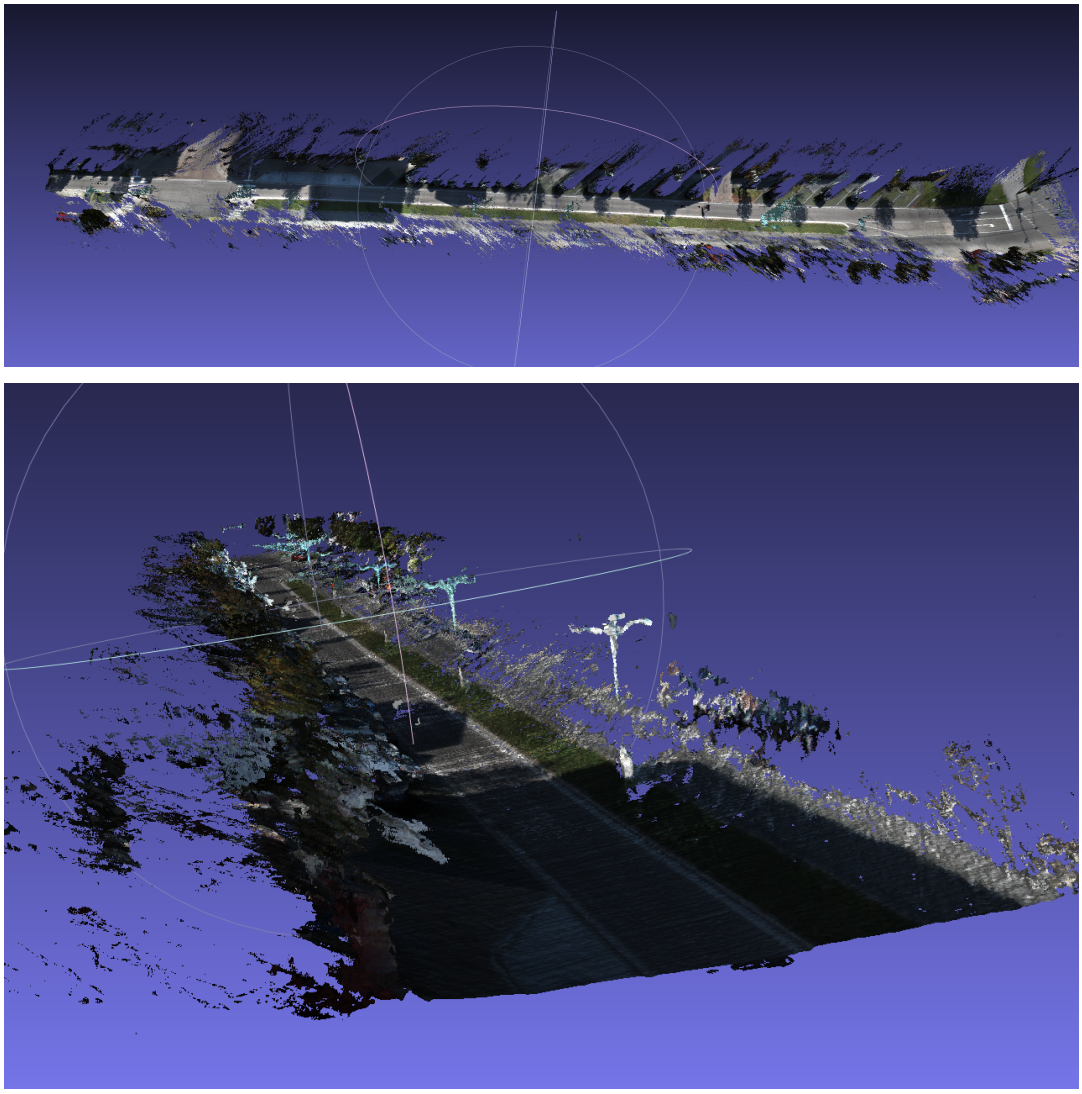


Figure 7.7: *We show large scale reconstruction of urban sequence from KITTI dataset generated from two different views. This sequence corresponds to around 200 metres and generated by around 150 stereo image pairs.*



Figure 7.8: *We show rendered RGB views on sequence taken from the KITTI dataset.*

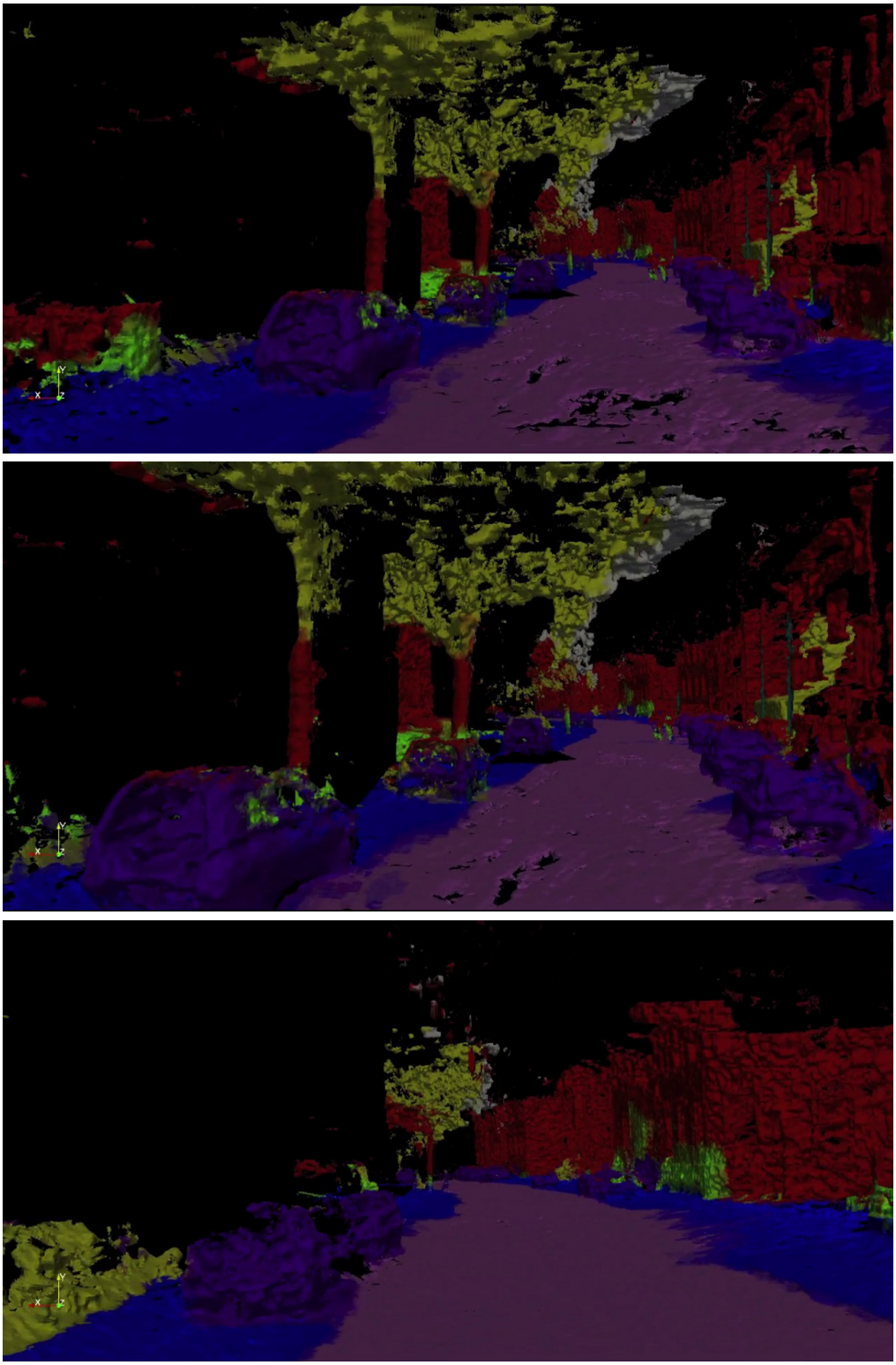


Figure 7.9: *In this figure again we show how our approach recovers very good reconstruction and labelling of the scenes taken from the KITTI dataset We recover very smooth reconstruction and labelling of road, building, cars, pavement, vegetation.*

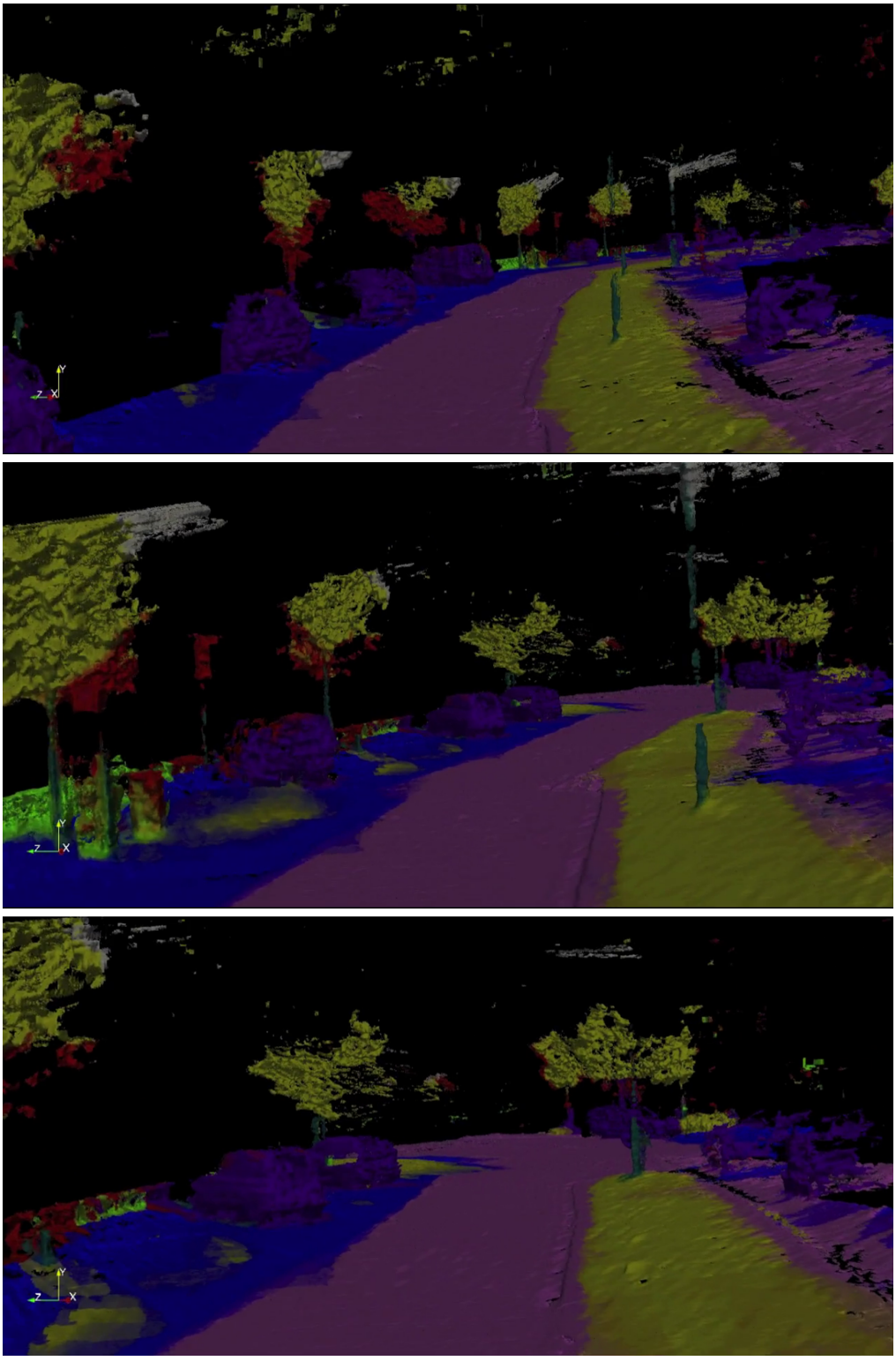


Figure 7.10: *More results to highlight that our approach recovers very good reconstruction and labelling of the scenes taken from the KITTI dataset We recover very smooth reconstruction and labelling of road, building, cars, pavement, vegetation.*

