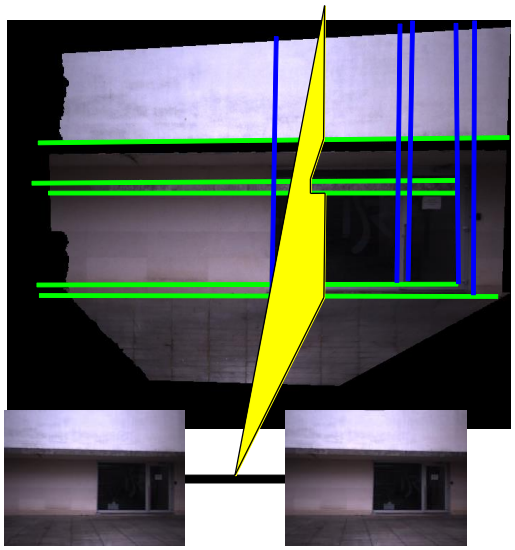


Michel Antunes

Stereo Reconstruction using Induced Symmetry and 3D scene priors

Tese de doutoramento em Engenharia Electrotécnica e de Computadores, orientada por João Pedro Barreto e apresentada no Departamento de Engenharia Electrotécnica e de Computadores da Universidade de Coimbra

2013

· U C ·

FCTUC

University of Coimbra
Faculty of Sciences and Technology
Department of Electrical and Computer Engineering

Stereo Reconstruction using Induced Symmetry and 3D scene priors

Michel Antunes

PhD Thesis

Coimbra, 2013

Stereo Reconstruction using Induced Symmetry and 3D scene priors

By
Michel Antunes

Advisor:
Professor Doutor João Pedro de Almeida Barreto

PhD Thesis

Department of Electrical and Computer Engineering
Faculty of Sciences and Technology, University of Coimbra

September 2013

Contents

1 Introduction **1**
- 1.1 Motivation . . . 1
- 1.2 Organization . . . 4
- 1.3 Contributions . . . 6

2 Stereo Matching using Induced Symmetry: a geometric account **8**
- 2.1 Introduction . . . 8
  - 2.1.1 Notation and Terminology . . . 9
- 2.2 Mirroring effect and Stereo from Induced Symmetry . . . 10
- 2.3 Geometric Analysis . . . 13
  - 2.3.1 Condition for a virtual plane $\mathbf{\Pi}$ to intersect the baseline . . . 13
  - 2.3.2 Proof of the Mirroring Effect . . . 15
  - 2.3.3 Singular Configuration . . . 16
- 2.4 Perfect symmetry in the presence of surface slant . . . 17
- 2.5 Mapping $\mathbf{\Pi}$ into a plane $\mathbf{\Gamma}$ in the DSI domain . . . 19
- 2.6 Sweeping the scene by vertical virtual planes . . . 20
- 2.7 Generating the symmetry/anti-symmetry images in 3D . . . 22
- 2.8 Conclusions . . . 23

3 SymStereo for dense matching and Stereo-Rangefinding **24**
- 3.1 Introduction . . . 24
- 3.2 Measuring local symmetry and anti-symmetry . . . 26
  - 3.2.1 SymBT . . . 27
  - 3.2.2 SymCen . . . 30
  - 3.2.3 logN . . . 31
- 3.3 Experiments in dense stereo matching . . . 36
  - 3.3.1 Methodology and tuning of parameters . . . 36
  - 3.3.2 Tests in Middlebury . . . 40
  - 3.3.3 Tests in Oxford Corridor . . . 42
  - 3.3.4 Experiments in wide-baseline stereo . . . 44

3.4 Experiments in Stereo-Rangefinding (SRF) 44
3.4.1 Methodology and tuning of parameters 45
3.4.2 Tests in Middlebury 46
3.4.3 Experiments in wide-baseline stereo 48
3.5 Stereo-Rangefinding vs. Laser-Rangefinding 49
3.5.1 Experimental Setup 50
3.5.2 Detection of the profile cut 50
3.5.3 Experimental results 53
3.6 Conclusions 55

4 Vanishing points and mutually orthogonal vanishing directions 57
4.1 Introduction 57
4.2 The Facility Location Problem 60
4.2.1 Uncapacited Facility Location (UFL) 60
4.2.2 Hierarchical Facility Location (HFL) 61
4.2.3 Solving UFL and HFL using the max-sum algorithm . . . 62
4.3 Algorithm for vanishing point (VP) detection 63
4.3.1 Vanishing point detection as a UFL problem 63
4.3.2 The consistency function $D(\mathsf{e}, \mathbf{v})$ 64
4.3.3 The function $W(\mathcal{S})$ for updating the VP estimate 65
4.4 Detection of multiple orthogonal triplets 66
4.5 Experiments with synthetic data 68
4.6 Experiments in real images 69
4.6.1 YUD using the supplied lines 69
4.6.2 YUD using extracted edges 71
4.6.3 Scenes containing multiple orthogonal triplets 72
4.7 Conclusions 73

5 Piecewise Planar Reconstruction using two views 74
5.1 Introduction 74
5.1.1 Planarity prior for SymStereo 76
5.2 Related Work 77
5.3 Background 78
5.3.1 Energy-based multi-model fitting using PEARL 78
5.3.2 MRF for Plane Labeling 79
5.4 Reconstruction of lines along a single cut plane 80
5.4.1 Line cut detection using Hough and PEARL 81
5.4.2 Experiments in line cut detection 82
5.5 PPR using SymStereo and PEARL 84
5.5.1 Formulation of the global framework 84
5.5.2 Initial plane hypotheses 86
5.5.3 Data and smoothness term 86
5.5.4 Plane refinement 88

5.5.5 Plane refinement after PEARL . . . 88
5.5.6 Results in semi-dense PPR . . . 89
5.5.7 Independent line cut reconstruction vs. semi-dense PPR . 91
5.6 Experiments in Piecewise Planar Reconstruction (PPR) . . . 91
5.6.1 Compared Algorithms . . . 92
5.6.2 Accuracy analysis and parameter tuning . . . 92
5.6.3 Comparison results . . . 93
5.6.4 Two view piecewise planar models . . . 94
5.7 Conclusion . . . 95

6 Stereo Matching using Multiple Slant Hypotheses 99
6.1 Introduction . . . 99
6.2 Related work . . . 101
6.3 Local stereo using Histogram Aggregation (HA) . . . 102
6.3.1 Why is disparity selection useful? . . . 103
6.4 Aggregation with different window orientations $\boldsymbol{\alpha}$ . . . 103
6.4.1 Mapping slants into support window orientations . . . 104
6.4.2 Visibility limits for the orientation $\boldsymbol{\alpha}$ . . . 105
6.4.3 Discretization of the aggregation window . . . 106
6.5 HA with multiple slant hypotheses . . . 107
6.5.1 Cost aggregation in the $(\mathbf{p}, d, \boldsymbol{\alpha})$ domain . . . 108
6.5.2 Sampling the space of the aggregation orientations $\boldsymbol{\alpha}$ . . . 108
6.5.3 Standard aggregation vs. HA . . . 109
6.6 Experimental Results . . . 110
6.6.1 Comparison of different aggregation configurations . . . 111
6.6.2 Evaluation in Middlebury . . . 112
6.7 Conclusions . . . 114

7 Conclusions 115

A Additional Results for Chapter 4Vanishing points and mutually orthogonal vanishing directionschapter.4 117
A.1 Results on YUD using detected edges . . . 117

List of Figures

1.1 Applications using stereo vision 2
1.2 Problematic situations to stereo matching 3

2.1 Plane Sweeping vs SymStereo 9
2.2 Stereo matching costs based on photo-similarity 11
2.3 The SymStereo framework 12
2.4 Geometric analysis of SymStereo 14
2.5 Singular configuration of SymStereo 17
2.6 Refinement using slant prior 18
2.7 Back-projection onto a virtual cut plane 22

3.1 SymBT 26
3.2 SymCen 29
3.3 logN 32
3.4 Efficient implementation of *logN* 33
3.5 (Qualitative) space-frequency behavior of log-Gabor wavelets . . 35
3.6 Tuning the number of wavelets scales N for dense stereo 38
3.7 Results for the standard Middlebury benchmark 38
3.8 Short-baseline stereo dataset 39
3.9 Average percentage of disparity errors 39
3.10 Average percentage of disparity errors in semi-dense disparity maps 39
3.11 Normalized number of disparity errors 39
3.12 Overlay of the disparity errors 40
3.13 Percentage of disparity errors in the *Oxford Corridor* 43
3.14 Disparity maps obtained on the *Oxford Corridor* 43
3.15 Mean errors on the fountain-P11 dataset (dense stereo) 44
3.16 Tuning of parameters for SRF 45
3.17 Benchmark of the cost functions for Stereo-Rangefinding (SRF) . 45
3.18 Pros and cons of logN 46
3.19 Mean errors on the fountain-P11 dataset [1] 48
3.20 Experimental setup 50
3.21 Example of the estimation of the profile cut location 51
3.22 Image matches from SRF and LRF 54

3.23 Multi-cut example. 55
3.24 Qualitative comparison between SRF and LRF 56

4.1 Two images of man-made environments. 58
4.2 The UFL problem 60
4.3 The HFL problem 61
4.4 Consistency function and VP estimator 64
4.5 Clustering of line pencils in synthetic data 68
4.6 Accuracy of the estimation of VPs given a pencil of lines 69
4.7 Comparison between our UFL approach with Tardif [2] 70
4.8 Cumulative consistency error 70
4.9 Cumulative consistency error - edges are automatically extracted . 71
4.10 Two cases from the YUD. 71
4.11 Scenes containing multiple orthogonal triplets - digital camera . . 72
4.12 Scenes containing multiple orthogonal triplets - Flickr image . . . 73

5.1 Line prior for SymStereo 76
5.2 Reconstruction of 3D line cuts using SymStereo 80
5.3 Results produced by our line cut detection algorithm 83
5.4 Results for two different parameter settings 83
5.5 Pipeline for PPR 85
5.6 The cyclopean eye 86
5.7 Crease edges and line segment clustering 87
5.8 Results produced by our semi-dense piecewise planar algorithm . 90
5.9 Independent line cut reconstruction and semi-dense PPR 91
5.10 Comparison between DS, SymS and SS for PPR. 96
5.11 Indoor results produced by our PPR algorithm. 97
5.12 Outdoor results produced by our PPR algorithm. 98

6.1 Aggregation in slanted surfaces 100
6.2 Disparity selection before HA 104
6.3 Implications of varying α_1 105
6.4 Differences between standard and HA 109
6.5 Results in Middlebury 113

A.1 We show 8 examples from YUD - Example 1 118
A.2 We show 8 examples from YUD - Example 2 119
A.3 We show 8 examples from our dataset - Example 1 120
A.4 We show 6 examples from our dataset - Example 2 121

List of Tables

3.1 Summary of the parameters 37
3.2 Runtime for evaluating the DSI 42
3.3 Computational complexity of the matching costs 42
3.4 Runtime of SRF 48
3.5 Specifications of the camera and the LRF 49

6.1 Spatial matching distribution 107
6.2 Aggregation configurations used 110
6.3 Comparison of 4 aggregation configurations 111
6.4 Evaluation in Middlebury 112

Acknowledgments

During these five years of PhD research, that in terms of motivation could be described as being a sinusoid of quite high frequency, a lot of persons were important to me. Some of them helped me to become a better researcher and engineer, others helped simply standing by my side during this long journey.

I want to thank my Advisor João Pedro Barreto for the observation ”What if the virtual planes in plane-sweeping intersect the baseline? There is some type of symmetry that could be a cue for estimating depth!!! Investigate!!!”. And so was born SymStereo, the main research of this PhD. I also what to thank his constant support in all stages of my PhD, as well as letting me ”divagate” and follow my own research ideas (some of them quite ”strange”).

I acknowledge the Portuguese Science Foundation (FCT) that generously funded my PhD through the grant SFRH/BD/47488/2008, and also the support of Professor Doutor Urbano Nunes and FCT under the project grants PTDC/SEN-TRA/099413/2008 and PTDC/EEA-AUT/113818/2009.

I want to thank Mom and Dad. Without them I could never be here. Thank you for all your efforts. I want to thank my Lab mates and friends Melo, Miguel, Defeat Saints, Francis, Vitor, DogAndYou, Butters, Abed two beds three beds, Pacheco, Salty, Lamb, Carolina,... and many others that in one way or another made this time more enjoyable. I want to thank my sister and my friends. Thank you friends for being my friends.

The biggest acknowledgment goes to my girlfriend and since 2012 my wife Teresa. This PhD was a long and demanding journey, but you make every day look fantastic.

Abstract

Recovering the 3D geometry from two or more views, known as stereo reconstruction, is one of the earliest and most investigated topics in computer vision. The computation of 3D models of an environment is useful for a very large number of applications, ranging from robotics, consumer utilization to medical procedures. The principle to recover the 3D scene structure is quite simple, however, there are some issues that considerable complicate the reconstruction process. Objects containing complicated structures, including low and repetitive textures, and highly slanted surfaces still pose difficulties to state-of-the-art algorithms.

This PhD thesis tackles this issues and introduces a new stereo framework that is completely different from conventional approaches. We propose to use symmetry instead of photo-similarity for assessing the likelihood of two image locations being a match. The framework is called SymStereo, and is based on the mirroring effect that arises whenever one view is mapped into the other using the homography induced by a virtual cut plane that intersects the baseline. Extensive experiments in dense stereo show that our symmetry-based cost functions compare favorably against the best performing photo-similarity matching costs. In addition, we investigate the possibility of accomplishing Stereo-Rangefinding that consists in using passive stereo to exclusively recover depth along a scan plane. Thorough experiments provide evidence that Stereo from Induced Symmetry is specially well suited for this purpose.

As a second research line, we propose to overcome the previous issues using priors about the 3D scene for increasing the robustness of the reconstruction process. For this purpose, we present a new global approach for detecting vanishing points and groups of mutually orthogonal vanishing directions in man-made environments. Experiments in both synthetic and real images show that our algorithms outperform the state-of-the-art methods while keeping computation tractable. In addition, we show for the first time results in simultaneously detecting multiple Manhattan-world configurations. This prior information about the scene structure is then included in a reconstruction pipeline that generates piece-wise planar models of man-made environments from two calibrated views. Our formulation combines SymStereo and PEARL clustering [3], and alternates between a discrete optimization step, that merges planar surface hypotheses and discards detections with poor support, and a continuous optimization step, that refines the plane poses. Experiments with both indoor and outdoor stereo pairs show significant improvements over state-of-the-art methods with respect to accuracy and robustness.

Finally, and as a third contribution to improve stereo matching in the presence of surface slant, we extend the recent framework of Histogram Aggregation [4]. The original algorithm uses a fronto-parallel support window for cost aggregation, leading to inaccurate results in the presence of significant surface slant. We

address the problem by considering discrete orientation hypotheses. The experimental results prove the effectiveness of the approach, which enables to improve the matching accuracy while preserving a low computational complexity.

Resumo

Recuperar a geometria 3D a partir de dois vistas, conhecida como reconstrução estéreo, é um dos tópicos mais antigos e mais investigado em visão por computador. A computação de modelos 3D do ambiente é útil para uma grande número de aplicações, desde a robótica, passando pela sua utilização do consumidor comum, até a procedimentos médicos. O princípio para recuperar a estrutura 3D cena é bastante simples, no entanto, existem algumas situações que complicam consideravelmente o processo de reconstrução. Objetos que contêm estruturas pouco texturadas ou repetitivas, e superfícies com bastante inclinação ainda colocam em dificuldade os algoritmos state-of-the-art.

Esta tese de doutoramento aborda estas questões e apresenta um novo framework estéreo que é completamente diferente das abordagens convencionais. Propomos a utilização de simetria em vez de foto-similaridade para avaliar a verosimilhança de pontos em duas imagens distintas serem uma correspondência. O framework é chamado SymStereo, e baseia-se no efeito de espelhagem que surge sempre que uma imagem é mapeada para a outra câmara usando a homografia induzida por um plano de corte virtual que intersecta a baseline. Experiências em estéreo denso comprovam que as nossas funções de custo baseadas em simetria se comparam favoravelmente com os custos baseados em foto-consistência de melhor desempenho. Param além disso, investigamos a possibilidade de realizar Stereo-Rangefinding, que consiste em usar estéreo passivo para recuperar exclusivamente a profundidade ao longo de um plano de varrimento. Experiências abrangentes fornecem evidência de que estéreo baseada em simetria induzida é especialmente eficaz para esta finalidade.

Como segunda linha de investigação, propomos superar os problemas descritos anteriormente usando informação *a priori* sobre o ambiente 3D, com o objectivo de aumentar a robustez do processo de reconstrução. Para tal, apresentamos uma nova abordagem global para detectar pontos de desvanecimento e grupos de direções de desvanecimento mutuamente ortogonais em ambientes Manhattan. Experiências quer em imagens sintéticas quer em imagens reais demonstram que os nossos algoritmos superaram os métodos state-of-the-art, mantendo a computação aceitável. Além disso, mostramos pela primeira vez resultados na detecção simultânea de múltiplas configurações de Manhattan. Esta informação a priori sobre a estrutura da cena é depois usada numa pipeline de reconstrução que gera modelos piecewise planares de ambientes urbanos a partir de duas vistas calibradas. A nossa formulação combina SymStereo e o algoritmo de clustering PEARL [3], e alterna entre um passo de otimização discreto, que funde hipóteses de superfícies planares e descarta detecções com pouco suporte, e uma etapa de otimização contínua, que

refina as poses dos planos. Expêriencias com pares estéreo de ambientes interiores e exteriores confirmam melhorias significativas sobre métodos state-of-the-art relativamente a precisão e robustez.

Finalmente, e como terceira contribuição para melhorar a visão estéreo na presença de superfícies inclinadas, estendemos o recente framework de agregação estéreo baseada em histogramas [4]. O algoritmo original utiliza janelas de suporte fronto-paralelas para a agregação de custo, o que leva a resultados imprecisos na presença de superfícies com inclinação significativa. Nós abordamos o problema considerando hipóteses de orientação discretas. Os resultados experimentais obtidos comprovam a eficácia do método, permitindo melhorar a precisão de correspondência, preservando simultaneamente uma baixa complexidade computacional.

Chapter 1

Introduction

1.1 Motivation

Photographs capture moments, places and events that are important and useful for a wide variety of personal and economical purposes. However, an image taken by a single camera is a low-dimensional representation of a rich 3D world, meaning that there is information lost during the physical imaging process. Computer vision tackles the inverse problem of recovering the 3D shape of objects and geometry of the environment from one or more images taken from the same object or scene. The computation of 3D models of an environment is advantageous for a very large number of applications, like city modeling and 3D mapping (e.g. [5, 6, 7, 8, 9]), robot navigation and autonomous drivers (e.g. [10, 11, 12, 13, 14, 15]), tele-immersion and view synthesis (e.g. [16, 17, 18, 19]). Computer vision is also applied in many medical procedures, for example in Minimally Invasive Surgery (MIS) it is used for registering the endoscopic video to a pre-operative 3D model of the anatomies of the patient (e.g. [20, 21, 22]).

Recovering the 3D geometry from two or more views, known as stereo reconstruction, is one of the earliest and most investigated topics in computer vision [23, 24, 25, 26]. The principle to recover the scene structure from a set of images is quite simple. Knowing the location of the same scene point in images taken by calibrated cameras, then it is possible to obtain the position of the point in 3D space using a technique known as triangulation [27]. This process of finding pixels in two or more images that correspond to the same 3D scene point is called stereo matching. Given that the parameters of the cameras (intrinsic calibration) and the relative position between the cameras (extrinsic calibration) is known, then the 2D dimensional pixel search ($\mathbb{R}^2 \rightarrow \mathbb{R}^2$) can be transformed to a 1D dimensional matching process ($\mathbb{R} \rightarrow \mathbb{R}$) based on the epipolar constraint [24].

This thesis focuses exclusively on the minimal problem of computing depth

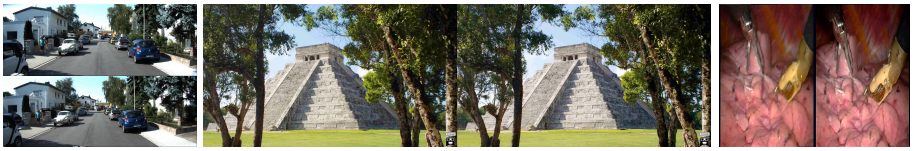


Figure 1.1: Three application scenarios that use stereo vision, from left to right: autonomous driving [14]; stereo pair downloaded from Flickr that was captured using a Sony Bloggie 3D; and stereo laparoscopy [28]

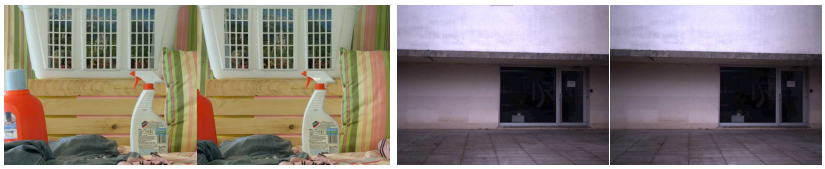
.

from a single pair of calibrated images. This has the disadvantage of having much less scene information available when compared to multi-view approaches [29], making the process considerably more error prone. But there are many advantages. First, and the main reason why we decided to explore the two view approach, is due to its larger amount of real application scenarios (refer to Figure 1.1). Setting up a (binocular) stereo vision system is nowadays quite uncomplicated, and camera calibration is a well studied topic [30, 31, 32, 33]. This makes stereo vision widely used in a large number of robotic tasks (e.g. [34, 11, 35, 36, 37, 13, 38]), many of which can work indoors as well as outdoors. This is important to stress, because the recent research in RGB-D cameras is not a viable alternative in case large depth ranges of operation or variable lighting conditions are considered. Besides these applications in robotics, very recently started to appear stereo cameras as consumer electronics being available either as standalone hand-held cameras (e.g. Fujifilm Finepix 3D, Sony Bloggie 3D) or integrated into smartphones (e.g. HTC Evo 3D). The work described in this thesis is also motivated by this proliferation of stereo cameras that we believe will create an urge for robust algorithms able to render complete, photo-realistic 3D models in an automatic manner. Finally, the computation of the 3D structure of organs, tissues and surgical instruments during MIS is important for computer assisted interventions (e.g. [20, 21, 22, 39, 40]). In the last few years, endoscope manufacturers started to produce equipments with two lenses, and in particular stereoscopic laparoscopes (e.g. Olympus 3D endoscopy). We are in the opinion that the research reported in this thesis will also have impact in this field, where robust approaches for extracting depth information from two views are required.

As stated, the research in this thesis will only consider two view stereo. In this case, the procedure of stereo matching is considerably simplified when the input images are previously rectified so that the epipolar lines are horizontally aligned [41, 23, 24, 26]. After rectification, the similarity of the pixels are compared at corresponding pixel locations (x, y) in the left view and $(x', y') = (x - d, y)$ in the right view, where d is the disparity, and is inversely proportional to the depth. The computation of the matching cost for each pixel (x, y) and each disparity hypothesis d creates a volume called Disparity Space Image (DSI) [42]. The final disparity map is obtained by analyzing the DSI and selecting a disparity d for each pixel (x, y).

The principle to recover the 3D scene structure is simple, however, and as discussed in my thesis project [43], there are some issues that considerable complicate the reconstruction process. This thesis mainly tackles two of them, which are mostly due to the captured 3D scene, other problems like errors in the camera calibration are not considered.

(a) Textured scenes



(b) Low and repetitive textured scenes

(c) Scenes containing slanted surfaces

Figure 1.2: Problematic situations to stereo matching. (a) The objects contained in these scenes are mostly textured and it is quite easy for stereo matching to identify the correct matches [23, 44, 45]; (b) These scenes contain mostly low and repetitive textures, it is difficult even for humans to identify correctly all the matching pixels [44, 45]; (c) surfaces that are non-frontal to the cameras pose additional difficulties to stereo matching, as we will see later, this occurs because a particular region in one view is warped in the right view depending on the slant of the surface. Note that all stereo scenarios depicted in Figure 1.1 contain these type of difficulties.

The first issue is due to the ambiguity in the stereo matching process that occurs in scene regions with low and/or repetitive textures (refer to Figure 1.2). All stereo algorithms need somehow to measure the likelihood of pixels in different views being a match, which is usually done by quantifying the photo-consistency between pixels or image regions. It was concluded in [23, 45] that the choice of this matching cost is crucial for the final performance of the stereo algorithm, but top-performing metrics still have difficulties in handling this type of complicated textures.

The second problem is posed by slanted surfaces contained in the scene, which difficult almost all stages of traditional stereo matching pipelines (refer to Figure 1.2). This occurs because most existing approaches assume, for simplicity

and tractability purposes, that the surfaces to be reconstructed are fronto-parallel to the cameras. As we will discuss later, there is a vast literature [46, 47, 48, 49, 50, 51, 52, 53] that tries to overcome this problem by estimating simultaneously depth and the *local* slant around the pixel being analyzed. I am in the opinion that the most recent approaches already do a good job in terms of accuracy, but better matching costs would certainly improve their performance. However, must of these approaches are not often used because they are complex and time consuming.

In contrast to standard stereo matching techniques, many authors propose to overcome the previous issues using priors about the 3D scene during the regularization process (e.g. [5, 7, 8, 9]). These approaches have the advantage of providing 3D models of the scene that are perceptually pleasing and geometrically simple, and, thus, their rendering, storage and transmission is computationally less complex. These approaches use multiple views, however, and as referred previously, our focus is on stereo reconstruction from a single pair of images. I think that, for this specific case, the literature lacks an appropriate pipeline.

Finally, and after a thorough review of the stereo literature (mostly documented in [43]), we wondered if it is always necessary to compute a complete disparity map whenever stereo vision is used? We are in the opinion that, in contrast to 3D modeling purposes, there are many applications (specially in robotics) in which a less dense 3D reconstruction could be sufficient for accomplishing the objectives (e.g. navigation, pedestrian detection, self-localization). How can this be done without loosing much accuracy, and being more efficient that computing the complete depth map (of course, it is useless if it is slower)?

1.2 Organization

This thesis is divided into three main parts. The first part faces the two issues described previously (matching ambiguity and surface slant), and also investigates the possibility of efficiently estimating depth only for a subset of image pixels without severely affecting the accuracy. It is organized as follows:

- Chapter 2 presents the SymStereo framework, providing an intuitive description of the mirroring effect that is induced by a virtual plane intersecting the baseline. The mirroring effect is the cornerstone of SymStereo because it enables the rendering of image signals that are either symmetric or anti-symmetric with respect to the contour where a virtual plane cuts meets the scene. Following this, stereo matching is achieved by finding the image of this contour in the two views using symmetry cues. A geometric analysis of the framework is performed, providing a formal proof of the mirroring effect, discussing singular configurations, and explaining how to select an appropriate set of virtual cut planes.

- Chapter 3 derives suitable symmetry metrics for quantifying the likelihood

of a certain image pixel being locally symmetric and/or anti-symmetric. We experimentally evaluate the symmetry-based metrics against photo-similarity for the purpose of data association in dense stereo. Moreover, and since the symmetries are induced using virtual cut planes, these new matching functions are particularly well suited for recovering depth along pre-defined scan planes. The independent estimation of depth along a scan plane is called Stereo-Rangefinding (SRF). We evaluate symmetry against photo-consistency based matching costs for the purpose of SRF, and also compare the depth estimation obtained using a SRF pipeline against the readings provided by a Laser-Rangefinder.

The second part investigates the problem of extracting geometric information from the scene, and how to use this information to improve the 3D reconstruction process. It is composed by two chapters:

- Chapter 4 addresses the problem of detecting vanishing points (VPs) and their grouping into sets of mutually orthogonal vanishing directions (VDs). These problems are cast as Uncapacited Facility Location (UFL) and Hierarchical Facility Location (HFL) problems, respectively, and solved using a message passing approach. We provide experimental result in synthetic and real images, and compare the performance of our algorithms against state-of-the-art approaches.

- Chapter 5 investigates the use of the planarity prior about the 3D scene for enhancing the 3D reconstruction. We propose a pipeline that combines SRF and PEARL optimization [3] for this purpose, and use the VPs obtained from the approach described in Chapter 4 for constraining the planar segmentation. The experiments show that the plane hypotheses computed using our symmetry-based pipeline outperform the approaches based on dense stereo reconstruction and sparse feature matching.

Finally, Chapter 6 extends the work presented in [4], and proposes a new stereo aggregation scheme able to cope with surface slant. The strategy consists in selecting the most suitable aggregation direction within a pre-defined set of discrete hypotheses. The approach is able to combine high matching accuracy with small computational overhead when compared to the state-of-the-art.

1.3 Contributions

There are some challenges, identified previously, that this thesis has the ambition to overcome. This work makes the following contributions:

- **A new cue for stereo vision** - Chapter 2 presents the first work in the literature proposing to use symmetry instead of photo-similarity for assessing the

likelihood of two image locations being a match. The framework is called SymStereo, and is based on the mirroring effect that arises whenever one view is mapped into the other using the homography induced by a virtual cut plane that intersects the baseline.

- **New matching costs based on symmetry** - Chapter 3 proposes three symmetry-based matching costs. The new matching costs are benchmarked against the state-of-the-art metrics for accomplishing dense disparity labeling in both short and wide-baseline images. The results show that the symmetry-based functions consistently outperform their similarity-based counterparts, suggesting that symmetry is superior to standard photo-consistency as a stereo metric.

- **Stereo-Rangefinding (SRF)** - Chapter 2 and Chapter 3 investigate the use of passive stereo for estimating depth along a single scan plane. The technique, named SRF, provides profile cuts of the scene similar to the ones that would be obtained by a Laser-Rangefinder (LRF). We provide the first benchmark of SRF, and compare the depth estimates obtained using a SRF pipeline with the readings provided by a 2D Laser-Rangefinding (LRF). The experimental results demonstrate that SRF can be leveraged to meet the robustness and depth accuracy of laser range data.

- **A global approach for detecting VPs and groups of mutually orthogonal VDs** - Chapter 4 presents an automatic and global approach for the detection of VPs and mutual orthogonal VDs. The core of the framework is the formulation of these multi-model fitting problems as Uncapacited Facility Location (UFL) and Hierarchical Facility Location (HFL) instances, respectively. Its effectiveness is experimentally evaluated in real scenarios containing multiple Manhattan-world configurations.

- **A Piecewise Planar Reconstruction (PPR) pipeline** - A new pipeline (refer to Chapter 5) that combines the SymStereo framework and PEARL [3] for the purpose of PPR. The experimental results obtained with this system demonstrate that it is possible to obtain accurate and simple 3D models of indoor and outdoor scenes from only two calibrated images.

- **A Histogram Aggregation (HA) framework that accounts for surface slant** - The strategy described in Chapter 6 consists in selecting the most appropriate aggregation direction for HA within a set of discrete hypotheses. The approach is able to combine high matching accuracy with small computational overhead when compared to existing approaches.

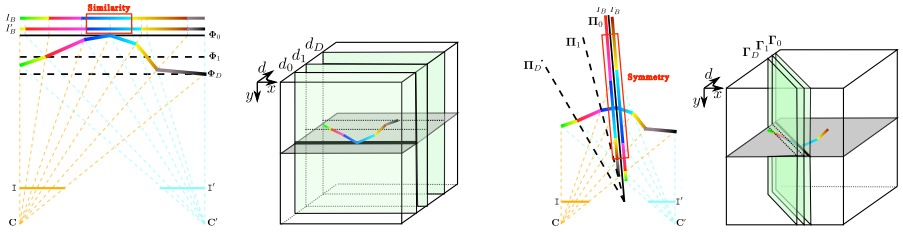
Chapter 2

Stereo Matching using Induced Symmetry: a geometric account

Stereo methods always require a matching function for assessing the likelihood of two pixels being in correspondence. Such functions, commonly referred as matching costs, measure the photo-consistency between image regions centered in putative matches. This chapter proposes a new framework from which a new family of stereo cost functions that measure symmetry instead of photo-similarity for associating pixels across views are derived. We start by observing that, given two stereo views and an arbitrary virtual plane intersecting the baseline, it is possible to render image signals that are either symmetric or anti-symmetric with respect to the contour where the virtual plane meets the scene. The fact is investigated in detail and used as cornerstone to develop a new stereo framework that relies in symmetry cues for solving the data association problem.

2.1 Introduction

Stereo correspondence methods require a metric for assessing the likelihood of two image locations being a match. Typically, the first step of a dense stereo algorithm is the evaluation of this matching function for all pixel locations and disparity range. The result is the DSI [42] over which is carried either local aggregation or global optimization with the objective of computing a depth map [23]. Local stereo methods aggregate the matching function over a support region for obtaining a spatially coherent DSI [54, 55]. This is usually followed by a Winner-Takes-All (WTA) procedure along the disparity dimension. In global stereo methods, the pixel correspondence between views is formulated as a global optimization prob-



(a) Plane Sweeping (b) DSI Plane Sweeping (c) SymStereo (d) DSI SymStereo

Figure 2.1: Plane Sweeping vs SymStereo. (a) and (b): Conventional stereo matching is a particular instance of plane sweeping [58]. The DSI is evaluated for increasing values of disparity d_i. Each disparity hypothesis d_i is associated with a virtual plane $\mathbf{\Phi}_i$ that is fronto-parallel. The chosen matching cost implicitly measures the photo-similarity between $\mathtt{I}_B$ and $\mathtt{I}'_B$, that are the results of back-projecting $\mathtt{I}$ and $\mathtt{I}'$ onto $\mathbf{\Phi}_i$; (c) and (d) - In SymStereo the virtual planes $\mathbf{\Pi}_i$ intersect the baseline, and the back-projection images are reflected with respect to the curve where $\mathbf{\Pi}_i$ intersects the scene structure (mirroring effect). This enables to perform stereo matching using symmetry instead of photo-similarity. In the same manner that each plane $\mathbf{\Phi}_i$ in (a) is associated with a constant disparity plane in (b), each plane $\mathbf{\Pi}_i$ in (c) corresponds to an oblique plane $\mathbf{\Gamma}_i$ in (d). Thus, the entire DSI domain can be fully covered by carefully choosing the set of virtual cut planes $\mathbf{\Pi}_i$.

lem over the DSI, which is solved using an energy minimization framework [56]. There is a third strategy called Semi-Global Matching (SGM) that minimizes a 2D energy function defined over the DSI by performing pathwise optimization along multiple directions [57].

This chapter revisits the construction of the DSI using a new type of matching functions. The functions described in the stereo literature rely in measuring the photo-consistency between two image locations. We show that, given a calibrated stereo pair, it is possible to render image signals that are either symmetric or anti-symmetric around the projection of the contour where an arbitrary virtual cut plane intersects the scene. This allows to use symmetry instead of photo-consistency for quantifying the likelihood of two pixels being a match.

2.1.1 Notation and Terminology

We represent scalars in italic, e.g. s, vectors in bold characters, e.g. $\mathbf{p}$, matrices in sans serif font, e.g. M, image signals in typewriter font, e.g. $\mathtt{I}$, and curves in calligraphic symbols, e.g. $\mathcal{C}$. Unless stated otherwise, we use homogeneous coordinates for points and other geometric entities, e.g. a point with non-homogeneous image coordinates $(p_1,\, p_2)$ is represented by $\mathbf{p} \sim (p_1\, p_2\, 1)^\mathsf{T}$, with $\sim$ denoting equality up to a scale. Finally, $[\mathbf{v}]_\times$ denotes the skew symmetric matrix defined by the 3-vector $\mathbf{v}$, and $\mathsf{I}_{3\times3}$ refers to the 3×3 identity matrix.

Although SymStereo can be used with any stereo pair, we assume, if not otherwise stated, rectified stereo for most derivations and experiments throughout this thesis. Thus, a generic 1-D line of the image signal $\mathtt{I}$ is denoted by $\mathtt{I}(p_1)$, with

p_1 being the free coordinate along the horizontal axis. The 1-D signal $\mathtt{I}(p_1)$ has a local symmetry about a point q_1 in its domain iff the following holds:

$$\mathtt{I}(q_1 + \delta) \;=\; \mathtt{I}(q_1 - \delta),\; \forall_{\delta \in \mathcal{N}}$$

with $\mathcal{N}$ being an interval centered in zero. In a similar manner, $\mathtt{I}(p_1)$ is said to be anti-symmetric in a local neighborhood around q_1 iff

$$\mathtt{I}(q_1) - \mathtt{I}(q_1 + \delta) \;=\; -(\mathtt{I}(q_1) - \mathtt{I}(q_1 - \delta)),\; \forall_{\delta \in \mathcal{N}}$$

The stereo matching will be carried by quantifying 1-D signal symmetry and anti-symmetry in successive pixel locations along epipolar lines.

We will often refer to a matching function as being a "matching cost" or a "cost function" without distinguishing if the function measures photo-similarity, photo-dissimilarity, local symmetry or lack of local symmetry. We will also employ the term "similarity-based matching cost" to designate matching functions that use conventional photo-consistency metrics, as opposed to the new stereo functions that exploit induced symmetry cues.

2.2 Mirroring effect and Stereo from Induced Symmetry

Let $\mathtt{I}$ and $\mathtt{I}'$ be a pair of rectified images acquired by two cameras with projection centers $\mathbf{C}$ and $\mathbf{C}'$. The scheme of Figure 2.1(a) is a top-view of this situation, where the two cameras observe a concave surface S with five regions identified with different colors. The 3D volume of Figure 2.1(b) is the corresponding DSI, with each point $(\mathbf{p}, d)$ representing the disparity hypothesis d for the pixel location $\mathbf{p} = (x, y)$ [42]. The matching cost is a scalar function with domain $(\mathbf{p}, d)$, and the DSI is the result of evaluating this function across the entire domain. Ideally, the cost function should be such that for each pixel $\mathbf{p}$ there is a single extremum along the disparity axis that signals the correct disparity value d. In this case, the set of all extrema define a surface in the DSI that enables the accurate 3D reconstruction of the scene. In practice, several ambiguities arise, and the evaluation of the matching cost usually leads to multiple incorrect extrema. The steps of local aggregation and/or global optimization over the DSI aim to overcome this problem by refining the matching surface taking into account spatial consistency.

It is well known that, for the case of rectified stereo, image points lying in a fronto-parallel plane $\mathbf{\Phi}_0$ are related by the same disparity amount d_0. Thus, the disparity plane d_0 in the DSI can be evaluated by back-projecting the two input views $\mathtt{I}$ and $\mathtt{I}'$ onto the virtual plane $\mathbf{\Phi}_0$, followed by comparing the results $\mathtt{I}_B$ and $\mathtt{I}'_B$ using some type of photo-similarity metric. As shown in the scheme of Figure 2.1 (a), the back-projected images $\mathtt{I}_B$ and $\mathtt{I}'_B$ overlap in the points where $\mathbf{\Phi}_0$ inter-

(a) Image $\mathtt{I}$ (b) 3D model (c) Image $\mathtt{I}'$ (d) SAD (e) ZNCC (f) Census

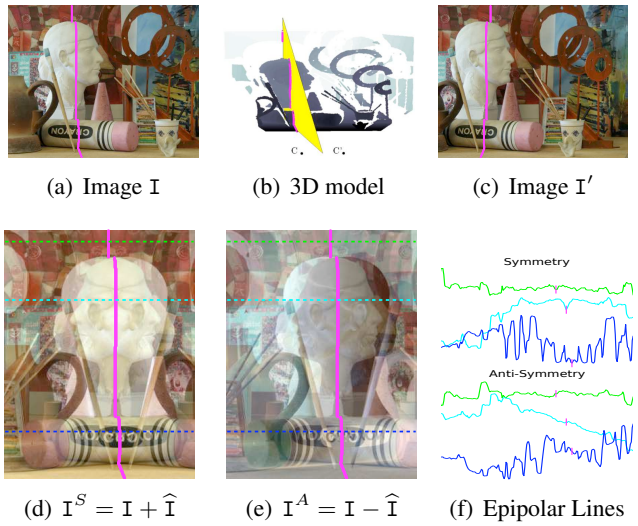
Figure 2.2: Conventional stereo matching costs based in photo-similarity. $\mathtt{I}$ and $\mathtt{I}'$ are stereo views of the 3D scene shown in (b). The virtual plane $\mathbf{\Phi}_0$ (yellow) corresponds to a constant disparity d_0 in the DSI domain. Let $\widehat{\mathtt{I}}$ be the result of mapping $\mathtt{I}'$ into $\mathtt{I}$ using the plane-homography. The disparity hypothesis d_0 is evaluated by measuring the photo-similarity between $\mathtt{I}$ and $\widehat{\mathtt{I}}$, such that the image of the regions where $\mathbf{\Phi}_0$ intersects the scene structure becomes highlighted (d)-(f).

sects the scene surface and, consequently, the quantification of photo-similarity tends to highlight these image locations enabling a correct disparity assignment. This way of addressing the problem was introduced by Collins, that suggested to find matches across multiple views by sweeping the 3D space with a pre-defined set of virtual planes [59]. The computation of the DSI in rectified stereo can be understood as a particular instance of *plane sweeping*, with the sweeping direction being parallel to the camera axis, and each plane $\mathbf{\Phi}_i$ corresponding to a constant disparity d_i (see Figure 2.1(a)-(b)).