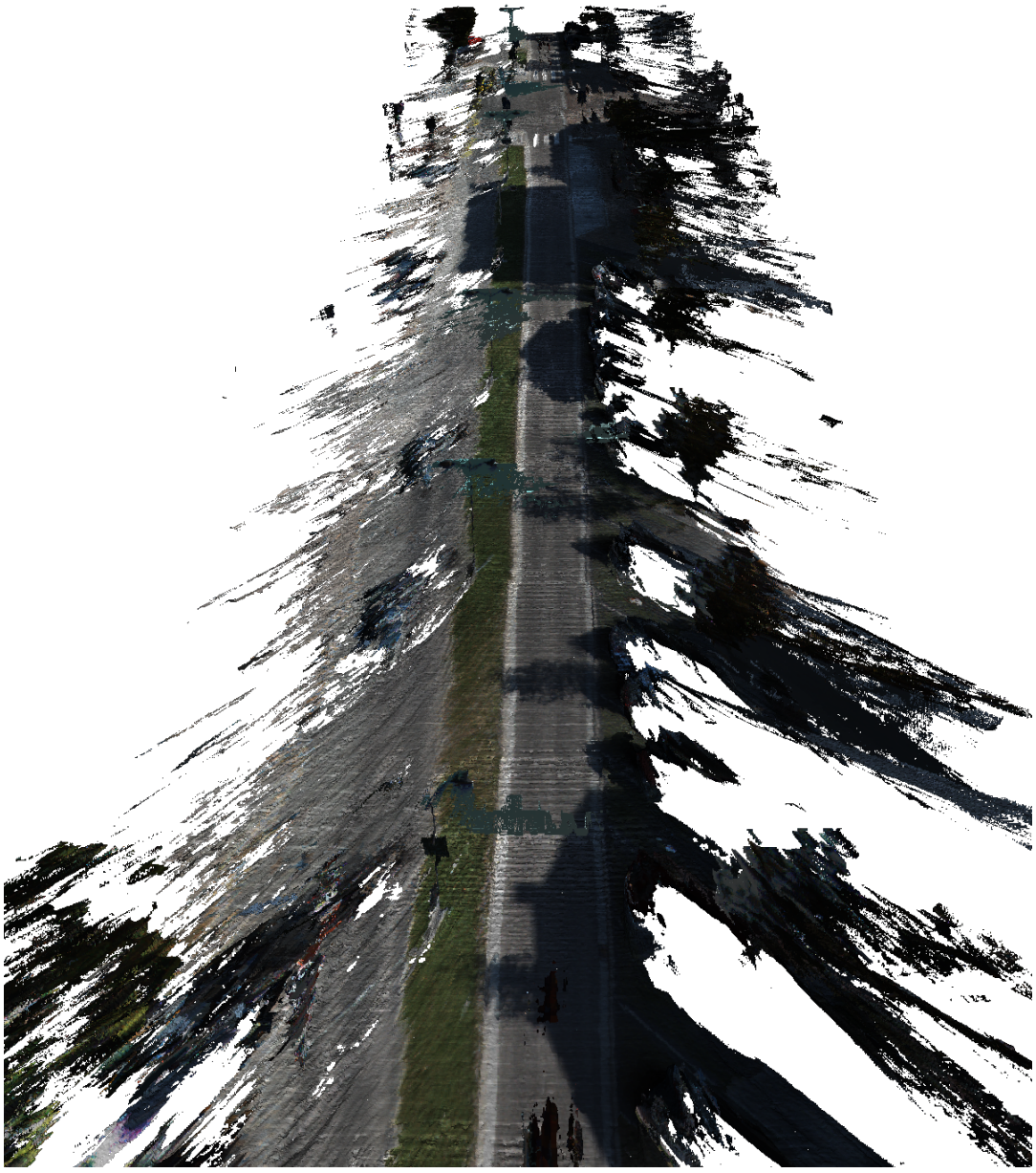


Figure 7.11: *We show more large scale reconstruction of urban sequence from KITTI dataset generated from two different views.*

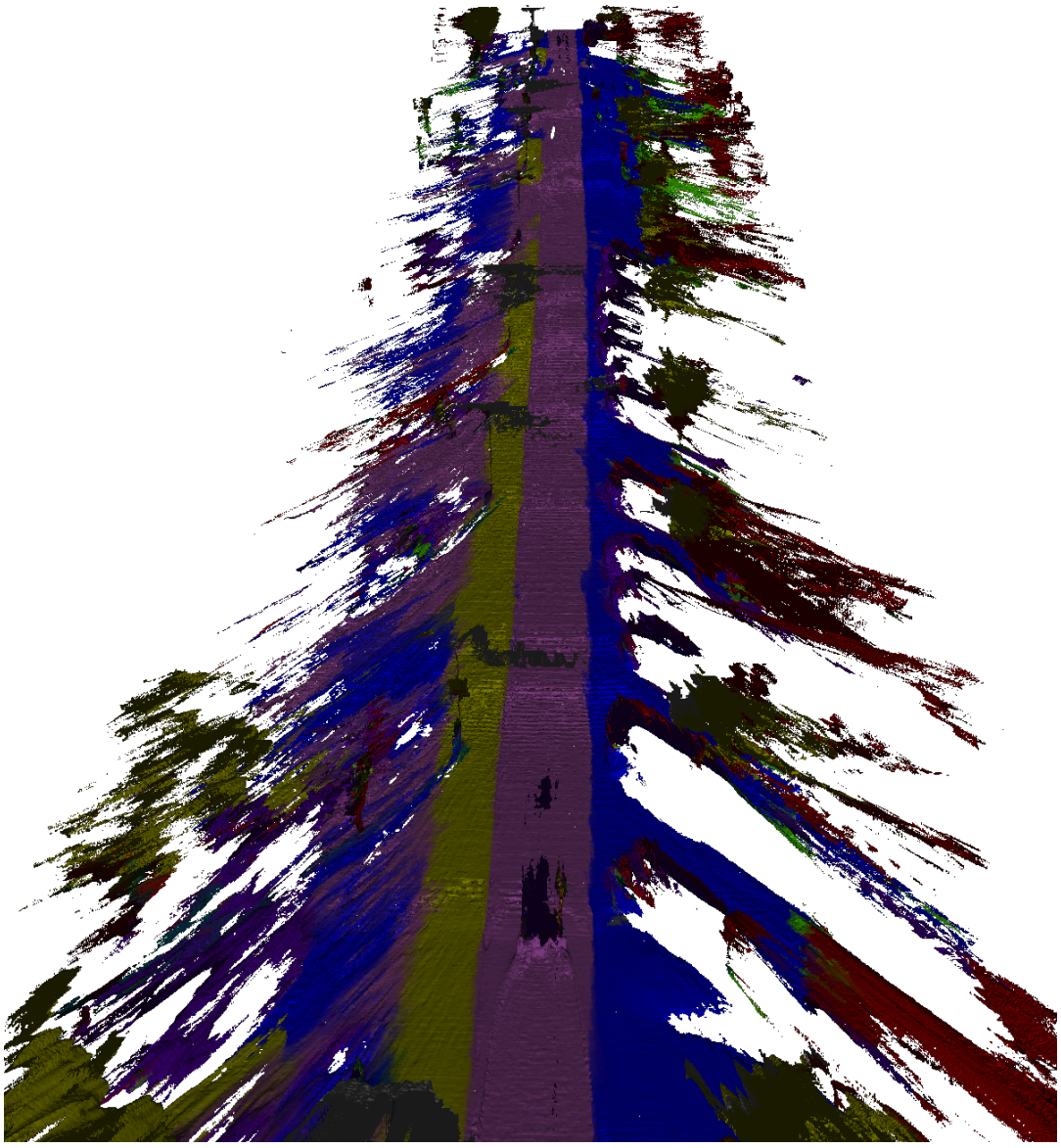


Figure 7.12: *We show more large scale reconstruction and labelling of urban sequence from KITTI dataset generated from two different views.*



Figure 7.13: *We show more large scale reconstruction of urban sequence from KITTI dataset generated from two different views.*

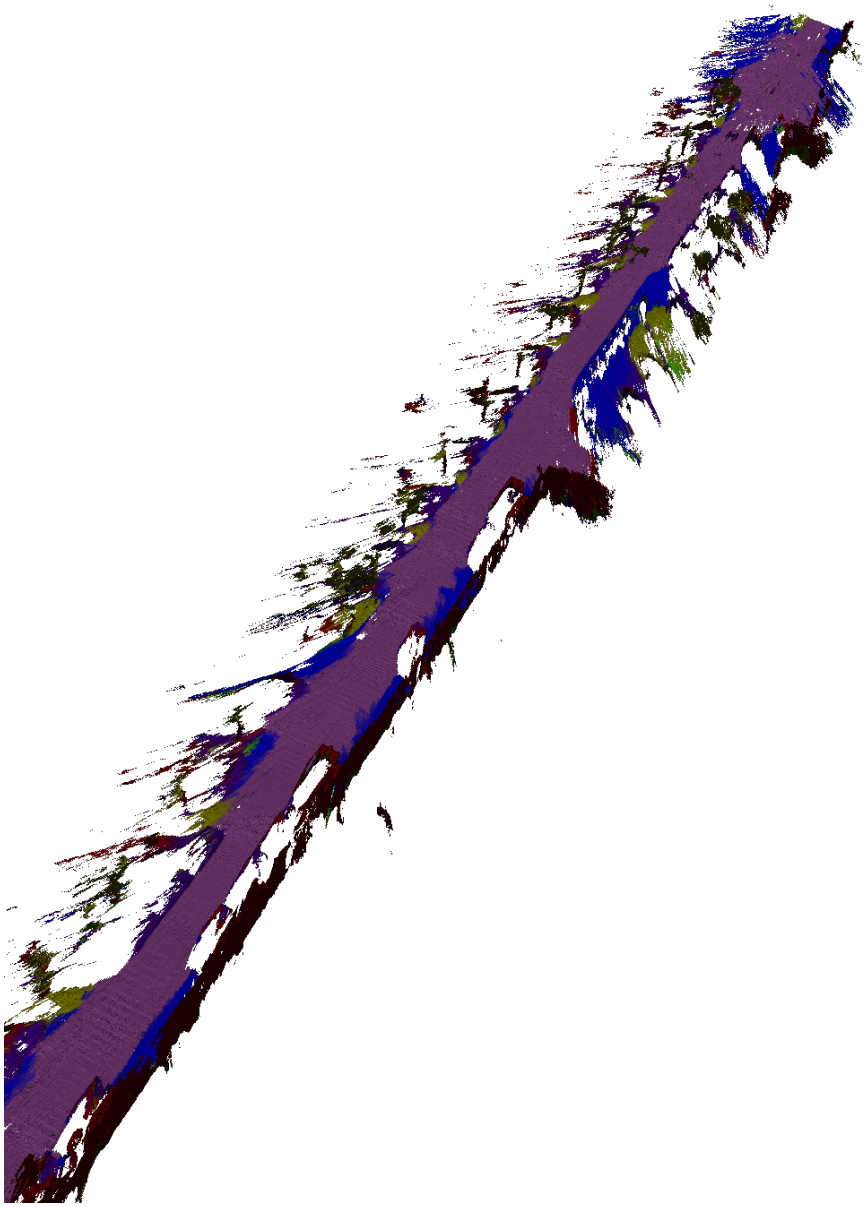


Figure 7.14: *We show more large scale reconstruction and labelling of urban sequence from KITTI dataset generated from two different views.*

Chapter 8

Applications

Many models have been proposed to estimate human pose and segmentation by leveraging information from several sources. A standard approach is to formulate it in a dual decomposition framework. However, these models generally suffer from the problem of high computational complexity. In this chapter, we propose **PoseField**, a new highly efficient filter-based mean-field inference approach for jointly estimating human segmentation, pose, per-pixel body parts, and depth given stereo pairs of images. We extensively evaluate the efficiency and accuracy offered by our approach on H2View [151], and Buffy [44] datasets. We achieve 20 to 70 times speedup compared to the current state-of-the-art methods, as well as achieving better accuracy in all these cases.

8.1 Introduction

Human pose estimation and segmentation have long been popular tasks in computer vision, and a large body of research has been developed on these problems [93, 159, 168, 187, 194]. Several of these methods model pose estimation and segmentation problems separately, and fail to capture the large variability and deformation in appearance and the structure of humans.

However, when segmentation and pose estimation results are considered together, one can observe discrepancies, for example a foreground region not corresponding to any detected body part, or vice versa. Joining the two problems together, either sequentially or simultaneously, can help to remove these discrepancies. Researchers have thus begun to consider the possibility of jointly estimating these outputs, leveraging the information from several high-level and low-level cues.

A number of methods insert various algorithms into a pipeline, where the result of one algorithm is used to initialize another. For example, Bray et al. tackle the problem of human segmentation by introducing a pose-specific MRF, encouraging the segmentation result to look "human-like" [21]. Similarly, Kumar et al. use layered pictorial structures to generate an object category specific MRF to improve segmentation [91]. The problem with this kind of approach is that errors in one part of the algorithm can propagate to later stages. Joint inference can be used to overcome this issue; Ladický et al. obtain an improvement in object class segmentation by incorporating global information from object detectors and object co-occurrence terms [96], solving detection and segmentation with one CRF. Further, Ladický et al. frame joint estimation of object classes and disparity as CRF problems in the product label space, and solve the two tasks together [98].

Additionally, in the context of human pose estimation and segmentation, Wang and Koller propose a dual-decomposition based inference method [84] in

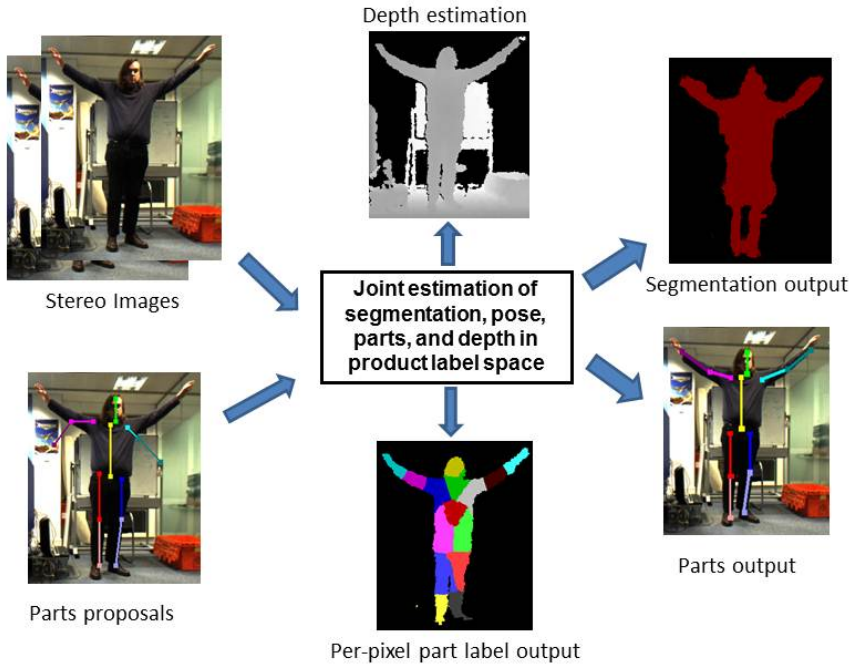


Figure 8.1: *Given stereo pairs and initial part proposals, our approach jointly estimates the human segmentation, pose, and depth, considering the relationships between per-pixel body part labels and part configurations.*

a multi-level CRF framework to jointly estimate pose and segmentation by introducing variables that capture the coupling between these two problems [184]. Extending their formulation, Sheasby et al. [151] add depth information, thus allowing human pose, segmentation and depth to be solved together [151, 152].

The complexity of such joint frameworks is a serious issue; if the framework is to be used for applications such as security and video gaming, fast output is required. In such situations, it might prove desirable to find an efficiently solvable approximation of the original problem. One such method that can be applied here is mean-field inference [81]. For a certain class of pairwise terms, mean-field inference has been shown to be very powerful in solving the object class segmentation problem, and object-stereo correspondence problems in CRF frameworks, providing an order-of-magnitude speedup [181]. In this chapter, we propose a highly efficient filter-based mean-field approach to perform joint estimation of human segmentation, pose, per-pixel part labels, and disparity in the product label space, producing a significant improvement in speed.

Further, to model the human skeleton, we propose a hierarchical model that captures relations on multiple levels. At the lowest level, we estimate part labels per pixel. Such a representation has been shown to be successful in generating body parts proposals and pose estimation by Shotton et al. [155]. Secondly, the higher level tries to find the best configuration from a set of part proposals. Our

framework is represented graphically in Fig 8.1.

Finally we extensively evaluate the efficiency and accuracy offered by our mean-field approach on two datasets: H2View [152], and Buffy [44]. We show results for segmentation, per pixel part labelling and pose estimation; disparity computation is used to improve these results, but is not quantitatively evaluated as it is not feasible to obtain dense ground truth data. We achieve 20-70 times speedup compared to the current state-of-the-art graph-cuts based dual-decomposition approach [151], as well achieving better accuracy in all cases.

The remainder of the chapter is structured as follows: an overview of dense CRF formulation is introduced in the next section, while our body part formulation is discussed in Section 8.2.1. We describe our joint inference framework in Section 8.3 and learning of different parameters is discussed in the Section 8.4. Results follow in Section 8.5, and Section 8.6 concludes the chapter.

8.2 Overview of Dense Random Field Formulation

The goal of our joint optimization framework is to estimate human segmentation and pose, together with part labels at the pixel level, and perform stereo reconstruction given a pair of stereo images. These problems however can be separately solved in a conditional random field (CRF) framework. Thus, before going into the details of the joint modelling and inference, we provide the models for solving them separately. Let $\mathcal{X}^{\mathcal{S}} = \{X_1^S, ..., X_N^S\}$, $\mathcal{X}^{\mathcal{J}} = \{X_1^J, ..., X_N^J\}$, $\mathcal{X}^{\mathcal{D}} = \{X_1^D, ..., X_N^D\}$ be the human segmentation, per-pixel part and disparity variables respectively. We assume each of these random variables is associated with each pixel in the image $\mathcal{N} = \{1, ..., N\}$. Further, each X_i^S takes a label from segmentation label set $\mathcal{L}^S \in \{0, 1\}$, X_i^D takes a label from $\mathcal{L}^D \in \{0...D\}$ disparity labels and X_i^J takes a label from $\mathcal{L}^J \in \{0, 1, ..., M\}$ where 0 represents background and M is the number of body parts.

First, we give details of the energy function for the segmentation variables. Assuming the true distribution of the segmentation variables is captured by the unary and pairwise terms, the energy function takes the following form:

$$E^S(\mathbf{x}^S) = \sum_{i \in V} \psi_u^S(x_i^S) + \sum_{i \in V, j \in N_i} \psi_p^S(x_i^S, x_j^S) \tag{8.2.1}$$

where N_i represents the neighborhood of the variable i, $\psi_u^S(x_i^S)$ represent unary terms for human segmentation class and $\psi_p^S(x_i^S, x_j^S)$ are pairwise terms capturing the interaction between a pair of segment variables. The human object specific

unary cost $\psi_u^S(x_i^S)$ is computed based on a boosted unary classifier on image-specific appearance using the model of Shotton et al. [156]. The pairwise terms between human segmentation variables ψ_p^S take the form of Potts models weighted by edge-preserving Gaussian kernels [87] as:

$$\psi_p^S(x_i^S, x_j^S) = \mu(x_i^S, x_j^S) \sum_{v=1}^{V} w^{(v)} k^{(v)}(\mathbf{f}_i, \mathbf{f}_j) \tag{8.2.2}$$

where $\mu(.,.)$ is an arbitrary *label compatibility function*, while the functions $k^{(v)}(.,.)$, $v = 1...V$ are Gaussian kernels defined on feature vectors $\mathbf{f}_i$, $\mathbf{f}_j$ derived from the image data at locations i and j (where Krahenbuhl and Koltun [87] form $\mathbf{f}_i$ by concatenating the intensity values at pixel i with the horizontal and vertical positions of pixel i in the image), and $w^{(v)}$, $m = 1...V$ are used to weight the kernels.

Similarly we define the energy functions over the per-pixel part and disparity variables as:

$$E^J(\mathbf{x^J}) = \textstyle\sum_{i \in V} \psi_u^J(x_i^J) + \sum_{i \in V, j \in N_i} \psi_p^J(x_i^J, x_j^J) \tag{8.2.3}$$

$$E^D(\mathbf{x^D}) = \textstyle\sum_{i \in V} \psi_u^D(x_i^D) + \sum_{i \in V, j \in N_i} \psi_p^D(x_i^D, x_j^D) \tag{8.2.4}$$

where $\psi_u^J(x_i^J)$ and $\psi_u^D(x_i^D)$ represent unary term for the per-pixel part and disparity variables respectively, and $\psi_p^J(x_i^J, x_j^J)$ and $\psi_p^D(x_i^D, x_j^D)$ are pairwise terms capturing the interaction between pairs of per-pixel part and disparity variables respectively. The per-pixel part variable dependent unary cost $\psi_u^J(x_i^J)$ is computed based on a boosted unary classifier on depth image. Further, if we do not have ground truth for the depth map, we can learn the unary cost for the per-pixel parts on image-specific appearance. The unary cost $\psi_u^D(x_i^D)$ for the disparity variables measures the color agreement of a pixel with its corresponding pixel i from the stereo-pair given a choice of disparity x_i^D. The pairwise terms for both these variables ψ_p^J and ψ_p^D take the form of contrast-sensitive Potts models as mentioned earlier.

8.2.1 Joint Formulation

The goal of our joint optimization framework is to estimate human segmentation and pose, together with part labels at the pixel level, and also perform stereo reconstruction. We formulate the problem in a conditional random field (CRF) framework as a product label space in a hierarchical framework. At the lower level, we define the random variables $\mathcal{X} = [\mathcal{X}^S, \mathcal{X}^J, \mathcal{X}^D]$, where $\mathcal{X}$ takes a label from the product label space $\mathcal{L} = \{(\mathcal{L}^S \times \mathcal{L}^J \times \mathcal{L}^D)^N\}$. For specifying human

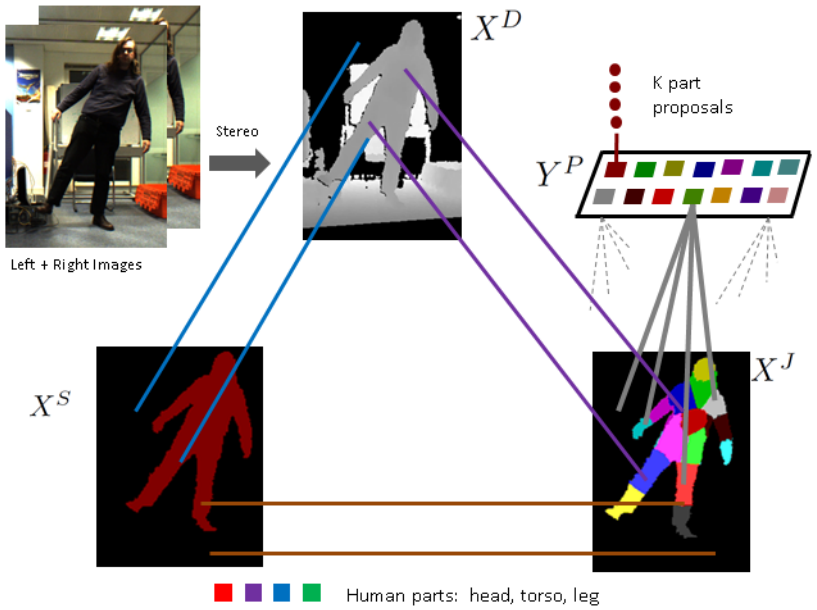


Figure 8.2: **PoseField** model jointly estimates the per-pixel human/background segmentation, body part, and disparity labels. Further, the relationship between per pixel body part label, and part configurations are captured in a hierarchical model with information propagating between these different layers. (Best viewed in color)

pose, we define a second layer, represented by a set of latent variables $\mathcal{Y} = \{Y_1, Y_2, ..., Y_M\}$ corresponding to the M body parts, each taking labels from $\mathcal{L}^P \in \{0, ..., K\}$ where $1, 2, ..., K$ corresponds to the K part proposals generated for each body part, and zero represents the background class. We generate K part proposals using the model of Yang and Ramanan [194]. The graphical model explaining our hierarchical joint model is shown in the Fig 8.2.

8.2.2 Joint energy function

Given the above model, we wish to define an energy function which is general enough to capture sufficient mutual interaction between the variables while still providing scope for efficient inference. For this reason, we assume our energy function to take the following form:

$$E(\mathbf{x}, \mathbf{y}) = E^S(\mathbf{x}^S) + E^D(\mathbf{x}^D) + E^J(\mathbf{x}^J) + E^P(\mathbf{y}) + E^{SJ}(\mathbf{x}^S, \mathbf{x}^J) + E^{SD}(\mathbf{x}^S, \mathbf{x}^D) + E^{DJ}(\mathbf{x}^D, \mathbf{x}^J) + E^{JP}(\mathbf{x}^J, \mathbf{y}) \quad (8.2.5)$$

Here, our joint model has been factorized into separate layers representing human segmentation, disparity, per pixel part and latent part variables. The in-

dividual terms at the layers are captured by E^S, E^D, E^J as defined earlier and E^P, the energy function for the latent part variables, details of which are provided later in this section. Further, in order to incorporate the dependency between these variables, we add pairwise interactions between these CRF layers. E^{SJ}, E^{SD}, E^{DJ} captures the interaction between (*segment,per-pixel part*), (*segment,disparity*) and (*disparity, per-pixel part*) variables. The term E^{JP} captures the mutual interaction between the (*per-pixel part, latent part*) variables. We design the forms of these pairwise interactions to allow efficient and accurate inference; details are provided below.

Per-part terms E^P : In our hierarchical model, the top layer corresponds to the human part variables Y, which involve per-part unary cost $\psi_u^P(x_i = k)$ for associating the ith part to the kth proposal or to the background [151], and the pairwise term $\psi_p^P(y_i, y_j)$ penalizes the case where parts that should be connected are distant from one another in image space. The per-part unary term $\psi_u^P(y_i = k)$ is the score generated by the Yang and Ramanan model [194].