SymStereo relates with plane sweeping in the sense that it also samples the 3D space by a set of virtual planes. However, there are two major differences: (i) the virtual planes intersect the baseline, which is considered a degenerate configuration in plane sweeping [60]; and (ii) the pixel association between views is achieved using symmetry cues instead of photo-similarity metrics.

Consider the scheme of Figure 2.1(c), with $\mathbf{\Pi}_0$ being a plane that intersects the baseline, and $\mathtt{I}_B$ and $\mathtt{I}'_B$ being the result of back-projecting views $\mathtt{I}$ and $\mathtt{I}'$ onto $\mathbf{\Pi}_0$. Remark that, while in Figure 2.1(a) the back-projection images correlate in the pixel locations where the virtual plane meets the 3D surface, in Figure 2.1(c) the images $\mathtt{I}_B$ and $\mathtt{I}'_B$ are mirrored with respect to the curve $\mathcal{C}$ where $\mathbf{\Pi}_0$ intersects the scene structure. SymStereo explores this *mirroring effect* for accurately reconstructing the contour $\mathcal{C}$ (the *profile cut*) using symmetry analysis. As discussed next, the strategy is effective not only for recovering depth along a virtual cut plane (SRF), but also for achieving dense stereo matching. It can be proved that the mirroring effect holds for any plane $\mathbf{\Pi}_i$ intersecting the baseline, corresponding to an oblique plane $\mathbf{\Gamma}_i$ in the DSI domain. Thus, and in a similar manner to plane sweeping, it is possible to carefully select the virtual cut planes such that the DSI is fully evaluated and the correct disparity surface is recovered (Figure 2.1 (d)).

Figure 2.2 aims to illustrate the evaluation of the disparity hypothesis d_0 using a conventional stereo matching cost such as Sum of Absolute Differences (SAD), Zero-mean Normalized Cross-Correlation (ZNCC) or Census. The plane $d = d_0$ in the DSI domain (Figure 2.1(b)) corresponds to a fronto-parallel virtual plane $\mathbf{\Phi}_0$ that is marked in yellow in the 3D model of Figure 2.2(b). Let $\widehat{\mathtt{I}}$ be the warping



(a) Image $\mathtt{I}$ (b) 3D model (c) Image $\mathtt{I}'$

(d) $\mathtt{I}^S = \mathtt{I} + \widehat{\mathtt{I}}$ (e) $\mathtt{I}^A = \mathtt{I} - \widehat{\mathtt{I}}$ (f) Epipolar Lines

Figure 2.3: SymStereo: The *virtual cut plane* $\mathbf{\Pi}_0$ in yellow intersects the scene structure in a non-continuous 3D curve $\mathcal{C}$ marked in magenta (the profile cut). Let $\widehat{\mathtt{I}}$ be the result of warping $\mathtt{I}'$ by the plane-homography induced by $\mathbf{\Pi}_0$. The image signals $\mathtt{I}^S$ and $\mathtt{I}^A$, obtained by adding and subtracting $\mathtt{I}$ with $\widehat{\mathtt{I}}$, are respectively symmetric and anti-symmetric around the image of the profile cut $\mathcal{C}$ (d)-(e). In (f) we show the pixel intensities of $\mathtt{I}^S$ and $\mathtt{I}^A$ along three distinct epipolar lines (green, cyan and blue). Remark that the intersections with the locus where $\mathcal{C}$ is projected can be identified with almost no ambiguity by searching common pixel locations for which the top and bottom 1D-signals are respectively locally symmetric and anti-symmetric.

result of mapping the right view $\mathtt{I}'$ into the left reference view using the plane-homography induced by $\mathbf{\Phi}_0$. For the particular case of rectified stereo, the warping is a simple image shift by d_0 pixels along the horizontal axis. The DSI values of the points lying in the plane $d = d_0$ is determined by measuring the similarity between images $\mathtt{I}$ and $\widehat{\mathtt{I}}$ using a specific metric. As shown by the results of Figure 2.2(d)-(f), this enables depth recovery by highlighting the pixel locations corresponding to the regions where $\mathbf{\Phi}_0$ intersects the scene structure (magenta marks in Figure 2.2(a)-(c)).

In this chapter, we propose to evaluate the DSI using a different strategy. Consider the virtual cut plane $\mathbf{\Pi}_0$ that intersects the scene surfaces in the profile cut $\mathcal{C}$ marked with magenta in the model of Figure 2.3(b). Let H be the plane-homography associated with $\mathbf{\Pi}_0$ that maps the right image into the reference view. If $\widehat{\mathtt{I}}$ is the warping result of mapping $\mathtt{I}'$ by H, then it comes from the mirroring effect that $\mathtt{I}$ and $\widehat{\mathtt{I}}$ are reflected around the image of the profile cut. Thus, the sum of $\mathtt{I}$ and $\widehat{\mathtt{I}}$ yields an image signal $\mathtt{I}^S$ that is symmetric around the locus where $\mathcal{C}$ is projected (Figure 2.3(d)). In a similar manner, the difference between $\mathtt{I}$ and $\widehat{\mathtt{I}}$ gives rise to an image signal $\mathtt{I}^A$ that is anti-symmetric at the exact same location (Figure 2.3 (e)). SymStereo detects the image of the profile cut by jointly evaluating symmetry and anti-symmetry of $\mathtt{I}^S$ and $\mathtt{I}^A$ at every pixel location (Figure 2.3 (f)). This provides

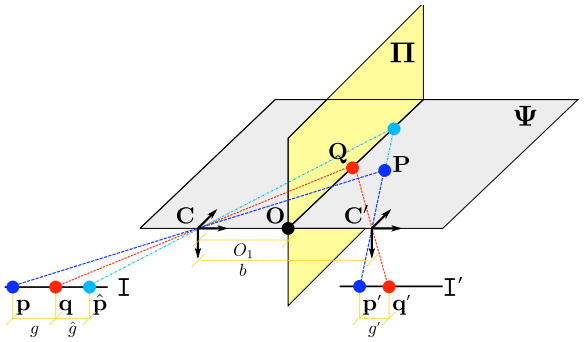


Figure 2.4: Geometric analysis of SymStereo. The analysis is carried in an arbitrary epipolar plane $\mathbf{\Psi}$ assuming that the images are rectified. The camera centers $\mathbf{C}$ and $\mathbf{C}'$ are separated by a distance b (baseline), and the world frame is coincident with the coordinate system of the left view (reference view). For the sake of graphical clarity the image points are projected behind the optical centers.

an implicit manner of recovering depth along $\mathbf{\Pi}_0$ and achieving data association across views. Since $\mathbf{\Pi}_0$ is mapped into an oblique plane $\mathbf{\Gamma}_0$ in the DSI domain, the joint symmetry and anti-symmetry metric assigns a matching cost to every point $(\mathbf{p}, d)$ lying on $\mathbf{\Gamma}_0$. Thus, and as stated above, the DSI can be fully evaluated by stacking the results of a set of planes $\mathbf{\Pi}_i$ such that the corresponding planes $\mathbf{\Gamma}_i$ cover the entire $(\mathbf{p}, d)$ domain (Figure 2.1(d)).

2.3 Geometric Analysis

This section derives the conditions for a generic 3D plane $\mathbf{\Pi}$ to intersect the baseline, proves that the mirroring effect holds for any virtual plane intersecting the baseline iff corresponding image pixels have the same order in both views, and discusses the mapping of planes $\mathbf{\Pi}_i$ in 3D space into planes $\mathbf{\Gamma}_i$ in the DSI domain.

2.3.1 Necessary and sufficient condition for a virtual plane $\mathbf{\Pi}$ to intersect the baseline

As shown in Figure 2.4, consider a rectified stereo pair acquired by two cameras with centers in $\mathbf{C}$ and $\mathbf{C}'$. Since the camera reference frames are aligned, the transformation T that maps the right view coordinates into left view coordinates is

$$\mathsf{T} = \begin{pmatrix} \mathsf{I}_{3\times 3} & \mathbf{t} \\ \mathbf{0}^{\mathsf{T}} & 1 \end{pmatrix}, \tag{2.1}$$

with

$$\mathbf{t} = \begin{pmatrix} b \\ 0 \\ 0 \end{pmatrix} .$$

We assume that the world coordinate system is coincident with the reference frame centered in $\mathbf{C}$. The virtual cut plane $\mathbf{\Pi}$, that passes between the cameras, is represented by the following homogeneous vector

$$\mathbf{\Pi} \sim \begin{pmatrix} \mathbf{n} \\ -h \end{pmatrix} , \tag{2.2}$$

where $\mathbf{n}$ indicates the direction orthogonal to the plane

$$\mathbf{n} \sim \begin{pmatrix} n_1 \\ n_2 \\ n_3 \end{pmatrix} .$$

In addition, the centers $\mathbf{C}$ and $\mathbf{C}'$ define a line $\mathbf{L}$ that contains the baseline and has Plücker coordinates [61]

$$\mathbf{L} \sim \begin{pmatrix} \mathbf{t} \\ \mathbf{0} \end{pmatrix} .$$

The intersection of the virtual cut plane with the baseline can be computed by multiplying the 4-vector $\mathbf{\Pi}$ with the Plücker matrix of the dual of $\mathbf{L}$ [62]. It follows that the homogeneous coordinates of the intersection point $\mathbf{O}$ are

$$\mathbf{O} \sim \begin{pmatrix} -[\mathbf{0}]_\times & \mathbf{t} \\ -\mathbf{t}^\mathsf{T} & 0 \end{pmatrix} \mathbf{\Pi} \sim \begin{pmatrix} \frac{h}{\mathbf{n}_1} \\ 0 \\ 0 \\ 1 \end{pmatrix} .$$

Using β to denote the ratio between the signed distances $\mathbf{CO}$ and $\mathbf{CC}'$ comes that the plane $\mathbf{\Pi}$ intersects the baseline *iff* the following condition holds

$$0 < \left(\beta = \frac{O_1}{b}\right) < 1 \quad \Longleftrightarrow \quad \beta^{-1} = \frac{b\, n_1}{h} > 1. \tag{2.3}$$

2.3.2 Proof of the Mirroring Effect

Consider a generic 3D point $\mathbf{P}$ that is projected into points $\mathbf{p}$ and $\mathbf{p}'$ in the stereo views as shown in Figure 2.4. Since we are assuming rectified stereo, then the non-

homogeneous coordinates p_2 and p'_2 have the same value y. In a similar manner, consider a point $\mathbf{Q}$ that lies in the intersection of the same epipolar plane $\mathbf{\Psi}$ with the virtual plane $\mathbf{\Pi}$. Since the image points $\mathbf{p}$, $\mathbf{q}$ in the left view and $\mathbf{p}'$, $\mathbf{q}'$ in the right view only differ in terms of the first coordinates, we define the following pair of signed distances:

$$\begin{array}{rcl} g & = & p_1 - q_1 \\ g' & = & p'_1 - q'_1 \end{array} \tag{2.4}$$

Remark that g and g' have the same sign *iff* the points $\mathbf{P}$ and $\mathbf{Q}$ are imaged with the same order in the two views. We assume henceforth that this condition holds.

The plane $\mathbf{\Pi}$ defines a homography H that maps points from the right view into the left view. Given the relative camera pose of Equation 2.1 and the homogeneous plane representation of Equation 2.2, it comes that [61]

$$\mathsf{H} \sim \left(\mathsf{I}_{3\times 3} + \frac{\mathbf{t}\,\mathbf{n}^{\mathsf{T}}}{h}\right)^{-1} \sim \begin{pmatrix} 1 + \frac{bn_1}{h-bn_1} & \frac{bn_2}{h-bn_1} & \frac{bn_3}{h-bn_1} \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \tag{2.5}$$

Using H to map $\mathbf{p}'$ in the right view onto $\widehat{\mathbf{p}}$ in the left view yields

$$\widehat{p}_1 \;=\; \left(1 + \frac{bn_1}{h - bn_1}\right) p'_1 \;+\; k_y,$$

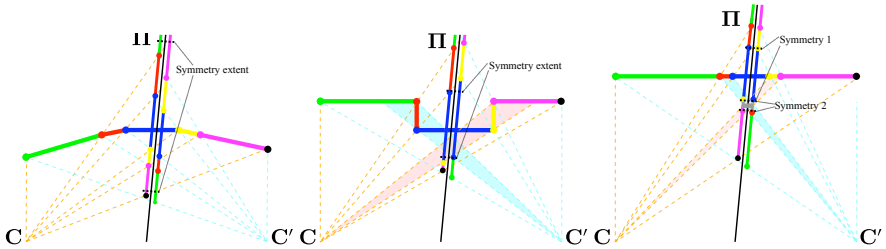
with k_y depending on the second coordinate y and being a constant for points on the same epipolar line. From Equation 2.4 comes that $p'_1 = g' + q'_1$ and the expression above can be re-written as

$$\widehat{p}_1 \;=\; \left(1 + \frac{bn_1}{h - bn_1}\right) q'_1 \;+\; k_y + \left(1 + \frac{bn_1}{h - bn_1}\right) g'. \tag{2.6}$$

In a similar manner, let $\widehat{\mathbf{q}}$ be the mapping result of $\mathbf{q}'$ such that $\widehat{\mathbf{q}} \sim \mathsf{H}\,\mathbf{q}'$. Since $\mathbf{Q}$ lies in the cut plane $\mathbf{\Pi}$ that defines the homography, then point $\widehat{\mathbf{q}}$ must be coincident with $\mathbf{q}$ and the following holds

$$q_1 \;=\; \left(1 + \frac{bn_1}{h - bn_1}\right) q'_1 \;+\; k_y\,.$$

Replacing the result above in Equation 2.6 comes that the signed image distance



(a) Continuous surface (b) Occlusion (c) Double Nail Illusion

Figure 2.5: (a) In the case the virtual cut plane $\mathbf{\Pi}$ intersects the scene in a continuous surface, most of the back-projected image regions contribute for the mirroring effect. (b) In the presence of occlusions (the yellow region is occluded in the left view and the red region is occluded in the right view), the symmetry extend is reduced and limited by the depth occlusion boundaries. (c) In the presence of double nail illusion, the virtual cut plane intersects two surfaces, in which case the mirroring effect occurs in two distinct regions - one corresponding to the surface in front (grey) and one corresponding to the surface in the back (blue).

between $\mathbf{q}$ and $\widehat{\mathbf{p}}$ is

$$\widehat{g} = \widehat{p}_1 - q_1 = \left(1 - \frac{bn_1}{h}\right)^{-1} g' \,. \tag{2.7}$$

For the case of the virtual plane $\mathbf{\Pi}$ passing between the cameras, the condition of Equation 2.3 holds, which means that g' and $\widehat{g}$ have opposite signs. Thus, and assuming that distances g and g' have always the same sign, we have just proved that points $\mathbf{p}$ and $\widehat{\mathbf{p}}$ must be on opposite sides of $\mathbf{q}$, so that the mirroring effect holds for any plane $\mathbf{\Pi}$ that intersects the baseline. Regarding the modulus of the distances g and $\widehat{g}$, it should be equal in order for the image symmetry of Figure 2.3(d) to be geometrically accurate. It can be analytically shown that in general $|g| \neq |\widehat{g}|$ (refer to Section 2.4), leading to a deviation in the rendered symmetry that depends both on the point where $\mathbf{\Pi}$ intersects the baseline, and on the position and slant of the imaged 3D surface.

2.3.3 Singular Configuration

We have proved that the homography associated with a cut plane causes a reflection *iff* the scene points are projected in the two views in the same order. For most stereo applications, the spatial order of corresponding points in the two views is the same, and the mirroring effect is verified (refer to Figure 2.5 (a) and (b)). However, there is a singular configuration for which the ordering constraint is not verified. This configuration, known as *double nail illusion*, typically arises in scenes with foreground objects that are finer than the baseline, or narrow holes [63]. Consider

the scheme of Figure 2.5 (c), in which case the thin foreground object (grey) causes a double nail illusion - the grey region is projected to the right of the blue region in the left view, while to the left in the right view. In this case, the virtual cut plane $\mathbf{\Pi}$ intersects the scene in two distinct regions (grey and blue) visible by both cameras. The mirroring effect occurs in both regions and two different symmetries are induced using SymStereo, each one precluding the detection of the other. Since the double nail illusion arises seldom in practice, we will ignore it for the rest of the paper, and consider that the mirroring effect is always verified, with the cut plane intersecting the scene in a single point per epipolar line.

2.4 Perfect symmetry in the presence of surface slant

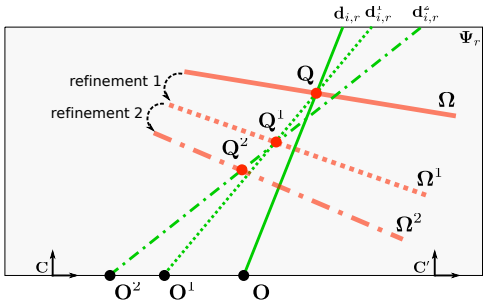


Figure 2.6: Refinement using slant prior (top view of scene in Figure 2.4). Assume that $\mathbf{Q}$ lies on the plane $\mathbf{\Omega}$. Then, we can determine the position on the baseline $\mathbf{0}^1$ (see Equation 2.11) that improves the induced symmetries. Using the vertical virtual cut plane defined by $\mathbf{0}^1$ and $\mathbf{Q}$, it is possible to induce new symmetries from which the refined point $\mathbf{Q}^1$ is estimated. $d_{i,r}$ is the intersection between the virtual cut plane $\mathbf{\Pi}_i$ and the epipolar plane $\mathbf{\Psi}_r$.

In plane-sweeping [58] it is possible to integrate prior knowledge of the scene to select the sweeping directions that maximize the performance of photo-consistency based stereo [60, 64]. We show in this section that slant priors can also be used in SymStereo for choosing the cut planes that render perfect signal symmetries, and improve the overall accuracy and robustness of the approach.

Consider again generic point $\mathbf{P}$ and a point $\mathbf{Q}$ that lies on the same epipolar plane $\mathbf{\Psi}$, and also assume that $\mathbf{Q}$ belongs to the virtual cut plane $\mathbf{\Pi}$ (see Figure 2.4). Let $d_p = p_1 - p'_1$ and $d_q = q_1 - q'_1$ be the disparities of $\mathbf{P}$ and $\mathbf{Q}$, respectively, and define

$$\Delta \;=\; d_p - d_q\,.$$

From Equation 2.4 follows that $g' = g - \Delta$, and Equation 2.7 can be written as

$$\widehat{g} = \left(\frac{\beta}{\beta - 1}\right)(g \;-\; \Delta). \tag{2.8}$$

The deviation in perfect mirroring ($\widehat{g}=-g$) around the projection of the profile cut is function of the differences in pixel disparities, which is directly related to the depth variation in the neighborhood of the 3D profile cut. Note that the virtual cut plane $\mathbf{\Pi}$ only affects the symmetry in terms of the intersection point with the baseline. For similar conditions of relative depth variation, any cut plane going through the same point $\mathbf{O}$ generates symmetries with equivalent quality, regardless of its orientation. Also note that, for the particular case of planes $\mathbf{\Pi}$ intersecting the baseline in the midpoint ($\beta=0.5$), the symmetry is perfect whenever the surfaces to be reconstructed are fronto-parallel to the stereo rig ($\Delta = 0$).

Assume that the points $\mathbf{P}$ and $\mathbf{Q}$ also lie on the same scene plane $\mathbf{\Omega} \sim \begin{pmatrix}\mathbf{m} & -l\end{pmatrix}^{\mathsf{T}}$ that defines a homography M, similar to Equation 2.5, mapping points in the right view into points in the left view. Following this, $\mathbf{q}=\mathsf{M}\mathbf{q}'$ and it can be shown that

$$d_q = \frac{m_1 b}{l} q_1 + \frac{m_2 b}{l} q_2 + \frac{-m_1 b q_1 - m_2 b q_2 + l d_q}{l}.$$

Since $\mathbf{p}$ is also the projection of the same planar surface, by applying the homography M comes that Δ_p differs from Δ_q by

$$\Delta = \alpha_1 (p_1 - q_1).$$

where

$$\alpha_1 = \frac{m_1 b}{l} \tag{2.9}$$

is proportional to the slant of the plane along the horizontal direction. Replacing in Equation 2.8 comes that

$$\widehat{g} = \left(\frac{\beta}{\beta - 1}\right)(1 - \alpha_1) g. \tag{2.10}$$

The conclusion that can be drawn is that having prior knowledge about the position and orientation of the surface to be reconstructed, we can determine the point of intersection between the virtual plane $\mathbf{\Pi}$ and the baseline that grants perfect induced symmetry. The image signals are perfectly symmetric whenever $\widehat{g} = -g$, so that solving with respect to β in Equation 2.10 yields

$$\beta = \frac{1}{2 - \alpha_1}. \tag{2.11}$$

Following the previous analysis, and in case there is slant information available a priori, we suggest a simple approach for refining the SymStereo depth estimates. Referring to Figure 2.6, we start by applying a virtual cut plane $\mathbf{\Pi}_i$ intersecting

the baseline in its midpoint $O_1 = 0.5b$, from which the 3D point $\mathbf{Q}$ is estimated. Assume that $\mathbf{Q}$ lies on the plane $\mathbf{\Omega}$, whose horizontal slant defines a particular direction α_1 (Equation 2.9). Using Equation 2.11, we can determine the position on the baseline $0_1^1 = \beta^1 b$ that a new virtual cut plane should intersect for enhancing the quality of the induced symmetries. This new vertical virtual cut plane $\mathbf{\Pi}_i^1$ is defined by the points $\mathbf{0}^1$ and $\mathbf{Q}$, from which a refined 3D estimation $\mathbf{Q}^1$ can be computed. Following this, the overall quality of the 3D points obtained using SymStereo can be iteratively refined by selecting appropriate virtual planes intersecting specific points on the baseline.

2.5 Mapping Π into a plane Γ in the DSI domain

In the same manner that a fronto-parallel plane $\mathbf{\Psi}$ induces a constant disparity d, a virtual cut plane $\mathbf{\Pi}$ defines a pixel association between views that corresponds to a particular surface $\mathbf{\Gamma}$ in the DSI domain (see Figure 2.1). Consider the inverse of the plane homography given by Equation 2.5. The transformation H^{-1} enables to map points $\mathbf{q}$ in the left image into points $\mathbf{q}'$ in the right image, such that

$$q_1' \;=\; (1 + \frac{bn_1}{h})\, q_1 + \frac{bn_2}{h}\, q_2 + \frac{bn_3}{h}\,. \tag{2.12}$$

It can be verified that the cut plane $\mathbf{\Pi}$ defines for each point $\mathbf{q}$ a putative disparity $d \;=\; q_1 - q_1'$ given by

$$d \;=\; -\frac{bn_1}{h}\, q_1 - \frac{bn_2}{h}\, q_2 - \frac{bn_3}{h}$$

The equation above specifies a plane in the 3D space parametrized by $(q_1,\, q_2,\, d)$. Thus, the matching hypotheses defined by $\mathbf{\Pi}$ (Equation 2.2) correspond to a plane $\mathbf{\Gamma}$ in the DSI domain, with homogeneous representation

$$\mathbf{\Gamma} \;\sim\; \begin{pmatrix} \frac{bn_1}{h} \\ \frac{bn_2}{h} \\ 1 \\ \frac{bn_3}{h} \end{pmatrix}\,. \tag{2.13}$$

2.6 Sweeping the scene by a pencil of vertical virtual planes bisecting the baseline

As stated previously, dense stereo matching with SymStereo requires using multiple virtual cut planes $\mathbf{\Pi}_i$ such that the corresponding planes $\mathbf{\Gamma}_i$ completely sweep the DSI domain. Assume that the planes $\mathbf{\Pi}_i$ belong to a vertical pencil with the axis intersecting the midpoint of the baseline. In this case, the homogeneous representation of each plane is given by

$$\mathbf{\Pi}_i \sim \begin{pmatrix} 1 \\ 0 \\ -\tan(\theta_i) \\ \frac{b}{2} \end{pmatrix},$$

with θ_i denoting the rotation angle around the vertical axis, and the plane homography of Equation 2.2 becomes

$$\mathsf{H}_i \sim \begin{pmatrix} -1 & 0 & 2\tan(\theta_i) \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}.$$

Consider now that the points $\mathbf{q}$ and $\mathbf{q}'$ are expressed in pixel coordinates, and that both cameras have the same intrinsic parameters

$$\mathsf{K} \sim \begin{pmatrix} f & 0 & c_1 \\ 0 & f & c_2 \\ 0 & 0 & 1 \end{pmatrix}.$$

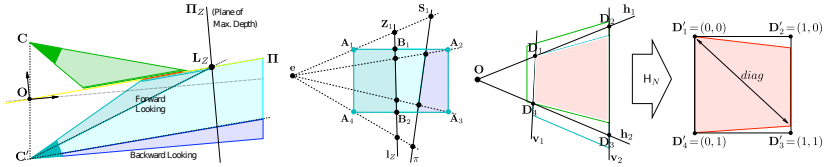
The homography mapping $\mathbf{q} \sim \mathsf{K}\,\mathsf{H}_i\,\mathsf{K}^{-1}\,\mathbf{q}'$ defines a possible pixel association between images that can be written as

$$q_1 = \underbrace{2\,c_1 - q_1'}_{flip} + \lambda_i\,, \tag{2.14}$$

with

$$\lambda_i = 2\,f\,\tan(\theta_i)\,.$$

Moreover, and from the discussion of Section 2.5, each virtual cut plane $\mathbf{\Pi}_i$ corre-



(a) Top view (b) Image of $\mathbf{C}'$ (c) Normalization using H_N

Figure 2.7: Back-projection onto a virtual cut plane $\mathbf{\Pi}$ with arbitrary normal orientation. (a) top view where the scene is assumed to lie between the cameras and the plane of maximum depth $\mathbf{\Pi}_Z$. The entire left image is considered an interest region because it back-projects in the area between the origin $\mathbf{O}$ and the line $\mathbf{L}_Z$. In the case of view $\mathbf{C}'$, the right image side is back-projected behind $\mathbf{O}$, while the middle part of the image is back-projected beyond $\mathbf{L}_Z$. Thus, the interest region is limited to the top most side (b). The search for the contour where $\mathbf{\Pi}$ cuts the scene surfaces needs only to be carried in the polygon of intersection of the left and right interest regions (red). (c) The alignment of the epipolar lines and the definition of a suitable tessellation are achieved by inscribing the search polygon into a unit square using the homography H_N.

sponds to a plane $\mathbf{\Gamma}_i$ in the DSI domain with homogeneous coordinates

$$\mathbf{\Gamma}_i \sim \begin{pmatrix} 2 \\ 0 \\ -1 \\ -2\,c_1 - \lambda_i \end{pmatrix}. \tag{2.15}$$

Two important conclusions can be drawn. The first is that the range of disparities in the DSI domain is fully covered by a set of planes $\mathbf{\Gamma}_i$ such that the parameters λ_i take successive integer values. This enables to choose the angles θ_i that define a suitable set of virtual planes $\mathbf{\Pi}_i$ in the 3D scene space. The second is that the homography mapping of Equation 2.14 considerably simplifies the rendering of images $\widehat{\mathtt{I}}_i$ required for generating the symmetries and anti-symmetries (see Figure 2.3). The warping can be efficiently achieved by flipping the original image $\mathtt{I}'$ around the vertical axis passing through the principal point, followed by shifting the result by an integer amount λ_i along the horizontal image direction.

2.7 Generating the symmetry/anti-symmetry images in the 3D virtual cut plane

In the previous sections, the homography induced by a vertical virtual cut plane $\mathbf{\Pi}$ intersecting the baseline was used for mapping the right view onto the left view for generating the symmetry and anti-symmetry signals. This section briefly explains how to explicitly back-project the input images onto the 3D virtual cut planes with

arbitrary orientation. This is important for future developments of the proposed framework, e.g. Multi-View SymStereo, where it might be interesting/necessary to back-project the views directly on the 3D planes.

Without lack of generality, assume a maximum value for the scene depth, which means that the profile cut $\mathcal{C}$ must lie on the area spreading between $\mathbf{O}$ and line $\mathbf{L}_z$ where the plane of maximum depth meets $\mathbf{\Pi}$ (Figure 2.7(a)). Thus, for each image we can define an interest region by the following steps (refer to Figure 2.7(b)):

1. Determine lines $\mathbf{l}_z$ and π ($\mathbf{l}'_z$ and π') by projecting $\mathbf{L}_Z$ and the line at infinity using the homography $\mathsf{H}_{\mathbf{C}}$ ($\mathsf{H}_{\mathbf{C}'}$) that relate the virtual plane $\mathbf{\Pi}$ with the left and right view, respectively

2. For each image corner $\mathbf{A}_i$, consider the line defined by the corner and the epipole $\mathbf{e}$, and determine the intersections $\mathbf{Z}_i$ and $\mathbf{S}_i$ with $\mathbf{l}_z$ and π

3. If the cross-ratio $\{\mathbf{Z}_i, \mathbf{e}; \mathbf{A}_i, \mathbf{S}_i\}$ is negative, then the corner $\mathbf{A}_i$ is in the interest region, otherwise it is outside; if the cross-ratios are all positive then the interest region is empty, if the cross-ratios are all negative then the interest region is the entire image, otherwise the interest region is the polygon defined by the corners $\mathbf{A}_i$ with negative cross-ratio and the intersections of $\mathbf{l}_z$ with the image borders.

The profile cut C can only be recovered if it is simultaneously seen in both views. Thus, the search region can be further constrained by back-projecting the boundaries of the bottom and top interest regions onto $\mathbf{\Pi}$ and finding their intersection polygon (Figure 2.7(c)). Mapping the polygon back into the input views yields the image regions that must be warped.

Two issues remain: (i) the epipolar lines are not vertically aligned, which can complicate subsequent processing and (ii) a uniform plane tessellation does account for the original image resolution, causing a magnification that increases with depth. We address these problems by rectifying the back-projections using a normalizing transformation H_N. H_N is a projective transformation on the cut plane $\mathbf{\Pi}$ that inscribes the search polygon in an unitary square as shown by Figure 2.7(c). Lines $\mathbf{h}_1$ and $\mathbf{h}_2$, that join the origin $\mathbf{O}$ with the top and bottom vertex of the polygon, are mapped into the top and bottom sides of the square. This grants that epipolar lines become vertically aligned. Lines $\mathbf{v}_1$ and $\mathbf{v}_2$ are chosen so that the transformed polygon is enclosed by the square and has maximum area. The resolution of the tessellation is determined by averaging the pixel length of the diagonal $diag$ that is mapped back in the two stereo images.

2.8 Conclusions

This chapter presented the first work in the literature proposing to use symmetry instead of photo-similarity for assessing the likelihood of two image locations be-

ing a match. Stereo from symmetry is possible because of the mirroring effect that arises whenever one view is mapped into the other using the homography induced by a virtual cut plane that intersects the baseline. We provided a formal proof of this effect, studied the singularities, and investigated its usage for solving the data association problem in stereo.

Chapter 3

SymStereo for dense matching and Stereo-Rangefinding

The SymStereo framework proposes to associate pixels across views by jointly using symmetry and anti-symmetry measurements. This chapter introduces metrics for quantifying symmetry and anti-symmetry that are used for matching pixels. We show through extensive experiments that symmetry-based metrics outperform photo-similarity metrics for the purpose of data association in dense stereo. Moreover, and since the symmetries are induced using virtual cut planes, these new matching functions are particularly well suited for recovering depth along a single scan plane. This is an effective way of probing into the 3D structure resulting in profile cuts of the scene that resemble the ones obtained with a 2D Laser-Rangefinder. The independent estimation of depth along a scan plane will be referred as SRF. The results confirm that, also in this case, symmetry-based matching costs are the top-performer.

3.1 Introduction

Dense stereo matching is a mature research topic and the literature reports a large number of matching functions. Following the taxonomy used in [45], the matching costs can be broadly divided into four types. *Pixel-wise matching costs*, like Absolute Differences (AD), measure the dissimilarity between single pixels, being popular because of their simplicity and fast computation. However, pixel-wise metrics tend to be ambiguous even when used in conjunction with local aggregation methods, e.g. SAD. Since pixel-wise matching functions do not make implicit assumptions about the image neighborhood surrounding the pixel, they are mostly

used for evaluating the DSI in global stereo approaches. In this case, the sampling-insensitive metric proposed by Birchfield-Tomasi (BT) is usually preferred to a straightforward AD implementation. BT computes the absolute difference between the pixel of interest in one view and a linear interpolation of the neighborhood of the hypothesized match in the other view [65]. A pre-processing step that significantly improves the stereo matching performance of BT is Bilateral Background Subtraction (BBS) that smooths the images without blurring the depth discontinuities [66].

Window-based matching costs evaluate the similarity (or dissimilarity) between 2D regions in the stereo images. Normalized Cross-Correlation (NCC) is an example of this type of matching functions that is widely used because of its good trade-off between accuracy and computational efficiency. ZNCC is a variant of NCC that compensates for gains and offsets [45].

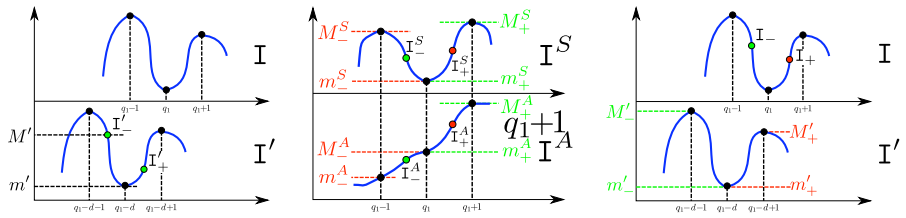
Non-parametric matching costs use the ordering of image intensities in a local neighborhood around the pixels of interest. The most popular metric of this type is probably the Census filter introduced in [67]. The approach consist in constructing a bit string where each bit corresponds to a pixel in a local neighborhood around the pixel of interest $\mathbf{q}$. The bit is set iff the pixel intensity value is lower than the intensity of $\mathbf{q}$. The filtered images are compared by computing the Hamming distance between corresponding bit strings.

Lastly, *Mutual Information* computed from the entropy of the input images can also be used as a stereo matching cost [57]. The idea is to transform views according to the disparity assignment such that the mutual information between the transformed images is maximized.

Several works in stereo have benchmarked not only competing matching costs [68, 23, 69, 70, 71, 45], but also cost aggregation methods [23, 70, 72, 54, 73, 55] and global optimization schemes [23, 70, 56]. In this chapter, we are only interested in the formers, among which the work of Hirschmuller and Scharstein [45] is of special relevance because of its systematic methodology and thorough evaluation using images of the Middlebury dataset [23, 44]. In their evaluation, each cost function gives rise to the DSI that leads to a final disparity map after using local aggregation, SGM or a straightforward Markov Random Field formulation with Graph-Cut (GC) optimization. The results show that BT with BBS, ZNCC, and Census are, respectively, the top-performers among pixel-wise, window-based, and non-parametric matching costs. In absolute terms, Census proved to have the best matching performance throughout the evaluation.

Following the discussion of the previous chapter, the objective of SymStereo is to associate pixels across views by jointly using symmetry and anti-symmetry measurements. This chapter proposes techniques for quantifying local symmetry and anti-symmetry for matching pixels. In Section 3.3, we use the same methodology of [45] for comparing our symmetry-based matching costs against BT with BBS, ZNCC and Census, in an effort to show that symmetry can be more effective than photo-similarity for solving the dense stereo problem.

Finally, and since the symmetries are induced using virtual cut planes, the



(a) BT (b) SymBT (c) Efficient SymBT

Figure 3.1: The SymBT metric: In (a) the standard BT cost compares value of pixel q_1 in the reference view against the intensity range $[m', M']$ around the putative match $q_1 - d$. The scheme (b) illustrates how SymBT quantifies the symmetry and anti-symmetry along the epipolar lines of $\mathtt{I}^S$ and $\mathtt{I}^A$. Given a particular pixel location q_1, the idea is to use the BT metric to compare the interpolated intensity value on one side against the intensity interval on the other side. Finally (c) shows how the SymBT metric can be efficiently implemented without requiring the explicit rendering of the image signals $\mathtt{I}^S$ and $\mathtt{I}^A$.

new framework is particularly well suited for recovering depth along pre-defined scan planes. This is an effective way of probing into the 3D structure resulting in profile cuts of the scene that resemble the ones obtained with a 2D Laser-Rangefinding (LRF). The independent estimation of depth along a scan plane will be referred as Stereo-Rangefinding (SRF) in order to be distinguished from conventional dense stereo. We show in Section 3.4 through extensive comparative experiments that symmetry-based metrics outperforms photo-similarity for the purpose of SRF. Moreover, we compare in Section 3.5 the depth estimates obtained using SRF with the ones provided by a LRF. As will be shown, SRF can be a plausible alternative to LRF in several application scenarios.

3.2 Measuring local symmetry and anti-symmetry

This section discusses techniques for quantifying local signal symmetry and anti-symmetry at every pixel of $\mathtt{I}^S$ and $\mathtt{I}^A$. We describe three alternative metrics: *SymBT* that adapts the famous BT matching cost for measuring signal asymmetry instead of dissimilarity [65]; *SymCen* that is a non-parametric symmetry metric inspired in the Census transform [67]; and *logN* that has been originally proposed by Kovesi in [74], and uses a bank of N log-Gabor wavelets for evaluating local symmetry.