***Segment, per-pixel part* terms (E^{SJ})** : The joint human segmentation and per-pixel body part term, E^{SJ}, encodes the relation between segmentation and per-pixel body part. Specifically, we expect a variable that takes a body part label to belong to the foreground class, and vice versa. We pay a cost of C^{SJ} for violation of this constraint, incorporated through a pairwise interaction between the segmentation and per-pixel part variables; this interaction takes the following form:

$$E^{SJ} = \psi_p^{SJ}(\mathbf{x}^S,\mathbf{x}^J) = \sum_{i=1}^{N} C^{SJ} \cdot [(x_i^S = 1) \wedge (x_i^J = 0)] + \sum_{i=1}^{N} C^{SJ} \cdot [(x_i^S = 0) \wedge (x_i^J \neq 0)] \quad (8.2.6)$$

***Segment, disparity* terms (E^{SD})** : Additionally, our joint object-depth cost E^{SD} encourages pixels with a high disparity to be classed as foreground, and pixels with a low disparity to be classified as background. We penalize the violation of this constraint by a cost C^{SD}. Following the formulation of [151], we first generate a segmentation map $\mathcal{F} = \{F_1, F_2, ..., F_N\}$ by thresholding the disparity map, thus each F_i takes a label from L^S. We would expect the prior map $\mathcal{F}$ to agree with the segmentation result, so that pixels taking human label ($f_i = 1$) are classified as human, and vice versa, otherwise we pay a cost C^{SD} for violation

of this constraint:

$$E^{SD} = \psi_p^{SD}(\mathbf{x}^S, \mathbf{x}^D) = \sum_{i=1}^{N} C^{SD} \cdot [(x_i^S = 1) \wedge (f_i = 0)] + \sum_{i=1}^{N} C^{SD} \cdot [(x_i^S = 0) \wedge (f_i = 1)] \quad (8.2.7)$$

***Per-pixel part, disparity* terms (E^{JD})** : The joint energy term E^{JD} encodes the relationship between the per-pixel body part variables and the disparity variables. As with the cost term E^{SD}, we use a flood fill to generate a segmentation map $\mathcal{F} = \{F_1, F_2, \ldots, F_N\}$ which gives us a prior based on disparity. We expect pixels classed as human by this prior (so $f_i = 1$) to be assigned to a body part label, so $x_i^J > 0$. Conversely, pixels classed as background ($f_i = 0$) should be assigned to the background label ($x_i^J = 0$). Therefore, the energy term has the following form:

$$E^{JD} = \psi_p^{JD}(\mathbf{x}^J, \mathbf{x}^D) = \sum_{i=1}^{N} C^{JD} \cdot [(x_i^J > 0) \wedge (f_i = 0)] + \sum_{i=1}^{N} C^{JD} \cdot [(x_i^J = 0) \wedge (f_i = 1)] \quad (8.2.8)$$

***Per-pixel part, latent part* terms (E^{JP})** : E^{JP} enforces the constraint that when a body part l is present in the solution at the pixel level, then the variable Y_l^P corresponding to the part l must be on, otherwise we pay a cost of C^{JP}.

$$E^{JP} = \psi_p^{JP}(\mathbf{x}^J, \mathbf{y}) = C^{JP} \cdot \sum_{l \in M} [(y_l = 0) \wedge (\sum_i [x_i^J = l]) > 0] \quad (8.2.9)$$

8.3 Inference in the Joint Model

Given the above complex hierarchical model, we now propose a new mean-field based inference approach to perform efficient inference for joint estimation. But, before going into details of our approach, we give a general form of mean-field update. We also highlight the work of Krahenbuhl and Koltun [87] for filter-based efficient inference in fully connected pairwise CRFs. This model was later extended by Vineet et.al. [181] to incorporate higher order potentials, and to solve jointly the object-stereo correspondence problems.

Let us consider a general form of energy function:

$$E(\mathbf{Z}|\mathbf{I}) = \sum_{c \in \mathcal{C}} \psi_c(\mathbf{z}_c|\mathbf{I}) \tag{8.3.1}$$

where $\mathbf{Z}$ is a joint assignment of the random variables $\mathcal{Z} = \{Z_1, ..., Z_{N_Z}\}$, $\mathcal{C}$ is a set of cliques each consisting of a subset of random variables $c \subseteq \mathcal{Z}$, and associated with a potential function ψ_c over settings of the random variables in c, $\mathbf{z}_c$. In Sec. 8.2 we have that $\mathcal{Z} = \mathcal{X}^S$, that each X_i takes values in the set $\mathcal{L}^S$ of human labels, and that $\mathcal{C}$ contains unary and pairwise cliques of the types discussed. In general, in the models discussed below we will have that $\mathcal{X}^S \subseteq \mathcal{Z}$, so that $\mathcal{Z}$ may also include other random variables (e.g. latent variables) which may take values in different label sets.

Considering this model, the general form of the mean-field update equation (see [81]) is:

$$Q_i(z_i = \nu) \quad = \quad \frac{1}{\tilde{Z}_i} \exp\{-\sum_{c \in \mathcal{C}} \sum_{\{\mathbf{z}_c | z_i = \nu\}} Q_{c-i}(\mathbf{z}_{c-i}) \cdot \psi_c(\mathbf{z}_c)\} \tag{8.3.2}$$

where ν is a value in the domain of the random variable z_i, $\mathbf{z}_c$ denotes an assignment of all variables in clique c, $\mathbf{z}_{c-i}$ an assignment of all variables apart from Z_i, and Q_{c-i} denotes the marginal distribution of all variables in c apart from Z_i derived from the joint distribution Q. $\tilde{Z}_i = \sum_\nu \exp\{-\sum_{c \in \mathcal{C}} \sum_{\{\mathbf{z}_c | z_i = \nu\}} Q_{c-i}(\mathbf{z}_{c-i}) \cdot \psi_c(\mathbf{z}_c)\}$ is a normalizing constant for random variable z_i. We note that the summations $\sum_{\{\mathbf{z}_c | z_i = \nu\}} Q_{c-i}(\mathbf{z}_{c-i}) \cdot \psi_c(\mathbf{z}_c)$ in Eq. 8.3.2 evaluate the expected value of ψ_c over Q given that Z_i takes the value ν.

Following this general update strategy, the updates for the densely connected pairwise model in Eq. 8.2.1 are derived by evaluating Eq. 8.3.2 across the unary and pairwise potentials defined in Sec. 8.2 for $z_i = x_{1...N}$ and $\nu = 0...L$. For the densely connected pairwise CRF model, the mean-field update takes the following form:

$$Q_i(z_i = l) = \frac{1}{\tilde{Z}_i} \exp\{-\psi_i(z_i) - \sum_{l' \in \mathcal{L}} \sum_{j \neq i} Q_j(z_j = l') \psi_{ij}(z_i, z_j)\} \tag{8.3.3}$$

With this mean-field update, Krahenbuhl and Koltun [87] proposed a filter-based method for performing fast inference thus reducing the complexity from $O(N^2)$ to $O(N)$ under the assumption that the pairwise potentials take the form of a linear combination of Gaussian kernels. They show how the expensive message passing update in the mean-field is approximated by a convolution with a bilateral filter in a high dimensional space. Given this Gaussian convolution, they

use a permutohedral lattice based bilateral filtering method [2] for performing efficient inference. They run the update equation for a fixed number of iterations, where each iteration leads to a decrease in the KL-divergence value. To extract a solution, they evaluate the approximate maximum posterior marginal as $z_i^* = \max_{z_i} Q_i(z_i)$.

8.3.1 Efficient Inference

In our framework, we need to jointly estimate the best possible configurations of the segmentation variables X^S, per-pixel part variables X^J, disparity variable X^D and part variable Y^P by minimizing the energy function $E(\mathbf{x}, y)$ in Eq. 8.2.5. We now provide the details of our mean-field updates for efficient joint inference.

Update for *segment* variables (X^S) : Given the energy function detailed in Sec. 8.2.2, the marginal update for human segmentation variable X_i^S takes the following form:

$$Q_i^S(x_{[i,l]}^S) = \frac{1}{Z_i^S} \exp\{-\psi^S(x_i^S) - \sum_{l' \in \mathcal{L}^\mathcal{J}} \sum_{j \neq i} Q_j^S(x_{[j,l']}^S)\psi(x_i^S, x_j^S) \\ - \sum_{l' \in \mathcal{L}^\mathcal{D}} Q_i^D(x_{[i,l']}^D)\psi(x_i^D, x_i^S) - \sum_{l' \in \mathcal{L}^\mathcal{J}} Q_i^J(x_{[i,l']}^J)\psi(x_i^S, x_i^J)\} \tag{8.3.4}$$

where $Q_i^D(x_{[i,l']}^D)\psi(x_i^D, x_i^S)$ and $Q_i^J(x_{[i,l']}^J)\psi(x_i^S, x_i^J)$ are the messages from disparity and per-pixel part variables respectively to the segmentation variables. Thus, these messages enforce the consistency between the segmentation, disparity and per-pixel part term variables. We write $x_{[i,l]}$ for $(x_i = l)$ and the same notation will be followed subsequently.

Update for *disparity* variables (X^D) : Similar to the updates for X_i^S, the marginal update for the per-pixel depth variables X_i^D takes the following form:

$$Q_i^D(x_{[i,l]}^D) = \frac{1}{Z_i^D} \exp\{-\psi^D(x_i^D) - \sum_{l' \in \mathcal{L}^D} \sum_{j \neq i} Q_j^D(x_{[j,l']}^D)\psi(x_i^D, x_j^D) \\ - \sum_{l' \in \mathcal{L}^S} Q_i^S(x_{[i,l']}^S)\psi(x_i^D, x_i^S) - \sum_{l' \in \mathcal{L}^J} Q_i^J(x_{[i,l']}^J)\psi(x_i^J, x_i^D)\} \tag{8.3.5}$$

where $Q_i^S(x_{[i,l']}^S)\psi(x_i^D, x_i^S)$ and $Q_i^J(x_{[i,l']}^J)\psi(x_i^J, x_i^D)$ correspond to the messages from the segmentation and per-pixel part variables to the disparity variables.

Update for *per-pixel part* variables (X^J) : The per-pixel part variable X_i^J takes messages from part configuration in the hierarchy along with the messages

from the other per-pixel part variables, segmentation variables and disparity variables. Thus, the marginal update for the per-pixel part variables X_i^J take the following form:

$$Q_i^J(x_{[i,l]}^J) = \frac{1}{Z_i^J}\exp\{-\psi_u^J(x_i^J) - \sum_{l'\in\mathcal{L}^\mathcal{J}}\sum_{j\neq i} Q_j^J(x_{[j,l']}^J)\psi(x_i^J, x_j^J) \\ - \sum_{l'\in\mathcal{L}^\mathcal{D}} Q_i^D(x_{[i,l']}^D)\psi(x_i^J, x_i^D) - \sum_{l'\in\mathcal{L}^\mathcal{S}} Q_i^S(x_{[i,l']}^S)\psi(x_i^J, x_i^S) - \sum_{l'\in\mathcal{L}^\mathcal{P}} Q_i^P(y_{[i,l']})\psi(y_i, x_i^J)\} \tag{8.3.6}$$

Here $Q_i^P(y_{[i,l']})\psi(y_i, x_i^J)$ carry messages from the valid part configuration in the hierarchy to the per-pixel part variables, and $Q_i^S(x_{[i,l']}^S)\psi(x_i^J, x_i^S)$ and $Q_i^D(x_{[i,l']}^D)$ $\psi(x_i^D, x_i^J)$ correspond to the messages from the segmentation and disparity variables to per-pixel part variables.

It is also to be noted that the required expectation update for messages from other joint variables, e.g. messages from segmentation variables to disparity variables, contribute a time complexity of $O(N)$. Thus, the marginal update steps do not increase the overall time complexity.

Update for *latent part* variables (Y) : Finally, the mean-field update for the part variables in the hierarchy corresponds to:

$$Q_i^P(y_{[i,l]}) \propto \exp\{-\psi_u(y_i) - \sum_{j'\in\mathcal{L}^\mathcal{J}}\sum_{j=1}^{N} Q_j^J(x_{[j,j']}^J)\psi(y_i, x_j^J) \tag{8.3.7}$$

where $Q_j^J(x_{[j,j']}^J)\psi(y_i, x_j^J)$ corresponds to the messages from the per-pixel part variables to the part configuration variables. Evaluation of the expectation for part variables contributes $O(N)$ to the overall complexity. Thus, our inference method does not increase the overall complexity of $O(N)$ for fully connected pairwise updates.

8.4 Learning

The weights $C^{SJ}, C^{SD}, C^{JD}, C^{JP}$ capturing the relationships between variables at different CRF layers are set through cross-validation. Our cross validation step to search for good set of parameters to weight these different terms in Eq. 5 is greedy in the sense that we set them one at a time sequentially. This way of sequential learning ensured an efficient way to search for a good set of the parameters without going through all the possible joint configurations of the

parameters. Structured learning [177] provides a possible future direction to learn these parameters, however our focus was efficient inference. Further, the Gaussian kernel parameters are set through cross-validation as well.

8.5 Experiments

In this section, we demonstrate the efficiency and accuracy provided by our approach on the H2View [151] dataset. Further, to highlight the generalization of our approach, we also conduct experiment on the Buffy [44] dataset where we do not have stereo pairs of images. In all experiments, timings are based on code run on an Intel® Xeon ® 3.33 GHz processor, and we fix the number of full mean-field update iterations to 5 for all models. As a baseline, we compare our approach for the joint estimation of human segmentation, pose, per-pixel part and disparity with the dual-decomposition based model of Sheasby et al. [151]. Further, we compare our joint approach against some other state-of-the-art approaches which do not perform any joint estimation. For example, we compare our human segmentation results against a graph-cuts based AHCRF [95] and the mean-field model of Krähenbühl et al. [87]. We assess human segmentation accuracy in terms of the overall percentage of pixels correctly labelled, the average recall and intersection/union score per class (defined in terms of the true/false positives/negatives for a given class as TP/(TP+FP+FN)). Similarly, for pose estimation, apart from comparing against the dual-decomposition based joint labelling model of Sheasby et.al. [151], we compare the probability of correct pose (PCP) criterion against the models of Yang and Ramanan [194], and Andriluka et al. [8], which do not perform joint labelling. In all these cases, we use the code provided by the authors for the AHCRF, Krähenbühl et al., Yang and Ramanan, Andriluka et al., and Sheasby et al. However we do not quantitatively evaluate the disparity results as we do not have the ground truth data for the disparity.