3.2.1 SymBT

Consider a pair of corresponding epipolar lines in the stereo images $\mathtt{I}$ and $\mathtt{I}'$, and let d be a putative disparity value that associates pixel q_1 in $\mathtt{I}$ with pixel $q_1 - d$ in $\mathtt{I}'$. The matching likelihood can be inferred by measuring the dissimilarity between

$\mathtt{I}(q_1)$ and $\mathtt{I}'(q_1 - d)$. In order to avoid sampling issues, Birchfield and Tomasi (BT) suggest to compare the intensity value $\mathtt{I}(q_1)$ in the reference view against a brightness interval $[m',\, M']$ around the putative image correspondence $\mathtt{I}'(q_1 - d)$ in the second view [65]. This is illustrated in Figure 3.1(a), where the boundaries of the intensity range are

$$\begin{aligned} m' &= \min\left(\mathtt{I}'(q_1 - d);\, \mathtt{I}'_-;\, \mathtt{I}'_+\right) \\ M' &= \max\left(\mathtt{I}'(q_1 - d);\, \mathtt{I}'_-;\, \mathtt{I}'_+\right), \end{aligned}$$

with $\mathtt{I}'_-$ and $\mathtt{I}'_+$ being interpolated brightness values at the sub-pixel locations around $q_1 - d$. The dissimilarity between $\mathtt{I}(q_1)$ and $\mathtt{I}'(q_1 - d)$ is quantified by

$$C = \max\left(0;\, \mathtt{I}(q_1) - M';\, m' - \mathtt{I}(q_1)\right).$$

Considering now that $\mathtt{I}'$ is the reference view, it comes in a similar manner that

$$C' = \max\left(0;\, \mathtt{I}'(q_1 - d) - M;\, m - \mathtt{I}'(q_1 - d)\right),$$

where

$$\begin{aligned} m &= \min\left(\mathtt{I}(q_1);\, \mathtt{I}_-;\, \mathtt{I}_+\right) \\ M &= \max\left(\mathtt{I}(q_1);\, \mathtt{I}_-;\, \mathtt{I}_+\right), \end{aligned}$$

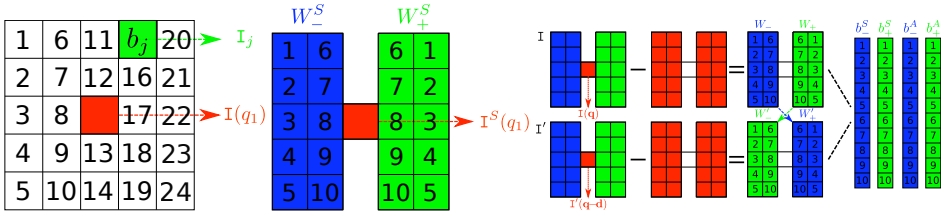
The final BT score handles the two views symmetrically and is given by

$$C_{BT}(q_1,\, d) = \min\left(C;\, C'\right)$$

3.2.1.1 Modifying BT to measure asymmetry

Inspired by the BT cost, we can define a metric for measuring asymmetry along the epipolar lines of the image signal $\mathtt{I}^S$ that is invariant to sampling issues. Let $\mathtt{I}^S_-$ and $\mathtt{I}^S_+$ be interpolated image values in the neighborhood of a particular pixel location q_1 in $\mathtt{I}^S$ (see Figure 3.1(b)). The 1-D image signal symmetry can be evaluated by verifying if the sub-pixel image value in one side of q_1 is within the brightness interval in the opposite side. Thus, we propose to quantify the asymmetry of the image signal $\mathtt{I}^S$ about the pixel location q_1 by

$$D^S_{BT} = \max\left(0, \mathtt{I}^S_- - M^S_+;\, m^S_+ - \mathtt{I}^S_-;\, \mathtt{I}^S_+ - M^S_-;\, m^S_- - \mathtt{I}^S_+\right),$$



(a) Census trasnform (b) SymCen (c) Efficient SymCensus

Figure 3.2: The SymCen transform. In (a) the standard Census transform defines a bit string $\mathbf{b}$ for each image point q_1, with each bit b_j corresponding to a particular pixel in a local patch centered in q_1. In (b) SymCen is used to quantify the signal symmetry in $\mathtt{I}^S$ by comparing the regions W_-^S and W_+^S on both sides of q_1. In (c) the SymCen is implemented without requiring the explicit rendering of $\mathtt{I}^S$ and $\mathtt{I}^A$. The bit strings b_-^S, b_+^S, b_-^A and b_+^A are computed by performing simple operations over W_-, W_+, W'_- and W'_+.

with

$$m_\pm^S = \min\left(\mathtt{I}^S(q_1);\, \mathtt{I}^S(q_1 \pm 1)\right)$$
$$M_\pm^S = \max\left(\mathtt{I}^S(q_1);\, \mathtt{I}^S(q_1 \pm 1)\right).$$

A similar approach can be used for scoring the anti-symmetry of the image signal $\mathtt{I}^A$ at particular pixel locations. Consider the scheme in the bottom of Figure 3.1(b), where $\mathtt{I}_-^A$, $\mathtt{I}_+^A$ are the interpolated image values at sub-pixel locations, and $[m_-^A,\, M_-^A]$, $[m_+^A,\, M_+^A]$ are the brightness intervals defined above. It is easy to understand that, if the image signal is anti-symmetric about q_1, then the following must hold:

$$\mathtt{I}^A(q_1) + (\mathtt{I}^A(q_1) - \mathtt{I}_-^A) \in [m_+^A,\, M_+^A]$$
$$\mathtt{I}^A(q_1) + (\mathtt{I}^A(q_1) - \mathtt{I}_+^A) \in [m_-^A,\, M_-^A].$$

Thus, we can modify the asymmetry score defined above for quantifying lack of signal anti-symmetry about q_1

$$\begin{aligned} D_{BT}^A = \quad & \max\left(0;\, 2\mathtt{I}^A(q_1) - \mathtt{I}_-^A - M_+^A;\, m_+^A - 2\mathtt{I}^A(q_1) + \mathtt{I}_-^A;\right. \\ & \left. \ldots 2\mathtt{I}^A(q_1) - \mathtt{I}_+^A - M_-^A;\, m_-^A - 2\mathtt{I}^A(q_1) + \mathtt{I}_+^A\right). \end{aligned}$$

Finally, the SymBT score for finding pixel locations that are simultaneously symmetric in $\mathtt{I}^S$ and anti-symmetric in $\mathtt{I}^A$ is defined as:

$$D_{BT}(q_1) = \max\left(D_{BT}^S;\, \mathsf{D}_{BT}^A\right). \tag{3.1}$$

3.2.1.2 Efficient Implementation

The SymBT metric described in the previous section has the inconvenient of requiring the explicit rendering of the image signals $\mathtt{I}^S$ and $\mathtt{I}^A$ for each considered virtual cut plane. As discussed in Section 2.6, a particular choice of cut plane implicitly assigns points q_1 in $\mathtt{I}$ to points $q_1 - d$ in $\mathtt{I}'$. It is now shown how to compute the SymBT score for a particular matching hypothesis $(q_1,\, d)$ without having to explicitly render the image signals $\mathtt{I}^S$ and $\mathtt{I}^A$. Let's consider the scheme of Figure 3.1(c) where $\mathtt{I}_-$, $\mathtt{I}_+$ are interpolated image values in the neighborhood of the pixel location q_1 in $\mathtt{I}$, and $[m'_-,\, M'_-]$, $[m'_+,\, M'_+]$ are the brightness intervals on the sides of the putative correspondence $q_1 - d$ in $\mathtt{I}'$. The metric S evaluates till which extent $\mathtt{I}_-$ and $\mathtt{I}_+$ are within the ranges $[m'_-,\, M'_-]$ and $[m'_+,\, M'_+]$, respectively.

$$\begin{aligned} S_- &= \max\left(0, \mathtt{I}_- - M'_-;\, m'_- - \mathtt{I}_-\right) \\ S_+ &= \max\left(0, \mathtt{I}_+ - M'_+;\, m'_+ - \mathtt{I}_+\right) \\ S &= S_- + S_+. \end{aligned}$$

Considering now that $\mathtt{I}'$ is the reference view, it comes in a similar manner that

$$\begin{aligned} S'_- &= \max\left(0, \mathtt{I}'_- - M_-;\, m_- - \mathtt{I}'_-\right) \\ S'_+ &= \max\left(0, \mathtt{I}'_+ - M_+;\, m_+ - \mathtt{I}'_+\right) \\ S' &= S'_- + S'_+. \end{aligned}$$

Finally, the SymBT score is given by

$$S_{BT}(q_1, d) = \max(\mathsf{S}, \mathsf{S}'). \tag{3.2}$$

It is important to note that Equation 3.1 and Equation 3.2 are not strictly equivalent. However, we verified experimentally that the metric of Equation 3.1 provides similar results than the metric of Equation 3.2, while avoiding the rendering of $\mathtt{I}^S$ and $\mathtt{I}^A$.

3.2.2 SymCen

The Census transform is a non-parametric filter that analyzes the differences between image intensity values in a $m \times n$ neighborhood around the pixel of interest. For illustration purposes consider a 5×5 patch centered in a pixel location denoted by q_1, and let $\mathtt{I}_j$ be the image intensity values for the entries j in this patch ($j = 1, \ldots, 24$) as shown in Figure 3.2(a)). The output of the Census transform is

a string $\mathbf{b}$, with 24 bits, where each bit b_j is set as follows:

$$b_j = \begin{cases} 1 & \text{if } \mathtt{I}(q_1) > \mathtt{I}_j \\ 0 & \text{if } \mathtt{I}(q_1) \leq \mathtt{I}_j \end{cases} . \tag{3.3}$$

Considering that the pixel q_1 in $\mathtt{I}$ corresponds to pixel $q_1 - d$ in $\mathtt{I}'$, we build a second bit string $\mathbf{b}'$ encoding the intensity values around $q_1 - d$ and compute the Census dissimilarity as

$$C_C(q_1, d) = H(\mathbf{b};\mathbf{b}') ,$$

with H denoting the Hamming distance.

3.2.2.1 Modifying Census to measure dissymmetry

Figure 3.2(b) shows how the Census transform can be used to quantify symmetry instead of dissimilarity. In this case, the 5×5 neighborhood is divided into two 5×2 regions, W_-^S and W_+^S, that are respectively in the left and right sides of the pixel of interest. The intensity values of the two patches are encoded in the bit strings $\mathbf{b}_-^S$ and $\mathbf{b}_+^S$ using Equation 3.3, and a new bit string is computed which describes the symmetry of the image signal $\mathtt{I}^S$ about the pixel location q_1

$$\mathbf{b}^S = (\mathbf{b}_-^S == \mathbf{b}_+^S),$$

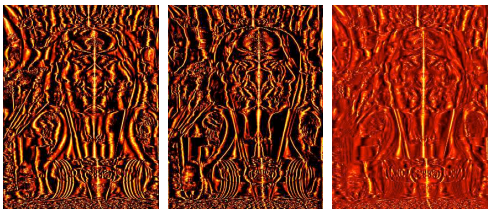
where $==$ is the bitwise equality operator. The anti-symmetry in image $\mathtt{I}^A$ can be encoded in a similar manner by

$$\mathbf{b}^A = (\mathbf{b}_-^A == \bar{\mathbf{b}}_+^A),$$

where $\mathbf{b}_-^A$ is the bit string of the left side region W_-^A, and $\bar{\mathbf{b}}_+^A$ is the binary complement of the bitstring of the right side patch W_+^A. The final SymCen score for the pixel q_1 is obtained by comparing corresponding symmetry and anti-symmetry bits $\mathbf{b}_j^S$ and $\mathbf{b}_j^A$, and then summing all the bit responses:

$$S_C(q_1) = \sum_j \mathbf{b}_j^S \& \mathbf{b}_j^A, \tag{3.4}$$

where $\&$ is the bitwise *and* operator. Remark that different from the Census metric, larger values of the SymCen cost correspond to higher matching likelihood.



(a) $\mathtt{E}^S$ (b) $\mathtt{E}^A$ (c) $\mathtt{E} = \mathtt{E}^S \cdot \mathtt{E}^A$

Figure 3.3: The logN metric: (a) is the symmetry energy $\mathtt{E}^S$ of the image signal $\mathtt{I}^S$, while (b) is the anti-symmetry energy $\mathtt{E}^A$ of image $\mathtt{I}^A$. The final joint energy $\mathtt{E}$ in (c) is obtained by pixel-wise multiplication of $\mathtt{E}^S$ and $\mathtt{E}^A$.

3.2.2.2 Efficient Implementation

The bit strings b^S_-, b^S_+, b^A_-, and b^A_+, required for evaluating the SymCen cost of Equation 3.4, can be directly computed from the stereo pair $\mathtt{I}$ and $\mathtt{I}'$ as shown in Figure 3.2(c). Let W_- and W_+ be the patches on both sides of pixel q_1 in the reference view $\mathtt{I}$, and W'_- and W'_+ be the patches around the putative correspondence $q_1 - d$ in the secondary view $\mathtt{I}'$. Subtract $\mathtt{I}(q_1)$ to the intensity values in regions W_- and W_+. Repeat the procedure in the secondary view using $\mathtt{I}'(q_1 - d)$. It can be proved that the bit strings for evaluating the score S_C can be determined as follows:

$$\begin{aligned} \mathbf{b}^S_- &= T(W_-;\, -W'_-) \\ \mathbf{b}^S_+ &= T(W_+;\, -W'_+) \\ \mathbf{b}^A_- &= T(W_-;\, W'_+) \\ \mathbf{b}^A_+ &= T(W_+;\, W'_-) \end{aligned}$$

with T being an operator that compares the intensity values of corresponding pixels in two patches W and W', generating a bit string with the j^{th} bit being given by

$$T_j(W;\, W') = \begin{cases} 1 & \text{if } \mathtt{I}_j > \mathtt{I}'_j \\ 0 & \text{if } \mathtt{I}_j \leq \mathtt{I}'_j \end{cases} .$$

This alternative scheme for computing the SymCen score has the obvious advantage of avoiding the explicit rendering of image signals $\mathtt{I}^S$ and $\mathtt{I}^A$, which substantially decreases the computational complexity.

3.2.3 logN

Kovesi shows that an intensity distribution that is symmetric about a particular pixel location gives rise to specific phase patterns in the Fourier series of the image signal [74]. Thus, he proposes to detect symmetry and anti-symmetry based on frequency information obtained using a bank of log-Gabor filters. This section describes the

joint application of Kovesi's algorithms with the SymStereo framework, leading to a new stereo matching cost that is referred as *logN*, with N standing for the number of wavelet scales that are considered for the signal analysis.

Since the log-Gabor wavelets are analytical signals, the image filtering is carried in the spectral domain. Let $\mathcal{G}_k$, with $k = 1, \ldots N$, be the frequency response of the pre-selected wavelet scales, and $\mathcal{I}^S$ be the spectrum of a generic epipolar line $\mathtt{I}^S(q_1)$ in the symmetry image (see Figure 2.3(d)). The filtering result is the following 1D complex signal

$$s_k^S(q_1) + \mathbf{i}\, a_k^S(q_1) = F^{-1}(\mathcal{I}^S \cdot \mathcal{G}_k)\,, \tag{3.5}$$

with F denoting the Fourier transform and $\mathbf{i}^2 = -1$. It can be shown that, if the image is symmetric about the pixel location q_1, then the real component $\mathtt{s}_k^S$ takes high values, while the imaginary component $\mathtt{a}_k^S$ takes small values [74]. Therefore, and given the N wavelet scale responses, we can establish the following energy of symmetry:

$$\mathtt{E}^S(q_1) \;=\; \frac{\sum_{k=1}^{N} \mid s_k^S(q_1) \mid - \mid a_k^S(q_1) \mid}{\sum_k \sqrt{\left(s_k^S(q_1)\right)^2 + \left(a_k^S(q_1)\right)^2}}, \tag{3.6}$$

where the normalization by the sum of the magnitudes provides invariance to changes in illumination [74]. Figure 3.3(a) shows the result of stacking the lines $\mathtt{E}^S(q_1)$ arising from each row of image $\mathtt{I}^S$ of Figure 2.3(d). It can be observed that the highlights correspond to pixel locations where the image signals is symmetric along the horizontal direction.

Consider now the anti-symmetric image $\mathtt{I}^A$ of Figure 2.3(e), we can use the same bank of wavelets and compute

$$s_k^A(q_1) + \mathbf{i}\, a_k^A(q_1) = F^{-1}(\mathcal{I}^A \cdot \mathcal{G}_k)\,. \tag{3.7}$$

By applying a similar approach for deriving an energy of anti-symmetry yields

$$\mathtt{E}^A(q_1) \;=\; \frac{\sum_{k=1}^{N} \mid a_k^A(q_1) \mid - \mid \; s_k^A(q_1) \mid}{\sum_k \sqrt{\left(s_k^A(q_1)\right)^2 + \left(a_k^A(q_1)\right)^2}}\,. \tag{3.8}$$

The resulting energy $\mathtt{E}^A$ is depicted in Figure 3.3(b), with the locations of image anti-symmetry being emphasized.

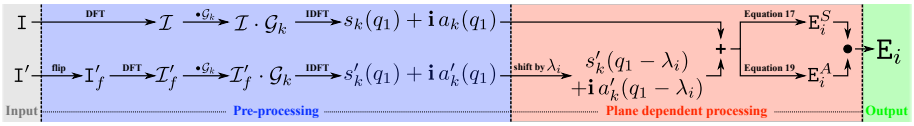


Figure 3.4: Efficient implementation of the *logN* stereo matching cost. In a first step the rectified stereo pair is filtered by the considered wavelet scales $\mathcal{G}_k$ in order to obtain the left and right complex signals $s_k(q_1) + \mathbf{i}\, a_k(q_1)$ and $s'_k(q_1) + \mathbf{i}\, a'_k(q_1)$ with $k = 1, 2 \ldots N$. In a second stage, and for each scale k, the right-side signal is shifted by an amount λ_i, which depends on the virtual cut plane $\mathbf{\Pi}_i$, and the result is added and subtracted to the left-side signal. The operation provides the input coefficients for computing the symmetry and anti-symmetry energies of Equations 3.6 and 3.8, ultimately leading to the energy $\mathtt{E}_i$.

Both $\mathtt{E}^S$ and $\mathtt{E}^A$ have several local maxima along the horizontal lines, which preclude a straightforward detection of the image of the profile cut $\mathcal{C}$, that is overlaid in Figure 2.3(d) and Figure 2.3(e). Since points in $\mathcal{C}$ must be simultaneously local maxima in $\mathtt{E}^S$ and $\mathtt{E}^A$, the pixel-wise multiplication of the two energies enables to discard most spurious detections. Thus, we consider the following joint energy $\mathtt{E}$

$$\mathtt{E} = \mathtt{E}^S \cdot \mathtt{E}^A \tag{3.9}$$

where the image of the contour $\mathcal{C}$ is clearly distinguishable as shown in Figure 3.3(c)

3.2.3.1 Efficient implementation

The joint energy $\mathtt{E}$ is computed from the images $\mathtt{I}^S$ and $\mathtt{I}^A$, which are rendered for a particular virtual cut plane $\mathbf{\Pi}$. As discussed in Section 2.6, each plane $\mathbf{\Pi}_i$ in the scene gives rise to a plane $\mathbf{\Gamma}_i$ in the DSI that is function of an integer parameter λ_i (see Equation 2.15). As discussed in this section, the energy $\mathtt{E}$ can be computed without explicitly rendering the image signals $\mathtt{I}^S$ and $\mathtt{I}^A$, and the evaluation of *logN* across the entire DSI domain can be carried very efficiently.

Let $\mathtt{I}^S(q_1)$ be the 1D signal arising from a generic epipolar line in the symmetry image $\mathtt{I}^S$. If $\mathtt{I}(q_1)$ and $\mathtt{I}'(q_1)$ are the corresponding lines in the rectified stereo pair, then it follows from Equation 2.14 that:

$$\begin{aligned} \mathtt{I}^S(q_1) &= \mathtt{I}(q_1) + \widehat{\mathtt{I}}(q_1) \\ &= \mathtt{I}(q_1) + \mathtt{I}'_f(q_1 - \lambda)\,, \end{aligned}$$

where λ is a shift amount that depends on the choice of the virtual plane $\mathbf{\Pi}$, and $\mathtt{I}'_f$ is a horizontally flipped version of the right image

$$\mathtt{I}'_f(q_1) = \mathtt{I}'(2c_1 - q_1)\,.$$

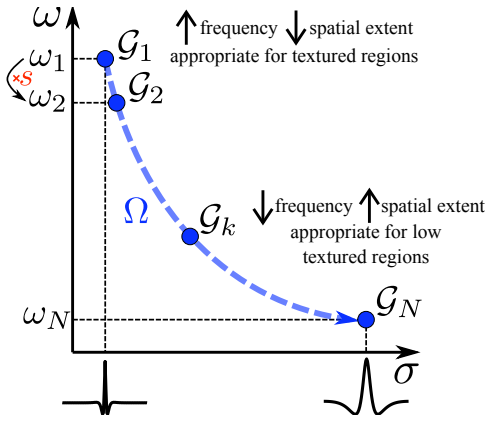


Figure 3.5: (Qualitative) space-frequency behavior of the log-Gabor wavelets $\mathcal{G}_k$. The horizontal axis refers to the spatial support σ of the filter kernel, while the vertical axis concerns the response frequency ω.

From the reasoning above, and exploring the linear properties of the Fourier transform, it comes that Equation 3.5 can be re-written as:

$$\begin{aligned} s_k^S(q_1) + \mathbf{i}\, a_k^S(q_1) = &\big(s_k(q_1) + s'_k(q_1 - \lambda)\big) \\ &+ \mathbf{i}\, \big(a_k(q_1) + a'_k(q_1 - \lambda)\big)\,, \end{aligned}$$

with

$$\begin{cases} s_k(q_1) + \mathbf{i}\, a_k(q_1) &= F^{-1}(\mathcal{I} \cdot \mathcal{G}_k) \\ s'_k(q_1) + \mathbf{i}\, a'_k(q_1) &= F^{-1}(\mathcal{I}'_f \cdot \mathcal{G}_k) \end{cases},$$

where $\mathcal{I}$ and $\mathcal{I}'_f$ stand for the Fourier transform of $\mathtt{I}(q_1)$ and $\mathtt{I}'_f(q_1)$, respectively. The response of Equation 3.7 for the anti-symmetric image signal $\mathtt{I}^A(q_1)$ can be computed in a similar manner by

$$\begin{aligned} s_k^A(q_1) + \mathbf{i}\, a_k^A(q_1) = &\big(s_k(q_1) - s'_k(q_1 - \lambda)\big) \\ &+ \mathbf{i}\, \big(a_k(q_1) - a'_k(q_1 - \lambda)\big)\,. \end{aligned}$$

Figure 3.4 is a schematic of the computation pipeline for obtaining the energy $\mathtt{E}_i$ for a particular choice $\mathbf{\Pi}_i$ of virtual cut plane. The new formulation avoids the explicit rendering of the symmetric and anti-symmetric images, but also enables to efficiently evaluate the entire DSI by simply varying the shifting amount λ_i with $i = 1, 2 \ldots M$.

3.2.3.2 Selection of wavelet scales

The choice of the log-Gabor wavelets for filtering the input images has a strong influence in the final stereo estimations. Despite of the fact that log-Gabor filters are analytical signals with no real representation in the spatial domain, the scheme of Figure 3.5 tries to provide an intuition about how the wavelet parameters relate with the space-frequency response of the filter. The horizontal axis refers to the spatial extent or support of the filter kernel, while the vertical axis concerns the frequency components of the image signal to which $\mathcal{G}_k$ responds. If the image region is very textured, then it is advisable to operate in the top-left corner of the $(\omega,\, \sigma)$ plane, and choose filters with high-frequency response and small space extent. On the other hand, if the image region is textureless, then we must consider wavelets that respond to low-frequency components, but that have a larger support which tends to diminish the pixel accuracy of the analysis.

As discussed in [75], the bank of log-Gabor wavelets $\mathcal{G}_k$ is usually parametrized by the shape-factor Ω, the center frequency of the mother wavelet ω_1, the scaling step s, and the total number N of wavelets. The shape-factor Ω can be related with the filter bandwidth, and defines a contour in the $(\omega,\, \sigma)$ domain containing the wavelets that can be selected (see Figure 3.5). The center frequency ω_1, together with the shape factor Ω, defines uniquely the first wavelet scale $\mathcal{G}_1$. The scaling step s sets the distance between the center frequencies of successive wavelet scales k and $k+1$ along the contour. In this chapter, we have manually set $\Omega = 0.55$, $\omega_1 = 0.25$, and $s = 1.05$, and kept these values constant throughout the entire set of experiments. The only parameter that is allowed to vary is the number of scales N that controls the ability of obtaining response in low textured image regions by using filters with a larger spatial support.

3.3 Experiments in dense stereo matching

We proposed in this chapter three matching costs - SymBT, SymCen, and logN - that use symmetry instead of photo-consistency for accomplishing data association. This section runs a set of experiments in dense stereo matching for comparing symmetry-based stereo with respect to state-of-the-art matching costs.

3.3.1 Methodology and tuning of parameters

Since the stereo literature is vast, it is virtually impossible to compare SymStereo against every possible method and approach. Thus, and in order to assure a rigorous and conclusive study, the evaluation herein presented follows the methodology and takes into account the results of the benchmark of Hirschmüller and Scharstein [45]. We compare three symmetry-based matching costs against the cost functions that were considered to be top-performers in [45]. These stereo cost functions are:

- *Birchfield-Tomasi (BT)* quantifies pixel dissimilarity by comparing 1-Dimensional (1D) neighborhoods defined along the epipolar lines [65]. According to [45], the BT metric combined with Bilateral Background Subtraction (BBS) [66] provides the best matching results among pixelwise parametric costs.

- *Zero-mean Normalized Cross-Correlation (ZNCC)* considers a 2-Dimensional (2D) support region for quantifying photo-similarity, and proved to be the a top-performer among window-based parametric matching costs.

- *Census* is a window-based non-parametric cost function [67] that consistently proved to be the top similarity measure for dense disparity estimation.

The evaluation is carried using stereo pairs with ground truth disparity that include challenging situations, e.g. slanted surfaces, low and repetitive textures. As in [45], most experiments are performed using the Middlebury dataset [23, 44, 45] but, while they run the benchmarking in 6 image pairs, we consider a set of 15 examples that covers a wider range of situations (see Figure 3.8). For each cost function under analysis, we build the DSI of the different image pairs, estimate the corresponding disparity maps using a particular stereo method, and score the estimation result by counting the number of pixel locations in non-occluded regions with a disparity error greater than one. The matching costs under benchmark are ranked by averaging the error score across all stereo pairs in the test set. Since the focus is in evaluating the performance of matching costs, the disparity estimation must be carried by the exact same stereo method for all costs in order to assure fair comparison. As in [45], we present results using three distinct approaches:

- *Local Aggregation* aggregates the DSI by summing the costs over a window and each image pixel is assigned with the disparity value that has the lowest cost.

- *Semi-Global Matching (SGM)* minimizes a 2D energy by solving multiple 1D minimization problems [57].

- *Graph-Cut (GC)* estimates a disparity map by global minimization of an energy function defined in the DSI using graph-cuts [76, 77, 78, 56].

GC and SGM are formulated in the standard manner, and post-processing steps, e.g. left-right consistency check or sub-pixel interpolation, are not considered.

It can be argued that using local aggregation is better suited for comparing different matching costs than using SGM or GC. It is a fact that global and semi-global methods, being more sophisticated techniques, can eventually hide issues and weaknesses of the cost function. Although we agree that local aggregation provides the most relevant benchmarking information, this section also presents the scores obtained with SGM and GC for the sake of completeness and to assure full compliance with the methodology and results described in [45].

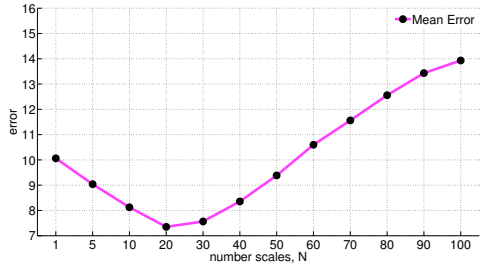


Figure 3.6: Tuning the number of wavelets scales N for dense stereo using the standard Middlebury dataset. The figure plots the average error in disparity estimation using local aggregation when N increases.

Table 3.1: Summary of the parameters used in the experiments throughout the article in Dense Stereo (DS), Stereo Rangefinder (SRF), and wide-baseline (WB) images.

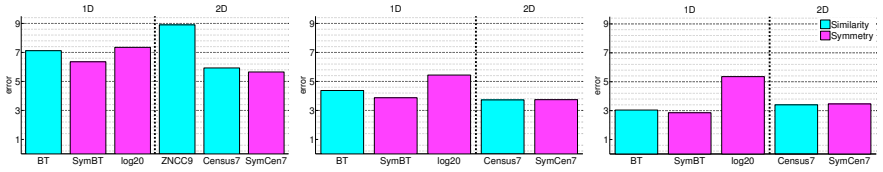
	DS (Sec. 3.3)	**DS-WB** (Sec. 3.3)	**SRF** (Sec. 3.4)	**SRF-WB** (Sec. 3.4)
BT	1×3	1×3	1×3	1×3
SymBT	1×3	1×3	1×3	1×3
*log***N**	20	50	40	70
*ZNCC***M**	9×9	7×7	15×15	9×9
*Census***H**	9×7	9×19	9×19	9×23
*SymCen***H**	9×7	9×19	9×9	9×23

It can also be argued that choosing adaptive-weight aggregation [79] instead of standard aggregation improves the disparity estimation in image regions that are close to depth discontinuities. This is true, but it is important to keep in mind that such improvements are transverse to all matching costs and do not necessarily change the relative disparity scores.

Finally, for every matching cost under study, the computation of the DSI is carried in C++ assuming input images with approximate size 460×370 and disparity range of 64 pixels. The C++ implementations are straightforward and only use the standard code optimizations described in the literature.

3.3.1.1 Tuning of parameters

As in [45], the parameters are manually tuned using the standard Middlebury dataset [23], which comprises the images *Tsukuba*, *Venus*, *Teddy* and *Cones* (from top to bottom in Figure 6.5). These pairs are not considered latter in the benchmark to avoid bias effects. Whenever applicable, we use the optimal values reported in [45], this is, the local aggregation window is 9×9, the ZNCC window is 9×9, and the Census window is 9×7. In order to allow a direct comparison between Census and SymCen, we also consider a window of 9×7 for the second. As shown in Figure 3.6, the number of wavelet scales to be used with logN is set to $N = 20$. As expected, increasing N does not necessarily improve the performance because low frequency wavelets have wider space support that decreases the accuracy of the



(a) Local Aggregation (b) SGM (c) GC

Figure 3.7: Result after tuning the parameters: the figure plots the percentage of errors in dense disparity estimation across the images of the standard Middlebury dataset.

disparity estimation (see Figure 3.5). For the case of BT and SymBT, we always apply bilateral filtering and consider a 3 pixel neighborhood. Table 3.1 summarizes the choice of parameters for this and the following section. For the latter experiments wide-baseline stereo and SRF, we will re-tune the window size of ZNCC, the horizontal window size of Census and SymCen, and the number of scales of logN.

After tuning the cost functions assuming local aggregation, we move to the setting of the parameters for SGM and GC that will be used with each matching cost. The tuning is carried by selecting the parameter values that provide the smallest percentage of disparity errors in the images of the standard dataset. These errors are plotted in Figure 3.7 where it can be observed that the results for BT, ZNCC, and Census are close to the ones reported in [45].

3.3.2 Tests in Middlebury

The matching costs are compared by analyzing the errors in dense disparity estimation in the Middlebury images of Figure 3.8. Figure 3.9 shows the mean of the percentage of pixels with incorrect disparity label for a particular combination of matching cost and stereo method. The first observation is that pixel-based 1D metrics tend to perform worse than window based 2D costs. This is to expect because most surfaces in the Middlebury dataset have moderate or no slant. More important is the fact that the symmetry-based metrics, SymBT and SymCen, consistently beat their similarity-based counterparts, BT and Census. Thus, the experimental evidence clearly suggests that the symmetry cues are more effective than the standard photo-consistency measurements for matching pixels across views.

It can also be observed that log20 has an erratic behavior ranking differently according to the stereo method that is considered. For the case of local aggregation, it is the most inaccurate metric among the 1D matching costs, although it performs significantly better than ZNCC. Apparently the use of global minimization changes the ranking of relative performances, with log20 becoming respectively the best and second best pixel-based cost function when combined with SGM and GC. The reasons for this behavior require a more detailed analysis of the experimental data. For this purpose, the input set is divided into two subsets:

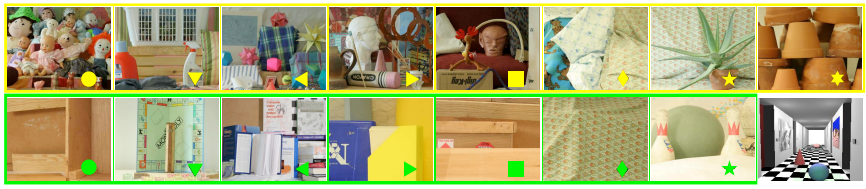
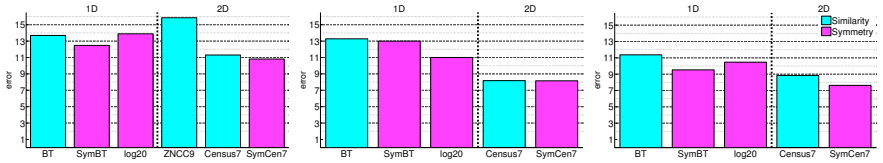
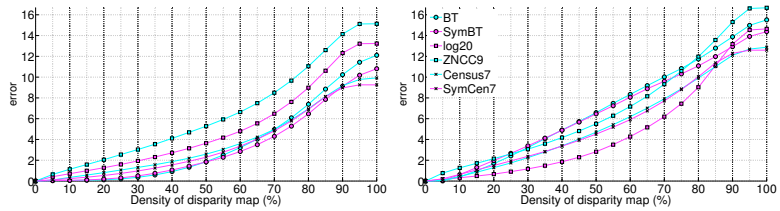


Figure 3.8: The stereo pairs that are used as input for the experiments of Sections 3.3 and 3.4. The benchmark is carried in 15 images of the Middlebury dataset [44, 45]. The top row shows the Set I comprising frames with several objects and depth discontinuities. The bottom row exhibits the Set II consisting in scenes dominated by continuous surfaces with low or repetitive texture. The image in the bottom right corner refers to the Oxford Corridor that is used in Section 3.3.3 for evaluating the performance in case of surface slant.



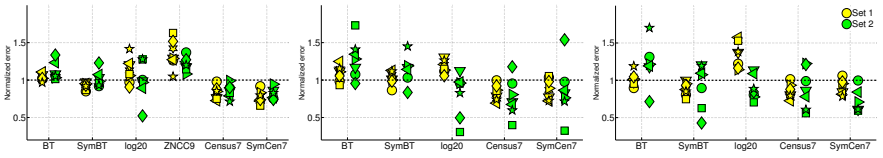
(a) Local Aggregation (b) SGM (c) GC

Figure 3.9: Average percentage of disparity errors in the dense disparity maps of the 15 images of the Middlebury dataset (Set I + Set II).



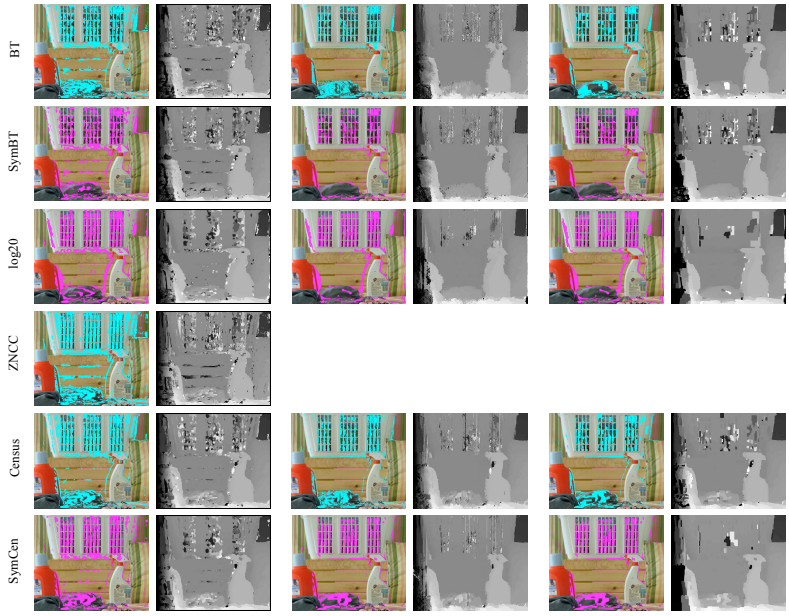
(a) Set I (b) Set II

Figure 3.10: Average percentage of disparity errors in the semi-dense disparity maps of Set I (a) and Set II (b) obtained by selecting the first $L\%$ matches with lowest cost [80].



(a) Local Aggregation (b) SGM (c) GC

Figure 3.11: The number of disparity errors for each input image normalized by the average number of errors across all matching costs [45].



(a) Local (b) SGM (c) GC

Figure 3.12: Overlay of the disparity errors (left) and disparity map (right) in the *Laundry* example for every possible combination of matching cost (rows) and stereo method (columns). Remark that there is no post-processing step after local stereo aggregation.

1. *Set I:* comprises the images with many objects and surface discontinuities (yellow in Figure 3.8).
2. *Set II:* contains the images that are dominated by large surfaces that mostly present poor or repetitive texture (green in Figure 3.8).

The estimation in the two sets is analyzed using the criterion introduced in [80] that tests the ability of a matching cost to rank the matches according to their reliability. After using local aggregation for the dense disparity labeling, the pixel locations are sorted in ascending order of cost, and a semi-dense disparity map is obtained by selecting the first $L\%$ pixels for which the matching confidence is higher. Figure 3.10 shows the mean percentage of errors in the semi-dense disparity estimation for increasing values of L. Looking to the scores for $L = 100\%$, it can be seen that all matching costs perform worse in Set II than in Set I, suggesting that the former dataset is more challenging than the latter. It can also be observed that SymBT and SymCen behave equal or better than BT and Census, respectively, for all levels of completeness L. The most striking difference between the two plots is the fact that log20 has the second worse reliability performance in the images of Set I, but it is clearly the most accurate matching cost for a completeness up to $L = 85\%$ in Set II, only loosing the advantage in the disparity labeling of the last 15% of pixels with highest cost scores. It happens that these pixels are usually located close to discontinuities and/or occlusion regions, suggesting that log20 is

Table 3.2: Runtime for evaluating the DSI assuming 375×450 images and a disparity range of 64 pixels.

Match. Cost	**Time (ms)**	**Match. Cost**	**Time (ms)**
BT (+BBS)	120 (+296)	*SymBT (+BBS)*	170 (+296)
Census7	160	*SymCen7*	185
ZNCC9	3200	*log20*	3900

Table 3.3: The left column shows how complexity scales with respect to image size $L{\times}W$, disparity range D, window size $l \times w$ or number of wavelet scales N. The right column reports the number of addition or subtraction (B), and comparison (C) operations required for evaluating each matching cost. We do not provide the last information for the case of logN and ZNCC because the analysis is difficult to carry and the result cannot be directly compared.

Match. Cost	**Big** O	Operations
BT	$O(LWD)$	$LWD \times (8B{+}11C)$
SymBT	$O(LWD)$	$LWD \times (14B{+}15C)$
Census	$O(LWDlw)$	$LWlw{\times}(2C) + LWDlw{\times}(1C)$
SymCen	$O(LWDlw)$	$LWl(w{-}1)/2{\times}(2B) + LWDl(w{-}1)/2{\times}(2B{+}4C)$
*log***N**	$O(LW(\log(W)N{+}D))$	
ZNCC	$O(LWDlw)$	

very effective in estimating the disparity along the continuous surfaces with low or repetitive texture, but has more difficulty than other matching costs in handling the depth discontinuities. This can also explain the improvements of log20 in the ranking of relative performances that were observed in Figure 3.9. Since the pixels in the continuous surfaces have lower cost values at the correct disparities, they have a stronger regularization effect during the SGM and GC minimizations that leverages the depth estimation close to the discontinuities.

Figure 3.11 shows, for each stereo pair and matching cost, the error normalized by the mean error over all matching functions [45]. The objective of the plots is to provide a perspective about the relative performance of the different matching cost in a particular input image. The results show that log20 always compares well for the images of Set II confirming the hypothesis that, despite of being a 1D matching cost, it is specially effective in scenes dominated by large surfaces with low and/or repetitive texture. It can also be seen that SGM and GC boost the relative accuracy of log20 in Set II but not in Set I, which is in accordance with the interpretation that the improvements in the ranking of Figure 3.9 are because of the low cost values at correct pixel disparities observed in Figure 3.10(b).

Figure 3.12 shows the disparity errors in the *Laundry* example. It is interesting to observe that SymBT and SymCen tend to outperform BT and Census in the continuous regions, while presenting similar performance close to discontinuities. In general the log20 is very accurate in the continuous surfaces, proving to be resilient to low and repetitive textures, but the error regions are considerably larger close to depth discontinuities and occlusions.

Table 3.2 summarizes the runtimes for evaluating the DSI of the *Teddy* stereo

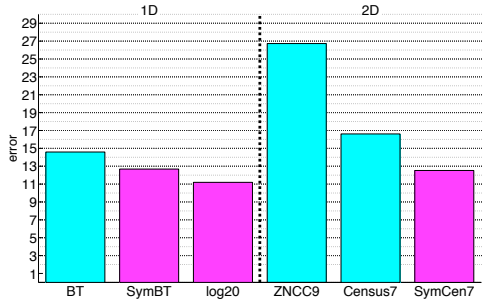
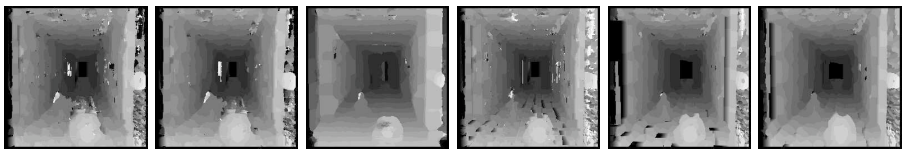


Figure 3.13: Percentage of disparity errors in the dense disparity map of the *Oxford Corridor*. The estimation was carried after local aggregation with a 9×9 window.

.



(a) BT (b) SymBT (c) log20 (d) ZNCC (e) Census (f) SymCen

Figure 3.14: Disparity maps obtained for each matching cost on the *Oxford Corridor*. Remark that there is no post-processing step after local stereo aggregation.

.

pair using the different matching functions, while Table 3.3 analyzes the computational complexity (*Big O* notation) and the principal operations required during the evaluation. As stated previously, BT and SymBT are always evaluated in a 1×3 region, while for the case of Census, SymCen and ZNCC we generalize the computational complexity analysis for a window of size $l \times w$. In general, the symmetry-based matching functions require more operations, but the magnitude of additional effort does not preclude the possibility of real-time dense disparity estimation, largely justifying the observed improvements in accuracy.

3.3.3 Tests in Oxford Corridor

Figure 3.13 shows the percentage of disparity errors for the *Oxford Corridor* that is exhibited in the bottom-right corner of Figure 3.8, while Figure 3.14 displays the disparity maps obtained using the different matching costs. The disparity estimation is carried by a WTA strategy after local aggregation of the DSI using a 9×9 window. The relative performance of the matching functions differs from the one observed in the equivalent experiment using the Middlebury dataset (see Figure 3.9(a)). First, for the *Oxford Corridor* the 1D matching costs outperform the 2D functions because now the scene is dominated by highly slanted surfaces. Second, the differences in accuracy between symmetry and similarity-based matching

functions are more striking in Figure 3.13 than in Figure 3.9(a). with the log20 being the top-performing metric. This is explained by the fact that most textures in the *Oxford Corridor* are either flat, e.g. the walls, or repetitive, e.g. the checkerboard pattern of the floor. Thus, the results of this experiment seem to confirm that the symmetry-based costs in general, and the logN metric in particular, are specially well suited for estimating the disparity in continuous regions with low or repetitive texture and high slant, clearly beating the similarity-based counterparts.

3.3.4 Experiments in wide-baseline stereo

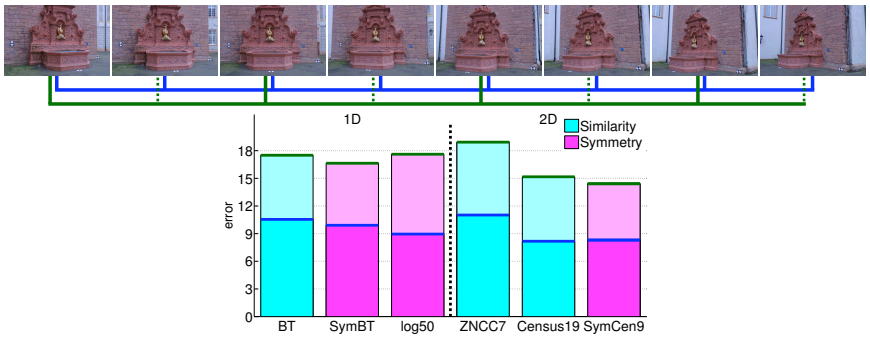


Figure 3.15: Mean errors on the fountain-P11 dataset [1]. The top row shows the 8 input images, while the bottom row shows the results of the different matching costs for dense stereo matching across the different image combinations (i) middle-baseline (blue), and (ii) wide-baseline (green).