8.5.1 H2View dataset

The H2View dataset [151] comprises 1108 training images and 1598 test images consisting of humans in different poses performing standing, walking, crouching, and gesticulating actions in front of a stereo camera. Ground truth human segmentation, and pose are provided; we augment these with a per-pixel part labels.

We first show the accuracy and efficiency achieved by our method on the human segmentation results. We observe an improvement of almost 3.5% over the dual-decomposition based joint inference model of Sheasby et al. [151], almost 4.5% compared to AHCRF [95] and almost 4% over dense CRF [87] in the I/U

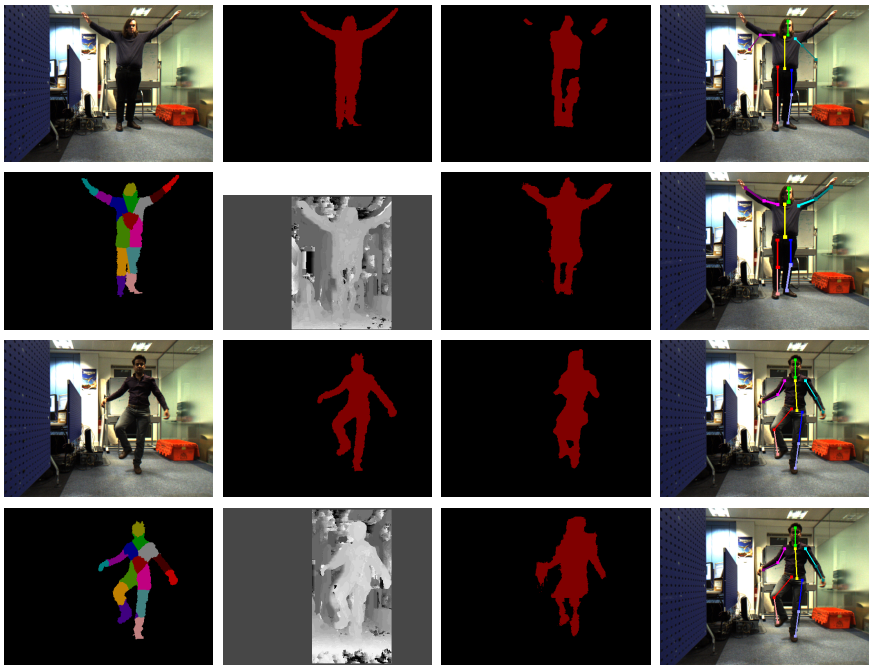


Figure 8.3: *Qualitative results on two sets of images from H2View dataset. First two rows correspond to the first image, and next two rows to the second image. From left to right: (top row) input image, ground truth for human segmentation, output from [151], pose estimation output from [151]; (second row): our per-pixel part label output, disparity estimation output, segmentation and pose-estimation outputs. Last two rows show the same set of images on the second input image. Our method is able to recover the limbs properly on both the segmentation and pose estimation problems. (Best viewed in color)*

score, shown in Tab. 8.1. Significantly, we observe a speed up of almost 3× over the model of Sheasby et al.. Further as far as pose estimation results are concerned, we achieve an improvement of almost 3.5% over Yang and Ramanan, 7% over Andriluka et al. in the PCP scores. Though these methods do not perform joint inference, we compare to highlight the importance of joint inference. Further compared to the model of Sheasby et al., we perform slightly worse in the PCP score, but we observe a speed up of almost 3× over their model. Quantitative results for pose estimation are as shown in Tab. 8.2. Additionally we observe qualitative improvement in both the segmentation and pose results, as shown in Fig. 8.3. As far as per-pixel part label accuracy is concerned, we achieve 94.43% of overall percentage of correctly labelled pixels, compared to 92.63% achieved by the dual-decomposition method of Sheasby et.al. [151], and 89.55% achieved by the graph-cuts based AHCRF method [95].

Method	Time (s)	Overall	Av.Re	I/U
Unary	0.36	96.12%	85.90%	78.94%
ALE [95]	1.5	96.14%	86.10%	80.14%
Sheasby [151]	35	96.67%	90.48%	81.52%
MF [87]	0.48	96.56%	86.12%	80.57%
Our	11.25	**97.14%**	92.32%	**84.60%**

Table 8.1: *Quantitative results on H2View dataset for human segmentation. The table compares timing and accuracy of our approach (last 2 lines) against the dual-decomposition model of Sheasby et.al. [151] as well as over other baselines. Note the significant improvement in class-average performance of our approach against the baselines.*

Method	T(s)	U/LL	U/FA	TO	Head	Overall
Sheasby [151]	35	83.43	54.56	90.05	89.8	73.18
Yang [194]	10	79.65	49.05	88.5	83.0	69.85
Andriluka [8]	35	74.85	47.7	83.9	76.0	66.03
Ours	11.2	82.86	55.16	89.05	86.20	73.12

Table 8.2: *The table compares timing and accuracy of our approach (last line) against the baseline for the pose estimation problem on H2View dataset [151]. Observe that our approach achieves almost* 3× *speedup against the dual-decomposition model of Sheasby et.al. [151] as well as over other baselines. U/LL represents average of upper and lower legs, and U/FA represents average of upper and fore arms.*

8.5.2 Buffy dataset

In order to show the generalization and effectiveness of our approach, we also evaluate our model on the Buffy dataset. We select a set of 476 images as training images, and 276 images as test images, using the same split as used in [44]. Since there are no depth images, we evaluate only on joint pose and segmentation problems. For the human segmentation case, our joint approach achieves a speed up of almost 20× compared to the dual-decomposition based method of Sheasby et al. [151], and 3× compared to AHCRF [95]. We also observe an improvement of almost 10% and 1% in the I/U scores respectively on segmentation results, shown in Tab. 8.3. Further, we observe an improvement of almost 0.4% over the Yang and Ramanan model and almost 7% over the model of Sheasby et al. model in the PCP score for the pose estimation problem, shown in Tab. 8.4. It should be noted that the results of the Yang and Ramanan model [194] reported in our paper is different than the one in their original paper since they first generate a set of detection windows by running an upper-body detector, and then evaluate



Figure 8.4: *Qualitative results on Buffy dataset [44]. From left to right: (first row:) input image, ground truth of segmentation, segmentation output before joint estimation, (second row:) segmentation output after joint estimation, pose output before joint estimation, and pose output after joint estimation. (Best viewed in color)*

Method	Time (s)	Overall	Av.Pr	I/U
Sheasby [151]	21	80.85%	85.80%	65.01%
ALE [95]	0.96	87.88%	86.05%	74.16%
MF [87]	0.26	88.40%	**86.47%**	75.01%
Ours	1.28	**88.79%**	86.45%	**75.18%**

Table 8.3: *Quantitative results on Buffy dataset for human segmentation problem. Observe the significant speedup (almost* $20\times$*) achieved compared to the dual-decomposition method of Sheasby et.al. [151] and over other approaches. Further, our approach achieves better accuracy than other methods as well.*

pose detection only on these detected windows. Here we evaluate the poses on whole image, thus a good detection of the non-detected person could be penalized. Further improvement through pose estimation within the detected boxes remains a possibility to our approach as well. However, our main goal is to show the efficiency achieved by our joint model without losing any accuracy given the same initial conditions. We also observe an improvement in qualitative results for both the segmentation and pose estimation problems, shown in Fig. 8.4.

8.6 Discussion

In this work, we proposed *PoseField*, an efficient mean-field based method for joint estimation of human segmentation, pose, per-pixel part and disparity. We

Method	T(s)	L	R	TO	H	Overall
Sheasby [151]	21	61.3	63.5	81.5	85.1	69.17
Yang [194]	1	66.6	71	87.3	90.5	75.6
Ours	1.28	68.2	71	87.6	90.2	76.05

Table 8.4: *The table compares timing and accuracy of our approach (last line) against the baseline for pose estimation problem. Observe that our approach achieves almost* 20× *speedup, and almost* 7% *improvement in accuracy over the dual-decomposition model of Sheasby et.al. [151].*

formulated this product label space problem in a hierarchical framework, which captures interactions between the pixel level (human/background, disparity, and body part labels), and the part level (head, torso, arm). Finally we have shown the value of our approach on the H2View and Buffy datasets. In each case, we have shown substantial improvement in inference speed (almost 20-70 ×) over the current state-of-the-art dual-decomposition methods, while also observing a good improvement in accuracies for both human segmentation and pose estimation problems. We believe our efficient inference algorithm would provide an alternative to some of the existing computationally expensive inference approaches in many other fields of computer vision where joint inference is required. Future directions include investigating new ways to improve the efficiency through parallelization and learning of the relationships between different layers of the hierarchy in a max-margin framework.

Chapter 9

Discussions

9.1 Contribution of the Thesis

In this thesis, we presented approaches and methods to solve several scene understanding problems. The main contributions are:

- **Higher Order Mean-field** We have introduced a set of techniques for incorporating higher-order terms into densely connected multi-label CRF models. As described, using our techniques bilateral filter-based methods remain suitable for inference in such models, effectively retaining the mean-field update complexity $O(MNL^2)$ as in [87] when higher-order P^n-Potts and co-occurrence models are used. This both increases the expressivity of existing fully connected CRF models, and opens up the possibility of using powerful filter-based inference in a range of models with higher-order terms. We have shown the value of such techniques for both joint object-stereo labelling and object class segmentation. In each case, we have shown substantial improvements in inference speed with respect to graph-cut based methods, particularly by using recent domain transform filtering techniques, while also observing similar or better accuracies.

- **Joint Human Segmentation-Pose Estimation** In this work, we proposed $PoseField$, an efficient mean-field based method for joint estimation of human segmentation, pose, per-pixel part and disparity. We formulated this product label space problem in a hierarchical framework, which captures interactions between the pixel level (human/background, disparity, and body part labels), and the part level (head, torso, arm). Finally we have shown the value of our approach on the H2View and Buffy datasets. In each case we have shown substantial improvement in inference speed (almost $20 - 70\times$) over the current state-of-the-art dual-decomposition methods, while also observing a good improvement in accuracies for both human segmentation and pose estimation problems. We believe our efficient inference algorithm would provide an alternative to some of the existing computationally expensive inference approaches in many other fields of computer vision where joint inference is required.

- **Joint Intrinsic Image-Object-Attribute Decomposition** In this work, we have explored the synergy effects between intrinsic properties of an images, and the objects and attributes present in the scene. We cast the problem in a joint energy minimization framework; thus our model is able to encode the strong correlations between intrinsic properties (*reflectance, shape, illumination*), objects (*table, tv-monitor*), and materials (*wooden,*

plastic) in a given scene. We have shown that dual-decomposition based techniques can be effectively applied to perform optimization in the joint model. We demonstrated its applicability on the extended versions of the NYU and Pascal datasets. We achieve both the qualitative and quantitative improvements for the object and attribute labeling, and qualitative improvement for the intrinsic images estimation.

- **Indoor Scene Reconstruction and Recognition** We have presented a system that allows a user to interactively segment and label an environment quickly and easily. A real-time algorithm reconstructs a 3D model of the surrounding scene as the user captures it. The user can interact with the world by touching surfaces and using voice commands to provide object category labels. A new, GPU-enabled mean-field inference algorithm then propagates the user's strokes through a volumetric random field model representing the scene. This results in a spatially smooth segmentation that respects object boundaries. In the background, the propagated labels are used to build a classifier, using our new streaming decision forest training algorithm. Once trained, the forest can predict a distribution over object labels for previously unseen voxels. These predictions are finally incorporated back into the 3D random field, and mean-field inference provides the final 3D semantic segmentation to the user.

- **Outdoor Scene Reconstruction and Recognition** Perceiving a 3D structure and recognizing the objects around us is central to our understanding of the world. In this chapter, we propose a robust and accurate approach for dense 3D reconstruction of outdoor and indoor environments along with associating them with object labels given stereo images pairs. At the core of our algorithm is a hash based fusion approach for reconstruction and a hash based approach to the mean-field inference for object labelling. In the process, we capture the synergy effects between the reconstruction and recognition tasks. Further, we harness the processing power of GPUs to provide us with the computation capabilities required to run the system at real-time rates. Thus our system scales well into large environments. We demonstrate the effectiveness of our system for high quality dense reconstruction and labelling of the scenes on the KITTI dataset.

- **Tiered Move Making Algorithm** In this work, we propose an iterative *tiered move making algorithm* which is able to handle general pairwise terms. Each move to the next configuration is based on the current labeling and an optimal tiered move, where each tiered move requires one application of the dynamic programming based tiered labeling method introduced in Felzenszwalb et. al. [43]. The algorithm converges to a local minimum

for any general pairwise potential, and we give a theoretical analysis of the properties of the algorithm, characterizing the situations in which we can expect good performance. We first evaluate our method on an object-class segmentation problem using the Pascal VOC-11 segmentation dataset where we learn general pairwise terms. Further we evaluate the algorithm on many other benchmark labeling problems such as stereo, image segmentation, image stitching and image denoising. Our method consistently gets better accuracy and energy values than α-expansion, loopy belief propagation (LBP), quadratic pseudo-boolean optimization (QPBO), and is competitive with TRWS.

9.2 Limitations

Despite very encouraging results, the works presented in the thesis is not without limitations which we discuss now.

- **Indoor Scene Reconstruction and Recognition** Currently the system uses a voice command to switch between training and test modes. We are planning an extension where both the learning and forest predictions are always turned on. This will require considerable care to avoid 'drift' in the learned category models: the feedback loop would mean that small errors could quickly get amplified. As with all recognition algorithms, the results are not always voxel-perfect. As we have demonstrated, allowing the user to interactively make corrections can help reduce such errors. We believe additional modes of interaction such as voice priors (e.g. 'walls are vertical'), as well as more intelligently sampling the training examples could further improve results. Algorithmic parameters such as the pairwise weights are currently set by hand. Given a small training set (perhaps boot-strapping), more reliable settings could be automatically selected.

- **Outdoor Scene Reconstruction and Recognition** As with all recognition algorithms, the segmentation results are not always voxel-perfect, as shown in the results and accompanying video. One possibility, however, is to allow the user to interactively make corrections to help reduce such errors. We believe additional modes of interaction such as voice priors (e.g. 'walls are vertical'), as well as more intelligently sampling the training examples could further improve results. From a computational standpoint, our system is fairly GPU heavy, which limits us to laptop only uses currently. With the advent of mobile GPGPU there are likely ways of addressing this in future work. Finally, algorithmic parameters such as the pairwise weights are

currently set at compile time (these are cross-validated and common across datasets shown). Given a small training set (perhaps boot-strapping), more reliable settings could be automatically selected online.

- **Joint Intrinsic Image-Object-Attributes Decomposition** In this chapter, we have assumed that we have a single global model of illumination, but natural scenes contain spatially-varying illumination due to attenuation, interreflection, cast and attached shadows. Such limitations could be solved by use of a mixture of illumination embedded in the soft segmentation of the scene. Further, the dual-decomposition optimisation solves the joint energy function only approximately. It would be good direction to design an algorithm that optimises the function globally. Finally such optimisation approach is too slow to be applied to solve any robotics related or interactive system where the results are expected at the real-time rate.

- **Higher Order Mean-field** There are two main limitations of our algorithms. First they are only able to handle P^N-Potts potentials, and secondly the parameters of the higher order models are set through cross-validation which limits it to be used for large scale problems including large number of classes.

9.3 Future Work

We now discuss some future directions which can extend the approaches proposed in this work to solve scene understanding problems.

Higher Order Mean-field

- We would like to investigate further ways to improve efficiency though parallelization, and learning techniques which can draw on high speed inference for joint parameter optimization in large-scale models.

- In this work, the higher order label consistency potential is restricted to P^N-Potts potentials. We have shown the importance of incorporating robust P^N-Potts potentials. Thus the next important extension of our work is to develop an efficient mean-field algorithm to incorporate such robust potentials. Currently we do not use any filtering approach to solve higher order potential. Thus another direction is to develop an algorithm based on a box-filtering approach to handle higher order potentials.

- As shown in the experiments, the fully-connected model helps to recover very fine object boundaries. This motivates us to look at solving object segmentation where objects are long, thin and wiry. In our day-to-day life we

encounter many such objects, such as pylons, wiry chairs, and table lamps. Current state-of-the-art algorithms generally smooth the object boundaries with the background. Thus, we would like to develop algorithms to perform very fine and efficient segmentation of wiry objects. The algorithms should be able to enforce topological constraints.

Joint Intrinsic Image-Object-Attribute Decomposition

- In outdoor environments shadows are common and are considered a nuisance for current vision and robotics systems. Intrinsic scene decomposition recovers an illumination independent image which will help in improving object recognition and other scene understanding tasks. Our intrinsic scene decomposition work will help in recovering such an illumination independent scene. In order for our algorithm to work in both indoor and outdoor settings, another important task required is to develop real-time version of it so that we can perform recognition on shadow independent images on the fly. Further, this synergy could be extended to handle specularity in the environments.

- Another extension of the work is to explore further synergy effects between the intrinsic scene properties and the objects and attributes present in the scene. Essentially we can incorporate the statistical prior on the shape/structure based on the objects and attributes. For example, knowledge about the structural properties - planar, vertical, boxy, rectangular and the objects table, airplane - can provide a lot of information about the structure of the scene, and thus can help in reducing the ambiguity present in the world and in better estimation of the shape and other intrinsic properties.

Indoor and Outdoor Scene Reconstruction and Recognition

- The first direction is to extend the current algorithm to handle multiple sensors. For example, it is desirable that our algorithms be able to use information for multiple cameras. In This will reduce both the computation and memory requirements. Further we would like to handle Kinect, stereo cameras, monocular cameras or lidar sensors.

- This will subsequently require us develop distributed learning and inference algorithms since the data is collected in a distributed fashion. For example, if two cameras gather information about an object from two different views, we would like to use the information from both these views to develop better models of the objects.

- Currently we have not incorporated any shape prior based on the object information to improve the quality of the reconstruction. Ideally we would like to develop an efficient algorithm that reconstructs the environments in real-time and incorporate object and attribute specific shape priors to improve the reconstruction. For example, we could use 3D models generated using CAD or downloaded from the Google 3D warehouse database to enforce a shape prior.

- Human interaction can play an important part in improving both the reconstruction and recognition of a scene. We have shown how interaction can help to improve object recognition for the indoor scenes. An extension of this would be to develop an interactive system for outdoor scenes to improve both the reconstruction and recognition tasks. To this end, one option would be use laser pointers to interact.

Bibliography

[1] M. Abdelrahman, M. Aono, M. El-Melegy, A. Farag, A. Ferreira, H. Johan, B. Li, Y. Lu, J. Machado, P. B. Pascoal, and A. Tatsuma. SHREC13: Retrieval of objects captured with low-cost depth-sensing cameras. In *Proc. Eurographics Workshop on 3D Object Recognition*, 2013.

[2] A. Adams, J. Baek, and M.A. Davis. Fast high-dimensional filtering using the permutohedral lattice. In *Computer Graphics Forum*, 2010.

[3] E. H. Adelson. Lightness perception and lightness illusions. In *The New Cognitive Neuroscience, 2nd Ed. MIT Press*, pages 339–351, 2000.

[4] E. H. Adelson. On seeing stuff: the perception of materials by humans and machines. In *SPIE*, pages 1–12, 2001.

[5] E. H. Adelson and A. P. Pentland. The perception of shading and reflectance. pages 409–423, 1996.

[6] S. Agarwal, Y. Furukawa, N. Snavely, I. Simon, B. Curless, S. M. Seitz, and R. Szeliski. Building rome in a day. *Commun. ACM*, 54(10):105–112, 2011.

[7] A. Anand, H. S. Koppula, T. Joachims, and A. Saxena. Contextually guided semantic labeling and search for three-dimensional point clouds. *The International Journal of Robotics Research*, 32(1):19–34, 2013.

[8] M. Andriluka, S. Roth, and B. Schiele. Pictorial structures revisited: People detection and articulated pose estimation. In *CVPR*, pages 1014–1021, 2009.

[9] J. Baek, A. Adams, and J. Dolson. Lattice-based high-dimensional gaussian filtering and the permutohedral lattice. volume 46, pages 211–237, 2013.

[10] J. T. Barron and J. Malik. Color constancy, intrinsic images, and shape estimation. In *ECCV*, pages 57–70, 2012.

[11] J. T. Barron and J. Malik. High-frequency shape and albedo from shading using natural image statistics. In *CVPR*, pages 2521–2528, 2012.

[12] J. T. Barron and J. Malik. Shape, albedo, and illumination from a single image of an unknown object. In *CVPR*, pages 334–341, 2012.

[13] J. T. Barron and J. Malik. Intrinsic scene properties from a single rgb-d image. In *CVPR*, 2013.

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

[15] H. Bay, A. Ess, T. Tuytelaars, and L. V. Gool. Surf: Speeded up robust features. In *Computer Vision and Image Understanding (CVIU)*, volume 110, pages 346–359, 2008.

[16] J. Besag. Spatial interaction and the statistical analysis of lattice systems. In *Journal of Royal Statistical Society*, 1974.

[17] A. Bifet, G. Holmes, B. Pfahringer, R. Kirkby, and R. Gavaldà. New ensemble methods for evolving data streams. In *Proc. SIGKDD*, 2009.

[18] C. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006.

[19] U. Bonde, V. Badrinarayanan, and R. Cipolla. Multi scale shape index for 3D object recognition. In *Scale Space and Variational Methods in Computer Vision*, pages 306–318. Springer, 2013.

[20] Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. *IEEE Trans. Pattern Anal. Mach. Intell.*, 23(11):1222–1239, 2001.

[21] M. Bray, P. Kohli, and P. Torr. Posecut: Simultaneous segmentation and 3d pose estimation of humans using dynamic graph-cuts. *ECCV*, pages 642–655, 2006.

[22] L. Breiman. Random forests. *Machine Learning*, 45(1), 2001.

[23] G. J Brostow, J. Shotton, J. Fauqueur, and R. Cipolla. Segmentation and recognition using structure from motion point clouds. In *Proc. ECCV*, 2008.

[24] R. O Castle, DJ Gawley, G. Klein, and D. W Murray. Towards simultaneous recognition, localization and mapping for hand-held and wearable cameras. In *Proc. ICRA*, 2007.

[25] A. Chambolle, V. Caselles, D. Cremers, M. Novaga, and T. Pock. An introduction to total variation for image analysis. In *Theoretical Foundations and Numerical Methods for Sparse Recovery*. De Gruyter, 2010.

[26] D. M. Chen, G. Baatz, K. Köser, S. S. Tsai, R. Vedantham, T. Pylvänäinen, K. Roimela, X. Chen, J. Bach, M. Pollefeys, B. Girod, and R. Grzeszczuk. City-scale landmark identification on mobile devices. In *CVPR*, pages 737–744, 2011.

[27] J. Chen, D. Bautembach, and S. Izadi. Scalable real-time volumetric surface reconstruction. *ACM TOG*, 32(4):113, 2013.

[28] X. Chen, A. Golovinskiy, and T. Funkhouser. A benchmark for 3D mesh segmentation. *ACM TOG*, 28(3):73, 2009.

[29] Y. Chen and G. G. Medioni. Object modelling by registration of multiple range images. *Image Vision Comput.*, 10(3):145–155, 1992.

[30] D. Comaniciu and P. Meer. Mean shift: A robust approach toward feature space analysis. In *IEEE PAMI*, 2002.

[31] C. Couprie, C. Farabet, L. Najman, and Y. LeCun. Indoor semantic segmentation using depth information. *arXiv preprint arXiv:1301.3572*, 2013.

[32] A. Criminisi and J. Shotton. Decision forests for computer vision and medical image analysis. *Springer*, 2013.

[33] A. Culotta. *Learning and inference in weighted logic with application to natural language processing.* 2008.

[34] B. Curless and M. Levoy. A volumetric method for building complex models from range images. pages 303–312, 1996.

[35] H. Dahlkamp, A. Kaehler, D. Stavens, S. Thrun, and G. R. Bradski. Self-supervised monocular road detection in desert terrain. 2006.

[36] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *Proc. CVPR*, 2005.

[37] P. Domingos and G. Hulten. Mining high-speed data streams. In *Proc. SIGKDD*, 2000.

[38] H. F. Durrant-Whyte and T. Bailey. Simultaneous localization and mapping: part 1. *IEEE Robot. Automat. Mag.*, 13(2):99–110, 2006.

[39] M. Everingham, L. J.V. Gool, C.K.I. Williams, and A. Zisserman. The pascal visual object classes (voc) challenge. In *IJCV*, pages 303–338, 2010.

[40] M. Everingham, L. Van Gool, C.K.I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge. In *VOC2011*, 2011.

[41] A. Farhadi, I. Endres, D. Hoiem, and D.A. Forsyth. Describing objects by their attributes. In *CVPR*, 2009.

[42] P. F. Felzenswalb and D. P. Huttenlocker. Efficient graph-based image segmentation. In *IJCV*, 2004.

[43] P. F. Felzenszwalb and O. Veksler. Tiered scene labeling with dynamic programming. In *CVPR*, pages 3097–3104, 2010.

[44] V. Ferrari, M. Marin-Jimenez, and A. Zisserman. Progressive search space reduction for human pose estimation. In *CVPR*, pages 1–8, 2008.

[45] N. Fioraio and L. Di Stefano. Joint detection, tracking and mapping by semantic bundle adjustment. In *Proc. CVPR*, 2013.

[46] E.S.SL. Gastla and M.M. Oliveira. Domain transform for edge-aware image and video processing. In *ACM Trans. Graph*, 2011.