In order to complete the evaluation for dense stereo, this section compares the performance of the matching functions in wide-baseline images. We consider the 8 frames of the fountain-P11 dataset [1] that are exhibited in the top row of Figure 3.15. The sequence gives rise to 7 *medium-baseline* examples, corresponding to pairwise consecutive frames, and 6 *wide-baseline* examples obtained by pairing the frames with one image interval. We randomly select one of the stereo pairs for tuning the matching functions, and later discard the example for the evaluation. The selected parameters are shown in the 3^{th} column of Table 3.1. The disparity range r is set by the minimum and maximum of the ground truth disparity maps for images with size 440×640, and the threshold e for deciding about the correctness of the disparity labeling is chosen such that the ratio e/r is the same as in Section 3.3.2.

The bottom plot of Figure 3.15 shows the percentage of errors for dense disparity labeling in *medium-baseline* and *wide-baseline* stereo pairs. The relative performance of the matching functions is in accordance with the observed in the previous sections, suggesting that all the conclusions drawn up to now hold for the case of wide-baseline imagery.

3.4 Experiments in Stereo-Rangefinding (SRF)

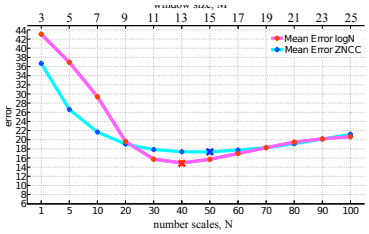
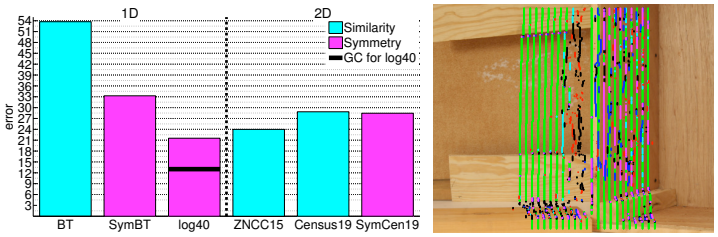


Figure 3.16: Tuning of parameters for SRF: average percentage of errors in the standard Middlebury dataset for logN and ZNNC when the spatial support increases. The disparity labeling is independently carried for each virtual cut plane by a WTA approach after local aggregation using a 9×1 window.



(a) Average percentage of disparity errors (b) log40 vs ZNCC15

Figure 3.17: Benchmark of the cost functions for Stereo-Rangefinding (SRF): (a) average percentage of disparity errors in the 15 Middlebury images of Figure 3.8 for the 6 matching costs; (b) disparity errors in the *Wood1* example when using log40 and ZNCC15. The disparity labeling is independently carried for each virtual cut plane by a WTA approach after local aggregation using a 9×1 window. The overlay refers to the image of the mirroring contour where green is correct estimation of both (log40 and ZNCC15), black is wrong detection of both, magenta and blue means log40 is correct and ZNCC15 is wrong, respectively, whereas red and cyan means log40 is wrong and ZNCC15 is correct, respectively.

Stereo-Rangefinding (SRF) consists in using passive stereo for estimating depth along a virtual cut plane (scan plane) in order to reconstruct the contour $\mathcal{C}$ where the plane meets the scene. As discussed previously, SRF enables a trade-off between runtime and 3D model resolution that, as we will see, does not interfere with depth accuracy. This section evaluates the performance of the matching functions for the purpose of SRF. Henceforth, we will only present the disparity estimation results obtained using local aggregation.

3.4.1 Methodology and tuning of parameters

From Section 2.6 follows that a virtual cut plane $\mathbf{\Pi}_i$ intersecting the baseline corresponds to a plane $\mathbf{\Gamma}_i$ in the DSI domain. While dense stereo evaluates the matching function for the entire DSI, SRF only considers the disparity hypotheses corresponding to 3D points lying in $\mathbf{\Pi}_i$, meaning that the cost is exclusively evaluated along the plane $\mathbf{\Gamma}_i$ in the DSI. In our experiments, the scores in $\mathbf{\Gamma}_i$ are locally aggregated using a vertical 9×1 window (no horizontal aggregation), and a disparity label is assigned to each epipolar line using WTA. Since the winning labels must always occur in the pixel locations where the profile cut $\mathcal{C}$ is projected, the number of errors in SRF is determined by counting the winners that are more than 1 pixel apart from the ground truth image contour (see Figure 2.3).

The performance of the matching functions is benchmarked by averaging the results obtained in the 15 Middlebury images of Figure 3.8. In each case, the scene depth is independently estimated along 201 vertical cut planes $\mathbf{\Pi}_i$ with uniformly distributed rotation angles θ_i (see Section 2.6). The objective of using such a large number of cut planes is to cover a broad range of possible SRF situations, with $\mathbf{\Pi}_i$ either intersecting the scene in a continuous surfaces or passing nearby a depth discontinuity. As in the dense stereo experiments, the parameters of the matching functions are manually tuned using the standard Middlebury dataset. Figure 3.16 plots the average percentage of errors for logN and ZNCC in case of increasing number of scales and window size, respectively. The choice of parameters is summarized in the second column of Table 3.1, where a comparison with dense stereo shows that SRF benefits from computing the matching costs across a wider pixel neighborhood. This is not surprising if we take into account that the larger image patches tend to compensate the fact that the aggregation is only carried in the 1D-vertical direction.

3.4.2 Tests in Middlebury

Figure 3.17(a) shows the percentage of disparity errors averaged across the 15 image pairs of Figure 3.8. Comparing with the dense stereo results of Figure 3.9, it comes that the disparity estimation in SRF is less accurate for all matching functions. The higher percentage of errors is justified by the fact that SRF uses less information than dense stereo for the disparity labeling, since it only evaluates and aggregates the cost along a plane $\mathbf{\Gamma}_i$ in the DSI domain. The second observation is that symmetry-based matching costs still outperform their similarity-based counterparts, with SymBT and SymCen19 having less 20% and 1% of errors than BT and Census, respectively. The relative lower performance of the BT family is largely due to the fact that the scores are computed across a small 3-pixel neighborhood, which seems to be an insufficient image support for handling the lack of horizontal aggregation. Finally, ZNCC15 is the most accurate metric among the similarity-based matching functions, but it is beaten by log40 that presents 4% less errors. The figure also shows the accuracy of log40 when the local aggregation is

(a) Strength (b) Weakness

Figure 3.18: Pros and cons of logN. Figures (a) and (b) show the symmetry images $\mathtt{I}^S$ for particular choices of $\mathbf{\Pi}_i$. The overlay refers to the image of the mirroring contour where blue is the ground truth, green is correct estimation and red wrong detection. The logN matching function performs well in low textured and slanted surfaces (a) but fails in flat regions close to depth discontinuities (b). In (b) the edge of the foreground object induces an apparent symmetry that misleads the logN estimation.

.

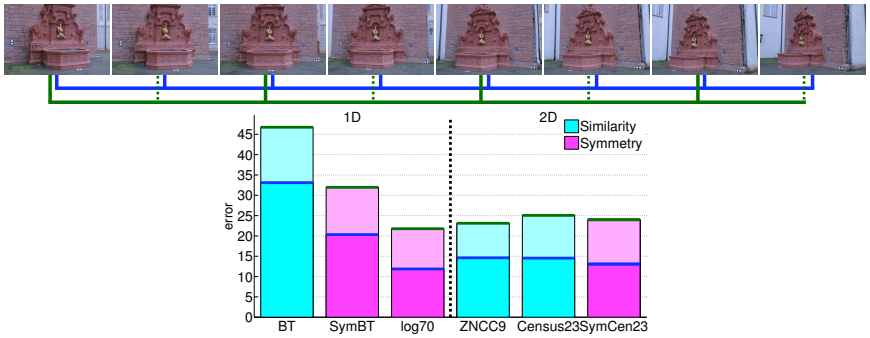
replaced by global optimization using a standard MRF formulation that enforces continuity in the mirroring contour. The error percentage becomes 13% which is about 5.8% more than the best result observed for dense stereo (SymCen7 with GC), and just 2% more than the best result accomplished with log20 (log20 with SGM).

Figure 3.17(b) compares the performance of logN and ZNCC in the *Wood1* stereo pair by overlaying the results in detecting the mirroring contours for the 201 virtual cut planes. It can be observed that the latter, being a 2D metric with a large window support, has difficulties in handling depth discontinuities (e.g. errors in the horizontal depth transition at the top of the image, and in the occlusion region at the image center) and surface slant (e.g. errors in the boards lying on the floor). On the other hand, logN seems to combine the benefits of being a pixel-based matching cost, with a good discriminative power for pairing pixels in low textured regions. This is illustrated in Figure 3.18(a) that shows the symmetry image $\mathtt{I}^S$ for a virtual cut plane that meets the scene in the vertical wooden board with significant slant. Since the pixel matching is accomplished using symmetry, the lack of local texture is partially compensated by nearby structures, such as edges and wood nodes that contribute to successfully detect the image of the mirroring contour. Thus, the good performance in the presence of low texture is explained by the global character of the induced symmetry cue. However, and as exemplified by the situation of Figure 3.18(b), such global character can become an issue whenever the contour passes in a flat region close to a depth discontinuity. In this case, the edge of the foreground object gives raise to an apparent image symmetry in the wrong location that, together with the absence of background texture, completely misleads the logN detection. It is also this phenomena that explains the poor performance of log20 close to discontinuities and occlusion regions during the dense stereo experiments (e.g. see the third column of Figure 3.12). The problem can eventually

Table 3.4: Runtime of SRF measured in the *Teddy* stereo pair. The column t_{oh} refers to the initialization overhead whenever applicable, and the column t_{Π} reports the time for estimating disparity along a single virtual cut plane. The total time for processing K independent profile cuts is given by $t = t_{oh} + K \cdot t_{\Pi}$.

Cost	t_{oh} (ms)	t_{Π} (ms)	**Cost**	t_{oh} (ms)	t_{Π} (ms)
BT	98	0.42	*SymBT*	98	0.60
Census19		32	*SymCen19*		33
ZNCC15		39	*log40*	378	13

be solved by using local texture information for selecting the wavelet scales at each pixel location, however the development of such a strategy was beyond the work of this research.



(a) Profile Cut Stereo

Figure 3.19: Mean errors on the fountain-P11 dataset [1]. The top row shows the 8 input images, while the bottom row shows the results of the different matching costs for SRF across the different stereo combinations (i) middle-baseline (blue), and (ii) wide-baseline (green).

Table 3.4 provides the average runtime for estimating the depth along a single virtual cut plane using SRF. Since the BBS filtering in BT and SymBT, and the spectral convolution in logN are executed only once independently of the number K of profile cuts, the workload required by these one time operations is accounted as an initialization overhead t_{oh}. The table shows that for $K = 1$ logN is about $10\times$ slower than Census, SymCen and ZNCC, but a quick calculation shows that for $K \geq 20$ the former becomes faster than the laters. Remark that there is no linear relationship between the runtimes of Tables 3.2 and 3.4 based on the number of image columns. The reasons are that the matching costs in SRF have larger window support and the scoring along a single plane in the DSI domain does not benefit from an efficient memory management.

3.4.3 Experiments in wide-baseline stereo

This section evaluates the performance of the matching functions when the input image pairs have a wide-baseline. As in the previous section concerning dense stereo matching, we use the fountain-P11 dataset [1] for the evaluation (see top row of Figure 3.19). The sequence gives rise to 7 *medium-baseline* and 6 *wide-baseline* examples. The selected parameters for SRF are shown in the 4^{th} columns of Table 3.1. Since the images are larger than the dataset used in Section 3.4, the scene depth is independently estimated along 401 vertical cut planes. The bottom plot of Figure 3.19 shows the percentage of errors for SRF in *medium-baseline* and *wide-baseline* images. Also in this case, the relative performance of the matching functions is in accordance with the observed in Figure 3.17(a) for the case of short-baseline stereo.

3.5 Stereo-Rangefinding vs. Laser-Rangefinding

There are many applications in robotics that make simultaneous use of visual data and laser-scans e.g. [35, 81, 82, 83]. Laser-Rangefinding is popular because it enables accurate depth measures in real-time, being effective under most operating conditions. On the other hand, passive vision is an extremely versatile sensor modality, providing rich image information. Replacing two sensor modalities by a single one without sacrificing skills or system capabilities is an appealing proposition. This would lead to savings in equipment with a positive impact in the overall cost of the final system. It is unlikely that LRF can ever replace passive vision without losses in versatility and system capabilities. Fortunately, the opposite seems much more feasible specially in cases where two calibrated cameras are available. As discussed in Chapter 2, stereo vision enables 3D reconstruction by associating pixels across images. Thus, it is plausible that it can succeed in estimating depth along a scan plane with an accuracy close to LRF.

This section is motivated by the possibility of replacing LRF by stereo vision in robotic applications. We compare the depth estimates obtained with SRF against range data acquired by a LRF. The experiment clearly shows the strengths and weaknesses of each technology.

3.5.1 Experimental Setup

We briefly introduce the experimental setup for the synchronous acquisition of stereo images and range data. The setup combines a 2D LRF with two perspective cameras for which the specifications are provided in Table 3.5. The sensors are mounted on a rigid mobile platform with the laser placed between the cameras as shown in Figure 3.20. The camera baseline is around $45cm$ and the distance between the top camera and the laser is roughly $19.5cm$. The cameras are not

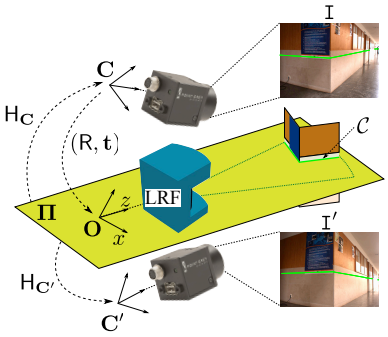


Figure 3.20: Experimental setup. The top camera $\mathbf{C}$ (reference view) points down, the bottom camera $\mathbf{C}'$ points up, and the LRF is positioned between the cameras. The LRF recovers the profile cut $\mathcal{C}$ that is the contour where the scan plane $\mathbf{\Pi}$ meets the scene structure. $\mathcal{C}$ is projected in the two images $\mathtt{I}$ and $\mathtt{I}'$ using the extrinsic calibration.

Table 3.5: Specifications of the camera and the LRF

Camera		LRF	
Manufacturer	Point-Grey	Manufacturer	Sick
Baseline	≈ 45 cm	Model	LMS200
Resolution	1280×960	Horiz. Res.	0.25^o

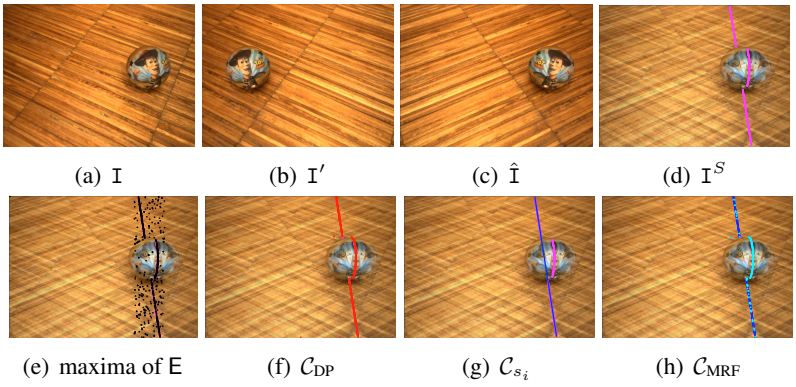
aligned, and $\mathbf{C}$ is the reference view.

Referring to Figure 3.20, the stereo cameras are calibrated using Bouguet's calibration toolbox [84], and the relative pose between the LRF and the reference camera $\mathbf{C}$ is estimated using the minimal solution proposed by Vasconcelos et al. [85]. This enables to determine the homogeneous representation of the scan plane $\mathbf{\Pi}$ in the stereo coordinate system, and compute the homographies $\mathsf{H}_{\mathbf{C}}$ and $\mathsf{H}_{\mathbf{C}'}$ that accurately map range data into images $\mathtt{I}$ and $\mathtt{I}'$, respectively.

3.5.2 Detection of the profile cut

In order to compare the depth estimates obtained using SRF and the readings provided by the LRF, the virtual cut plane of SymStereo is aligned with the known scan plane of the LRF. It is important to remark that, contrary to the properties described in Section 2.6, the homography induced by the cut plane, in this case, is not a simple flipping and shifting of the input images. Following the experimental results of the previous section, we select the logN matching cost, being the top-performer in SRF. It is also important to note that, given the particular stereo configuration of the experimental setup described in the previous section, the epipolar lines have vertical orientation. This section considers the traditional setup, where the stereo cameras are horizontally aligned, but the algorithm generalizes for any calibrated stereo setup.

A naive approach for locating the profile contour $\mathcal{C}$, would be to simply se-



(a) $\mathtt{I}$ (b) $\mathtt{I}'$ (c) $\hat{\mathtt{I}}$ (d) $\mathtt{I}^S$

(e) maxima of E (f) $\mathcal{C}_{\text{DP}}$ (g) $\mathcal{C}_{s_i}$ (h) $\mathcal{C}_{\text{MRF}}$

Figure 3.21: Example of the estimation of the profile cut location in a traditional stereo setup (baseline has horizontal displacement); magenta is ground truth. The global optimization correctly decides between straight (blue) and non-straight (cyan) points on the profile cut.

lect the maxima along each column in the joint energy E computed using logN. However, and as shown in Figure 3.21, this would lead to noisy estimates. This section proposes to improve the estimation of the profile cut by considering two soft constraints for SRF.

3.5.2.1 Dynamic Programming (DP)

We use a simple optimization approach for obtaining a binary labeling for $\mathtt{I}$, where each epipolar line has only one pixel y set to one, corresponding to the most likely pixel location lying in the image of the profile cut. This is accomplished using a straightforward Dynamic Programming (DP) approach [23]. The algorithm computes the maximum cost path in E, where the energy for each pixel $\mathbf{p}$ is re-defined as:

$$\mathsf{D}(p_1,p_2)=\mathsf{E}(p_1,p_2)+\min_{y}\left(\mathsf{D}(p_1-1,y)+V_{\text{DP}}(p_2,y)\right), \tag{3.10}$$

with V_{DP} being a smoothness term given by

$$V_{\text{DP}}(p_2,y)=\begin{cases}\frac{\lambda_{\text{DP}}}{\Delta I^S} & \text{if } |p_2-y|>0\\ 0 & \text{otherwise}\end{cases},$$

$\Delta I^S = |I^S(p_1,p_2) - I^S(p_1-1,y)|$ and λ_{DP} is a constant parameter. The binary labeling is accomplished by selecting for each column p_2, the pixel $\mathbf{p}$ with maximum cost D. Thus, we obtain a contour corresponding to a possible location of the profile cut expressed in discrete terms. In order to refine the contour estimation and obtain sub-pixel precision, we fit to each p_1 a parabola in E around the neighborhood of p_2. The output of this step is the contour $\mathcal{C}_{\text{DP}}$ (see Figure 3.21).

3.5.2.2 Line detection using the Hough transform

In order to detect straight lines in $\mathtt{I}$, which correspond to the intersection of $\mathbf{\Pi}$ with a plane in the scene, a weighted Hough Transform is applied to the joint energy E. We extract at most N_{HT} line parameters $\mathbf{s}_i$, where $i = 1, \ldots, N_{HT}$. Figure 3.21 shows the contours $\mathcal{C}_{\mathbf{s}_i}$, obtained from the intersection of $\mathbf{s}_i$ with each epipolar line of $\mathtt{I}$ ($C_{\mathbf{s}_i}(x)$ is the point of intersection between $\mathbf{s}_i$ and the epipolar line x).

3.5.2.3 MRF for straight and non-straight profile cut labeling

Given a particular epipolar line, there are $N_{HT}+1$ possible locations for the profile cut, N_{HT} corresponding to the extracted lines segments $\mathcal{C}_{s_i}$, and one corresponding to the estimation $\mathcal{C}_{\text{DP}}$ using DP. In order to decide which one is the most suitable point on each epipolar line, we formulate the decision as a labeling problem in a MRF. Following the notation used in Section 5.3.1, the objective is to assign to each image row $d \in \mathcal{D}$, a label f_d in the set $\mathcal{L}$, which is the union of all line segment labels $f_{\mathbf{s}_i}$ and the non-straight label f_{DP}. The energy to minimize is given by:

$$E = \sum_{d \in \mathcal{D}} D(f_d) + \lambda_{\text{MRF}} \sum_{d \in \mathcal{N}} V_{d,e}(f_d, f_e).$$

The data function is defined as:

$$D_d(f) = \begin{cases} -\left(\mathsf{E}_d(f) + \gamma_S(1 - \mathsf{S}_d(f))\right) & \text{if } f = f_{\text{DP}} \\ -\mathsf{E}_d(f) & \text{otherwise} \end{cases}$$

where $\mathsf{E}_d(f) = \mathsf{E}(d, C_f(d))$, S denotes the image entropy in the neighborhood of $C_f(d)$, and γ_S is a constant parameter. We use S for penalizing the label f_{DP} in low-textured regions. Finally, the smoothness term is given by:

$$V_{d,e}(f_d, f_e) = \begin{cases} 0 & \text{if } f_d = f_e \\ |\mathcal{C}_{f_d}(d) - \mathcal{C}_{f_e}(e)| & \text{if } (f_d \vee f_e) = f_{\text{DP}} \\ \min\left(|\mathcal{C}_{f_d}(d) - p_{f_{(d,e)}}|, |\mathcal{C}_{f_e}(e) - p_{f_{(d,e)}}|\right) & \text{if } (f_d \wedge f_e) = f_{\mathbf{s}} \end{cases},$$

where $p_{f_{(d,e)}}$ is the intersection point between lines $\mathbf{s}_d$ and $\mathbf{s}_e$, and $f_{\mathbf{s}}$ represent any line segment label. The third term aims to penalize transitions between line contours that are far away from the corresponding point of intersection. The energy is minimized using α-expansion [76, 77, 78], and the output is the profile cut $\mathcal{C}_{\text{MRF}}$, one point per epipolar line. Figure 3.21 present an example of the estimation of $\mathcal{C}_{\text{MRF}}$. As can be observed, the global optimization distinguishes between straight and non-straight segments in the scene. This ability is crucial for overcoming low

and repetitive textured surfaces, as will be shown next.

3.5.3 Experimental results

This section compares the depth estimates achieved with our algorithm for SRF against real range data obtained with a LRF. Figure 3.24 shows pairs of stereo images and corresponding top views of the scan plane with different depth estimates overlaid. The green contour refers to the laser readings, the red points concern the depth estimates obtained by SymStereo with DP refinement, and the blue contour represents the final results after MRF labeling, with dark blue denoting straight line segments (Hough Transform estimates) and light blue denoting non-straight segments (DP estimates). The different profile cuts are projected onto the stereo views for analysis purposes. The examples try to cover a broad range of operating conditions including indoor and outdoor scenes, planar and non-planar surfaces, variable illumination, low textured regions and slanted surfaces.

The overall results are quite encouraging. Referring to Figure 3.24, SymStereo followed by DP provides accurate depth estimates whenever the profile cut $\mathcal{C}$ lies in textured surfaces or is close to strong edges. On the other hand, the DP depth results are often inaccurate in low-textured regions because the joint energy E tends to become disperse around the contour location, and the path optimization is unable to handle the ambiguity. Fortunately, and for the case of planar surfaces, the line segment prior followed by MRF selection seems to be effective in correcting most of the errors.

Major failure occurs in the most distant walls (b), (d), (i), and (j). Curiously, these poor estimates do not happen in the cases (e), (g), and (h), despite of similar circumstances in terms of texture, slant and depth range. This apparent contradiction can be explained by the fact that the induced symmetry, that is quantified by SymStereo, is only perfect for a particular combination of surface slant and point $\mathbf{O}$ where the virtual scan plane intersects the baseline (for further details refer to Section 2.4). Whenever the orientation of the surface to be reconstructed differs from the surface slant that grants perfect induced symmetries for a particular $\mathbf{O}$, then the symmetry deviation is source of errors. This problem is usually handled by the log-Gabor wavelets with wider spatial support (refer to Section 3.2.3). However, in the absence of large textured support, the symmetry deviation is not compensated by the log-Gabor wavelets, and the energy E does not present a well-defined ridge along the contour $\mathcal{C}$. This is the main reason for the failures observed in examples (d), (i) and (j) of Figure 3.24. In the case (e), the scan plane intersects the surface further away from the white wall, which is enough for creating a wider textured support region. In the examples (g) and (h), the cameras are closer to the slanted surfaces and the texture is better perceived.

So far the comparison was carried in the scan plane considering metric depth estimates. Let us briefly analyze what happens in the image domain, where the projection of the profile cut $\mathcal{C}$ is supposed to go through corresponding pixels in

(a) $\mathtt{I}$ (b) $\mathtt{I}'$ (c) Zoom of $\mathtt{I}$ (d) Zoom of $\mathtt{I}'$

Figure 3.22: Image matches obtained in case (a) of Figure 3.24 for the region outlined in yellow in (a) and (b). As in the previous examples, (green) corresponds to the LRF measurements, (red) is $\mathcal{C}_{\text{DP}}$, and (blue and cyan) is $\mathcal{C}_{\text{MRF}}$; where in (cyan) the MRF decided for $\mathcal{C}_{\text{DP}}$, while in (blue) the labeling corresponds to $\mathcal{C}_{\mathbf{s}_i}$. In the case the virtual scan plane intersects a textured region or near strong edges, the matching obtained from DP is very accurate.

the two views. Figure 3.22 shows the zoom of a region in the stereo pair of example (a) in Figure 3.24. The DP estimation leads to the best matching results, proving that SymStereo can achieve accuracies of $1-2$ pixels for an image resolution of 1280×960 whenever the surface is textured. It is also interesting to observe that the projection of the range data obtained with LRF is slightly off in terms of stereo correspondence. This is explained by small errors in the extrinsic calibration between the cameras and the LRF that can hardly be avoided.

(a) Top Image ($\mathtt{I}$) (b) Bottom Image ($\mathtt{I}'$)

Figure 3.23: Multi-cut example. The correspondence between contours in the top and bottom images can be idenfied by the color coding.

Finally, Figure 3.23 shows that our algorithm enables independent depth estimate along multiple virtual scan planes, with the only constraint being that the scan planes must intersect the baseline (refer to Chapter 2).

3.6 Conclusions

This chapter proposed three symmetry-based matching costs for the SymStereo framework: *SymBT*, *SymCen* and *logN*. The first two are closely related with the top-performing cost function BT [65] and Census [67], being in a large extent mere modifications for measuring symmetry instead of similarity, while the later relies in wavelet transforms for detecting local signal symmetry. The new matching costs

were benchmarked against the state-of-the-art metrics for accomplishing dense disparity labeling in both short and wide-baseline images. The results showed that the symmetry based functions, SymBT and SymCen, consistently outperform their similarity-based counterparts, BT and Census, suggesting that symmetry is superior to standard photo-consistency as a stereo metric. The $\log N$ cost proved to be particularly effective in scenes with slanted surfaces and difficult textures, being the top-performer matching function in the Oxford Corridor dataset. The major weakness is its relative poor performance close to discontinuities and occlusion regions.

We also investigated the use of passive stereo for estimating depth along a single scan plane. The technique, named Stereo-Rangefinding (SRF), provides profile cuts of the scene similar to the ones that would be obtained by a LRF. For the purpose of SRF, logN was clearly the top-performing metric. Additionally, SRF was experimentally compared against LRF in several indoor and outdoor scenes. The results were encouraging in terms of showing that passive stereo can be leveraged to meet the robustness and depth accuracy of laser range data. SRF proved to be as accurate as LRF in most of the tests, but important issues remain for the case of the profile cut lying in slanted surfaces with very low texture.

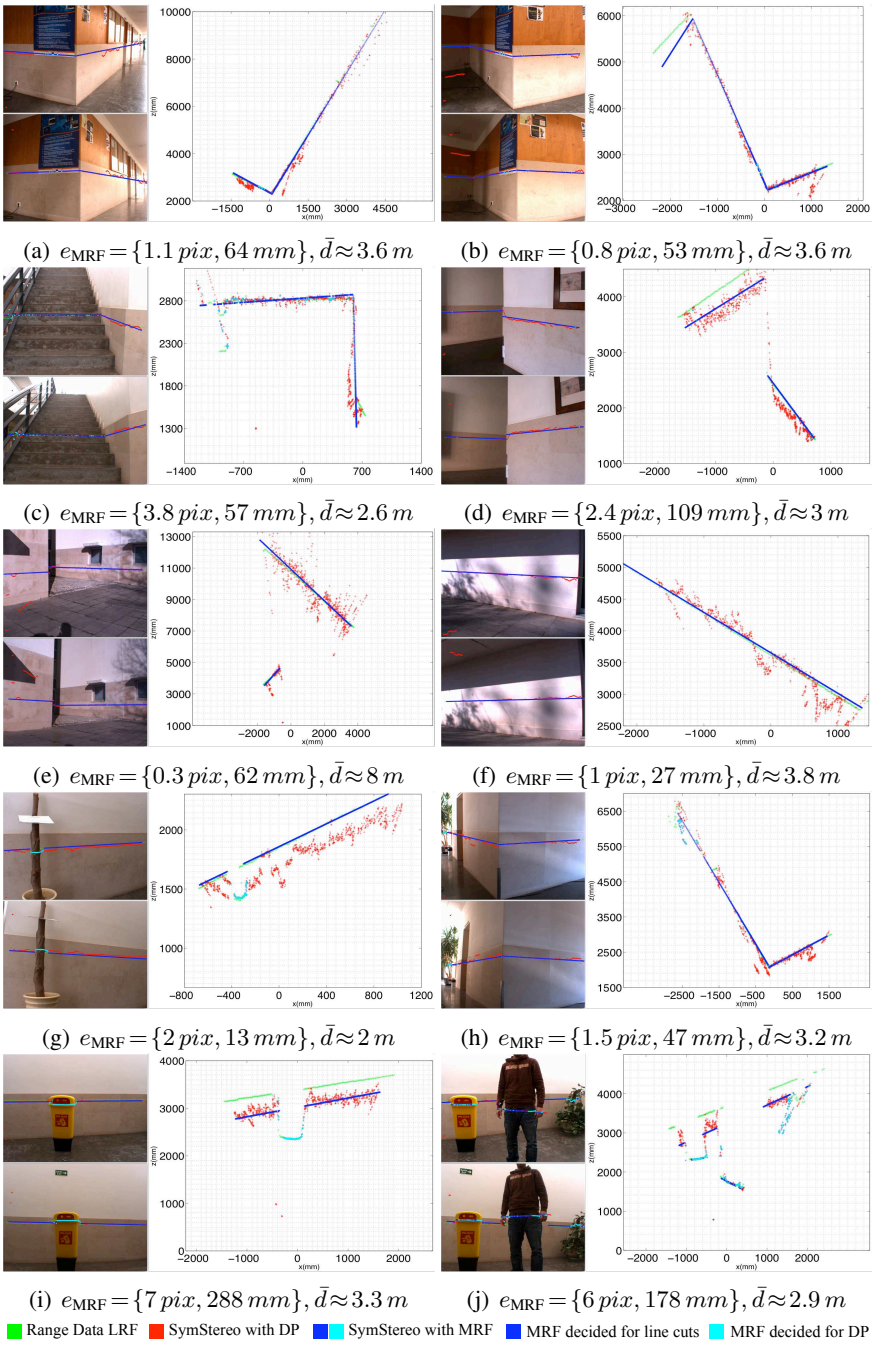


Figure 3.24: Qualitative comparison between $\mathcal{C}$ estimated using SRF and the measurements provided by a LRF. (Green) - Measurements of the LRF, (red) $\mathcal{C}_{\text{DP}}$ estimated using DP, (blue and cyan) labeling obtained from the MRF, where (blue) are points to which a line segment was assigned, while for the (cyan) points the MRF decided for the non-straight label ($\mathcal{C}_{\text{DP}}$). e_{MRF} (final estimation) - average distance between the range data provided by the LRF and the points C_{MRF} (the first value is measured in the reference image, while the second value concerns depth measurements). $\bar{d}$ represents the average distance of the LRF readings from the origin.

Chapter 4

Vanishing points and mutually orthogonal vanishing directions

This chapter presents a new global approach for detecting VPs and groups of mutually orthogonal VDs in man-made environments. These multi-model fitting problems are respectively cast as UFL and HFL instances that are solved using a message passing inference algorithm. We also propose new functions for measuring the consistency between an edge and a putative VP, and for computing the VP defined by a subset of edges. Experiments in both synthetic and real images show that our algorithms outperform the state-of-the-art methods while keeping computation tractable. In addition, we show for the first time results in simultaneously detecting multiple Manhattan-world configurations.

4.1 Introduction

A set of parallel lines in the scene project into a pencil of lines intersecting in the so-called vanishing point (VP). The VP is the image of the point at infinity where the parallel lines intersect and encodes their common direction. In the case of man-made environments, the sets of parallel lines are usually orthogonal to each other, and the detection of the corresponding VPs enables to accomplish different tasks. Applications include intrinsic camera calibration [86], estimation of the camera rotation with respect to the scene [87, 88], 3D reconstruction [89], and recognition [90].

The automatic detection of VPs using sparse edges [91] or edge gradients [92] is a problem of multi-model fitting where the models are line pencils. It is in general a "chicken-and-egg" problem because we neither know the number and

(a) Manhattan World (b) Multiple orthogonal triplets

Figure 4.1: Two images of man-made environments.

parameters of the models (the VPs), nor the edges that belong to each model (the membership). The first attempt of automatic detection of VPs goes back to the 80's when Barnard proposed to use the Hough transform on a quantized Gaussian sphere [93]. It was latter shown that the accuracy of such an approach highly depends on the choice of the voting bins, and that the detection results are often spurious. In [87], Antone and Teller suggests to carry the VP detection using Expectation-Maximization (EM) with the E-step computing the probability distributions of the input lines passing through the hypothesized VPs, and the M-step refining the VP models by maximizing the likelihood of the observed data. Later, the EM framework was successfully extended to the case of uncalibrated cameras [88, 92]. However, the process is iterative and requires a good initial estimate that is typically accomplished by clustering the edges assuming a world dominated by either 3 (Manhattan) [87, 88] or 5 (Atlanta) [92] mutually orthogonal VDs. In [94], Rother combines RANSAC search with several heuristics for recovering the VPs of Manhattan directions, but the final algorithm is computationally expensive and requires distinguishing between finite and infinite VPs. Finally, Tardif has recently proposed a new image-based consistency metric to be used with J-Linkage for clustering the edges into pencils of lines [2]. The algorithm is fast, robust, and accurate, being one of the best performing VP detectors that are currently available.

The works above perform the separate estimation of the VPs in the image, which, in many cases, is followed by grouping the result into directions that are mutually orthogonal [2]. A different approach is to consider *a priori* that the scene follows the Manhattan world assumption and determine the rotation that is aligned with the 3 dominant VDs. In this case, the VP detection is no longer a problem of multiple model fitting, but the problem of fitting a single triplet of mutually orthogonal VPs in the presence of edges that are outliers. Such fitting can be accomplished through EM [91], by using minimal solutions as hypothesis generator in a RANSAC paradigm [95], or by applying Branch-and-Bound to solve a consensus set maximization that assures global optimality [96]. The disadvantages of this type of approach are that additional VDs that might exist are passed undetected, and the methods have difficulty in handling images with more than one set of Manhattan-world directions for which multi-model fitting is again required (see Figure 4.1(b)).

This chapter addresses the problem of detecting VPs in uncalibrated images using either edgels (by edgel we mean a discrete set of edge points that are connected) or line segments, and (given the intrinsic calibration) the problem of grouping the detection results into sets of mutually orthogonal VDs. We propose three main contributions with respect to the state-of-the-art.

The first contribution is a a global solution for the detection of VP. As discussed in Section 5.3.1, methods that greedily search for models with most inliers (within a threshold) while ignoring the overall classification of data are in general a flawed approach to multi-model fitting, and that formulating the fitting as an optimal labeling problem with a global energy function is usually preferable [3]. Our research goes towards this direction and formulates for the first time the detection of VPs as an Uncapacited Facility Location (UFL) problem [97] that can be solved using a local message passing approach [97, 98]. Experiments show that such a global approach is very competitive with the state-of-the-art algorithm [2] that relies in J-Linkage and EM. Very recently, Tretyak et al. [99] presented a method that integrates the estimation of line segments, lines, VPs, the horizon and zenith in a single energy optimization framework. Besides of being complex and time consuming, this formulation already assumes that a discrete number of accurate VPs has been obtained.

Independently of the multi-model fitting approach, the detection of VPs always requires a consistency function $D(\mathsf{e}, \mathbf{v})$, which measures the likelihood of the edgel e being in a line $\mathbf{l}$ passing through the putative VP $\mathbf{v}$, and a function $W(\mathcal{S})$ that computes the most likely VP given a set of edges $\mathcal{S}$. Many prior works formulate the consistency function in the Gaussian sphere after back-projecting the edges and VPs [93, 87, 88, 96, 92]. However, and as argued in [2], measurements in the image space are usually preferred because the non-linear mapping into the sphere changes the statistics of noise ultimately leading to biased estimation results [94]. Therefore, Tardif proposes to formulate $D(\mathsf{e}, \mathbf{v})$ and $W(\mathcal{S})$ using the geometric distance measured in the image [2]. However, and in order to avoid iterative non-linear minimization, he works with the maximum orthogonal distances to the edge endpoints rather than considering the mean distance to all points. As our second contribution, we show that this minimization problem can be solved in closed-form and propose new functions $D(\mathsf{e}, \mathbf{v})$ and $W(\mathcal{S})$ that improve the overall fitting results while keeping computation tractable.

Finally, our last contribution is a global solution for detecting multiple sets of mutually orthogonal VDs. The existing methods for detecting mutually orthogonal VDs assume that the image depicts a single Manhattan-world configuration [95, 96]. In practice, these algorithms often become unstable and/or inaccurate whenever there is no image evidence for one of the Manhattan directions, and cannot cope with frames like the one of Figure 4.1(b) showing more than one group of mutually orthogonal directions. We propose for the first time an algorithm that, given an initial set of VPs, is able to detect multiple Manhattan-world configurations that can either be complete or incomplete (two directions), and be independent or have one direction in common (Atlanta-world). The multi-model fitting

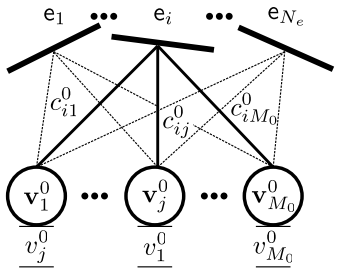


Figure 4.2: The UFL problem. The objective is to assign to each customer e_i a facility $\mathbf{v}^0_j$, minimizing the sum of the customer-facility costs c^0_{ij} plus the sum of facility opening costs v^0_j (see Equation 4.1).

is solved in a global manner by casting the problem as an Hierarchical Facility Location (HFL) problem [100].

4.2 The Facility Location Problem

This section briefly introduces the problems of UFL and HFL that play a key role in the global approaches for detecting VPs and clustering mutually orthogonal VDs. To the best of our knowledge these frameworks were seldom used in the context of computer vision. In [98] and [101] the problems of subspace segmentation and two-view motion segmentation are formulated as UFL problems, respectively, while in [102] Xiao et al. formulated the simultaneous segmentation of registered 2D images and 3D points as a hierarchical exemplar-based clustering instance [103], a problem that is closely related to UFL, and that was solved using a greedy bottom-up affinity propagation approach [102]. The UFL is a classical NP-hard problem that can be solved by applying an optimization method based on the max-sum algorithm [97, 98]. This method is more robust than the greedy solver for UFL discussed by Delong et al. in [104], and has been recently extended for also handling the HFL problem [100]. Since this extension has never been applied in computer vision, we briefly outline the solver that relies in local message passing.

4.2.1 Uncapacited Facility Location (UFL)

Suppose that you need to open a set of facilities $\mathbf{v}^0_j$ to serve N_e customers $\mathsf{e}_i \in \mathcal{E}$ whose locations are known (see Figure 4.1). Given a set $\mathcal{V}_0$ comprising M_0 possible facility locations, the cost $c^0_{ij} : \mathcal{E} \times \mathcal{V}_0 \rightarrow \mathbb{R}$ for assigning the facility $\mathbf{v}^0_j$ to the customer e_i, and the cost $v^0_j : \mathcal{V}_0 \rightarrow \mathbb{R}$ for opening the particular facility $\mathbf{v}^0_j$, the goal of the UFL problem is to select a subset of $\mathcal{V}_0$ such that each customer is served by one facility, and the sum of the customer-facility costs plus the sum of

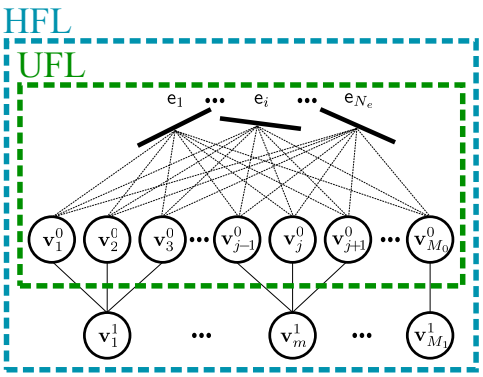


Figure 4.3: The HFL problem. The objective is to assign to each customer e_i a facility $\mathbf{v}^0_j$, minimizing the sum of the customer-facility costs c^0_{ij}, the sum of facility opening costs v^0_j, and the *storage* facilities opening costs v^1_k (see Equation 4.2).

facility opening costs is minimized. This leads to an integer programming problem that is usually formulated using unary indicator variables y^0_j and binary indicator variables x^0_{ij}, and whose objective is to find the vector $\mathbf{x}^0 = \{x^0_{11}...x^0_{ij}...x^0_{N_eM_0}\}$ such that :

$$\min_{\mathbf{x}^0} \quad \sum_{i=1}^{N_e}\sum_{j=1}^{M_0} c^0_{ij}\, x^0_{ij} + \sum_{j=1}^{M_0} v^0_j y^0_j$$

$$\text{subject to} \quad \begin{cases} x^0_{ij},\, y^0_j \in \{0,1\},\, \forall_{i,j} \\ \sum_{j=1}^{M_0} x^0_{ij} = 1,\, \forall_i \\ y^0_j \geq x^0_{ij},\, \forall_{i,j} \end{cases} \tag{4.1}$$

The equality in the second constraint ensures that each customer is assigned to exactly one facility, while inequality of the last constraint guarantees that each customer is only served by facilities that were opened.

4.2.2 Hierarchical Facility Location (HFL)

Let's now imagine that the facilities $\mathbf{v}^0_j$ need to be stocked by storage facilities (warehouses) $\mathbf{v}^1_k$, which in turn need to be stocked by larger warehouses $\mathbf{v}^2_m$, and so forth till the graph of the UFL problem is extended by L additional levels (Figure 4.3 shows an example of a HFL with two levels). Given a set of potential M_l facility locations $\mathcal{V}_l$ at layer l, the cost $v^l_j : \mathcal{V}_l \to \mathbb{R}$ for opening the facility $\mathbf{v}^l_j$, and the cost $c^l_{jk} : \mathcal{V}_{l-1} \times \mathcal{V}_l \to \mathbb{R}$ for the facility $\mathbf{v}^l_k$ supplying the facility $\mathbf{v}^{l-1}_j$, the goal of HFL is to find the vector $\mathbf{x} = \{\mathbf{x}^0...\mathbf{x}^l...\mathbf{x}^L\}$ that minimizes the following

function:

$$\min_{\mathbf{x}} \quad \sum_{i=1}^{N_e}\sum_{j=1}^{M_0} c_{ij}^0 x_{ij}^0 \; + \; \sum_{l=1}^{L}\sum_{j=1}^{M_{l-1}}\sum_{k=1}^{M_l} c_{jk}^l x_{jk}^l \; + \; \sum_{l=0}^{L}\sum_{j=1}^{M_l} v_j^l y_j^l$$

$$\text{s.t.:} \quad \begin{cases} x_{ij}^l, y_j^l \in \{0,1\} \\ \sum_{j=1}^{M_0} x_{ij}^0 = 1, \forall_i \quad \bigwedge \quad \sum_{k=1}^{M_l} x_{jk}^l = y_j^{l-1}, \forall_{j,\, l>0} \\ y_j^0 \geq x_{ij}^0, \forall_{i,\, j} \quad \bigwedge \quad y_k^l \geq x_{jk}^l, \; \forall_{j,\, k,\, l>0} \end{cases} \tag{4.2}$$

The additional restrictions compared to Equation 4.1 are that if a facility $\mathbf{v}_j^{l-1}$ is closed in layer $l-1$, then $\mathbf{v}_j^{l-1}$ will not need to be stocked by a storage facility $\mathbf{v}_j^l$. Whereas if a facility $\mathbf{v}_j^{l-1}$ is open, then it must be stocked by a facility in the next layer l. Note that in the case of a single layer, the HFL problem reduces to the UFL problem (see Figure 4.2).

4.2.3 Solving UFL and HFL using the max-sum algorithm

In [98, 97] Lazic et al show how to solve the UFL problem using a local message passing approach. They formulate the UFL problem as a maximum-a-posteriori (MAP) problem and represent it using a factor graph [105]. The MAP estimates for x_{ij}^0 can then be inferred using the max-sum algorithm [105], which is a log-domain equivalent of the max-product solver [105]. More recently, Givoni et al. [100] extended this message passing framework for solving the HFL problem. The basic idea is to iteratively update the following messages until convergence[1]:

$$\eta_{ij}^l = -c_{ij}^l + \min(\tau_i^l, -\max_{k \neq j}(\alpha_{ik}^l - c_{ik}^l)), l > 0$$
$$\eta_{ij}^l = -c_{ij}^l - \max_{k \neq j}(\alpha_{ik}^l - c_{ik}^l), l = 0$$
$$\alpha_{ij}^l = \min[0, -v_j^l + \phi_j^l + \sum_{k \neq i} \max(0, \eta_{kj}^l)], l < L+1$$
$$\alpha_{ij}^l = \min[0, -v_j^l + \sum_{k \neq i} \max(0, \eta_{kj}^l)], l = L+1$$

[1] Remark that initially $\eta = 0$ and $\alpha = 0$.

where the messages

$$\tau_k^{l+1} = \sum_{j=1}^{M^l} \max(0, \eta_{jk}^l) - v_k^l \quad , \quad \phi_j^{l-1} = \max_k(\alpha_{jk}^l - f_{jk}^l)$$

are required for connecting successive layers. The message τ_k^{l+1} is passed upwards from layer l to layer $l+1$, while the message ϕ_j^{l-1} goes down from layer l to layer $l-1$. The max-sum algorithm is guaranteed to converge on tree graphs, and has shown good performance for $L = 1$ on graphs with cycles in many applications, e.g. [98]. It is important to mention that a practical way of dealing with message oscillations is to *damp* the messages at each iteration [97]

$$\eta = \gamma\eta_{\text{prev}} + (1 - \gamma)\eta$$

where $\gamma \in [0, 1[$ is the damping factor and η_{prev} is the previous message. Upon convergence, the set of facilities $\mathcal{F}^l$ in layer l that are open are $\mathcal{F}^l = \{\mathbf{v}_j^l | \left(\alpha_{ij}^l + \eta_{ij}^l\right) > 0\}$. The optimal MAP estimation for $\mathbf{x}$ is given by

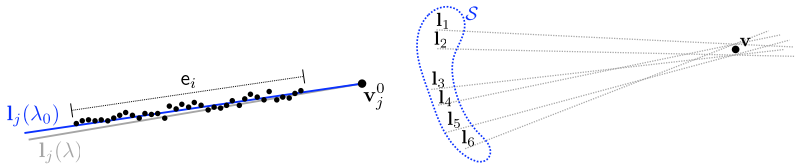
$$x_{ij}^l = \begin{cases} 1 & \text{if } c_{ij}^l \leq c_{ik}^l \bigwedge \mathbf{v}_j^l, \mathbf{v}_k^l \in \mathcal{F}, \forall_{j,k} \\ 0 & \text{otherwise} \end{cases}$$

4.3 Algorithm for VP detection

This section shows that the detection and estimation of VPs can be formulated as an instance of the UFL problem discussed in Section 4.2.1. Such formulation requires defining a consistency metric $D(\mathsf{e}_i, \mathbf{v}_j^0)$ that measures the consistency of an edgel e_i with a putative VP $\mathbf{v}_j^0$, and a function $W(\mathcal{S}, \mathbf{w})$ that, given a subset of edges $\mathcal{S}$, computes the most likely VP $\mathbf{v}$.

4.3.1 Vanishing point detection as a UFL problem

Let $\mathsf{e}_i \in \mathcal{E}$ with $i = 1...N$ be the ith edgel extracted from an image. The objective is to assign to each e_i a VP $\mathbf{v}_j^0 \in V^0$ using as few unique VP models as possible. This multi-model fitting problem can be cast as an instance of the UFL problem as follows: consider that the edgels e_i are the customers and the putative VPs $\mathbf{v}_j^0$ are the facilities. Let the cost c_{ij}^0 be given by the function $D(\mathsf{e}_i, \mathbf{v}_j^0)$ that evaluates the consistency between e_i and $\mathbf{v}_j^0$, and let v_j^0 be the cost for adding $\mathbf{v}_j^0$ in the final VP assignment. The goal is to select a subset of VPs in $\mathcal{V}^0$ such that sum of the consistency measures c_{ij}^0 and the costs $\mathbf{v}_j^0$ is minimized, which corresponds exactly



(a) Consistency function D (b) VP estimator W

Figure 4.4: Consistency function and VP estimator. (a) We compute the line $\mathbf{l}_j(\lambda_0)$ that intersects $\mathbf{v}_j^0$ and minimizes the sum of the squares of the geometric distances to the points on e_i for measuring the consistency between e_i and $\mathbf{v}_j^0$; (b) We compute $\mathbf{v}$ by finding the point location that minimizes the weighted sum of the square of geometric distances to the lines $\mathbf{l}_i$.

to the minimization of Equation 4.1. There are however some issues that must be addressed:

1. the selection of the set $\mathcal{V}^0$ of VP hypotheses
2. the definition of the function $D(\mathsf{e}_i, \mathbf{v}_j^0)$ that provides the client-facility cost c_{ij}^0 by measuring the consistency between e_i and $\mathbf{v}_j^0$
3. the choice of the function $W(\mathcal{S}, \mathbf{w})$ that, after clustering a subset $\mathcal{S}$ of line segments, updates the VP location.

The set $\mathcal{V}^0$ containing the initial VP hypotheses depends mainly on the time constraints of the particular application. In the case of no time limitations, $\mathcal{V}^0$ can comprise all the point intersections between pairs of lines $\mathbf{l}_i$, $\mathbf{l}_j$ fitting every possible pair of edgels e_i, e_j, respectively. Otherwise, a fast RANSAC procedure can be used for quickly extract model hypotheses using minimal sample sets. The issues 2 and 3 are addressed next.

4.3.2 The consistency function $D(\mathsf{e}, \mathbf{v})$

Given an edgel e_i comprising P_i points $\mathbf{e}_k$ with $k = 1...P_i$ and a putative VP $\mathbf{v}_j^0$, the objective is to find a cost function $D(\mathsf{e}_i, \mathbf{v}_j^0)$ that evaluates how well a line $\mathbf{l}_j$ in the pencil centered in $\mathbf{v}_j^0$ can fit the edge points in $\mathbf{e}_k$ (see Figure 4.4(a)). We propose to determine the line $\mathbf{l}_j$ that minimizes the sum of the squares of the geometric distances to the points, and use the root mean value of this sum as the client-facility cost c_{ij}^0. Contrary to what is suggested in [2], the minimization problem can be solved in a closed-form manner. Any line $\mathbf{l}_j$ going through $\mathbf{v}_j^0$ can be parametrized as follows

$$\mathbf{l}_j(\lambda) \sim (1-\lambda)[\mathbf{a}]_\times \mathbf{v}_j^0 + \lambda[\mathbf{b}]_\times \mathbf{v}_j^0 \,,$$

with $\mathbf{a}$ and $\mathbf{b}$ being any two points non-collinear with $\mathbf{v}_j^0$, and λ being a free parameter. For the sake of convenience, the points $\mathbf{a}$ and $\mathbf{b}$ are typically chosen as being the endpoints of a line segment orthogonal to e_i and passing through its midpoint.

We want to find the λ value such that:

$$\min_{\lambda} \quad \sum_{k=1}^{P_i} d_{\perp}^2(\mathbf{e}_k,\, \mathbf{l}_j(\lambda))$$

From the formula for the orthogonal distance $d_\perp$, it comes after some algebraic manipulations that

$$\sum_{k=1}^{P_i} d_{\perp}^2(\mathbf{e}_i^k, \mathbf{l}_j) = \frac{(\mathbf{v}_j^{0\mathsf{T}} \mathsf{A}_2 \mathbf{v}_j^0)\lambda^2 + (\mathbf{v}_j^{0\mathsf{T}} \mathsf{A}_1 \mathbf{v}_j^0)\lambda + \mathbf{v}_j^{0\mathsf{T}} \mathsf{A}_0 \mathbf{v}_j^0}{(\mathbf{v}_j^{0\mathsf{T}} \mathsf{B}_2 \mathbf{v}_j^0)\lambda^2 + (\mathbf{v}_j^{0\mathsf{T}} \mathsf{B}_1 \mathbf{v}_j^0)\lambda + \mathbf{v}_j^{0\mathsf{T}} \mathsf{B}_0 \mathbf{v}_j^0} \tag{4.3}$$

where

$$\mathsf{A}_0 = \sum_{k=1}^{P_i} [a]_\times \mathbf{e}_k \mathbf{e}_k^T [a]_\times \qquad \mathsf{A}_1 = \sum_{k=1}^{P_i} ([a]_\times \mathbf{e}_k \mathbf{e}_k^T [b]_\times + [b]_\times \mathbf{e}_k \mathbf{e}_k^T [a]_\times) - 2\mathsf{A}_0$$

$$\mathsf{B}_0 = \sum_{k=1}^{P_i} [a]_\times \mathsf{I}_s [a]_\times \qquad \mathsf{B}_1 = \sum_{k=1}^{P_i} ([a]_\times \mathsf{I}_s [b]_\times + [b]_\times \mathsf{I}_s [a]_\times) - 2\mathsf{B}_0$$

$$\mathsf{B}_2 = \sum_{k=1}^{P_i} ([b]_\times \mathsf{I}_s [b]_\times) - \mathsf{B}_0 - \mathsf{B}1 \qquad \mathsf{A}_2 = \sum_{k=1}^{P_i} ([b]_\times \mathbf{e}_k \mathbf{e}_k^T [b]_\times) - \mathsf{A}_0 - \mathsf{A}_1$$

The minima and maxima of the objective function are the λ values for which the derivative is zero. By differentiating the expression of Equation 4.3, it comes that these extrema can be easily computed by solving a second order equation. Given the particular arrangement between $\mathbf{a}$, $\mathbf{b}$, and e_i, we choose the root λ_0 that is closest to 0.5, and replace the result in the equation below:

$$c_{ij}^0 \equiv D(\mathsf{e}_i, \mathbf{v}_j^0) \;=\; \sqrt{\frac{\sum_{k=1}^{P_i} d_{\perp}^2(\mathbf{e}_k, \mathbf{l}_j(\lambda_0))}{P_i}} \tag{4.4}$$

4.3.3 The function $W(\mathcal{S})$ for updating the VP estimate

After solving the UFL problem, the edgels sharing the same label are clustered into a subset $\mathcal{S}$, and the objective is to determine the most likely intersection point $\mathbf{v} \sim W(\mathcal{S})$ for the lines $\mathbf{l}_i$ fitting the edgels $\mathsf{e}_i \in \mathcal{S}$ (see Figure 4.4(b)). We propose to update the VP by finding the point location that minimizes the weighted sum of the square of geometric distances to the lines $\mathbf{l}_i$. Taking into account the formula

of the orthogonal distance, it comes after some algebraic manipulations that

$$W(\mathcal{S}) = \begin{array}{ll} \min\limits_{\mathbf{v}} & \mathbf{v}^\mathsf{T}\,\mathsf{Q}\,\mathbf{v} \\ \text{subject to :} & \mathbf{v}^\mathsf{T}\,\mathbf{p} = 1 \end{array}$$

with

$$\mathsf{Q} = \sum_{i=1}^{N_i} w_i^2 \frac{\mathbf{l}_i\,\mathbf{l}_i^\mathsf{T}}{\mathbf{l}_i^\mathsf{T}\mathsf{l}_s\mathbf{l}_i}$$

where w_i is the length of each edge e_i, and $\mathbf{p} = (0\ 0\ 1)^\mathsf{T}$. Remark that the purpose of the constraint is to assure that $v_3 = 1$ complies with the formula for computing the orthogonal distance $d_\perp$. We can rewrite the constrained minimization problem as an unconstrained one:

$$W(\mathcal{S}) = \min_{\mathbf{v},\,\lambda} \ \mathbf{v}^\mathsf{T}\mathsf{Q}\mathbf{v} - \lambda(\mathbf{v}^T\mathbf{p} - 1).$$

with λ being a Lagrange multiplier. By differentiating the objective function, it comes that the minima can be determined by solving following matrix equation

$$\underbrace{\begin{bmatrix} 2\mathsf{Q} & -\mathbf{p} \end{bmatrix}}_{\mathsf{Q}'} \begin{pmatrix} \mathbf{v} \\ \lambda \end{pmatrix} = \mathbf{0},$$

Note that if the lines $\mathbf{l}_i$ are quasi-parallel, the problem becomes undetermined, which can be observed by the matrix Q' becoming poorly conditioned. In this case, the VP $\mathbf{v}$ is at infinity, and its direction can be computed by simply averaging over the directions of $\mathbf{l}_i$.

4.4 Detection of multiple orthogonal triplets

We assume in this section that a set of VPs has already been extracted using any type of VP detection approach e.g. the approach proposed in Section 4.3, and the objective is to detect multiple mutually orthogonal directions in the scene. As will be shown, this problem can be easily cast as a HFL problem.

Given the intrinsic calibration matrix K, two VPs $\mathbf{v}_j^0$ and $\mathbf{v}_k^0$ are orthogonal if the following relation is verified

$$\mathbf{v}_j^{0\mathsf{T}}\,\omega\,\mathbf{v}_k^0 = 0, \tag{4.5}$$

where $\omega = \mathsf{K}^{-\mathsf{T}}\mathsf{K}^{-1}$ is the image of the absolute conic [106]. Let the set

$$\mathbf{v}_m^1 = \{\mathbf{v}_j^0, \mathbf{v}_k^0, \mathbf{v}_l^0\}$$

be a mutually orthogonal triplet, meaning that each pair of VPs in $\mathbf{v}_m^1$ verifies Equation 4.5. Consider again a set of edgels $\mathsf{e}_i \in \mathcal{E}$, a set of VPs $\mathbf{v}_j^0 \in \mathcal{V}^0$ and a set of orthogonal triplets $\mathsf{v}_k^1 \in \mathcal{V}^1$, whose VP elements are known and are contained in $\mathcal{V}^0$. The objective is to assign a VP to each e_i, minimizing not only the number of VPs, but also the number of orthogonal triplets. This problem is cast as a HFL instance with two different layers (see Figure 4.3): at the bottom layer $l = 0$ we have e_i and $\mathbf{v}_j^0$, and at the top layer $l = 1$ we have the orthogonal triplets v_k^1. In addition to the costs c_{ij}^0 and v_j^0 described in Section 4.3, there is a new penalization $v_k^1 : \mathcal{V}^1 \rightarrow \mathbb{R}$ for $\mathbf{v}_k^1$ being contained in the scene. The connection costs c_{jk}^1 between $\mathbf{v}_j^0$ and $\mathbf{v}_k^1$ are given by

$$c_{jk}^1 = \begin{cases} 0 & \text{if } \mathbf{v}_j^0 \in \mathbf{v}_k^1 \\ \infty & \text{otherwise} \end{cases}$$

There are three issues that must be addressed:

1. how to propose an initial set of orthogonal triplets $\mathcal{V}^1$
2. there might exist VPs in $\mathcal{V}^0$ that are not part of any orthogonal triplet $\mathbf{v}_k^1$
3. the orthogonal triplets can share a common VD.

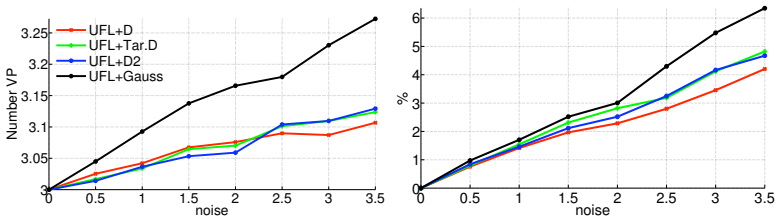
The issue 1 is solved as follows: for each pair $\mathbf{v}_j^0, \mathbf{v}_k^0$ in $\mathcal{V}^0$ whose angle is in the range $[\frac{\pi}{2}-\theta, \frac{\pi}{2}+\theta]$, we obtain an exact orthogonal triplet $\mathbf{v}_m^1$ computed as follows

$$\mathbf{v}_m^1 = \{\mathbf{v}_1^0, \mathbf{v}_2^0, \mathbf{v}_3^0\} \begin{array}{|ll} \mathbf{v}_1^0 & = \mathbf{v}_j^0 \\ \mathbf{v}_2^0 & = \text{Null}(\omega(\mathbf{v}_1^0\ \mathbf{v}_k^0)) \\ \mathbf{v}_3^0 & = \text{Null}(\omega(\mathbf{v}_1^0\ \mathbf{v}_2^0)) \end{array}$$

which is added to $\mathcal{V}^1$, and where the operator $\text{Null}(\mathsf{M})$ returns the left nullspace of the matrix M. Note that the additional created VPs are also added to $\mathcal{V}^0$, which implies having very similar or even equal VPs in $\mathcal{V}^0$. This problem is easily handled by the HFL solver that prefers assignments with less VPs.

For solving 2, we add the groups $\mathbf{v}_m^1 = \{\mathbf{v}_j^0\}$ containing a single VP to $\mathcal{V}^1$ whenever there is no VP in $\mathcal{V}^0$ whose direction makes up an angle in the range $[\frac{\pi}{2}-\theta, \frac{\pi}{2}+\theta]$ with $\mathbf{v}_j^0$. The costs v_m^1 for sets $\mathbf{v}_m^1$ containing a single VP are always less than for orthogonal triplets, keeping these VPs in the final labeling. Finally, the issue 3 is solved by noting that, since we construct each orthogonal triplet $\mathbf{v}_m^1$ individually, we can keep track of similar VPs in $\mathbf{v}_m^1$ after the HFL labeling.

4.5 Experiments with synthetic data



(a) Number of VP (b) Labeling Error (%)

Figure 4.5: Clustering of line pencils in synthetic data. We compare UFL using four different consistency metrics.

In this section, we conduct a set of experiments in a simulation environment that considers an image of size 640×480 and the intrinsic parameters provided by the York Urban Database (YUD) [91]. We randomly generate 3 VDs in the Gaussian sphere, with the angles between them being always less than $20°$. For each VP $\mathbf{v}_j$, we generate a pencil of N line segments in the image, which are sampled into a discrete set of points $\mathbf{e}_k$ with $k = 1...P_i$. Each set e_i has a length between 20 to 200 pixels. The points are then perturbed with Gaussian noise of different magnitudes and 200 trials are run for each noise level.

Figure 4.5 compares four different consistency metrics for quantifying c_{ij}^0 for the UFL clustering method:

1. **UFL+D** - our measure D described in Equation4.4 using all the points in $\mathbf{e}_i$
2. **UFL+D2** - the same measure D using only the end points of $\mathbf{l}_i$
3. **UFL+Tar.D** - the consistency metric of Tardif described in [2]
4. **UFL+Gauss** - operate on the Gaussian sphere by analyzing the angle between the normal to the line $\mathbf{l}_i$ and $\mathbf{v}_j^0$.

Clearly, UFL operating on the Gaussian sphere provides the worst labeling results with increasing magnitude of noise. The performance of the three metrics operating in the image plane are similar for low noise, but our metric $D(\mathsf{e}_i, \mathbf{v}_j^0)$, which uses all the points in e_i being clearly the top-performer for higher noise magnitudes. The consistency metric $D(\mathsf{e}_i, \mathbf{v}_j^0)$ operating on the two end points of $\mathbf{l}_i$ performs slightly worse, but with a high increase in computational efficiency. By taking this results in consideration, we decided to select **UFL+D2** for measuring the consistency between edgels and VPs, being a good trade-off between accuracy and computational efficiency.

Given a cluster $\mathcal{S}$ containing N lines $\mathbf{l}_i$, we need to compute a better VP estimation. As in the previous experiment, we randomly generate a pencil $\mathcal{S}$ containing N lines, sample the lines into a discrete set of points, perturb the points using Gaussian noise of different magnitudes, and then fit a line $\mathbf{l}_i$ to these points in the least-squares sense. We compare in Figure 4.6:

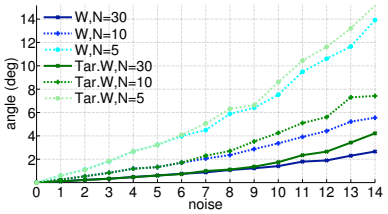


Figure 4.6: Accuracy of the estimation of VPs given a pencil of N lines. The error corresponds to the angle in the Gaussian sphere between the ground truth and the estimated VP.

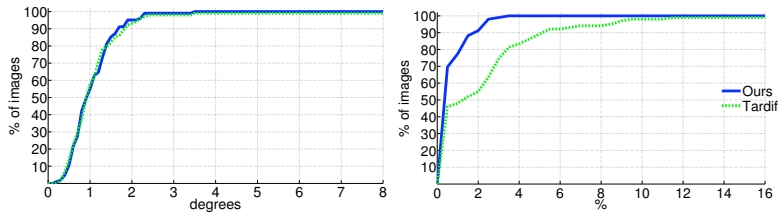
1. **W** - our function $W(\mathcal{S})$ described in Equation 4.3.3
2. **Tar.W** - VP estimator proposed by Tardif in [2]

A careful analysis of the graphic shows that our VP estimator provides better estimates for the same pencil of lines, being considerably more robust to the noise level. These results justify our choice for selecting $W(\mathcal{S})$ as VP estimator.

4.6 Experiments in real images

This section presents experimental results carried in real data. Our algorithm was implemented in Matlab, being the UFL and HFL solver run in MEX files. We compute an initial set of 5000 VP hypotheses for UFL using RANSAC over a minimal set of two edges. In order to handle possible outlier edges detected in the images, we added the empty sets $\mathbf{v}_{\emptyset}^{0}$ and $\mathbf{v}_{\emptyset}^{1}$ to both UFL and HFL, which have the facility costs $\mathbf{v}_{\emptyset}^{0,1} = 0$ and constant connections costs.

4.6.1 YUD using the supplied lines



(a) Cumul. Angular Deviation (b) Cumul. Error Labeling

Figure 4.7: Comparison between our UFL approach with the method proposed by Tardif [2] for the detection of VPs.

We tested our algorithm for VP detection in the YUD [91], which consists in 102 calibrated images of man-made environments. Each image contains two or

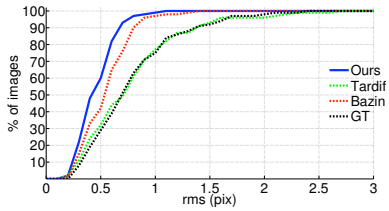


Figure 4.8: Cumulative consistency error computed using our D for the three groups of ground truth edges (belonging to orthogonal VDs). For each image we compute the rms consistency error across all lines fitting the estimated VPs; **GT** corresponds to the ground truth VPs provided by YUD.

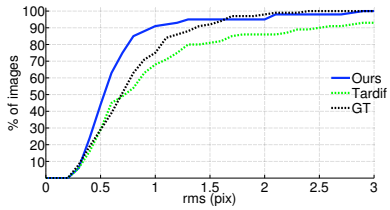


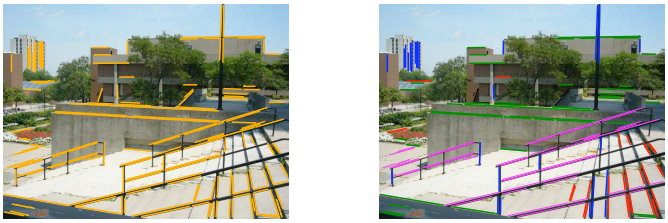
Figure 4.9: Cumulative consistency error computed using D for the three groups of ground truth edges. The results for **Ours** and **Tardif** were obtained using edges automatically detected.

three VPs, line segments that were manually extracted, and whose VP membership is provided. Given the set of line segments, we run our UFL algorithm and compared the results against the ones obtained using the algorithm proposed by **Tardif** [2]. The results are shown in Figure 4.7. The accuracy of the estimation of the VPs positions is very similar, whereas concerning the clustering of the lines, our approach shows some improvements, having in 92% of the images less than 2% of the lines wrongly labeled. In terms of computation time, **Tardif** takes on average 0.5 seconds on images of YUD, while our UFL approach needs 1 second (note that the number of initial VPs is the same for both).

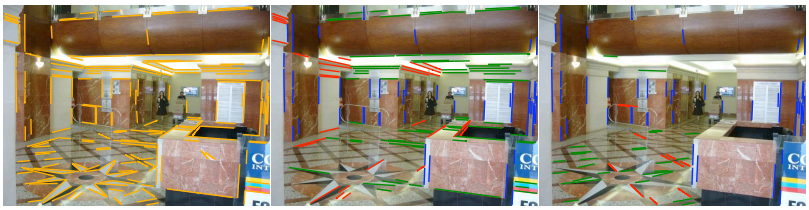
Given the initial set of VPs obtained using the UFL algorithm, the objective now is to detect the Manhattan directions, or similarly, a single rotation. We run our HFL method and compared it against (1) the globally optimal line clustering approach proposed by **Bazin** et al. [96], and (2) the rotation obtained using the three most orthogonal VDs of **Tardif** after fitting a perfect orthogonal frame [91]. The results are shown in Figure 4.8. Despite of the close performance in terms of estimating the three orthogonal VDs, our method is computational more efficient, running more than 50 times faster than **Bazin**.

4.6.2 YUD using extracted edges

In this section, we test our HFL algorithm for detecting the Manhattan frame in the YUD, but using edges extracted trough Tardif's detector [2] instead of the line



(a) We simultaneously detected the Manhattan directions (red,green,blue) and 1 non-orthogonal VD (magenta).



(b) Our algorithm detected 2 orthogonal triplets (middle and right), and assigned the blue direction as being common for both.

Figure 4.10: Two cases from the YUD. The left images show the extracted edges (orange), while the detection results are shown on the right. Black lines were assigned to the empty set (no VP).

segments supplied by the database (see Figure 4.10). The comparison with respect to **Tardif** is shown in Figure 4.9. We consistently outperform **Tardif**, reaching 100% of success only approximately 1.5 pixels later than using the ground truth lines, which proves the robustness of our approach. The **Bazin** method was not included in this experiment due to its higher computational cost when compared to **Ours** and **Tardif**.

Figure 4.10 shows two particularly interesting results obtained by our approach (refer to Figure A.1 and Figure A.2 for additional results). Using the HFL, we correctly identified in Figure 4.10(a) the Manhattan frame and simultaneously estimated the VDs corresponding to the handrails of the stairs. In Figure 4.10(b), we identified two different mutually orthogonal triplets (remark that for the analysis in Figure 4.9, the orthogonal triplet with more lines was automatically selected), one corresponding to the Manhattan frame and the other is due to the squares on the floor. We also identified that both orthogonal frames share the same vertical direction.

4.6.3 Scenes containing multiple orthogonal triplets

This section shows experiments on real images containing more than one orthogonal triplet of VDs. The images shown in Figure 4.11 were obtained using a Panasonic DMC digital camera (refer to Figure A.3 and Figure A.4 for additional results), while the image shown in Figure 4.12 was downloaded from Flickr. The

Figure 4.11: (Left) extracted edges, and (right) detection results. In each example (row) we detected 2 groups of orthogonal triplets with the blue VD in common. The VD in magenta (row 3) was detected but (incorrectly) not assigned to any triplet.

Figure 4.12: (Row 1) extracted edges (left) and clustering obtained using UFL (right); (Row 2) 3 groups of orthogonal triplets were detected using HFL, the 2 on the left have the blue VD in common

input edges for our UFL and HFL algorithms are obtained using Tardif's edge detector. We are able to handle high-resolution images containing many edges, detecting simultaneously both multiple orthogonal triplets as well as single VDs. Figure 4.12 shows results for both the UFL labeling (top,right) and the following HFL procedure (bottom). There is one error in the hierarchical clustering (bottom, middle). Our approach mistakenly assigns the edges on the roof to one orthogonal

triplet, but this issue can be a consequence of either a poor estimation of the focal length or an ineffective tuning of the facility costs for HFL.

4.7 Conclusions

We presented an automatic and global approach for the detection of VPs and mutual orthogonal VDs. The core of the framework is the formulation of these multi-model fitting problems as UFL and HFL instances, which are solved using a message passing approach. The effectiveness of the framework is proved by challenging real scenarios containing multiple Manhattan-world configurations.

Chapter 5

Piecewise Planar Reconstruction using two views

This chapter describes a reconstruction pipeline that generates piecewise-planar models of man-made environments from two calibrated views. The 3D space is sampled by a set of virtual cut planes using SymStereo, implicitly defining possible pixel correspondences. The likelihood of these possible correspondences is measured using logN (see Chapter 2), obtaining profile contours of the 3D scene that become lines whenever the virtual cut planes intersect planar surfaces. The detection and estimation of these lines cuts *is formulated as a global optimization problem over the symmetry matching cost, and pairs of reconstructed lines are used to generate plane hypotheses that serve as input to PERL clustering [3]. Our PERL formulation alternates between a discrete optimization step, that merges planar surface hypotheses and discards detections with poor support, and a continuous optimization step, that refines the plane poses. The pipeline outputs a semi-dense PPR of the 3D scene. In addition, the input images can then be segmented into piecewise-planar regions by using a standard MRF formulation for assigning pixels to plane detections. Experiments with both indoor and outdoor stereo pairs show significant improvements over state-of-the-art methods with respect to accuracy and robustness.*

5.1 Introduction

As discussed in previous chapters, stereo reconstruction is a classical problem in computer and robot vision that deserved the attention of thousands of authors [23, 29]. Despite of the many advances in the field, situations of poor texture, vari-

able illumination, severe surface slant or occlusion are still challenging for most stereo matching methods, making it difficult to find a tuning that provides good results under a broad variety of acquisition circumstances [107]. Since man-made environments are dominated by planar surfaces, several authors suggested to overcome the above mentioned difficulties by using the planarity assumption as a prior for the stereo reconstruction [64, 108, 7, 8, 9]. These approaches have the advantage of providing piecewise-planar 3D models of the scene that are perceptually pleasing and geometrically simple, and, thus, their rendering, storage and transmission is computationally less complex. This chapter proposes a pipeline for two-view Piecewise Planar Reconstruction (PPR) understood as the detection and reconstruction of dominant planar surfaces in the scene[1].

As many multi-model fitting problems (e.g. Chapter 4), PPR is in a large extent a *chicken-and-egg* problem. If there is accurate 3D evidence about the scene, such as points, lines, VDs, etc, then the problem of detecting, segmenting, and estimating the pose of dominant planes can be potentially solved using standard model fitting techniques [109, 3]. On the other hand, if there is a prior knowledge about the dominant planes in the scene, then the matching process can be constrained to improve the accuracy of the final 3D reconstruction, e.g. the known plane orientations can be used to guide the stereo aggregation [51]. Existing methods for PPR typically comprise three steps that are executed sequentially:

- *3D Reconstruction:* the objective is to collect 3D evidence about the scene from multiple views. This evidence can either be obtained from *sparse stereo* that matches a sparse set of features across views, or from *dense stereo* (refer to Chapter 2).

- *Plane Hypotheses Generation:* given the 3D data, the objective is to detect and estimate the pose of planar surfaces using some sort of multi-model fitting approach.

- *Plane Labeling:* the goal is to assign to each image pixel one of the plane hypotheses generated in the previous step.

While most methods were originally designed to receive multiple views as input [64, 110, 108, 7, 8, 9], we propose a pipeline that uses only two views and makes no assumptions about the scene other than the fact of being dominated by planar surfaces. The novelty is mainly in the steps of *3D Reconstruction* and *Plane Hypothesis Generation*, and the contributions are twofold.

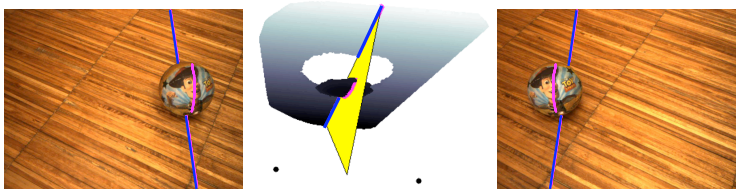
First, we propose a new approach for the reconstruction of line cuts using Sym-Stereo. As discusses previously, establishing dense stereo correspondence is computationally expensive specially when dealing with high-resolution images. On the other hand, sparse stereo applied to only two views tends to provide insufficient 3D

[1] We mean by PPR something that is different from approximating surfaces by small planes, as typically done in several dense stereo methods (e.g. [51, 53])

data for establishing accurate plane hypotheses. Thus, we propose to carry a semi-dense reconstruction of the scene by independently recovering depth along a set of pre-defined virtual planes using SRF (refer to Chapter 2 and Chapter 3). Since the virtual scan planes must intersect the plane surfaces into lines, we extract line segments from the profile cuts and use these *line cuts* to generate plane hypotheses.

The second contribution is a global plane fitting formulation based on PEARL [3]. Most methods for PPR treat stereo matching and plane detection in a sequential and independent manner [64, 110, 108, 7, 8, 9]. This is problematic because the accuracy of the plane hypotheses is inevitably limited by the accuracy of the initial 3D reconstruction that does not take into account the fact of the scene being dominated by planar surfaces. We carry the 3D reconstruction and the plane fitting in a simultaneous and integrated manner using the recent PEARL framework proposed in [3]. The algorithm alternates between a global discrete optimization step, that considers VD and crease edges to merge plane surface hypotheses and discards spurious detections, and a continuous optimization step over the symmetry energy, that refines the plane pose estimation taking into account surface slant. The output of the proposed pipeline is a set of plane hypotheses and a semi-dense PPR of the 3D scene where the reconstructed line cuts are labeled according to the plane detections.

5.1.1 Planarity prior for SymStereo



(a) Left Image $\mathtt{I}$ (b) 3D view (c) Right Image $\mathtt{I}'$

Figure 5.1: As discussed in Section 2, the objective of SRF is to estimate the profile cut (magenta and blue), corresponding to the intersection of a virtual cut plane $\mathbf{\Pi}$ (yellow) with the scene. In the case $\mathbf{\Pi}$ intersects a planar surface, then the profile cut is a 3D line segment (blue).

Let's assume a particular virtual cut plane that intersects a planar surface in the scene (refer to Figure 5.1). The profile cut of intersection between these two planes is a 3D line. The pipeline that is proposed in this chapter uses this prior for performing PPR. The basic idea is to search for *line cuts* along a discrete set of virtual cut planes, which are then used for posing plane hypotheses in the scene.

5.2 Related Work

Several works in PPR start by obtaining a sparse 3D reconstruction of the scene (e.g. point clouds, edge lines, etc), then establish plane hypotheses by applying multi-model fitting to the reconstructed data, and finally use these hypotheses to guide the dense stereo process and/or perform a piecewise planar segmentation of the input images [64, 110, 108]. Werner and Zisserman use multiple cues and assumptions to find dominant surface orientations, and then perform plane-sweep reconstruction along the detected normal directions. Pollefeys et al [108] detect planar surfaces in urban environments from 3D point features obtained from Structure from Motion (SfM), and use the estimated normals for guiding plane-sweep stereo.

Furukawa et. al [7] propose to perform PPR assuming a Manhattan-world model. They reconstruct 3D patches in textured image regions from multiple views using [111], and use the normals of these patches to establish plane hypotheses. These hypotheses are then used in a MRF formulation for pixel-wise plane labeling. In [8], Sinha et al. propose a probabilistic framework for assigning plane hypotheses to pixels with the evidences of planar surfaces being provided by the estimation of VPs, and the reconstruction of sparse feature points and line segments. Gallup et al [9] propose a stereo method capable of handling both planar and non-planar objects contained in the scene. A robust procedure based on RANSAC is used for fitting plane hypotheses to dense depth maps, followed by a MRF formulation for plane labeling of the input images.

An alternative strategy is to over-segment the stereo images based on color information and fit a 3D plane to each non-overlapping region. The number of planes to be considered is defined by the segmentation result, which acts as a smoothness prior during the global optimization. This segmentation information is either used as a hard minimization constraint [50, 112, 48] or as a soft constraint [113]. The main weakness of this type of strategy is the assumption that planar surfaces in the scene have different colors, which is often not the case in most man-made environments (e.g. walls, doors, windows, etc).

The drawback of the approaches described so far is the fact that depth estimation and plane fitting are carried in a sequential and decoupled manner. The errors in the extracted 3D evidence may affect the accuracy of the plane pose estimation, and the inferred planar surfaces are not used for refining the initial depth estimates.

There are a few approaches [114, 115, 116] that perform PPR by carrying stereo matching and 3D plane fitting iteratively. The strategy consists in alternating between segmenting the input images into non-overlapping regions and estimating the plane parameters for each region. However, and as stated by the authors of [115], these types of algorithms can become stuck in a local minimum whenever they face challenging surface structures e.g. surfaces with low and/or repetitive texture.

5.3 Background

This section briefly reviews two background concepts that are used throughout the chapter, namely the energy-based multi-model fitting framework called PEARL (Section 5.3.1), and a global pixel-wise plane labeling formulation (Section 5.3.2). There is no major novelty, so that readers that are familiar with these concepts can skip the section.