[47] P. V. Gehler, C. Rother, M. Kiefel, L. Zhang, and S. Bernard. Recovering intrinsic images with a global sparsity prior on reflectance. In *NIPS*, pages 765–773, 2011.

[48] A. Geiger, P. Lenz, and R. Urtasun. Are we ready for autonomous driving? The KITTI vision benchmark suite. In *Proc. CVPR*, 2012.

[49] A. Geiger, P. Lenz, and R. Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. pages 3354–3361, 2012.

[50] A. Geiger, J. Ziegler, and C. Stiller. Stereoscan: Dense 3d reconstruction in real-time. pages 963–968, 2011.

[51] R. B. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. *CoRR*, abs/1311.2524, 2013.

[52] G. Gkioxari, B. Hariharan, R. B. Girshick, and J. Malik. R-CNNs for pose estimation and action detection. *CoRR*, abs/1406.5212, 2014.

[53] J.M. Gonfaus, X. Boix, J. Van De Weijer, A.D. Bagdanov, J. Serrat, and J. Gonzalez. Harmony potentials for joint classification and segmentation. In *IEEE CVPR*, pages 1–8, 2010.

[54] S. Gould, R. Fulton, and D. Koller. Decomposing a scene into geometric and semantically consistent regions. In *ICCV*, pages 1–8, 2009.

[55] A. Gupta, A. A Efros, and M. Hebert. Blocks world revisited: Image understanding using qualitative geometry and mechanics. In *Proc. ECCV*. 2010.

[56] S. Gupta, P. Arbelaez, and J. Malik. Perceptual organization and recognition of indoor scenes from rgb-d images. In *CVPR*, pages 564–571, 2013.

[57] C. Häne, C. Zach, A. Cohen, R. Angst, and M. Pollefeys. Joint 3D scene reconstruction and class segmentation. In *Proc. CVPR*, 2013.

[58] H. Hirschmüller. Accurate and efficient stereo processing by semi-global matching and mutual information. In *CVPR (2)*, pages 807–814, 2005.

[59] H Hirschmuller. Stereo processing by semiglobal matching and mutual information. *Trans. PAMI*, 30(2):328–341, 2008.

[60] B.K.P. Horn. Shape from shading: a method for obtaining the shape of a smooth opaque object from one view. In *Technical Report, MIT*.

[61] C.Q. Howe and D. Purves. *Perceiving Geometry*. Springer, 2005.

[62] Y. Ioanou, B. Taati, R. Harrap, and M. Greenspan. Difference of normals as a multi-scale operator in unorganized point clouds. In *Proc. 3DIMPVT*, 2012.

[63] S. Izadi, D. Kim, O. Hilliges, D. Molyneaux, R. A. Newcombe, P. Kohli, J. Shotton, S. Hodges, D. Freeman, A. J. Davison, and A. W. Fitzgibbon. KinectFusion: real-time 3D reconstruction and interaction using a moving depth camera. In *UIST*, pages 559–568, 2011.

[64] T. S. Jaakkola and M. I. Jordan. Improving the mean field approximation via the use of mixture distributions. In *Learning in Graphical Models*.

[65] A. Johnson. *Spin-Images: A Representation for 3-D Surface Matching*. PhD thesis, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, August 1997.

[66] A. Johnson. Spin-images: A representation for 3-d surface matching. 1997.

[67] M. Kaess, H. Johannsson, R. Roberts, V. Ila, J. J. Leonard, and F. Dellaert. isam2: Incremental smoothing and mapping using the bayes tree. *I. J. Robotic Res.*, 31(2):216–235, 2012.

[68] O. Kähler and I. Reid. Efficient 3D scene labeling using fields of trees. In *Proc. ICCV*, 2013.

[69] E. Kalogerakis, A. Hertzmann, and K. Singh. Learning 3D mesh segmentation and labeling. *ACM TOG*, 29(4):102, 2010.

[70] J. H. Kappes, B. Andres, F. A. Hamprecht, C. Schnörr, S. Nowozin, D. Batra, S. Kim, B. X. Kausler, J. Lellmann, N. Komodakis, and C. Rother. A comparative study of modern inference techniques for discrete energy minimization problems. In *CVPR*, pages 1328–1335, 2013.

[71] A. Karpathy, S. Miller, and L. Fei-Fei. Object discovery in 3D scenes via shape analysis. In *Proc. ICRA*, 2013.

[72] A. Khosla, T. Zhou, T. Malisiewicz, A. A. Efros, and A. Torralba. Undoing the damage of dataset bias. In *ECCV (1)*, pages 158–171, 2012.

[73] B. Kim, P. Kohli, and S. Savarese. 3D scene understanding by voxel-CRF. In *Proc. ICCV*, 2013.

[74] V. G Kim, W. Li, N. J Mitra, S. Chaudhuri, S. DiVerdi, and T. Funkhouser. Learning part-based templates from large collections of 3D shapes. *ACM TOG*, 2013.

[75] Y. M. Kim, N. J Mitra, D. Yan, and L. Guibas. Acquiring 3D indoor environments with variability and repetition. *ACM TOG*, 31(6):138, 2012.

[76] G. Klein and D. W. Murray. Parallel tracking and mapping for small ar workspaces. In *ISMAR*, pages 225–234, 2007.

[77] P. Kohli, M. P. Kumar, and P. H. S. Torr. P3 & beyond: Move making algorithms for solving higher order functions. In *IEEE Trans. Pattern Analysis and Machine Intelligence*, pages 1645–1656, 2009.

[78] P. Kohli, M.P. Kumar, and P.H.S. Torr. P3 & beyond: Solving energies with higher order cliques. In *CVPR*, pages 1–8, 2007.

[79] P. Kohli, L. Ladicky, and P. H. S. Torr. Robust higher order potentials for enforcing label consistency. *International Journal of Computer Vision*, 82(3):302–324, 2009.

[80] P. Kohli and C. Rother. Higher-order models in computer vision. 2012.

[81] D. Koller and N. Friedman. *Probabilistic Graphical Models - Principles and Techniques*. MIT Press, 2009.

[82] V. Kolmogorov. Convergent tree-reweighted message passing for energy minimization. *IEEE Trans. Pattern Anal. Mach. Intell.*, 28(10):1568–1583, 2006.

[83] N. Komodakis and N. Paragios. Beyond pairwise energies: Efficient optimization for higher-order mrfs. In *IEEE CVPR*, pages 2985–2992, 2009.

[84] N. Komodakis, N. Paragios, and G. Tziritas. MRF energy minimization and beyond via dual decomposition. *IEEE Trans. Pattern Anal. Mach. Intell.*, 33(3):531–552, 2011.

[85] H. S Koppula, A. Anand, T. Joachims, and A. Saxena. Semantic labeling of 3D point clouds for indoor scenes. In *Proc. NIPS*, 2011.

[86] P. Kornprobst, J. Tumblin, and F. Durand. Bilateral filtering: Theory and applications. In *Foundations and Trends in Computer Graphics and Vision*, pages 1–74, 2009.

[87] P. Krähenbühl and V. Koltun. Efficient inference in fully connected crfs with gaussian edge potentials. In *NIPS*, pages 109–117, 2011.

[88] A. Krizhevsky, I. Sutskever, and G.E. Hinton. ImageNet classification with deep convolutional neural networks. In *Proc. NIPS*, 2012.

[89] K. A. Krueger and P. Dayan. Flexible shaping: how learning in small steps helps. volume 110, pages 380–394, 2009.

[90] M. P. Kumar, V. Kolmogorov, and P. H. S. Torr. An analysis of convex relaxations for map estimation of discrete mrfs. *Journal of Machine Learning Research*, 10:71–106, 2009.

[91] M.P. Kumar, P.H.S. Torr, and A. Zisserman. Objcut: Efficient segmentation using top-down and bottom-up cues. *PAMI*, 32(3):530–545, 2010.

[92] M.P. Kumar, O. Veksler, and P.H.S. Torr. Improved moves for truncated convex models. In *JMLR*, 2011.

[93] M.P. Kumar, A. Zisserman, and P.H.S. Torr. Efficient discriminative learning of parts-based models. In *CVPR*, pages 552–559, 2009.

[94] L. Ladicky, C. Russell, P. Kohli, and P. H. S. Torr. Associative hierarchical CRFs for object class image segmentation. In *ICCV*, pages 739–746, 2009.

[95] L. Ladickỳ, C. Russell, P. Kohli, and P.H.S. Torr. Associative hierarchical CRFs for object class image segmentation. In *ICCV*, pages 739–746, 2009.

[96] L. Ladickỳ, C. Russell, P. Kohli, and P.H.S. Torr. Graph cut based inference with co-occurrence statistics. In *ECCV*, pages 239–253, 2010.

[97] L. Ladickỳ, P. Sturgess, C. Russell, S. Sengupta, Y. Bastanlar, W. Clocksin, and P. HS Torr. Joint optimization for object class segmentation and dense stereo reconstruction. *International Journal of Computer Vision*, 100(2):122–133, 2012.

[98] L. Ladickỳ, P. Sturgess, C. Russell, S. Sengupta, Y. Bastanlar, W.F. Clocksin, and P.H.S. Torr. Joint optimisation for object class segmentation and dense stereo reconstruction. In *BMVC*, pages 1–11, 2010.

[99] J. D. Lafferty, A. McCallum, and F. C. N. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *ICML*, pages 282–289, 2001.

[100] E.H. Land and J.J. McCann. *Lightness and retinex theory.* JOSA, 1971.

[101] À. Lapedriza, H. Pirsiavash, Z. Bylinskii, and A. Torralba. Are all training examples equally valuable? *CoRR*, abs/1311.6510, 2013.

[102] Y. LeCun, F. J. Huang, and L. Bottou. Learning methods for generic object recognition with invariance to pose and lighting. In *CVPR*, pages 97–104, 2004.

[103] V. Lepetit and P. Fua. Keypoint recognition using randomized trees. *IEEE Trans. PAMI*, 2006.

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

[105] M. Levoy, K. Pulli, B. Curless, S. Rusinkiewicz, D. Koller, L. Pereira, M. Ginzton, S. Anderson, J. Davis, and J. Ginsberg. The digital Michelangelo project: 3D scanning of large statues. In *Proc. SIGGRAPH*. ACM, 2000.

[106] Point Cloud Library. http://pointclouds.org/.

[107] D. Lin, S. Fidler, and R. Urtasun. Holistic scene understanding for 3D object detection with RGBD cameras. In *Proc. ICCV*, 2013.

[108] H. Lin, J. Gao, Y. Zhou, G. Lu, M. Ye, C. Zhang, L. Liu, and R. Yang. Semantic decomposition and reconstruction of residential scenes from LiDAR data. *ACM TOG*, 32(4), 2013.

[109] C. Liu, L. Sharan, E.H. Adelson, and R. Rosenholtz. Exploring features in a bayesian framework for material recognition. In *CVPR*, pages 239–246, 2010.

[110] C. Liu, J. Yuen, and A. Torralba. Nonparametric scene parsing: Label transfer via dense scene alignment. In *CVPR*, 2009.

[111] D. G Lowe. Object recognition from local scale-invariant features. In *Proc. ICCV*, 1999.

[112] M. R. Luettgen, W. Clement Karl, A. S. Willsky, and R. R. Tenney. Multiscale representations of Markov random fields. *IEEE Transactions on Signal Processing*, 41(12):3377–3396, 1993.

[113] J. Malik. The 3 R's of computer vision: Recognition, reconstruction and reorganization. In *The Stanford Center for Image Systems Engineering*, 2013.

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

[115] C. Mei, G. Sibley, M. Cummins, P. Newman, and I. Reid. Rslam: A system for large-scale mapping in constant-time using stereo. *International Journal of Computer Vision*, pages 1 – 17, June 2010.

[116] P. Merrell, E. Schkufza, Z. Li, M. Agrawala, and V. Koltun. Interactive furniture layout using interior design guidelines. *ACM TOG*, 2011.

[117] K. Mikolajczyk and C. Schmid. A performance evaluation of local descriptors. *IEEE Trans. Pattern Anal. Mach. Intell.*, 27(10):1615–1630, 2005.

[118] L. Nan, K. Xie, and A. Sharf. A search-classify approach for cluttered indoor scene understanding. *ACM TOG*, 31(6):137, 2012.

[119] R. A Newcombe, S. Izadi, O. Hilliges, D. Molyneaux, D. Kim, A. J Davison, P. Kohli, J. Shotton, S. Hodges, and A. Fitzgibbon. KinectFusion: Real-time dense surface mapping and tracking. In *Proc. ISMAR*, 2011.

[120] R. A. Newcombe, S. Lovegrove, and A. J. Davison. DTAM: Dense tracking and mapping in real-time. In *ICCV*, pages 2320–2327, 2011.

[121] R. A Newcombe, S. J Lovegrove, and A. J Davison. DTAM: Dense tracking and mapping in real-time. In *Proc. ICCV*, 2011.

[122] M. Nießner, M. Zollhöfer, S. Izadi, and M. Stamminger. Real-time 3D reconstruction at scale using voxel hashing. *ACM Trans. Graph.*, 32(6):169, 2013.

[123] A. Oliva and A. Torralba. Modeling the shape of the scene: A holistic representation of the spatial envelope. In *IJCV*, pages 145–175, 2001.