5.3.1 Energy-based multi-model fitting using PEARL

As briefly discussed in Chapter 4, Isack and Boykov argued in [3] that methods that greedily search for models with most inliers while ignoring the overall classification of data are a flawed approach to multi-model fitting, and that formulating the fitting as an optimal labeling problem with a global energy function is preferable. For this purpose, they propose the PEARL algorithm consisting in three main steps:

1. *Propose* an initial set of plausible models (labels) $\mathcal{L}_0$ from the observations
2. *Expand* the label set for estimating its spatial support (inlier classification)
3. *Re-estimate* the inlier models by minimizing some error function.

Given the initial model set $\mathcal{L}_0$, the multi-model fitting is cast as a global optimization where each model in $\mathcal{L}_0$ is interpreted as a particular label f. Consider that $d \in \mathcal{D}$ is a data point and that f_d is a particular label in $\mathcal{L}_0$ assigned to d. The objective is to compute the labeling $\mathbf{f} = \{f_d | d \in \mathcal{D}\}$ such that the following energy is minimized:

$$E(\mathbf{f}) = \underbrace{\sum_{d \in \mathcal{D}} D_d(f_d)}_{\text{data term}} + \lambda_S \underbrace{\sum_{d,e \in \mathcal{N}} V_{d,e}(f_d, f_e)}_{\text{smoothness term}} + \underbrace{\lambda_L \cdot |\mathcal{F}_f|}_{\text{label term}} \tag{5.1}$$

where $\mathcal{N}$ is the neighborhood system considered for d, $D_d(f_d)$ is some error that measures the likelihood of point d belonging to model f_d, and $V_{d,e}$ is the spatial smoothness term that encourages piecewise smooth labeling by penalizing configurations $\mathbf{f}$ that assign to neighboring nodes d and e different labels. The label term is used for describing the data points using as few unique models as possible, with $\mathcal{F}_f$ being the subset of different models assigned to the nodes d by the labeling $\mathbf{f}$ (see [3] for further details). In order to handle outlier data points in $\mathcal{D}$, the outlier label $f_\emptyset$ is added to $\mathcal{L}_0$. Any point d to which is assigned the label $f_\emptyset$ is considered an outlier, and has a constant likelihood measure $D_d(f_d = f_\emptyset) = \tau$. The energy of Equation 5.1 is efficiently minimized using α-expansion [3].

Finally, the third step of PEARL consists in re-estimating the model labels f in $\mathcal{L}_0$ with non-empty set of inliers $\mathbf{D}(f) = \{d \in \mathcal{D} | f_d = f\}$. Let $\mathbf{m}_f$ be the model

associated to the label f. Each model $\mathbf{m}_f$ is refined by minimizing the error cost over its parameters:

$$\mathbf{m}_f^* = \min_{\mathbf{m}_f} \sum_{d \in \mathbf{D}(f)} D_d(f).$$

The models with non-empty set in $\mathcal{L}_0$ are replaced with the refined models $\mathbf{m}_f^*$, and the labels with empty set are discarded. The new set of labels $\mathcal{L}_1$ is then used in a new expand step, and we iterate between discrete labeling and plane refinement until the α-expansion optimization does not decrease the energy of Equation 5.1.

5.3.2 MRF for Plane Labeling

Given a set of plane hypotheses contained in the scene, many PPR algorithms perform a pixel-wise plane labeling of the input images. We follow a standard MRF formulation for comparing all the tested algorithms. The objective is to minimize an energy involving data, smoothness and labeling terms (refer to Equation 5.1). In this case, the nodes $d \in \mathcal{D}$ are the image pixels, and the labels $f \in \mathcal{P}$ are the plane hypotheses. A 4×4 neighborhood $\mathcal{N}_4$ is assumed for neighboring pixels $\mathbf{d}$ and $\mathbf{e}$, and the data term is defined as

$$D_d(f) = \begin{cases} \min(\rho_d(f), \rho_{max}) & \text{if } f \in \mathcal{P} \\ \gamma \rho_{max} & \text{if } f = f_\emptyset \end{cases}$$

where $\rho_d(f)$ is the photo-consistency between the pixels in the two views put into correspondence by the plane associated to label f. For measuring the photo-consistency, we use ZNCC (refer to Chapter 2). The photo-consistency metric is given by $\rho_d(f) = (1 - ZNCC(f))/2$, where $ZNCC(f)$ is the cost obtained using ZNCC for the plane hypothesis f (ρ_{max} and γ are constant parameters).

The smoothness term is defined as:

$$V_{d,e}(f_d, f_e) = g \cdot \begin{cases} 0 & \text{if } f_d = f_e \\ M & \text{if } (f_d \vee f_e) = f_\emptyset \\ D' & \text{otherwise} \end{cases}$$

where

$$D' = \min(D, M) + m \quad \text{and}$$
$$g = \frac{1}{\Delta I^2 + 1},$$

D is the 3D distance between neighboring points according to their plane f_d and

f_e, respectively, M and m are constant parameters, and

$$\Delta I = |I(d) - I(e)|.$$

is the image gradient.

The data and the smoothness terms are, with minor differences, similar to the ones used in the graph-cut labeling of Gallup et al. [9]. We additionally add the labeling term $\lambda_L|\mathcal{F}_f|$ for avoiding very close plane hypotheses in $\mathcal{P}$ to be assigned in $\mathbf{f}$. This has the effect of simplifying the 3D model using as few unique planes as possible.

5.4 Reconstruction of lines along a single cut plane

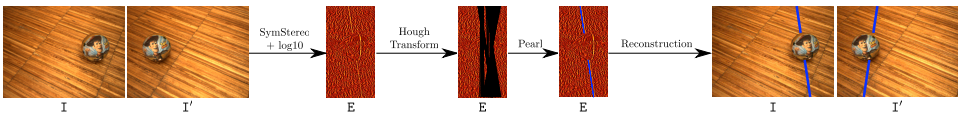


Figure 5.2: Reconstruction of 3D line cuts from a stereo pair using SymStereo along a virtual cut plane $\mathbf{\Pi}$. The log10 cost is employed for obtaining the joint energy $\mathtt{E}$, which is used as input to a weighted Hough transform for extracting line cuts (black lines). The most appropriate hypotheses (in this example only one line cut (blue) is detected) are then selected using a global framework constituted by data, smoothness and label costs. Finally, the line cuts are mapped back onto the right view using the inverse of H.

The reconstruction of lines from two or more views has in the vast majority of existing algorithms one common denominator: the detection of line segments in the input views that are matched in subsequent steps. In the case there are no (salient) line segments in the input images, then no 3D line reconstructions can be obtained.

This section describes an algorithm that reconstructs a set of 3D line cuts along a single virtual cut plane $\mathbf{\Pi}$ using SRF. This is achieved by noting that the intersection of $\mathbf{\Pi}$ with a plane in the scene is a line (e.g. in Figure 5.2, the intersection of $\mathbf{\Pi}$ with the floor plane is the blue line cut). The 3D lines corresponding to the intersection of $\mathbf{\Pi}$ with multiple planes are projected onto the stereo views as line segments, whose locations in most of the cases cannot be perceived only from the input images alone (there are no visible edges). However, these lines can be reliably detected and estimated from the joint symmetry and anti-symmetry energy E that is obtained from $\mathbf{\Pi}$. Remark that each line cut that is detected from a virtual cut plane corresponds to a particular plane contained in the scene. However, the corresponding parameters cannot be estimated from a single cut plane (we will see in Section 5.5 how to detect and estimate planes based on the information of more than one virtual cut plane).

5.4.1 Line cut detection using Hough and PEARL

As shown in Figure 5.2, we use the SymStereo framework along a virtual cut plane $\mathbf{\Pi}$ and employ the log10 symmetry metric for computing the joint energy $\mathtt{E}$. Each pixel in $\mathtt{E}$ provides the matching likelihood of a particular pair of pixels in the stereo views, being an indirect measurement of the occupancy probability in 3D along $\mathbf{\Pi}$. The energy $\mathtt{E}$ is used as input to a weighted Hough transform for extracting a set of line cut hypotheses $\mathcal{L}_0$ (a similiar approach was also used in Section 3.5). This is accomplished by selecting the N_H local maxima in the Hough voting space.

Next, we formulate the line cut detection as a global labeling problem in a PEARL framework, in which the objective is to assign to each epipolar line (image row) a line cut hypothesis in $\mathcal{L}_0$. Following the notation of Section 5.3.1, the data points d of the graph are the epipolar lines, with the size of the set $\mathcal{D}$ being equal to the number of image rows, and the goal is to assign a line segment label f to each epipolar line d. The data term is defined as

$$D_d(f) = \begin{cases} \min(1 - \mathtt{E}(d, x_f), \tau) & \text{if } f \neq f_\emptyset \\ \alpha_\emptyset \tau & \text{otherwise} \end{cases}$$

where $\mathtt{E}(r, c)$ is the joint energy value for row r and column c. The coordinate x_f corresponds to the intersection between the epipolar line d and the line segment $\mathbf{l}_f$ associated to label f. Remark that the truncation parameter τ is used for handling poorly matching surfaces e.g. containing low and/or repetitive textures, while the discard label $f_\emptyset$ indicates that no satisfactory line cut hypothesis can be assigned to d. In this case, the virtual cut plane $\mathbf{\Pi}$ has high probability of not intersecting a planar surface along the epipolar plane associated to d.

The smoothness term of neighboring nodes d and e is given by

$$V_{de}(f_d, f_e) = \begin{cases} 0 & \text{if } f_d = f_e \\ \lambda_\emptyset & \text{if } (f_d \vee f_e) = f_\emptyset \\ \frac{1}{\Delta I^2 + 1} & \text{otherwise} \end{cases}$$

where

$$\Delta I = |I(d, x_{f_d}) - I(e, x_{f_e})|$$

is the image gray-scale gradient. No penalization is assigned to neighboring image rows d and e receiving the same label, while in the case one node receives the label $f_\emptyset$, then a non-zero cost $\lambda_\emptyset$ is added to $\mathbf{f}$. The smoothness term V prefers label transitions at locations of larger image gradient (lower smoothness cost), which usually occurs at the boundaries of two different surfaces. We use a constant label term λ_L in Equation 5.1 for favoring line cut assignments $\mathbf{f}$ with fewer labels.

Finally, and after computing an initial labeling solution $\mathbf{f}$ for nodes d, the line cuts $\mathbf{l}$ are refined by minimizing their parameters over the energies $\mathtt{E}$ via Levenberg-

Marquardt (LM) [106]

$$\mathbf{l}_f^* = \min_{\mathbf{l}_f} \sum_{d \in \mathbf{D}(f)} \left(1 - \mathtt{E}(d, x_f)\right), \tag{5.2}$$

where $\mathbf{D}(f)$ is a subset of image rows d to which the label f was assigned. Remark that at each solver iteration, the point x_f on d is recomputed according to the current line cut hypothesis $\mathbf{l}_f$. The new set of line cuts $\mathbf{l}_f^*$ are then used in a new global line cut assignment (expand) step, and we iterate between discrete labeling and line cut refinement until the energy of Equation 5.1 stops decreasing (which usually occurs after $2-3$ iterations).

5.4.2 Experiments in line cut detection

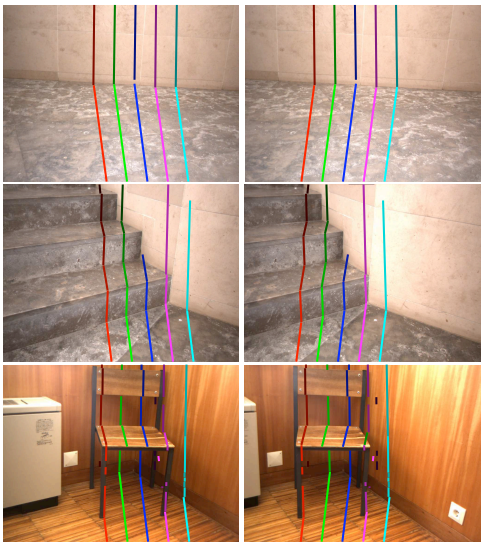
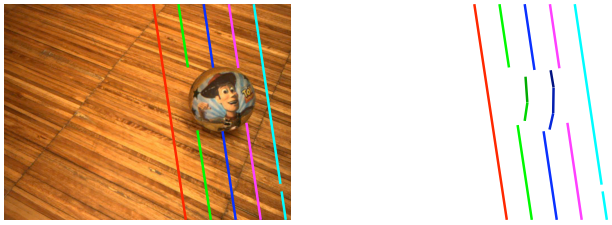


Figure 5.3: Results produced by our line cut detection algorithm along 5 virtual cut planes. We show for each example the left and right views with the detected line cuts overlaid; different colors indicate different cut planes, while different shades identifying different line cuts.

We performed experiments of our line cut detection approach[2] on various indoor scenes (see Figure 5.2-5.4) acquired using a Bumblebee stereo camera from PointGrey, which has a baseline of 24 cm and image resolution of 1024×768 pixels.

In the first example of Figure 5.3, we detect 2 different line cuts for each virtual cut plane, one corresponds to the intersection of the cut planes with the floor

[2]We used for all the experiments the same parameters: $N_H = 200$, $\lambda_S = 1, \tau = 0.8$, $\alpha_\emptyset = 0.7$, $\lambda_\emptyset = 0.9$ and $\lambda_L = 20$ that were empirically selected.



(a) $\alpha_\emptyset = 0.7$ and $\lambda_L = 20$ (b) $\alpha_\emptyset = 0.8$ and $\lambda_L = 10$

Figure 5.4: Results for two different settings of $\alpha_\emptyset$ and λ_L. By varying these parameters, we can control the algorithm to be more permissive with respect to what is considered a line cut (b), while for lower values of $\alpha_\emptyset$ and higher values of λ_L the algorithm only detects line segments with high probability of belonging to planar surfaces (a).

and one is due to the intersection with the wall. Remark that the matching of the line segments across the views is almost perfect and consistent for all virtual cut planes, even tough the line cut detection was carried for each virtual cut plane independently. In the example (b), the scene consists of multiple planar surfaces, some containing quite complicated textures. In this case, the line cut estimation approach along a single virtual cut plane begins to have difficulties. In situations the cut plane intersects the scene in low-textured regions, the symmetry based matching using log10 does not provide a well defined ridge at the locations of the image of the profile cut. Following this, the algorithm prefers to label those regions with the $f_\emptyset$ label (e.g. blue cut plane), since it has low confidence about the location of the image of the profile cut. Finally, the example (c) presents some failure cases of this approach, namely slanted surfaces with low-texture. In this cases, the algorithm tends to (i) assign more than one line cut label that corresponds to the same planar surface (noisy energy E), (ii) does not detect the line cut at all, or (iii) computes wrong matches. Note that (i) could be handled by increasing the label cost λ_L, however this would imply that line cuts corresponding to close planes (e.g. chair backs and wall in example (c)) are assigned the same label. We will show in Section 5.5 that most of these difficulties are handled by our PPR algorithm that jointly estimates plane hypotheses from multiple virtual cut planes simultaneously.

So far most of the examples contained only planar surfaces. We show in Figure 5.4 a scene containing a non-planar object above the floor plane. The control of labeling just *strict* planes (example (a)) or approximate non-planar surfaces by an appropriate set of planes (example (b)) is achieved using different settings of the weighting factor $\alpha_\emptyset$ and the label cost λ_L. Using low values of $\alpha_\emptyset$ and high values of λ_L implies that only line cuts belonging to planar surfaces are reconstructed, while higher values of $\alpha_\emptyset$ and low values of λ_L enable to approximate non-planar surfaces by various plausible line cuts.

5.5 PPR using SymStereo and PEARL

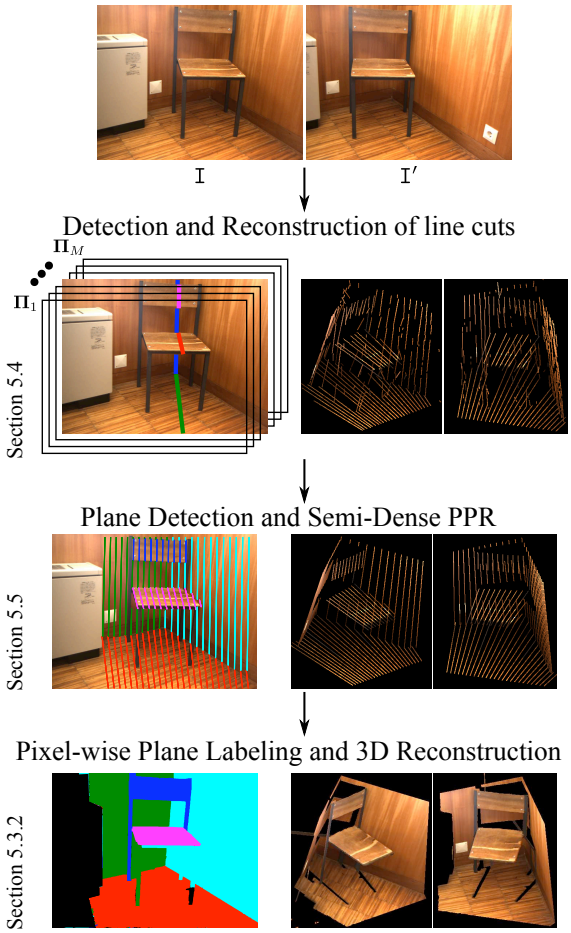


Figure 5.5: Pipeline for PPR using a pair of calibrated images: (1) apply the line cut detection algorithm described in Section 5.4 along M virtual cut planes for obtaining a sparse set of 3D line cuts; then (2) use the global semi-dense PPR algorithm described in Section 5.5 for computing planar surfaces and obtain a semi-dense PPR; use the line cuts estimated in (1) for obtaining plane hypotheses; and (3) use the global pixel-wise plane labeling (Section 5.3.2) for computing a dense PPR model from the plane hypotheses in (2).

This section describes an algorithm that combines the SymStereo framework (refer to Chapter 2) with the geometric multi-model fitting algorithm PEARL [3] for semi-dense PPR (see Figure 5.5). The input to this algorithm are M joint energies $\mathtt{E}_i$ that were computed using $log10$ from a set of M virtual cut planes $\mathbf{\Pi}$ that belong to a vertical pencil intersecting the baseline in its midpoint. The output are a discrete set of planar surfaces and a semi-dense 3D reconstruction, where each reconstructed point belongs to a particular plane. The detected planes can then be used as plane hypotheses in a global plane labeling strategy for computing

a dense model (see Section 5.3.2).

5.5.1 Formulation of the global framework

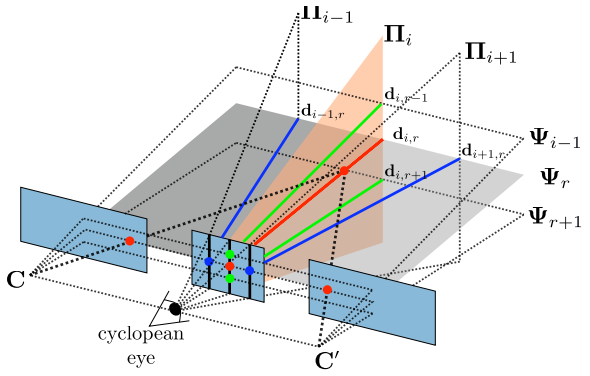


Figure 5.6: The scene is sampled by a discrete set of virtual cut planes $\mathbf{\Pi}_i$. This can be thought as an image created by a virtual camera that is located between the cameras (cyclopean eye), where each epipolar plane $\mathbf{\Psi}_r$ projects onto one row and each $\mathbf{\Pi}_i$ projects onto one column of the image. Each pixel of the cyclopean eye is originated from the back-projection ray $\mathbf{d}_{i,r}$ (red), corresponding to the intersection between $\mathbf{\Pi}_i$ and $\mathbf{\Psi}_r$.

Consider that the midpoint of the baseline is the center of projection of a virtual camera, which will be called the cyclopean eye (see Figure 5.6). The image that is perceived by the cyclopean eye has height equal to the number of epipolar planes $\mathbf{\Psi}_r$ with $r = 1, ..., R$ (one epipolar plane per image row), and the width is given by number of virtual cut planes $\mathbf{\Pi}_i$ with $i = 1, ..., M$ (one cut plane for each column). Each pixel of the cyclopean eye is originated by the back-projection ray $\mathbf{d}_{i,r}$, which corresponds to the line of intersection between $\mathbf{\Pi}_i$ and $\mathbf{\Psi}_r$. The objective is to estimate the point on each $\mathbf{d}_{i,r}$ that most likely belongs to a planar surface. This problem is cast as a labeling problem following a PEARL framework, as described in Section 5.3.1. The nodes of the graph are the back-projection rays $\mathbf{d}_{i,r}$ of the cyclopean eye, and to each $\mathbf{d}_{i,r}$ we want to assign a plane label f_d. The set of possible labels is $\mathcal{L}_0 = \{\mathcal{P}_0, f_\emptyset\}$, with $f_\emptyset$ meaning that no point on $\mathbf{d}_{i,r}$ belongs to a planar surface. Note that we use $\mathbf{d}$ instead of $\mathbf{d}_{i,r}$ whenever the virtual and epipolar plane specifications are not strictly necessary. We assume a $\mathcal{N}_4$ neighborhood for $\mathbf{d}_{i,r}$ that is defined by the four back-projection rays $\mathbf{d}_{i\pm1,r}$ and $\mathbf{d}_{i,r\pm1}$ (see Figure 5.6).

5.5.2 Initial plane hypotheses

As discussed in Section 5.4, each line cut is a possible location of intersection of a virtual cut plane with a planar surface in the scene. In order to propose an initial set of plane models $\mathcal{P}_0$ for PEARL, we could generate all possible planes that can

be obtained from two line cuts belonging to different planes $\mathbf{\Pi}$. However and depending on the number of cut planes that are used, the set $\mathcal{P}_0$ can easily become very large. We noticed that using only pairs of line cuts from neighboring cut planes $\mathbf{\Pi}_{i\pm1,2}$ drastically decreases the size of $\mathcal{P}_0$ and is enough for initializing our piecewise-planar labeling approach. Since it is unlikely that line cuts intersecting different epipolar planes correspond to the same planar surface, we further reduce $\mathcal{P}_0$ and only use pairs of line cuts that have a minimum of N_E epipolar lines of overlap ($N_E = 10$).

5.5.3 Data and smoothness term

The data term $D_{\mathbf{d}_{i,r}}$ for the back-projection ray $\mathbf{d}_{i,r}$ is defined as

$$D_{\mathbf{d}_{i,r}}(f) = \begin{cases} \min(1 - \mathtt{E}_i(r, x_f), \tau) & \text{if } f \in \mathcal{P}_0 \\ \tau & \text{if } f = f_\emptyset \end{cases}$$

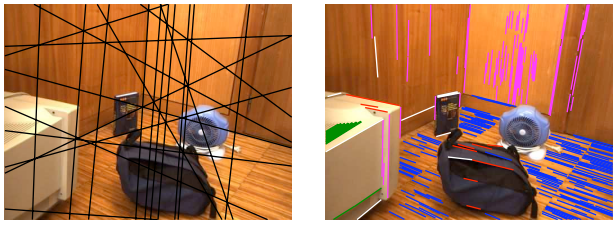
where $\mathtt{E}_i$ is the joint energy associated with the virtual cut plane $\mathbf{\Pi}_i$, r is the row corresponding to the epipolar plane $\mathbf{\Psi}_r$ and τ is a constant. The coordinate x_f is the column defined by the plane hypothesis f, corresponding to the intersection of $\mathbf{d}_{i,r}$ with the plane indexed by f. Note that similarly to [9], the non-planar $f_\emptyset$ label indicates that no satisfactory plane hypothesis can be assigned to $\mathbf{d}_{i,r}$. In this case, the back-projection ray $\mathbf{d}_{i,r}$ has high probability of not intersecting the scene in a planar surface.

Inspired by the work of Sinha et al. [8], the smoothness term for neighboring nodes $\mathbf{d}$ and $\mathbf{e}$ is given by

$$V_{\mathbf{de}}(f_\mathbf{d}, f_\mathbf{e}) = \begin{cases} 0 & \text{if } f_\mathbf{d} = f_\mathbf{e} \\ \lambda_1 & \text{if } (\mathbf{d}, \mathbf{e}, f_\mathbf{d}, f_\mathbf{e}) \in \mathsf{S}_1 \\ \lambda_2 & \text{if } (\mathbf{d}, \mathbf{e}, f_\mathbf{d}, f_\mathbf{e}) \in \mathsf{S}_2 \\ \lambda_3 & \text{if } (\mathbf{d}, \mathbf{e}) \in \mathsf{S}_3 \\ \lambda_4 & \text{if } (f_\mathbf{d} \vee f_\mathbf{d}) = f_\emptyset \\ 1 & \text{else} \end{cases}$$

where $0 < \lambda_1 < \lambda_2 < \lambda_3 < 1$, and the content of the sets S_1, S_2 and S_3 is described next. Remark that no penalization is assigned to neighboring nodes receiving the same plane label, while in the case of one node obtaining the discard label $f_\emptyset$, a non-zero cost λ_4 is added to the plane configuration $\mathbf{f}$.

Following a similar reasoning [8], plane transitions between neighboring nodes $\mathbf{d}$ and $\mathbf{e}$ are more likely to occur in the presence of crease or occlusion edges. A crease edge corresponds to the projection of the 3D line of intersection between two different planes in the scene, while occlusion boundaries arise from spatially separated objects in 3D whose image projections interfere with each other.



(a) Crease Edges (b) Detected Line Segments

Figure 5.7: We show in (a) some crease edges obtained from intersections of two different planes contained in $\mathcal{P}_0$, while in (b) the result of the clustering of concurrent lines is shown. Each group of lines (different groups have different colors) provides a possible vanishing point location. The white line segments did not received any vanishing point label.

Let the point $\mathbf{p}_{\mathbf{d},f_\mathbf{d}}$ ($\mathbf{p}_{\mathbf{e},f_\mathbf{e}}$) be the projection onto $\mathtt{I}$ of the intersection between $\mathbf{d}$ ($\mathbf{e}$) and the plane associated to f_d (f_e). In order to encourage plane label transitions at crease edges, we store in the set S_1 the quadruples $(\mathbf{d}, \mathbf{e}, f_\mathbf{d}, f_\mathbf{e})$ in which the points $p_{\mathbf{d},f_d}$ and $p_{\mathbf{e},f_e}$ are located on different sides of the crease edge defined by f_d and f_e. Whenever a labeling configuration $\mathbf{f}$ contains assignments located in S_1, then it incurs a penalization λ_1 (Figure 5.7(a) shows some crease edges that are estimated from real imagery).

Occlusion edges are usually coincide with visible 2D line segments in the input views and often are also aligned with the vanishing directions of scene planes (Figure 5.7(b)). In order to find possible occlusion edges, we detect 2D line segments in the left view $\mathtt{I}$ using the Line Segment Detector [117]. Each line segment is a possible location of an occlusion boundary. For clustering concurrent lines we use the global vanishing point (VP) detection algorithm proposed in Chapter 4. The set S_2 contains the quadruples $(\mathbf{d}, \mathbf{e}, f_\mathbf{d}, f_\mathbf{e})$ where the image points $p_{\mathbf{d},f_d}$ and $p_{\mathbf{e},f_e}$ are located on different sides of a line segment that was clustered to a particular VP, whose direction is orthogonal either to the planes associated to $f_\mathbf{d}$ or $f_\mathbf{e}$. Finally, S_3 contains the remaining pairs $(\mathbf{d}, \mathbf{e})$ whose projections are on different sides of a line segment to which no VP was assigned. Remark that in contrast to [8], we do not perform any line matching between the stereo views, substantially decreasing the complexity of the algorithm.

5.5.4 Plane refinement

The third step of the PEARL algorithm (see Section 5.3.1) is to re-estimate the plane model parameters using the inliers of the discrete labeling $\mathbf{f}$. Let $\mathbf{\Omega}_f$ be the plane associated to f to which has been assigned a non-empty set of inliers $\mathbf{D}(f) = \{\mathbf{d} \in \mathcal{D} | f_\mathbf{d} = f\}$. Each plane $\mathbf{\Omega}_f$ is refined by minimizing its plane

parameters over the energies $\mathtt{E}$ via LM [106]:

$$\mathbf{\Omega}_f^* = \min_{\mathbf{\Omega}_f} \sum_{\mathbf{d}_{i,r} \in \mathbf{D}(f)} \left(1 - \mathtt{E}_i(r, x_{\mathbf{\Omega}})\right), \tag{5.3}$$

where $x_{\mathbf{\Omega}}$ is the column defined by the intersection of $\mathbf{d}_{i,r}$ with $\mathbf{\Omega}$. The new set of labels $\mathcal{P}_1 = \left\{\mathbf{\Omega}_f^*\right\}$ is then used in a new expand step, and we iterate between discrete labeling and plane refinement until the α-expansion optimization does not decrease the energy of Equation 5.1 (which usually takes $2-3$ iterations).

5.5.5 Plane refinement after PEARL

We have discussed in Section 2.4 that SymStereo can be enhanced in case there is slant information available. The output of the global algorithm described previously, is the labeling $\mathbf{f}$ that assigns to each back-projection ray $\mathbf{d}$ a plane $\mathbf{\Omega}$. The intersection of $\mathbf{d}$ with $\mathbf{\Omega}$ defines a 3D point $\mathbf{Q}$, and $\mathbf{\Omega}$ also defines α_1 that is proportional to the 3D slant in the neighborhood of $\mathbf{Q}$. Following this, and as described in Section 2.4, the position $\mathbf{Q}$ can be refined by iteratively optimizing β.

Let $\mathbf{\Omega}$ be the plane associated to label f to which has been assigned a non-empty set of inliers $\mathbf{D}(f) = \left\{\mathbf{d}_{i,r} \in \mathcal{D} | f_{\mathbf{d}_{i,r}} = f\right\}$, and consider that $\mathbf{Q}_{i,r}$ is the intersection between the ray $\mathbf{d}_{i,r}$ and $\mathbf{\Omega}$ (refer to Figure 2.6). For each $\mathbf{d}_{i,r}$, we compute the "ideal" β_1 and obtain a new back-projection ray $\mathbf{d}_{i,r}^1$. The new ray $\mathbf{d}_{i,r}^1$ is located on the same epipolar plane, but on the virtual cut plane intersecting the point $\mathbf{O}^1$ and the previously reconstructed point $\mathbf{Q}_{i,r}$. Given the new plane $\mathbf{\Omega}^1$, a new homography mapping (see Equation 2.5) can be used for inducing improved symmetries, and from which the joint energy $\mathtt{E}_{i,r}^1$ is re-calculated. The new joint energies $\mathtt{E}_{i,r}^1$ are used in a new refinement step using LM (Section 5.5.4). We iterate between re-computing new back-projection rays $\mathbf{d}_{i,r}^n$ and refining $\mathbf{\Omega}^n$ four times.

5.5.6 Results in semi-dense PPR

Figure 5.8 shows 10 different indoor and outdoor results obtained using our semi-dense PPR algorithm. We show for all the stereo pairs some crease edges that can be used as indicators of the accuracy of the plane estimation. Concerning $\tau = 0.8$, the first two scenes (rows) are only composed by planar surfaces that are accurately reconstructed. We added to the scenes of rows 3 and 4 non-planar objects, which are well approximated by one or more plane surfaces. Additionally, the scene of row 4 contains, besides large planar and non-planar surfaces, a small plane corresponding to the blue book, which is well estimated from only 4 virtual cut planes (please see the crease edge between the book and the floor).

We correctly detect in the first example of the stairs data set 9 planar surfaces.

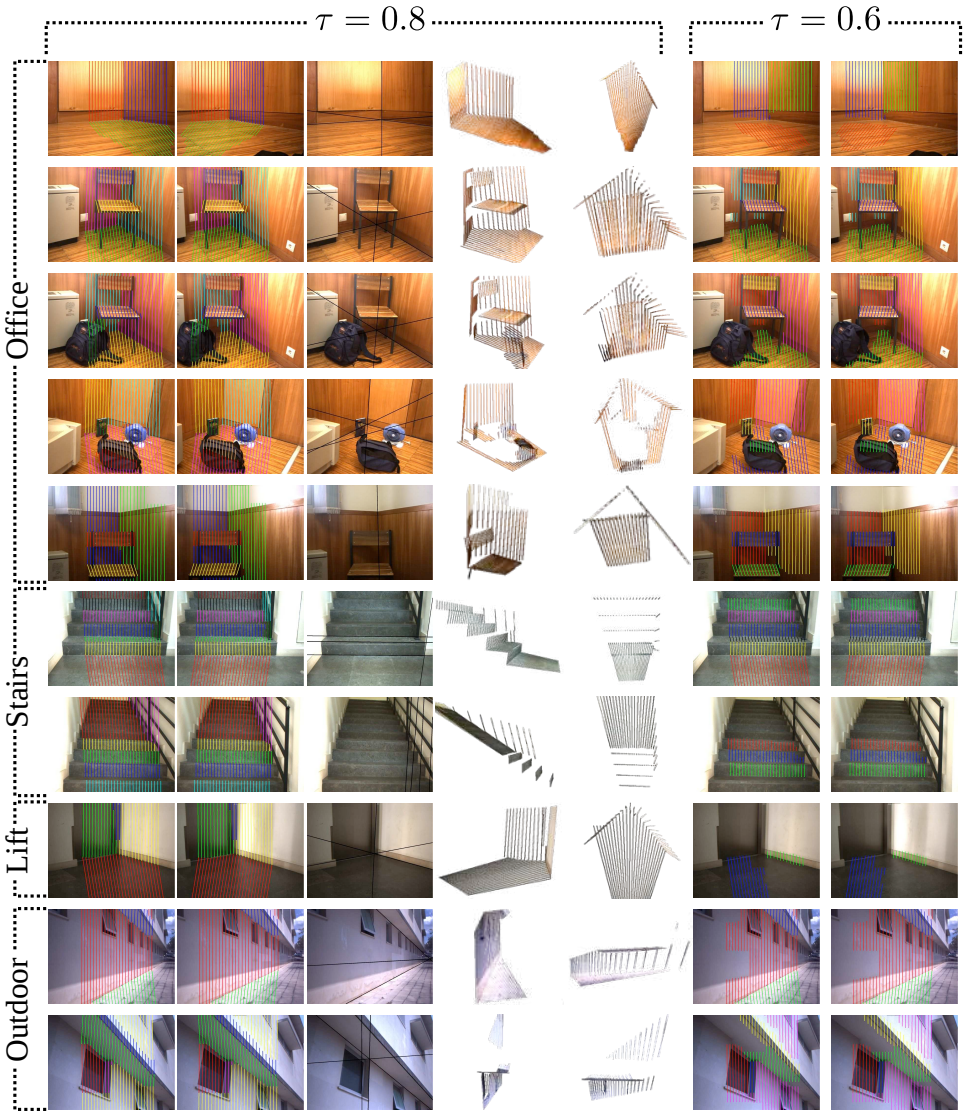


Figure 5.8: Results produced by our semi-dense piecewise planar algorithm. From left to right: the left and right views with the images of the profile cuts overlaid, different colors indicate different planes; the left view with crease edges (black) that can be used as indicators of the estimation accuracy; and two views of the textured 3D reconstruction rendered from different viewpoints.

Whereas in the second example, the top steps are approximated by a single plane (red). This occurs because the image resolution is not sufficient for SymStereo to discriminate depth at such large distances. Additionally, we are able to detect a plane on the right of the stairs that apparently corresponds to the white wall (magenta). However and since the estimation is deceived by the handrail, the plane model seems to be inaccurate. Finally, in the outdoor dataset, we show non-trivial

3D scenarios containing high slant and very low texture, which seem to be correctly handled by our approach.

So far, most of the back-projection rays $\mathbf{d}$ received a plane label even though belonging to non-planar objects (see rows 3 and 4). The control of labeling just *strict* planes can be achieved using the truncation parameter τ. We show on the two right columns of Figure 5.8 results on the same datasets, however decreasing τ from 0.8 to 0.6. In this case, the non-planar objects in rows 3 and 4, as well as the red plane in the second stairs example are not reconstructed, because the algorithm only outputs plane models of which has high confidence of being correct. However, this has the drawback of discarding planes in regions of low texture or containing specular reflections (e.g. rows 5 and 8).

5.5.7 Independent line cut reconstruction vs. semi-dense PPR

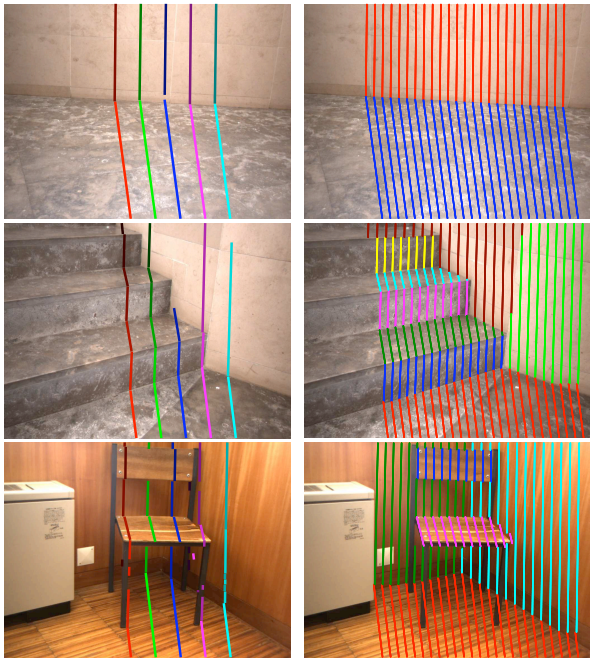


Figure 5.9: Comparison between independent line cut reconstruction along virtual planes and the semi-dense PPR algorithm. For each example, we show the (independent) detection results along 5 virtual cut planes (left), and the final labeling results of the semi-dense PPR for 25 cut planes (right).

We show in Figure 5.9 a brief comparison of the line cut reconstruction algorithm described in Section 5.4 with the semi-dense PPR approach described in this section. In the case the virtual cut planes intersect planar surfaces with some texture and far from object discontinuities, the independent reconstruction along single virtual planes provides accurate results (example (a)). In scenarios contain-

ing multiple planes and complicated textures (examples (b-d)), the independent line cut reconstruction has some difficulties. These problems are solved using our semi-dense PPR pipeline that estimates planar surfaces along different virtual cut planes simultaneously and in a global manner.

5.6 Experiments in PPR

We proposed a new and original algorithm for detecting and estimating planar surfaces in the scene that combines the SymStereo framework and PEARL optimization. For showing the effective advantages with respect to the existing approaches, this section runs a set of experiments in PPR from a pair of stereo images, and compares the performance of the proposed algorithm with respect to the state-of-the-art.

The evaluation is carried on a new data-set comprising challenging indoor and outdoor scenes (some examples are shown in Figures 5.10-5.12). The stereo pairs were acquired using a Bumblebee stereo camera from PointGrey, with a baseline of 24 cm and image resolution of 1024×768 pixels. The scenes contain mostly planar surfaces, including a variety of complicated situations to traditional stereo methods e.g. low and/or repetitive textures, and high surface slant.

5.6.1 Compared Algorithms

The output of our algorithm (**SymS**) is a discrete set of plane hypotheses $\mathcal{P}^{SymS}$ and a semi-dense 3D reconstruction. We compare these plane hypotheses with the ones obtained using two different approaches.

The first applies dense stereo (**DS**) for PPR and was proposed by Gallup et al. [9]. The authors start by obtaining a dense depth map with respect to the left view `I` using local stereo. Then, plane hypotheses are generated using a sequential RANSAC procedure over the disparity map (refer to [9] for details). Finally, a plane linking step is performed for combining near planes and/or single planes that are disjoint in the image. The output of this algorithm is the set $\mathcal{P}^{DS}$ of plane hypotheses and a dense PPR.

The second approach was proposed by Sinha et al. [8], and is based on sparse stereo (**SS**). It detects and computes sparse correspondences, line segments and VDs from the images. From these data, plane hypotheses are generated from specific histogram votings and RANSAC procedures. The output is the set $\mathcal{P}^{SS}$ and a sparse PPR composed by 3D points and 3D line segments.

5.6.2 Accuracy analysis and parameter tuning

The objective is to compare the performance of DS, SymS and SS for generating plane hypotheses for the MRF plane labeling described in Section 5.3.2. Con-

cerning the accuracy analysis, it is difficult to obtain the ground truth (GT) model parameters in each stereo pair of the dataset, which would involve a error prone and time consuming manual selection of point matches in the stereo views. We decided to use a different indicator for measuring the accuracy.

For each stereo pair, we manually define the planar region $\mathcal{R}_k$ in the left view $\mathtt{I}$ that is associated to a particular plane $\mathbf{\Omega}_k$ in the scene (see Figure 5.10). Given the pixel-wise plane labeling $\mathbf{f}$, computed using the plane hypotheses generated from the algorithms described in Section 5.6.1, the accuracy of the estimation of $\mathbf{\Omega}_k$ is evaluated using the following metric:

$$\mathrm{P}_k = \frac{\sum\limits_{\mathbf{p}\in\mathcal{R}_k} \rho_{\mathbf{p}}(f_{\mathbf{p}})}{\#\mathcal{R}_k}, \tag{5.4}$$

where $\#\mathcal{R}_k$ is the number of pixels in the region. Remark that the accuracy analysis using P_k must be performed with caution. There is no guarantee that $\mathrm{P}_k < \mathrm{P}_l$ means the plane $\mathbf{\Omega}_k$ was better estimated than $\mathbf{\Omega}_l$. The proposed metric depends largely on the textures and illumination of the surfaces e.g. planar surfaces with low-texture and specularities will have a large P_k even tough the corresponding plane model is well estimated. On the contrary, we are in the opinion that the metric P_k is adequate for comparing different estimations of the same plane $\mathbf{\Omega}_k$.

Assume that we use two different algorithms for obtaining two different sets of plane hypotheses, say $\mathcal{P}^{A1}$ and $\mathcal{P}^{A2}$, which are used as input to the global plane labeling described in Section 5.3.2. After the graph-cut optimization, we have the assignments $\mathbf{f}^{A1}$ from $\mathcal{P}^{A1}$ and $\mathbf{f}^{A2}$ from $\mathcal{P}^{A2}$ for each image pixel. Following this, we can compute for each GT plane $\mathbf{\Omega}_k$ the photo-consistency metrics P_k^{A1} and P_k^{A2}. In case $\mathrm{P}_k^{A1} < \mathrm{P}_k^{A2}$, then the first algorithm generated a plane hypothesis that better fits the input images, which most probably means that $\mathbf{\Omega}_k^{A1}$ is more accurate than $\mathbf{\Omega}_k^{A2}$. We noticed in practice that this empirical comparison is a very good accuracy indicator in real-world scenarios.

The parameters that are used in the different algorithms were manually tuned using the GT labeling on a subset of stereo pairs of the dataset, whose results are not shown in the experimental comparison. These values are kept constant for all the remaining experiments. Concerning our SymS algorithm, we decided to use $M = 25$ virtual cut planes for the best compromise between accuracy and runtime. Concerning the MRF labeling (see Section 5.3.2), the parameters are constant and the same for all three plane hypotheses generators, namely $\rho_{max} = 0.8$, $\gamma = 0.6$, $m = 1$ and $M = 2$.

5.6.3 Comparison results

The dense PPR results obtained using DS, SymS and SS as plane hypotheses generators for the pixel-wise plane labeling are shown in Figure 5.10.

In the first two examples, the scene is composed by two and three planes, respectively, which are mostly fronto-parallel to the cameras. In these cases, the three algorithms work well and provide approximately similar results. SS has some problems distinguishing the vertical planes in the example (b), which is mainly due to lack of features in the wall on the right. Both examples shown in the second row contain, besides other planes, a highly slanted surface (blue in example (c) and green in example (d)). Our algorithm is able to detect and accurately reconstruct this surfaces, whereas DS and SS clearly have difficulties handling this amount of slant. The examples (e) and (f) show scenes containing many planes at different distances from the camera. SymS is able to detect all the planes and provides the most accurate plane hypotheses, being less sensitive to the surface-camera distance when compared to DS and SS.

The last row shows two examples containing scene with difficult textures and illumination conditions. SS is not able to provide acceptable plane hypotheses for the MRF labeling so that no plane assignment is obtained. DS is still able to cope with the complicated texture of example (g), but completely fails in the example (h), where the joint effect of high slant and repetitive texture are major challenges for dense stereo matching. Our approach recovers all the planes, and can even distinguish the close planes of example (g) corresponding to the floor and the carpet.

Finally and for the sake of completeness, the run-times (without the final MRF labeling) for each algorithm in the images shown in Figure 5.10 are: $1-2$ min for SymS (the runtime mostly depends on the number of line cuts that are estimated (Section 5.4)), 2 min for DS, and approximately 3 min for SS. These are straightforward and unoptimized implementations in Matlab, except for α-expansion optimization, for which the public available code of [76, 77, 78, 118] in C++ is used.

5.6.4 Two view piecewise planar models

As discussed in [119], the depth error in stereo vision is related with the correspondence error by a multiplication factor known as the geometric resolution that depends on the baseline and on the focal length. We will assume that the maximum allowed relative depth error should be 2%. From our evaluation, this value is reached for the case of our algorithm and in the images shown in Figure 5.10 for a depth of around 12 m. This will be our depth reconstruction limit, so that we will not reconstruct surfaces further away from this bound.

Figure 5.11 and Figure 5.12 show plane labeling and 3D reconstruction results in indoor and outdoor scenes, respectively. This is the type of environments targeted by the PPR algorithms described in [110, 7, 8, 9]. While these methods require multiple views, our approach is able to reach competitive results using only a stereo pair. The labeling results are exclusively based on photo-consistency and proximity, which explains the poorly defined region borders in some examples. Such issue can be easily solved using a more sophisticated pixel-wise plane la-

beling MRF, similar to the one used in Section 5.5 that incorporates crease and occlusion edge information. We chose not to do so in order to better assess the accuracy of our plane pose estimation.

5.7 Conclusion

This chapter presented an automatic piecewise planar reconstruction algorithm from two views. Unlike other existing approaches, the stereo depth estimation and the detection of planar surfaces are accomplished in a tight and coupled manner by combining SymStereo with PEARL [3]. The effectiveness of the scheme is proved by comparison with two different state-of-the-art approaches in several challenging indoor and outdoor scenarios.

As a final comment, it can be claimed that the energy-based model fitting can either be applied to dense stereo reconstruction or to a sparse point-cloud model. The former would substantially increase the computational complexity without bringing obvious benefits, while the latter would avoid the use of the smoothness term for regularizing the PEARL energy minimization. Thus, the symmetry-based semi-dense stereo provides a trade-off between the two, playing a key role in the success of the overall approach.

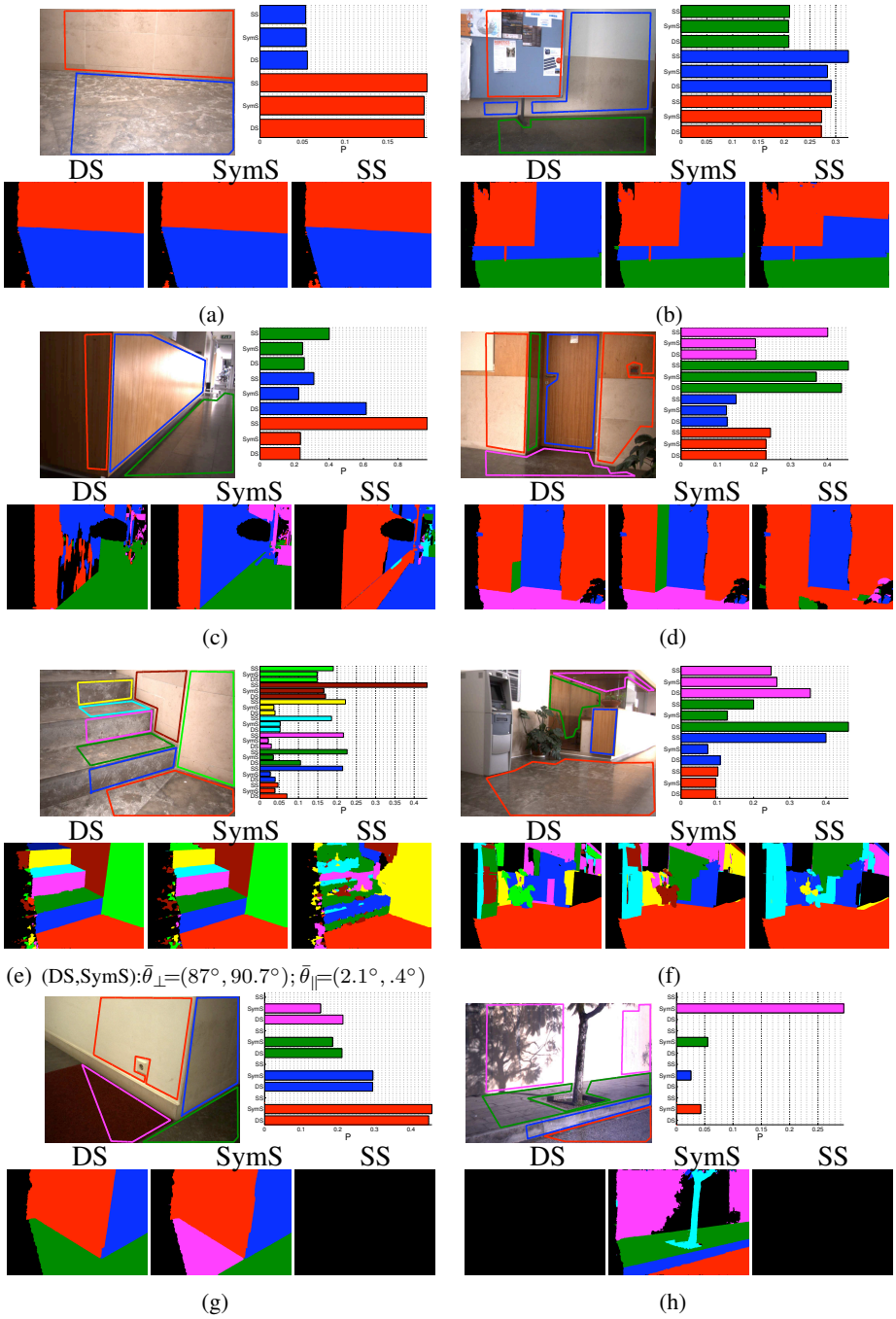
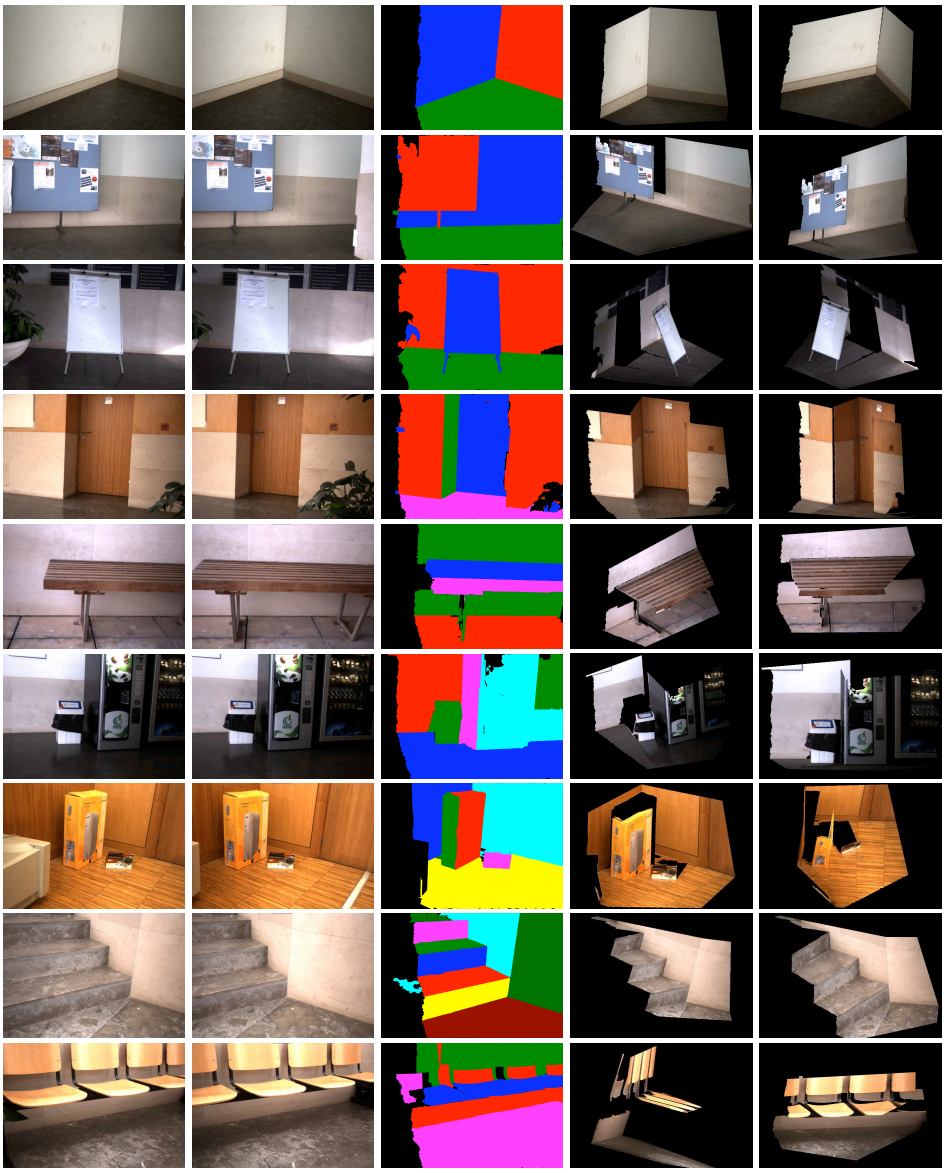
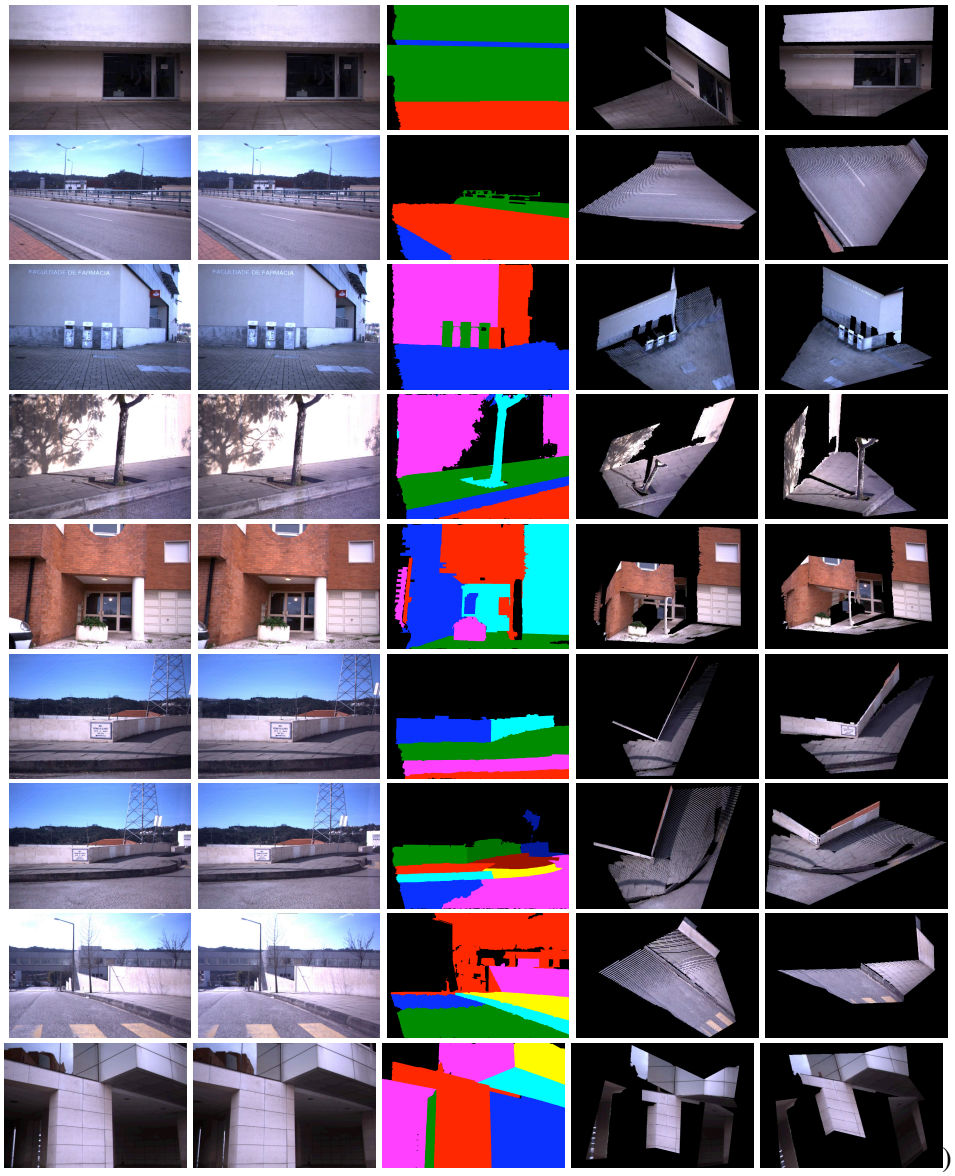


Figure 5.10: (top, left) I with GT labeling, different colors correspond to different planes; (top, right) mean photo-consistency P in the GT region for each algorithm, each color identifies a particular plane; and (bottom) pixel-wise plane assignment obtained using the different algorithms as plane hypotheses generators. The black label refers to the discard label $f_{\emptyset}$. We manually identified for the example (e) the planes that are mutually orthogonal (e.g. blue and red) and parallel (e.g. green and red); we present the mean angles $\bar{\theta}_{\perp}$ and $\bar{\theta}_{\parallel}$ between the perpendicular and the parallel planes, respectively.



(a) Stereo pair (b) Plane labeling (c) Textured 3D reconstruction

Figure 5.11: Indoor results produced by our PPR algorithm.



(a) Stereo pair (b) Plane labeling (c) Textured 3D reconstruction

Figure 5.12: Outdoor results produced by our PPR algorithm.

Chapter 6

Stereo Matching using Multiple Slant Hypotheses

This chapters extends the recent framework of Histogram Aggregation (HA) [4], which enables to improve the matching accuracy while preserving a low computational complexity. The original algorithm uses a fronto-parallel support window for cost aggregation, leading to inaccurate results in the presence of significant surface slant. We address the problem by considering a pre-defined set of discrete orientation hypotheses for the aggregation window. It is shown that a single orientation hypothesis in the DSI is usually representative of a large interval of possible 3D slants, and that handling slant in the DSI has the advantage of avoiding visibility issues. Additionally, we propose a fast recognition scheme in the DSI volume for selecting the most likely orientation hypothesis for aggregation. The experiments prove the effectiveness of the approach.

6.1 Introduction

As was discussed in Chapter 2 and in Chapter 3, dense stereo matching consists in assigning to each pixel in one view the corresponding pixel in the other view [23]. This requires using a matching cost for comparing image pixel locations and quantifying their likelihood of being a correspondence. This chapter focuses exclusively in local methods, that aggregate the matching cost over a support region in the DSI [42], as a way to enforce spatially coherence and improve the final depth estimates.

It is well known that the aggregation window must be aligned with the surface of the pixel being analyzed in order to maximize the matching performance[46,

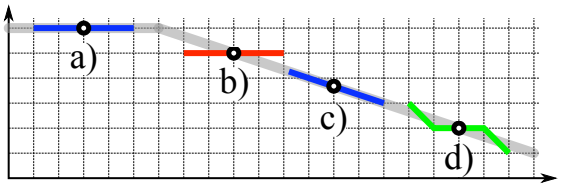


Figure 6.1: The objective is to reconstruct the points on the gray surface. The aggregation windows are overlaid (blue, red and green). Traditional stereo aggregation assumes that all the pixels in the support window have the same disparity. This works fine for fronto-parallel (FP) surfaces (a), however this assumption is incorrect for slanted surfaces (b). In this case, the support window should be aligned with the surface in gray (c). We propose an aggregation scheme that accommodates surface slant by considering a pre-defined set of possible orientations for the support window. Slant hypotheses involving sub-pixel disparities are approximated by discrete directions of aggregation in the DSI (d).

47, 48, 49, 50, 5, 51, 52, 53]. We revisit ahead several stereo methods that account for the surface slant by either working in terms of 3D space or in terms of the DSI, which is conceptually equivalent. Their objective is the estimation of the orientation of the 3D plane that approximates the surface region that is projected in the pixel or group of pixels under analysis. This usually involves the estimation of sub-pixel matches for each hypothesized planar region. Thus, these algorithms tend to be complex and time consuming.

This chapter presents a simple but effective approach for increasing the robustness to surface slant during stereo cost aggregation. Our strategy consists in avoiding the errors in pixel matching caused by surface slant without having to explicitly infer the normal orientation of the original 3D surfaces in the scene. This completely avoids sub-pixel matching and interpolation issues, enabling to improve the global stereo accuracy without substantially increasing the computational complexity. We explore the DSI and propose the discretization of slanted aggregation windows as it is done for disparity vs. 3D depth (Figure 6.1). It is demonstrated that an initial small set of aggregation orientations improves the stereo aggregation even for surfaces contained in the scene that are only approximated by those orientations. In order to improve the efficiency of the proposed stereo aggregation, we use a simple and fast recognition scheme for selecting the most appropriate aggregation orientation $\boldsymbol{\alpha}$ for each pixel-disparity pair $(\mathbf{p}, d)$. The Histogram Aggregation (HA) technique [4] is used (refer to Section 6.3), which is conceptually different from the standard cost aggregation in those cases where only one aggregation orientation is considered for each pixel-disparity pair. In a certain sense, we enhance the HA technique proposed in [4] with slant information, boosting the accuracy at the expense of a small computational overhead. The experimental results in terms of integer pixel disparity accuracy are close to [53] (highly ranked in Middlebury), but with some orders of magnitude less computation time.

6.2 Related work

In recent years, three main research topics concerning cost aggregation were addressed:

1. handling depth discontinuities [79]
2. reducing the computational complexity [4]
3. handling surface slant [46, 47, 48, 49, 50, 51, 52, 53].

The first issue is solved by the adaptive weight strategy of Yoon and Kweon [79], while the second was recently addressed in [4] by eliminating redundant computations. We focus on the 3D slant issue and, to keep computation tractable, on the second by following similar sampling schemes as [4]. We will briefly review the stereo methods that take the surface slant into account, and distinguish between four types of approaches.

The first group uses fronto-parallel stereo for the initialization. In [49], the authors use an iterative optimization for estimating the disparities and the partial derivatives of the disparities simultaneously. In [51], Zhang et al presented an alternative which consists in estimating a disparity plane orientation for each pixel using a disparity map computed using fronto-parallel aggregation. In the second step, a new adaptive cost aggregation is performed along the estimated plane orientation. The limitation of these approaches is that the initial estimation is poor in the presence of highly slanted surfaces.

The objective of the second group is to assign a 3D plane to each image pixel from a pre-defined set of plane hypotheses [46, 47]. These approaches have several drawbacks: (i) a good slant coverage always requires a large number of initial plane hypotheses, (ii) there are impossible plane hypotheses (due to visibility issues) that must be pre-calculated, and (iii) they are time consuming due to the exhaustive plane search and pixel interpolation.

A different approximation is to fit 3D planes using image segmentation. Conceptually, this is equivalent to the previous group, but with the segmentation defining the 3D plane space to be considered, and working as smoothness prior for the global optimization. The segmentation-based stereo methods over-segment the input images into homogeneous colored regions, and then perform a disparity-plane fitting for each segment. The extracted disparity planes are then used in an energy minimization framework either using the segmentation information as a hard constraint [50, 48] or as a soft constraint [52]. The disadvantage are that (i) it assumes that planar surfaces have different colors, and (ii) it is computational complex.

More recently, Bleyer et al. [53] proposed an algorithm that estimates a 3D plane at each pixel onto which the support region is projected. They start by assigning to each pixel a random plane, and then apply suitable spatial and view propagations. It provides high sub-pixel accuracy, being the top-performer in the Teddy pair. The drawback is its complexity due to the propagation process.

Our new stereo aggregation strategy is more closely related to the second group, however with two conceptual differences. The first is that we work in the DSI without the ambition of correctly estimate the 3D slant. In practical terms, we avoid interpolation issues, at the expense of no explicit sub-pixel matching accuracy. The second concerns the quantization of the 3D plane space similarly to [46, 47], and by doing it in the DSI, we are able to cover the slant space with less plane samples, as well as to implicitly handle visibility/impossible configuration issues.

6.3 Local stereo using HA

This section formulates the local stereo framework to be used as a starting point for the developments of the subsequent sections. As discussed in Chapter 2, the goal of stereo matching is to assign to each pixel $\mathbf{p}$ in $\mathtt{I}$ a disparity d from a pre-defined set of discrete values $\mathbf{D} = [0, ..., D-1]$. This assignment implicitly associates $\mathbf{p}$ with the pixel $\mathbf{p}' = (p'_1 - d, p_2)$ in $\mathtt{I}'$. As in [53], we choose as pixel matching cost the so-called truncated color and gradient differences (TD) [1]:

$$c(\mathbf{p}, d) = (1-\epsilon)\max(\tau_{col} - ||\mathtt{I}_{\mathbf{p}} - \mathtt{I}'_{\mathbf{p}'}||, 0) + \epsilon \max(\tau_{grad} - ||\Delta\mathtt{I}_{\mathbf{p}} - \Delta\mathtt{I}'_{\mathbf{p}'}||, 0),$$

where $||\mathtt{I}_{\mathbf{p}} - \mathtt{I}'_{\mathbf{p}'}||$ corresponds to the L_2-distance of the RGB colors of pixels $\mathbf{p}$ and $\mathbf{p}'$, $||\Delta\mathtt{I}_{\mathbf{p}} - \Delta\mathtt{I}'_{\mathbf{p}'}||$ is the L_2-distance of the gray-value gradients, the parameter ϵ balances the influence of color and gradient, and τ_{col} and τ_{grad} serve to truncate the cost in order to improve robustness near discontinuities.

The cost aggregation is defined as a joint histogram voting as suggested in [4]:

$$\mathsf{C}(\mathbf{p}, d) = \sum_{\mathbf{q} \in \mathcal{N}(\mathbf{p})} \omega(\mathbf{p}, \mathbf{q}) c(\mathbf{q}, d),$$

with C being the aggregated DSI, $\mathcal{N}(\mathbf{p})$ denoting the pixel neighborhood of $\mathbf{p}$ defined by the size B of the aggregation window, and $\omega(\mathbf{p}, \mathbf{q})$ corresponding to the adaptive support weighting function proposed in [79]. This function is defined as:

$$\omega(\mathbf{p}, \mathbf{q}) = \exp\left(-\frac{\sqrt{(\mathtt{I}_{\mathbf{p}} - \mathtt{I}_{\mathbf{q}})^2}}{\delta_{col}} - \frac{\sqrt{(\mathbf{p} - \mathbf{q})^2}}{\delta_{sp}} \right),$$

with δ_{col} and δ_{sp} being constant parameters.

The complexity of histogram-based cost aggregation can be substantially reduced by applying two sampling strategies [4]. The first consists in independently

[1] *The* $\min$ *operator is replaced by the* $\max$ *operator for the sake of convenience*

selecting for each pixel $\mathbf{p}$ a small subset of disparity hypotheses that have better support. This is accomplished by using a small square window for filtering the cost $c(\mathbf{p}, d)$ along the disparity dimension, and then choosing the $P\%$ local maxima of the obtained 1-D signal. The result is a subset $\mathbf{D}_P^{\mathbf{p}} = \{P\% \text{ best disparities of } \mathbf{p}\}$ comprising the disparities to be considered in subsequent steps. The operation reduces the complexity of the stereo aggregation from $O(HWBD)$ to $O(HWBDP)$, where HW is the number of image pixels, D is the size of disparity range, and $0 < P \leq 1$.

The second strategy samples the image grid by a factor of $S \times S$, which enables to reduce the complexity of the stereo aggregation to $O(HWBDP/S^2)$. Taking into account these sampling strategies, the aggregated cost C can be re-written as:

$$\mathsf{C}_{P,S}(\mathbf{p}, d) = \sum_{\mathbf{q} \in \mathcal{N}(\mathbf{p})} \omega(\mathbf{p}, \mathbf{q}) c(\mathbf{q}, d) o_P(\mathbf{q}, d) s_S(\mathbf{q}) \tag{6.1}$$

where

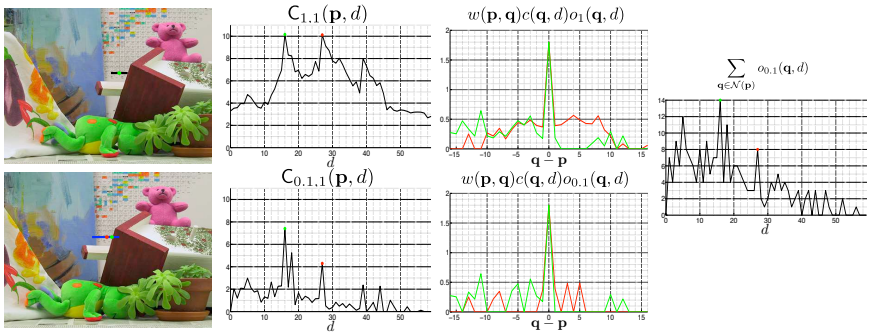
$$o_P(\mathbf{q}, d) = \begin{cases} 1 & \text{if } d \in \mathbf{D}_P^{\mathbf{q}} \\ 0 & \text{otherwise} \end{cases} \qquad \text{and} \qquad s_S(\mathbf{q}) = \begin{cases} 1 & \text{if } \mathbf{q} \% S = 0 \\ 0 & \text{otherwise} \end{cases}$$

6.3.1 Why is disparity selection useful?

It is obvious that disparity selection in HA decreases the computational complexity, since less voting steps in the histogram are required. The interesting fact presented in the experimental results in [4] is that the accuracy does not degrade, and in many cases even increases when less disparity hypotheses are used. The authors of [4] justify this as *unnecessary disparity candidates contaminate the aggregation process*. We reinforce this observation using Figure 6.2. The pixels in ambiguous regions vote in the aggregation histogram in a chaotic manner. However, the main point is that even in ambiguous image regions the correct disparity for $\mathbf{p}$ appears more times as local maxima in the neighborhood $\mathcal{N}(p)$ than other disparities, so that the disparity selection step leads to an improved disparity voting.

6.4 Aggregation with different window orientations α

This section derives the mapping between the 3D surface slant and the orientation α of the aggregation window, shows that surface visibility is easily enforced when handling the slant problem in the DSI, and proposes the discretization of the slanted aggregation window without requiring interpolation.



(a) Input Images (b) DSI along $\mathbf{p}$ (c) Adapt. Weight (d) d count

Figure 6.2: Disparity selection before HA decreases the errors in ambiguous regions and near discontinuities. (a) Top: left image with the aggregation window (black) centered in the pixel $\mathbf{p}$ under analysis (green). Bottom: right image with the matching candidates (blue); the green and red points are respectively correct and false matches. (b) Aggregated DSI results for pixel $\mathbf{p}$. It is notorious that the disparity selection (bottom) avoid the existence of multiple maxima (top) that create ambiguity. (c) Adaptive aggregation for the neighboring pixels of $\mathbf{p}$ [79]. Green corresponds to the correct disparity, while red corresponds to a false match. If no disparity selection is used (top), the two cost aggregation results will be similar because of the low texture of the roof in the case of the correct disparity. The disparity selection (bottom) removes for the false match non-discriminative contributions caused by the textured wall in background. (d) The correct disparity for $\mathbf{p}$ is voted more times in $D_P^{\mathbf{q}}$.

6.4.1 Mapping 3D surface slants into support window orientations α

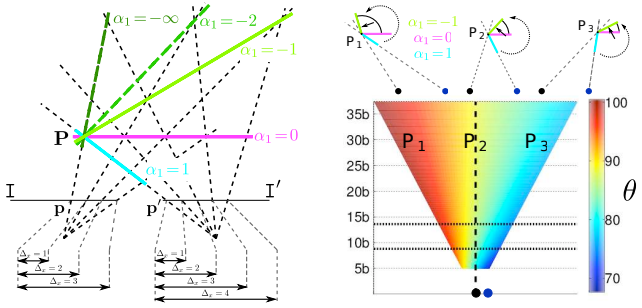
As in Chapter 2, assume a rectified stereo setup with a relative camera translation of

$$\mathbf{t} = \begin{pmatrix} b \\ 0 \\ 0 \end{pmatrix},$$

and a generic scene point $\mathbf{Q}$ that lies in a surface with normal $\mathbf{m}$. As discussed in Section 2.4, this surface can be locally approximated by a plane that defines a homography M mapping points $\mathbf{q}'$ in the right view into points $\mathbf{q}$ on the left view, and whose the disparity is given by

$$d_q = \frac{m_1 b}{l} q_1 + \frac{m_2 b}{l} q_2 + \frac{-m_1 b q_1 - m_2 b q_2 + l d_q}{l}. \tag{6.2}$$

Consider now a generic image point p in the neighborhood $\mathcal{N}(\mathbf{q})$ (unlike the case analyzed in Section 2.4, the neighboring points can lie on different epipolar lines) that is the projection of the same plane. Applying the homography M comes that



(a) Plane orientation vs. α_1 (b) $\alpha_1 = -1$

Figure 6.3: Implications of varying α_1. (a) Independently of the point $\mathbf{P}$, the surface is fronto-parallel for $\alpha_1 = 0$ (magenta), aligned with the back-projection ray of $\mathbf{p}'$ for $\alpha_1 = 1$ (blue), and aligned with the back-projection ray of $\mathbf{p}$ for $\alpha_1 = -\infty$ (dark green). (b) The 3D slant θ varies with the point location $\mathbf{P}$. We show θ for $\alpha_1 = -1$ for different locations in 3D space, the color coding identifies the variation of θ.

the disparity d_p of this neighboring point differs from d_q by

$$\Delta = d_p - d_q = \alpha_1(p_1 - q_1) + \alpha_2(p_2 - q_2) \tag{6.3}$$

with

$$\alpha_1 = \frac{m_1 b}{l} \quad \text{and} \quad \alpha_2 = \frac{m_2 b}{l}. \tag{6.4}$$

Equation 6.3 shows that the orientation of the aggregation window in the DSI must be in accordance with the 3D surface slant. A standard window along a constant disparity direction cannot account for the variation Δ in the neighborhood of the pixel under analysis. The ideal window must be slanted around a vertical axis by an angle with tangent α_1, and a horizontal axis by an angle with tangent α_2. Henceforth, we will parametrize the orientation of the aggregation window by $\boldsymbol{\alpha} = (\alpha_1, \alpha_2)$, with $\boldsymbol{\alpha} = (0,\, 0)$ being the standard situation of aggregation along a constant disparity.

6.4.2 Visibility limits for the orientation $\boldsymbol{\alpha}$

Most stereo methods that handle surface slant explicitly estimate a 3D plane for each pixel onto which the neighborhood is projected (refer to Section 6.2). In order to accomplish this, they analyze for each point $\mathbf{P}$ wheter the hypothesized surface is visible in both cameras. We show in this section that this visibility issue is implicitly solved for each pair $(\mathbf{p}, d)$ in the DSI using the parametrization $\boldsymbol{\alpha}$. Since our objective is not to accurately estimate the surface slant for each pixel, but only to obtain an appropriate approximation, we consider horizontal and vertical surface slants separately. Following [47] and as shown in Table 6.1, by horizontal

slant we mean the surfaces on which the disparity changes as we move along the x-axis, which is related to the aggregation orientation α_1 (Equation 6.4). Similarly, the disparity on a vertical slanted surface varies as we move along the y-axis, corresponding to the orientation α_2.

We show in Figure 6.3(a) the horizontal slant of the plane going through the 3D point $\mathbf{P}$ when α_1 takes different values. The neighboring pixels of $\mathbf{p}$ have the same disparity whenever $\alpha_1 = 0$, which corresponds to a fronto-parallel surface in 3D. For the case of $\alpha_1 = 1$, the surface is aligned with the back-projection ray of $\mathbf{p}'$. In this case, all the neighboring pixels of $\mathbf{p}$ are matched with the same pixel $\mathbf{p}'$ in the right view. This is the limit for which the surface is visible in the right camera, so that $\alpha_1 > 1$ makes no sense. The second limit for the aggregation orientation corresponds to $\alpha_1 = -\infty$, whose corresponding surface is aligned with the back-projection ray of $\mathbf{p}$, and represents the visibility limit for the surface slant in the left camera. It is important to emphasize that independently of the matching pair hypothesis $(\mathbf{p}, \mathbf{p}')$ being considered, $\alpha_1 = -\infty$ and $\alpha_1 = 1$ are always the visibility limits for the left and right cameras, respectively.

Following the previous observations, the range of α_1 such that the surface is visible in both cameras is $]-\infty, 1[$ (note that, since a one-to-one assumption is used, the visibility limits are not included in our analysis). Figure 6.3(b) shows the plane orientations for $\alpha_1 = -1$. As can be observed and from the previous discussion, the interval $\alpha_1 \in [-1, 1[$ covers the majority of situations in real application scenarios, so that we will use this slant range for our experiments. Following a similar reasoning, we set the working range for the vertical slant as $\alpha_2 \in [-1, 1]$.

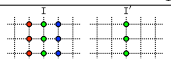
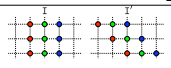
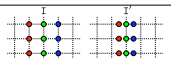
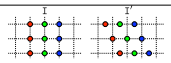
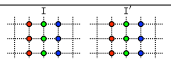
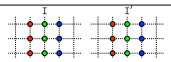
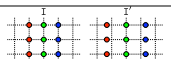
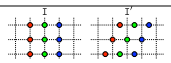
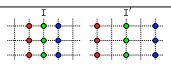
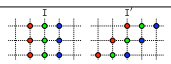
For the sake of completeness, Table 6.1 shows a 3×3 neighborhood in the left view with the corresponding variation in the right view for different values of α_1 (the green center pixel represents the reference point $\mathbf{p}$). As discussed previously, $\alpha_1 = 0$ corresponds to the fronto-parallel case, so that all the neighboring pixels have the same disparity. For positive values of α_1, the matching region is contracted, while for negative values the neighborhood is stretched in the right view.

6.4.3 Discretization of the aggregation window

The DSI is inherently a discrete 3D space so that considering continuous window orientations requires the interpolation of the cost volume or of the input images before the matching cost calculation. This provides depth estimations at a sub-pixel accuracy level, however with the drawback of increased computational cost. We avoid the interpolation issues by discretizing the slanted window in the DSI, proposing a very simple approximation, where the incremental disparity between successive pixels is given by

$$\Delta = (int)(\boldsymbol{\alpha} \cdot (\mathbf{p} - \mathbf{q})^{\mathsf{T}}). \tag{6.5}$$

Table 6.1: We show the spatial matching distribution for different values of α_1 and α_2, the range of values for α_1 and α_2 which correspond to the same aggregation pattern p_i in a volume of size B, and the aggregation patterns that can be obtained for B=5. Note that k=(B–1/2) and $j=[1,...,k-1]$.

α_1, α_2	Match distribution for α_1	Match distribution for α_2	Aggregation pattern	$B=5$
1			$p_0 : \alpha_1, \alpha_2 = 1$	
$0 < \alpha_1, \alpha_2 < 1$			$p_j : \frac{1}{k-(j-1)} \leq \alpha_1, \alpha_2 < \frac{1}{k-j}$	
0			$p_k : -\frac{1}{k} < \alpha_1, \alpha_2 < \frac{1}{k}$	
$-1 < \alpha_1, \alpha_2 < 0$			$p_{j+k} : -\frac{1}{k-(j-1)} \leq \alpha_1, \alpha_2 < -\frac{1}{k-j}$	
-1			$p_{B-1} : \alpha_1, \alpha_2 = -1$	

As described in Section 6.4.2, we assume the working ranges $\alpha_1 \in [-1, 1[$ and $\alpha_2 \in [-1, 1]$, and consider vertical and horizontal surface slants separately. Following this, it can be verified that using the support windows discretization proposed in Equation 6.5, there are $B-1$ distinguishable horizontal and B distinguishable vertical aggregation patterns for a window of size B. We depicted in Table 6.1 the range of values for α_1 and α_2 that represent the same aggregation pattern p_i, where i represents the index of a specific pattern. Finally, Table 6.1 also shows the different aggregation patterns that can be obtained for an aggregation window of size $B=5$.

6.5 HA with multiple slant hypotheses

This section describes a new scheme for HA that takes into account the surface slant. This is achieved by considering a pre-defined set of N_α window orientations in the DSI. In addition, we propose a simple recognition approach for selecting the best aggregation direction for each pixel, and discuss the differences between using standard and HAs in conjunction with orientation selection.

6.5.1 Cost aggregation in the $(\mathbf{p}, d, \boldsymbol{\alpha})$ domain

In order to accommodate surface slant in the framework of HA, we reformulate the function of Equation 6.1 to consider an additional dimension $\boldsymbol{\alpha}=(\alpha_1, \alpha_2)$ that accounts for the orientation of the support window:

$$\mathsf{C}_{r,P,S}(\mathbf{p}, d, \boldsymbol{\alpha}) = \sum_{\mathbf{q} \in \mathcal{N}(\mathbf{p})} \omega(\mathbf{p}, \mathbf{q}) c(\mathbf{p}, d + \Delta_d) h_{r,P}(\mathbf{q}, d + \Delta_d, \boldsymbol{\alpha}) s_S(\mathbf{q}), \quad (6.6)$$

where Δ_d is proportional to $\boldsymbol{\alpha}$ (see Equation 6.5). The look-up table o_P for the disparity selection is now replaced by $h_{r,P}$ that enables selecting the aggregation direction in addition to disparity selection. Please note that the histogram voting is only performed if $(d+\Delta_d) \in \mathbf{D}$. Before proceeding, there are two important aspects in the new cost aggregation function that must be emphasized. First, we need to define in advance a configuration $\mathbf{A}_{N_\alpha}$ for the orientations $\boldsymbol{\alpha}$ that are considered for cost aggregation. The number N_α of possible aggregation directions is bounded according to the discussion of Section 6.4.2, and the maximum number of possible aggregation patterns depends on the size B of the aggregation window (Section 6.4.3). Second, an exhaustive evaluation of a certain configuration $\mathbf{A}_{N_\alpha}$ with N_α possible aggregation directions increase the overall complexity to $O(HWBDPN_\alpha/S^2)$, which may become intractable even for a small N_α.