[124] M. Osadchy, D. W. Jacobs, and R. Ramamoorthi. Using specularities for recognition. In *ICCV*, 2003.

[125] D. Parikh, A. Kovashka, A. Parkash, and K. Grauman. Relative attributes for enhanced human-machine communication. In *AAAI*, 2012.

[126] G. B. Peterson. A day of great illumination: B. F. Skinners discovery of shaping. 2004.

[127] M. Pollefeys, D. Nistér, J.M. Frahm, A. Akbarzadeh, P. Mordohai, B. Clipp, C. Engels, D. Gallup, S.J. Kim, P. Merrell, et al. Detailed real-time urban 3D reconstruction from video. *IJCV*, 78(2):143–167, 2008.

[128] J. Ponce, T. L. Berg, M. Everingham, D. A. Forsyth, M. Hebert, S. Lazebnik, M. Marszalek, C. Schmid, B. C. Russell, A. Torralba, C. K. I. Williams, J. Zhang, and A. Zisserman. Dataset issues in object recognition. In *Toward Category-Level Object Recognition*, pages 29–48, 2006.

[129] I. Posner, M. Cummins, and P. Newman. Fast probabilistic labeling of city maps. In *Robotics: Science and Systems*, 2008.

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

[131] B. Potetz and T.S. Lee. Efficient belief propagation for higher-order cliques using linear constraint nodes. In *CVIU*, pages 39–54, 2008.

[132] V. Pradeep, C. Rhemann, S. Izadi, Ch. Zach, M. Bleyer, and S. Bathiche. Monofusion: Real-time 3D reconstruction of small scenes with a single web camera. In *Proc. ISMAR*, 2013.

[133] D. Purves and R.B. Lotto. Why we see what we do: An empirical theory of vision. 2003.

[134] F. Ramos, J. Nieto, and H. Durrant-Whyte. Combining object recognition and SLAM for extended map representations. In *Experimental Robotics*, pages 55–64. Springer, 2008.

[135] X. Ren, L. Bo, and D. Fox. RGB-(D) scene labeling: Features and algorithms. In *Proc. CVPR*, 2012.

[136] C. Rhemann, A. Hosni, M. Bleyer, C. Rother, and M. Gelautz. Fast cost-volume filtering for visual correspondence and beyond. In *CVPR*, pages 3017–3024, 2011.

[137] Lawrence G Roberts. *Machine perception of three-dimensional solids*. PhD thesis, Massachusetts Institute of Technology, 1963.

[138] C. Rother, P. Kohli, W. Feng, and J. Jia. Minimizing sparse higher order energy functions of discrete variables. In *CVPR*, pages 1382–1389, 2009.

[139] C. Rother, V. Kolmogorov, and A. Blake. GrabCut - Interactive foreground extraction using iterated graph cuts. *ACM Transactions on Graphics (TOG)*, 23(3), 2004.

[140] C. Rother, V. Kolmogorov, and A. Blake. GrabCut: Interactive foreground extraction using iterated graph cuts. In *ACM TOG*, pages 309–314, 2004.

[141] H. A. Rowley, S. Baluja, and T. Kanade. Neural network-based face detection. *IEEE Trans. Pattern Anal. Mach. Intell.*, 20(1):23–38, 1998.

[142] S. Rusinkiewicz, O. Hall-Holt, and M. Levoy. Real-time 3D model acquisition. *ACM TOG*, 21(3):438–446, 2002.

[143] A. Saffari, C. Leistner, J. Santner, M. Godec, and H. Bischof. On-line random forests. In *3rd IEEE ICCV Workshop on On-line Learning for Computer Vision*, 2009.

[144] R. F Salas-Moreno, R. A Newcombe, H. Strasdat, P. HJ Kelly, and A. J Davison. SLAM++: Simultaneous localisation and mapping at the level of objects. In *Proc. CVPR*, 2013.

[145] S. Sengupta, E. Greveson, A. Shahrokni, and P. H. S. Torr. Urban 3D semantic modelling using stereo vision. In *ICRA*, pages 580–585, 2013.

[146] S. Sengupta, P. Sturgess, L. Ladicky, and P. H. S. Torr. Automatic dense visual semantic mapping from street-level imagery. In *IROS*, pages 857–862, 2012.

[147] Q. Shan, R. Adams, B. Curless, Y. Furukawa, and S. M Seitz. The visual turing test for scene reconstruction. In *Proc. 3DTV*, 2013.

[148] T. Shao, W. Xu, K. Zhou, J. Wang, D. Li, and B. Guo. An interactive approach to semantic modeling of indoor scenes with an RGBD camera. *ACM TOG*, 31(6):136, 2012.

[149] L. Shapira, S. Shalom, A. Shamir, D. Cohen-Or, and H. Zhang. Contextual part analogies in 3D objects. *International Journal of Computer Vision*, 89(2-3):309–326, 2010.

[150] T. Sharp. Implementing decision trees and forests on a gpu. In *Computer Vision–ECCV 2008*, pages 595–608. Springer, 2008.

[151] G. Sheasby, J. Valentin, N. Crook, and P.H.S. Torr. A robust stereo prior for human segmentation. *ACCV*, 2012.

[152] G. Sheasby, J. Warrell, Y. Zhang, N. Crook, and P.H.S. Torr. Simultaneous human segmentation, depth and pose estimation via dual decomposition. *BMVW*, 2012.

[153] C. Shen, H. Fu, K. Chen, and S. Hu. Structure recovery by part assembly. *ACM TOG*, 31(6):180, 2012.

[154] J. Shotton, A. W. Fitzgibbon, M. Cook, T. Sharp, M. Finocchio, R. Moore, A. Kipman, and A. Blake. Real-time human pose recognition in parts from single depth images. In *CVPR*, pages 1297–1304, 2011.

[155] J. Shotton, A. W. Fitzgibbon, M. Cook, T. Sharp, M. Finocchio, R. Moore, A. Kipman, and A. Blake. Real-time human pose recognition in parts from single depth images. In *CVPR*, pages 1297–1304, 2011.

[156] J. Shotton, J. M. Winn, C. Rother, and A. Criminisi. Textonboost: Joint appearance, shape and context modeling for multi-class object recognition and segmentation. In *ECCV (1)*, pages 1–15, 2006.

[157] J. Shotton, J. M. Winn, C. Rother, and A. Criminisi. Textonboost for image understanding: Multi-class object recognition and segmentation by jointly modeling texture, layout, and context. *International Journal of Computer Vision*, 81(1):2–23, 2009.

[158] J. Shotton, J. M. Winn, C. Rother, and A. Criminisi. Textonboost for image understanding: Multi-class object recognition and segmentation by jointly modeling texture, layout, and context. In *IJCV*, 2009.

[159] L. Sigal and M.J. Black. Humaneva: Synchronized video and motion capture dataset for evaluation of articulated human motion. *Brown Univertsity TR*, 120, 2006.

[160] N. Silberman and R. Fergus. Indoor scene segmentation using a structured light sensor. In *Proc. ICCV Workshop on 3D Representation and Recognition*, 2011.

[161] N. Silberman, D. Hoiem, P. Kohli, and R. Fergus. Indoor segmentation and support inference from rgbd images. In *ECCV*, pages 746–760, 2012.

[162] B. F. Skinner. Reinforcement today. 1958.

[163] P. P. Sloan, J. Kautz, , and J. Snyder. Precomputed radiance transfer for real-time rendering in dynamic, low-frequency lighting environments. In *SIGGRAPH*, pages 527–536, 2002.

[164] N. Snavely, S. M Seitz, and R. Szeliski. Photo tourism: exploring photo collections in 3D. *ACM TOG*, 25(3), 2006.

[165] J. Stückler, B. Waldvogel, H. Schulz, and S. Behnke. Dense real-time mapping of object-class semantics from RGB-D video. *Journal of Real-Time Image Processing*, pages 1–11, 2013.

[166] J. Stühmer, S. Gumhold, and D. Cremers. Parallel generalized thresholding scheme for live dense geometry from a handheld camera. In *ECCV Workshops (1)*, pages 450–462, 2010.

[167] D. Sun, J. Wulff, E. B. Sudderth, H. Pfister, and M. J. Black. A fully-connected layered model of foreground and background flow. In *CVPR*, pages 2451–2458, 2013.

[168] M. Sun, P. Kohli, and J. Shotton. Conditional regression forests for human pose estimation. In *CVPR*, pages 3394–3401. IEEE, 2012.

[169] C. A. Sutton and A. McCallum. Piecewise training for undirected models. In *UAI*, 2005.

[170] 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 with smoothness-based priors. In *IEEE Trans. Pattern Analysis and Machine Intelligence*, pages 1068–1080, 2008.

[171] A. Taneja, L. Ballan, and M. Pollefeys. City-scale change detection in cadastral 3d models using images. In *CVPR*, pages 113–120, 2013.

[172] J. Tighe and S. Lazebnik. Understanding scenes on many levels. In *ICCV*, pages 335–342, 2011.

[173] A. Torralba and A. A. Efros. Unbiased look at dataset bias. In *CVPR*, pages 1521–1528, 2011.

[174] A. Torralba, K. P. Murphy, and W. T. Freeman. Contextual models for object detection using boosted random fields. In *NIPS*, 2004.

[175] A. Torralba, K.P. Murphy, and W.T. Freeman. Sharing visual features for multiclass and multiview object detection. In *IEEE PAMI*, 2007.

[176] B. Triggs, P. F. McLauchlan, R. I. Hartley, and A. W. Fitzgibbon. Bundle adjustment - a modern synthesis. In *Workshop on Vision Algorithms*, pages 298–372, 1999.

[177] I. Tsochantaridis, T. Hofmann, T. Joachims, and Y. Altun. Support vector machine learning for interdependent and structured output spaces. In *ICML*, 2004.

[178] J. PC Valentin, S. Sengupta, J. Warrell, A. Shahrokni, and P. HS Torr. Mesh based semantic modelling for indoor and outdoor scenes. In *Proc. CVPR*, 2013.

[179] V. Vineet and PJ Narayanan. Cuda cuts: Fast graph cuts on the gpu. In *Computer Vision and Pattern Recognition Workshops, 2008. CVPRW'08. IEEE Computer Society Conference on*, pages 1–8. IEEE, 2008.

[180] V. Vineet, J. Warrell, and P. H. S. Torr. Filter-based mean-field inference for random fields with higher-order terms and product label-spaces. In *ECCV (5)*, pages 31–44, 2012.

[181] V. Vineet, J. Warrell, and P.H.S. Torr. Filter-based mean-field inference for random fields with higher-order terms and product label-spaces. ECCV, 2012.

[182] J. S. Vitter. Random sampling with a reservoir. *ACM Trans. Math. Software*, 11(1), 1985.

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

[184] H. Wang and D. Koller. Multi-level inference by relaxed dual decomposition for human pose segmentation. In *CVPR*, pages 2433–2440, 2011.

[185] J. V. D. Weijer, C. Schmid, and J. Verbeek. Using high-level visual information for color constancy. In *ICCV*, pages 1–8, 2007.

[186] Y. Weiss. Comparing the mean field method and belief propagation for approximate inference in mrfs. In *Advanced Mean Field Methods: Theory and Practices*, 2001.

[187] J. Winn and J. Shotton. The layout consistent random field for recognizing and segmenting partially occluded objects (pdf). 2006.

[188] O. Woodford, P.H.S. Torr, I. Reid, and A. Fitzgibbon. Global stereo reconstruction under second-order smoothness priors. In *IEEE PAMI*, pages 239–253, 2009.

[189] O. J. Woodford, P. H. S. Torr, I. D. Reid, and A. W. Fitzgibbon. Global stereo reconstruction under second order smoothness priors. In *CVPR*, 2008.

[190] K. M. Wurm, R. Kümmerle, C. Stachniss, and W. Burgard. Improving robot navigation in structured outdoor environments by identifying vegetation from laser data. In *IROS*, pages 1217–1222, 2009.

[191] J. Xiao. *A 2D + 3D Rich Data Approach to Scene Understanding*. PhD thesis, Massachusetts Institute of Technology, 2014.

[192] J. Xiao, J. Hays, K. A Ehinger, A. Oliva, and A. Torralba. SUN database: Large-scale scene recognition from abbey to zoo. In *Proc. CVPR*, 2010.

[193] J. Xiao, A. Owens, and A. Torralba. Sun3d: A database of big spaces reconstructed using sfm and object labels. In *ICCV*, pages 1625–1632, 2013.

[194] Y. Yang and D. Ramanan. Articulated pose estimation with flexible mixtures-of-parts. In *CVPR*, pages 1385–1392, 2011.

[195] J. S. Yedidia, W. T. Freeman, and Y. Weiss. Generalized belief propagation. In *NIPS*, pages 689–695, 2000.

[196] J. S. Yedidia, W. T. Freeman, and Y. Weiss. Generalized belief propagation. In *NIPS*, pages 689–695, 2000.

[197] A. Yuille. Belief propagation, mean-field, and bethe approximation. In *Saad and Opper (ed) Advanced Mean Field Methods, MIT Press*, 2001.

[198] Q. Zhou, S. Miller, and V. Koltun. Elastic fragments for dense scene reconstruction. In *Proc. ICCV*, 2013.