6.5.2 Sampling the space of the aggregation orientations $\boldsymbol{\alpha}$

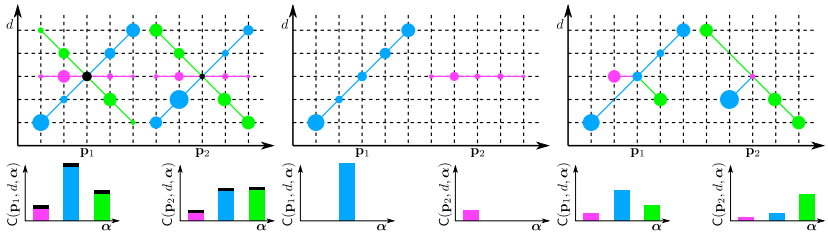
We propose a simple and fast recognition approach for an efficient implementation of the HA formulated in Equation 6.6. The objective is to select for each pixel $\mathbf{p}$ and disparity d, the best aggregation orientation among the hypotheses in the configuration $\mathbf{A}_{N_\alpha}$ under consideration. The recognition is accomplished by correlating the cost $c(\mathbf{p}, d)$ with the window of size R slanted according to $\boldsymbol{\alpha}$. It is important to distinguish between the size B of the aggregation window and the size R of the recognition window. This operation is carried whenever the parameter r is set (Equation 6.6) and is defined by the following scoring function

$$\rho(\mathbf{p}, d, \boldsymbol{\alpha}) = \frac{\sum\limits_{\mathbf{q} \in \mathcal{N}_R(\mathbf{p})} c(\mathbf{q}, d + \boldsymbol{\alpha} \cdot (\mathbf{p}-\mathbf{q})^\mathsf{T})}{\sum\limits_{\mathbf{q} \in \mathcal{N}_R(\mathbf{p})} \left(d + \boldsymbol{\alpha} \cdot (\mathbf{p}-\mathbf{q})^\mathsf{T}\right) \in \mathbf{D}}, \tag{6.7}$$

For each pixel and disparity pair $(\mathbf{p}, d)$, the orientation $\boldsymbol{\alpha}$ with highest score defines the set $\mathbf{A}_r^{\mathbf{p},d} = \{\text{best } \boldsymbol{\alpha} \text{ for } (\mathbf{p}, d)\}$. In the case the parameter r is zero, then $\mathbf{A}_r^{\mathbf{p},d} = \mathbf{A}_{N_\alpha}$ and all orientations are considered for the aggregation. The new look-up table h is defined as:

$$h_{r,P}(\mathbf{p}, d, \boldsymbol{\alpha}) \quad = \quad \begin{cases} 1 & \text{if } \boldsymbol{\alpha} \in A_r^{\mathbf{p},d} \wedge d \in D_P^{\mathbf{p}}. \\ 0 & \text{otherwise} \end{cases}$$

Remark that the selection of $P\%$ of the most likely disparities d for each pixel $\mathbf{p}$ ($D_P^{\mathbf{p}}$) only makes sense in conjunction with orientation selection ($r = 1$). In this case, the scoring function ρ is the new metric for choosing the best disparities. The selection of a single window for cost aggregation restores the overall complexity to



(a) $r=0$ (b) $r=1$ Standard (c) $r=1$ Histogram

Figure 6.4: Differences between standard and HA using 3 aggregation orientations (magenta, blue and green). We show two examples for two different reference points $\mathbf{p}_1$ and $\mathbf{p}_2$ (black). (b,c) Blue slant assigned to $\mathbf{p}_1$ and magenta to $\mathbf{p}_2$.

$O(HWBDP/S^2)$. There is obviously an overhead due to the recognition process but, since $R << B$ is considered, this computational cost is very small.

6.5.3 Standard aggregation vs. HA

There is a difference between standard [79] and HA [4] in cases where the aggregation orientation is pre-selected ($r = 1$). As shown in Figure 6.4, for $r = 0$ both approaches obtain the same cost $\mathsf{C}(\mathbf{p}, d, \boldsymbol{\alpha})$, corresponding to the sum of all neighboring costs along the N_α aggregation orientations $\boldsymbol{\alpha}$. However, if the recognition parameter is set to $r = 1$, then for standard aggregation, $\mathsf{C}(\mathbf{p}, d, \alpha)$ is obtained by aggregating the neighborhood of $\mathbf{p}$ along the assigned orientation $\boldsymbol{\alpha}$ for $(\mathbf{p}, d)$. In HA, each neighbor votes along the orientation to which it was assigned. This means that the N_α bins $\mathsf{C}(\mathbf{p}, d, \boldsymbol{\alpha})$ of $(\mathbf{p}, d)$ are voted by the neighboring pixels for which the aggregation direction $\boldsymbol{\alpha}$ intersects $(\mathbf{p}, d)$.

6.6 Experimental Results

In this section, we study the performance of the proposed stereo aggregation using different sets of aggregation orientations, and compare it against two state-of-the-art methods. Following the standard evaluation [23], the disparity maps are scored by counting the number of *nonocc* (pixels in non-occluded regions), *all* (all pixels), and *disc* (visible pixels near occluded regions) pixels that differ in more than one pixel from the ground truth. We set the matching cost parameters $\{\epsilon = 0.8,\ \tau_{col} = 9,\ \tau_{grad} = 3\}$ and the adaptive weight parameters $\{B = 37,\ \delta_{col} = 16,\ \delta_{sp} = 3\}$ constant. Occluded pixels are detected by left/right consistency check, and the disparity values of background regions are propagated to the invalidated pixels using a simple line-by-line approach. The experiments are performed on the standard Middlebury dataset (see Figure 6.5), on the Wood1 stereo pair (bottom left of Figure 3.8), and on the Oxford Corridor stereo pair (bottom right of Figure 3.8).

Table 6.2: We use 4 aggregation configurations (FP - Fronto-parallel, ver. - vertical, hor. - horizontal).

	A_1	A_3		A_7				A_{11}			
α_1	0	A_1	0 0	A_3	0 0	−.5 .5		A_7	0 0	−.2 .2	
α_2	0		−1 1		−.5 .5	0 0			−.2 .2	0 0	
	FP		2 ver.		2 ver.	2 hor.			2 ver.	2 hor.	

Table 6.3: Comparison of 4 configurations $\mathbf{A}_N$(Figure6.2). No spatial sampling is applied ($S=1$). The under-script values in $(P=1, r=1)$ correspond to the errors for conventional aggregation (Section 6.5.3).

	Stereo Pair	Teddy			Cones			Wood1			Corridor		
	P	0.1	1		0.1	1		0.1	1		0.1	1	
	r	1	1	0	1	1	0	1	1	0	1	1	0
FP	$\mathbf{A}_1$	5.29	6.04		2.95	3.4		8.18	10.6		28	19	
SLANT	$\mathbf{A}_3$	**2.84**	$\mathbf{3.05}_{4.21}$	**3.32**	**2.71**	$\mathbf{2.98}_{2.94}$	3.48	4.19	$12.6_{3.52}$	7.73	22.6	$\mathbf{28.3}_{23.7}$	18.9
	$\mathbf{A}_7$	4.93	$8.41_{6.03}$	3.88	2.93	$3.74_{4.02}$	4.09	6	$18.7_{3.60}$	7.54	26.7	$40.5_{24.8}$	20.4
	$\mathbf{A}_{11}$	5.89	$13.3_{4.87}$	3.78	3.4	$9.85_{3.15}$	**3.09**	**1.78**	$\mathbf{4.43}_{1.89}$	**2.06**	**12.7**	$30.5_{16.8}$	**15.7**

Concerning the possible orientations for aggregation, we only assume vertical or horizontal slants separately. The Table 6.2 specifies the configurations $\mathbf{A}_{N_\alpha}$ to be considered, indicating the $\boldsymbol{\alpha}=(\alpha_1, \alpha_2)$ values that define the orientations of the N_α window hypotheses. These values were selected according to the discussion of Section 6.4.2. The experimental results shown next indicate that in general our small discrete set of orientations $\boldsymbol{\alpha}$ are able to approximate different 3D slants in the scene. We compare the results for the 4 configurations $\mathbf{A}_N$ of Table 6.2 in an attempt to assess the influence of the number of considered direction hypotheses for aggregation. Please note that for $\mathbf{A}_3$ and $\mathbf{A}_7$ a window of $R=5$ is sufficient for the recognition step, while for $\mathbf{A}_{11}$ we use $R=11$ to discriminate between the different aggregation orientations.

6.6.1 Comparison of different aggregation configurations

We show in Table 6.3 the results of the disparity labeling for nonocc pixels in 4 stereo pairs. As expected from [4], the selection of the best disparities improves the disparity estimation in most cases.

6.6.1.1 Effect of considering various aggregation orientations ($r=0$)

Considering various aggregation orientations improves the accuracy in the majority of the cases when compared to fronto-parallel aggregation. This does not happen for one case ($\mathbf{A}_7$) in the cones dataset, however this scene does not contain any slanted planes, and more aggregation orientations tend to amplify the chaotic voting referred previously. Concerning the selection of $\mathbf{A}_N$, considering plane hypotheses that are not in the scene degrade the results. $\mathbf{A}_7$ considers, in addition to the hypotheses in $\mathbf{A}_3$, horizontal and vertical orientations that are not present in any of the datasets, leading to systematically degradation of the estimations.

$\mathbf{A}_{11}$ takes into account finer vertical and horizontal hypotheses that are not present neither in $\mathbf{A}_3$ nor $\mathbf{A}_7$. This leads to a dramatic improvement in datasets where the existing slants are well approximated, being the top-performer in three stereo pairs. In summary, the selection $\mathbf{A}_N$ must take into account the stereo conditions (likely relevant slants), however considering various aggregation orientations is in most of the cases better than only fronto-parallel aggregation.

It is important to refer that for the Corridor pair, we only manage to accurately estimate depth using $\mathbf{A}_{11}$. This happens because the relationship between f and b are different from the Middlebury stereo pairs (refer to Equation 6.4 and Figure 6.3), so that finer aggregation orientations are needed. This reinforces the fact that the configuration of the stereo setup must be taken into account when selecting the orientations for $\mathbf{A}_N$.

6.6.1.2 Effect of selecting one slant hypothesis ($r=1$)

There are two different effects that must be accounted: (i) the effectiveness of the recognition scheme in selecting the most suitable orientation hypothesis $\boldsymbol{\alpha}$, and (ii) the effect of the HA. It can be observed that the results tend to be significantly worse than for ($r=0, P=1$). This is not because of the slant selection process, but rather because of the fact that HA is not effective without disparity sampling. We show in under-script the results when there is aggregation orientation selection, but the aggregation is performed in the standard manner (see Section 6.5.3). The accuracy degrades slightly but doubts concerning the effectiveness of the recognition can be discarded. Finally, and as can be seen in column ($r=0, P=0.1$), the HA is effective if we use both disparity sampling and slant selection.

There are two take home messages considering HA taking into account surface slant. The first is that slant selection in HA works always well if the surface slants contained in the scene are well approximated by the hypothesis considered in $\mathbf{A}_N$. Otherwise, the decision process can assign different values $\boldsymbol{\alpha}$ to points on the same 3D surface that are equally well approximated by the discrete aggregation directions. This creates contradictory contributions in the histogram voting for neighboring pixels, enhancing the ambiguity described in Figure 6.2. The second observations is that the previous effect can be compensated by pursuing both slant selection and disparity sampling. The disparity sampling discards the contributions of neighbors of (x, d) for which the decision of slant can be equally fitted by more than one hypothesis, so that their votes are diluted in the histogram voting. Note that the results in Wood1 show that, by chance, one of our slant hypothesis is consistent with the scene. This does not happen in the others, which proves that the framework generalizes and the experimental values are not result of coincidence between the orientation sets $\mathbf{A}_N$ in the DSI and effective slants in the 3D space.

Table 6.4: Evaluation in Middlebury (Consulted in 11/2012.). We set ($P=0.1, r=1$).

	Algorithm	Rank	Tsukuba			Venus			Teddy			Cones			Runtime (Tsukuba)
			nonocc	all	disc	nonocc	all	disc	nonocc	all	disc	nonocc	all	disc	
	PatchMatch[53]	12	2.09_{66}	2.33_{52}	9.31_{63}	0.21_{22}	0.39_{18}	2.62_{31}	2.99_{1}	8.16_{8}	9.62_{2}	2.47_{5}	7.8_{9}	7.11_{7}	$\approx 60s$
$S=1$	HistAggr+TD	25	2.44_{72}	2.69_{56}	9.17_{62}	0.25_{30}	0.34_{16}	3.24_{38}	5.29_{16}	10.7_{22}	14_{16}	2.95_{24}	8.59_{24}	8.24_{25}	$16.9s$
$S=1$	**HistAggr+TD+Slant**	17	2.38_{70}	2.62_{56}	9.33_{64}	0.26_{32}	0.36_{17}	3.32_{41}	2.84_{1}	8.19_{9}	8.51_{1}	2.71_{13}	8.16_{16}	7.52_{13}	$18s$
$S=3$	HistAggr+TD	30	2.27_{70}	2.52_{55}	9.14_{62}	0.24_{28}	0.31_{13}	2.92_{35}	5.90_{19}	11.6_{31}	15.4_{22}	3.16_{29}	8.81_{30}	8.73_{34}	$1.7s$
$S=3$	**HistAggr+TD+Slant**	**22**	$\mathbf{2.25}_{68}$	$\mathbf{2.50}_{55}$	$\mathbf{9.77}_{68}$	$\mathbf{0.29}_{34}$	$\mathbf{0.37}_{17}$	$\mathbf{3.30}_{41}$	$\mathbf{3.44}_{3}$	$\mathbf{8.82}_{13}$	$\mathbf{9.77}_{4}$	$\mathbf{2.90}_{20}$	$\mathbf{8.40}_{20}$	$\mathbf{7.97}_{20}$	**2s**
$S=3$	HA+Census[4]	61	2.47_{72}	2.71_{58}	11.1_{77}	0.74_{62}	0.97_{56}	3.28_{39}	8.31_{68}	13.8_{58}	21_{86}	3.86_{48}	9.47_{45}	10.4_{53}	$0.34s$

6.6.2 Evaluation in Middlebury

We compare the proposed aggregation with PatchMatch[53] as being one of the most accurate local algorithms that take into account the surface slant, and with the original HA approach [4], which has very low computational complexity.

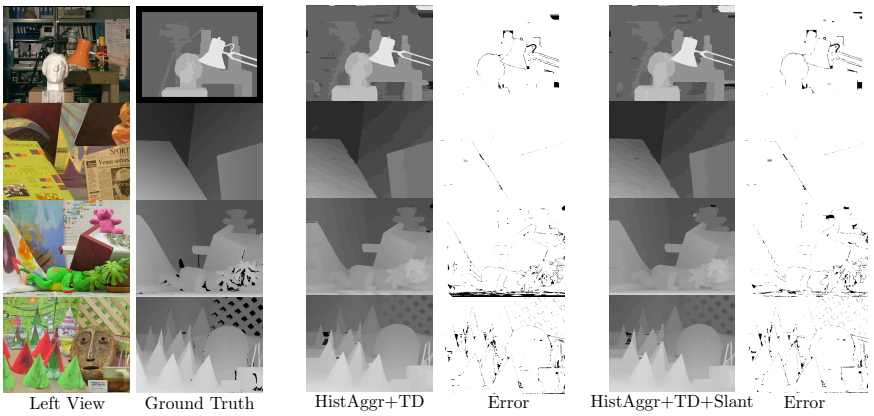


Figure 6.5: Results in Middlebury (Tsukuba, Venus, Teddy and Cones) [23]. The results correspond in Table 6.4 to $P=0.1$ and $S=3$.

The results are presented in Table 6.4 and the disparity maps are shown in Figure 6.5. HA+TD corresponds to the fronto-parallel aggregation ($\mathbf{A}_1$), whereas HA+TD+Slant takes into account 3 aggregation orientations ($\mathbf{A}_3$). HA+Census is the original algorithm [4] (they used $P=0.1$ and $S=3$), the only difference with respect to HA+TD is the matching cost: we use TD instead of Census[2]. The higher computational time for $S=3$ is due to the matching cost (Census is faster than TD), as well as the higher level of code optimization (we used a straightforward C implementation).

Our algorithm combines the advantages of both, the accuracy of PatchMatch by considering surface slant hypotheses, and the speed of the HA technique. We dramatically improve with respect to fronto-parallel HA+TD at the expense of a computational overhead of $15-20\%$. We take the first position in the ranking for the Teddy stereo pair, which is more relevant, since it is the only one containing considerable slant. This is achieved with less than $1/3$ of the runtime of PatchMatch. The spatial sampling $S=3$ is just slightly more inaccurate, but with a

[2]PatchMatch also uses TD, which is in theory better suited for dealing with slant.

speedup of 30 times [3]. The floor of the Teddy dataset in Figure 6.5 is accurately reconstructed using a single aggregation pattern ($\alpha_2 = 1$), even if the ground truth values vary between $0.7 \leq \alpha_2 \leq 1.3$. This proves that the selection of a small set of aggregation orientations is sufficient to improve cost aggregation to surface slant. As finally remark, we propose to use **HA+TD+Slant** with $\mathbf{S = 3}$, being the best compromise between accuracy and runtime (approach submitted in the Middlebury online evaluation and that was called HistAggr+TD+Slant).

6.7 Conclusions

This chapter presented a new HA framework that accounts for surface slant. The strategy consisted in choosing the most suitable aggregation direction within a discrete set of hypotheses. The approach is able to combine high matching accuracy with small computational overhead when compared to [4]. In the line with what has been discussed in [4] for the sampling strategies, we demonstrated that increasing the number of slant hypotheses does not necessarily improve the depth map accuracy. Nevertheless, we manage to prove that a fixed set of hypotheses, even when non coincident with the existing plane surfaces in the scene, improves the results. Finally, we converge to the accuracy of PatchMatch [53] with much less computation time. The reader could argue that eventually the spatial sampling strategy can also be applied to PatchMatch. We think that this observation is not obvious, since PatchMatch is based on spatial propagation, which most likely worsens with the spatial sampling.

[3] It takes about $6s$ for processing the Teddy pair.

Chapter 7

Conclusions

This thesis presented the research in computer vision carried during my PhD. The work makes the following contributions:

- **A new cue for stereo vision** - Chapter 2 presented the first work in the literature proposing to use symmetry instead of photo-similarity for assessing the likelihood of two image locations being a match. This framework was called SymStereo, and is based on the mirroring effect that arises whenever one view is mapped into the other using the homography induced by a virtual cut plane that intersects the baseline.

- **New matching costs based on symmetry** - Chapter 3 proposed three symmetry-based matching costs: *SymBT*, *SymCen*, and *logN*. The first two are closely related to existing metrics based on photo-similarity, while the later relies in wavelet transforms for detecting local signal symmetry. The new matching costs were benchmarked against the state-of-the-art metrics for accomplishing dense disparity labeling in both short and wide-baseline images. The results showed that the symmetry-based functions, SymBT and SymCen, consistently outperform their similarity-based counterparts, BT and Census, suggesting that symmetry is superior to standard photo-consistency as a stereo metric. The logN cost proved to be particularly effective in scenes with slanted surfaces and difficult textures. The major weakness is its relative poor performance close to discontinuities and occlusion regions.

- **Stereo-Rangefinding (SRF)** - Chapter 2, Chapter 3 and Chapter 5 investigated the use of passive stereo for estimating depth along a scan plane. The technique, named Stereo-Rangefinding (SRF), provides profile cuts of the scene similar to the ones that would be obtained by a LRF. We provided the first benchmark of SRF, which showed that logN is the best performing matching cost for this purpose. Moreover, we compared the depth estimates

obtained using SRF with the readings provided by a 2D LRF. The experimental results demonstrated that SRF can be leveraged to meet the robustness and depth accuracy of laser range data.

- **A global approach for detecting VPs and groups of mutually orthogonal VDs** - Chapter 4 presented an automatic and global approach for the detection of VPs and mutual orthogonal VDs. The core of the framework is the formulation of these multi-model fitting problems as Uncapacited Facility Location (UFL) and Hierarchical Facility Location (HFL) instances, which are solved using a message passing approach. The effectiveness of the framework is proved in real scenarios containing multiple Manhattan-world configurations.

- **A Piecewise Planar Reconstruction (PPR) pipeline** - The pipeline described in Chapter 5 combines the SymStereo framework and the PEARL algorithm [3] for PPR. The experimental results obtained with this system demonstrated that it is possible to obtain very accurate piecewise planar models of indoor and outdoor scenes from only two calibrated images.

- **A Histogram Aggregation (HA) framework that accounts for surface slant** - The strategy described in Chapter 6 consisted in choosing the most suitable aggregation direction for HA within a pre-defined set of discrete hypotheses. The approach is able to combine high matching accuracy with small computational overhead when compared to existing approaches.

Appendix A

Additional Results for Chapter 4

A.1 Results on YUD using detected edges

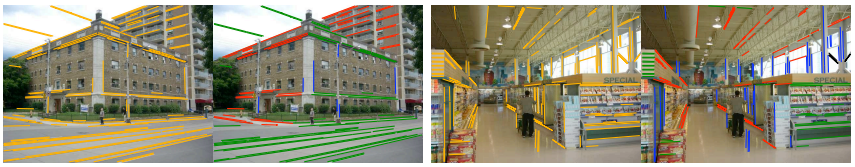
We show in Figure A.1 and Figure A.2 the output of our HFL algorithm (see Chapter 4) on 16 examples of the YUD database using edges extracted trough Tardif's detector [2]. The colors red, green and blue correspond to the directions of the Manhattan frame, while the other colors e.g. magenta, cyan, yellow, indicate non-orthogonal VPs. The edges marked in black received the empty (no VP) label. The left image shows the input edges (orange), while the right image shows the detected Manhattan directions and the non-orthogonal VPs.

The images shown in Figure A.3 and Figure A.4 were captured using a Panasonic DMC digital camera that was calibrated in advance. We run Tardif's edge detector [2] for obtaining the input edges for our UFL and HFL algorithms. The output of our HFL algorithm is shown in Figure A.3 and Figure A.4. The colors red, green and blue correspond to the directions of the mutually orthogonal triplets, while the other colors e.g. magenta, cyan, yellow, indicate non-orthogonal VPs. The edges marked in black received the empty (no VP) label.

(a) Example 1 (b) Example 2

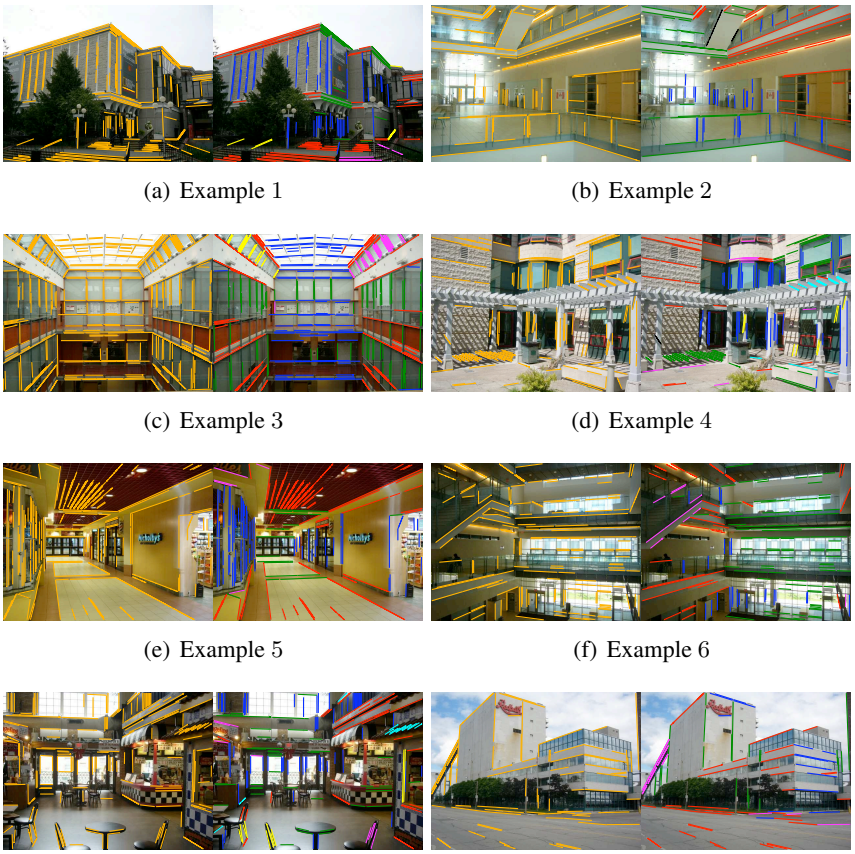
(c) Example 3 (d) Example 4

(e) Example 5 (f) Example 6



(g) Example 7 (h) Example 8

Figure A.1: We show 8 examples from YUD. The left images show the input edges (orange), while the right images show the clustering results. The 3 directions of the Manhattan frame are detected (red, green and blue).



(a) Example 1 (b) Example 2

(c) Example 3 (d) Example 4

(e) Example 5 (f) Example 6

(g) Example 7 (h) Example 8

Figure A.2: We show 8 examples from YUD. The left images show the input edges (orange), while the right images show the clustering results. We detect the VDs of the Manhattan frame (red, green and blue), VPs that are non-orthogonal to the Manhattan triplet (yellow, magenta and cyan), and lines that are considered outliers (black, no VP is assigned). This is performed simultaneously by our algorithm.

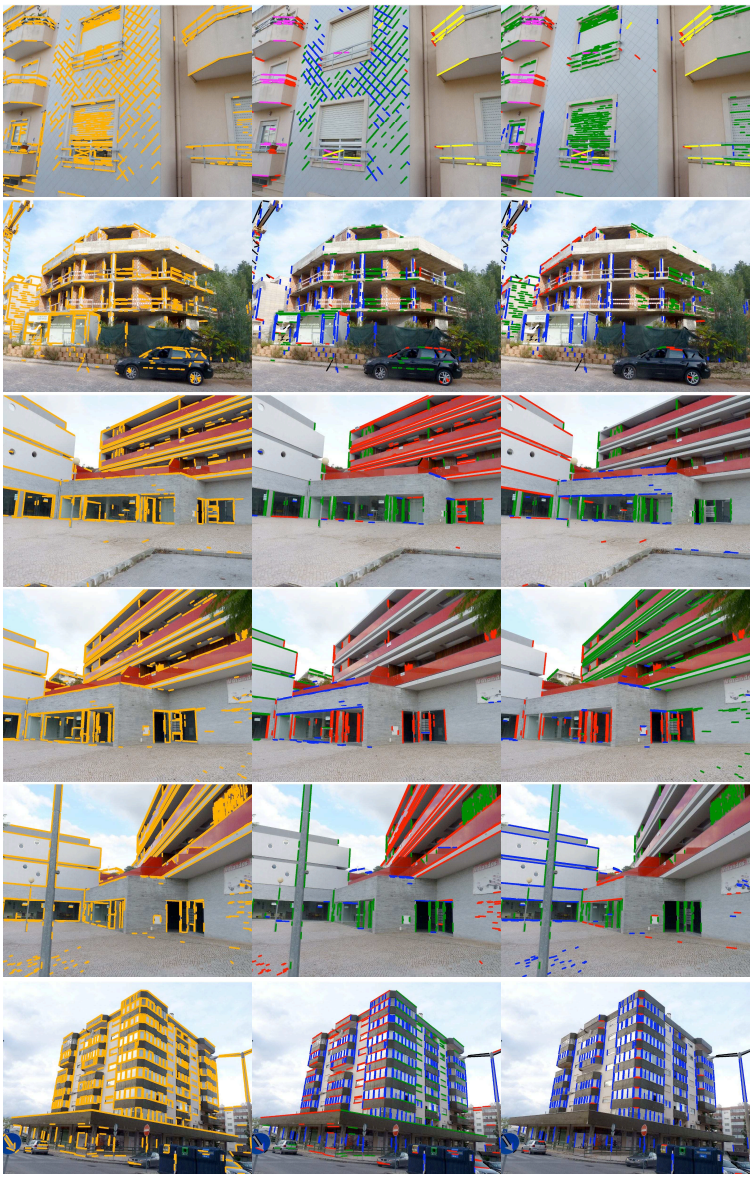
(a) Example 1 (b) Example 2

(c) Example 3 (d) Example 4

(e) Example 5 (f) Example 6

(g) Example 7 (h) Example 8

Figure A.3: We show 8 examples from our dataset. The left images show the input edges (orange), while the right images show the clustering results. We detect the VDs of the Manhattan frame (red, green and blue), VPs that are non-orthogonal to the Manhattan triplet (yellow, magenta and cyan), and lines that are considered outliers (black, no VP is assigned). This is performed simultaneously by our algorithm.



(a) Input edges (b) Orthogonal triplet 1 (c) Orthogonal triplet 2

Figure A.4: We show 6 examples from our dataset. The left images show the input edges (orange), while the right images show the clustering results. We detect the VDs of two different mutually orthogonal triplets. The lines clustered to VDs belonging to these mutually orthogonal triplets are marked in red, green and blue, and lines labeled with the same color in different triplets have the same VD. Lines assigned to non-orthogonal VPs are labeled in yellow and magenta, while lines that are considered outliers are black (no VP is assigned). We show two figures for the two different orthogonal triplets for the sake of clarity, however the detection of the mutually orthogonal triplets and the non-orthogonal VPs is performed simultaneously.

Bibliography

[1] C. Strecha, W. von Hansen, L. J. V. Gool, P. Fua, and U. Thoennessen, "On benchmarking camera calibration and multi-view stereo for high resolution imagery," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2008.

[2] J.-P. Tardif, "Non-iterative approach for fast and accurate vanishing point detection," in *International Conference on Computer Vision*, 2009.

[3] H. Isack and Y. Boykov, "Energy-based geometric multi-model fitting," *International Journal of Computer Vision*, 2012.

[4] D. Min, J. Lu, and M. N. Do, "A revisit to cost aggregation in stereo matching: How far can we reduce its computational redudancy?" *International Conference on Computer Vision*, 2011.

[5] D. Gallup, J.-M. Frahm, P. Mordohai, Q. Yang, and M. Pollefeys, "Real-Time Plane-Sweeping Stereo with Multiple Sweeping Directions," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2007.

[6] J. Xiao, T. Fang, P. Zhao, M. Lhuillier, and L. Quan, "Image-based street-side city modeling," in *SIGGRAPH*, 2009.

[7] Y. Furukawa, B. Curless, S. Seitz, and R. Szeliski, "Manhattan-world stereo," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2009.

[8] S. Sinha, D. Steedly, and R. Szeliski, "Piecewise planar stereo for image-based rendering," in *International Conference on Computer Vision*, 2009.

[9] D. Gallup, J.-M. Frahm, and M. Pollefeys, "Piecewise planar and non-planar stereo for urban scene reconstruction," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2010.

[10] A. J. Davison, I. D. Reid, N. D. Molton, and O. Stasse, "Monoslam: Real-time single camera slam," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2007.

[11] F. Bonin-Font, A. Ortiz, and G. Oliver, "Visual navigation for mobile robots: A survey," *Journal of Intelligent and Robotic Systems*, 2008.

[12] R. Siegwart and I. R. Nourbakhsh, *Introduction to Autonomous Mobile Robots*. Bradford Company, 2004.

[13] B. Kitt, A. Geiger, and H. Lategahn, "Visual odometry based on stereo image sequences with ransac-based outlier rejection scheme," in *Intelligent Vehicles Symposium*, 2010.

[14] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The kitti dataset," *International Journal of Robotics Research*, 2013.

[15] D. Scaramuzza, M. Achtelik, L. Doitsidis, F. Fraundorfer, E. Kosmatopoulos, A. Martinelli, M. Achtelik, M. Chli, S. Chatzichristofis, L. Kneip, D. Gurdan, L. Heng, G. Lee, S. Lynen, L. Meier, M. Pollefeys, A. Renzaglia, R. Siegwart, J. Stumpf, P. Tanskanen, C. Troiani, and S. Weiss, "Vision-controlled micro flying robots: from system design to autonomous navigation and mapping in gps-denied environments," *IEEE Robotics and Automation Magazine*, 2013.

[16] K. Daniilidis, J. Mulligan, R. McKendall, G. Kamberova, D. Schmid, and R. Bajcsy, "Real-time 3d tele-immersion," in *The Confluence of Vision and Graphics*, 2000.

[17] W. Matusik and H. Pfister, "3d tv: A scalable system for real-time acquisition, transmission and autostereoscopic display of dynamic scenes," *SIGGRAPH*, 2004.

[18] C. L. Zitnick, S. B. Kang, M. Uyttendaele, S. Winder, and R. Szeliski, "High-quality video view interpolation using a layered representation," *ACM Transactions on Graphics*, 2004.

[19] B. Petit, J.-D. Lesage, C. Menier, J. Allard, J.-S. Franco, B. Raffin, E. Boyer, and F. Faure, "Multicamera real-time 3d modeling for telepresence and remote collaboration," *International Journal of digital multimedia broadcasting*, 2009.

[20] C. Wengert, P. C. Cattin, J. M. Duff, and G. Székely, "Markerless endoscopic registration and referencing," in *Conference on Medical Image Computing and Computer Assisted Intervention*, 2006.

[21] P. Markelj, D. Tomazevic, B. Likar, and F. Pernus, "A review of 3d/2d registration methods for image-guided interventions," *Medical Image Analysis*, 2010.

[22] J. Barreto, R. Melo, M. Antunes, M. Lourenco, and F. Vasconcelos, "Arthronav:computer assisted navigation system for orthopedic surgery using endoscopic images," in *First Portuguese Meeting in Biomedical Engineering*, 2009.

[23] D. Scharstein and R. Szeliski, "A taxonomy and evaluation of dense two-frame stereo correspondence algorithms," *International Journal of Computer Vision*, 2002.

[24] R. I. Hartley and A. Zisserman, *Multiple View Geometry*. Cambridge University Press, 2004.

[25] S. M. Seitz, B. Curless, J. Diebel, D. Scharstein, and R. Szeliski, "A comparison and evaluation of multi-view stereo reconstruction algorithms," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2006.

[26] R. Szeliski, *Computer Vision: Algorithms and Applications*. Springer, 2010.

[27] R. I. Hartley and P. F. Sturm, "Triangulation," in *International Conference on Computer Analysis of Images and Patterns*, 1995.

[28] S. Giannarou, M. Visentini-Scarzanella, and G.-Z. Yang, "Probabilistic tracking of affine-invariant anisotropic regions," *Transactions on Pattern Analysis and Machine Intelligence*, 2013.

[29] S. Seitz, B. Curless, J. Diebel, D. Scharstein, and R. Szeliski, "A comparison and evaluation of multi-view stereo reconstruction algorithms," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2006.

[30] Z. Zhang, "Flexible camera calibration by viewing a plane from unknown orientations," in *International Conference on Computer Vision*, 1999.

[31] J. Y. Bouguet, "Camera calibration toolbox for matlab," 2008. [Online]. Available: http://www.vision.caltech.edu/bouguetj/calib_doc/.

[32] D. Nistér, "An efficient solution to the five-point relative pose problem," *Transactions on Pattern Analysis and Machine Intelligence*, 2004.

[33] J. Barreto, J. Roquette, P. Sturm, and F. Fonseca, "Automatic camera calibration applied to medical endoscopy," in *British Machine Vision Conference*, 2009.

[34] D. Murray and J. Little, "Using real-time stereo vision for mobile robot navigation," in *Autonomous Robots*, 2000.

[35] Y. Negishi, J. Miura, and Y. Shirai, "Mobile robot navigation in unknown environments using omnidirectional stereo and laser rangefinder," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2004.

[36] A. Howard, "Real-time stereo visual odometry for autonomous ground vehicles," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2008.

[37] S. Nedevschi, S. Bota, and C. Tomiuc, "Stereo-based pedestrian detection for collision-avoidance applications," *IEEE Transactions on Intelligent Transportation Systems*, 2009.

[38] L. Meier, P. Tanskanen, L. Heng, G. H. Lee, F. Fraundorfer, and M. Pollefeys, "Pixhawk: A micro aerial vehicle design for autonomous flight using onboard computer vision," *Autonomous Robots*, 2012.

[39] D. Stoyanov, M. V. Scarzanella, P. Pratt, and G.-Z. Yang, "Real-time stereo reconstruction in robotically assisted minimally invasive surgery," in *Conference on Medical Image Computing and Computer Assisted Intervention*, 2010.

[40] D. Stoyanov, "Stereoscopic scene flow for robotic assisted minimally invasive surgery," in *Conference on Medical Image Computing and Computer Assisted Intervention*, 2012.

[41] A. Fusiello, E. Trucco, and A. Verri, "A compact algorithm for rectification of stereo pairs," *Machine Vision and Applications*, 2000.

[42] R. Szeliski and D. Scharstein, "Sampling the disparity space image," *Transactions on Pattern Analysis and Machine Intelligence*, 2004.

[43] M. Antunes, "3d reconstruction and registration in medical endoscopy," in *Thesis Project - as stipulated by FCT-UC*, 2010.

[44] D. Scharstein and C. Pal, "Learning conditional random fields for stereo," *IEEE Conference on Computer Vision and Pattern Recognition*, 2007.

[45] H. Hirschmüller and D. Scharstein, "Evaluation of stereo matching costs on images with radiometric differences," *Transactions on Pattern Analysis and Machine Intelligence*, 2009.

[46] X. Zabulis and K. Daniilidis, "Multi-camera reconstruction based on surface normal estimation and best viewpoint selection," in *3DPVT*, 2004.

[47] A. S. Ogale and Y. Aloimonos, "Stereo correspondence with slanted surfaces: Critical implications of horizontal slant," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2004.

[48] M. Bleyer and M. Gelautz, "A layered stereo algorithm using image segmentation and global visibility constraints," in *International Conference on Image Processing*, 2004.

[49] G. Li and S. W. Zucker, "Stereo for slanted surfaces: First order disparities and normal consistency," in *IEEE Conference on Computer Vision and Pattern Recognition (EMMCVPR)*, 2005.

[50] A. Klaus, M. Sormann, and K. Karner, "Segment-based stereo matching using belief propagation and a self-adapting dissimilarity measure," in *International Conference on Pattern Recognition*, 2006.

[51] Y. Zhang, M. Gong, and Y.-H. Yang, "Local stereo matching with 3d adaptive cost aggregation for slanted surface modeling and sub-pixel accuracy," in *International Conference on Pattern Recognition*, 2008.

[52] M. Bleyer, C. Rother, and P. Kohli, "Surface stereo with soft segmentation," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2010.

[53] M. Bleyer, C. Rhemann, and C. Rother, "Patchmatch stereo - stereo matching with slanted support windows," in *British Machine Vision Conference*, 2011.

[54] M. Gong, R. Yang, L. Wang, and M. Gong, "A performance study on different cost aggregation approaches used in real-time stereo matching," *International Journal of Computer Vision*, 2007.

[55] F. Tombari, S. Mattoccia, L. D. Stefano, and E. Addimanda, "Classification and evaluation of cost aggregation methods for stereo correspondence," *IEEE Conference on Computer Vision and Pattern Recognition*, 2008.

[56] R. Szeliski, R. Zabih, D. Scharstein, O. Veksler, V. Kolmogorov, A. Agarwala, M. Tappen, and C. Rother, "A comparative study of energy minimization methods for markov random fields with smoothness-based priors," *Transactions on Pattern Analysis and Machine Intelligence*, 2008.

[57] H. Hirschmüller, "Accurate and efficient stereo processing by semi-global matching and mutual information," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2005.

[58] R. Collins, "A space-sweep approach to true multi-image matching," in *IEEE Conference on Computer Vision and Pattern Recognition*, 1996.

[59] ——, "A space-sweep approach to true multi-image matching," in *IEEE Conference on Computer Vision and Pattern Recognition*, 1996.

[60] D. Gallup, J. Frahm, P. Mordohai, Q. Yang, and M. Pollefeys, "Real-time plane-sweeping stereo with multiple sweeping directions," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2007.

[61] Y. Ma, S. Soatto, J. Kosecka, and S. S. Sastry, *An Invitation to 3-D Vision: From Images to Geometric Models*. SpringerVerlag, 2003.

[62] J. Ponce, K. McHenry, T. Papadopoulo, M. Teillaud, and B. Triggs, "On the absolute quadratic complex and its application to autocalibration," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2005.

[63] J. Sun, Y. Li, S. Bing, and K. H. yeung Shum, "Symmetric stereo matching for occlusion handling," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2005.

[64] T. Werner and A. Zisserman, "New techniques for automated architectural reconstruction from photographs," in *European Conference on Computer Vision*, 2002.

[65] S. Birchfield and C. Tomasi, "A pixel dissimilarity measure that is insensitive to image sampling," *Transactions on Pattern Analysis and Machine Intelligence*, 1998.

[66] A. Ansar, A. Castano, and L. Matthies, "Enhanced real-time stereo using bilateral filtering," in *3D Data Processing, Visualization and Transmission*, 2004.

[67] R. Zabih and J. Woodfill, "Non-parametric local transforms for computing visual correspondence," in *European Conference on Computer Vision*, 1994.

[68] S. Gautama, S. Lacroix, and M. Devy, "Evaluation of stereo matching algorithms for occupant detection," in *Proceedings of the International Workshop on Recognition, Analysis, and Tracking of Faces and Gestures in Real-Time Systems*, 1999.

[69] J. Banks and P. Corke, "Quantitative evaluation of matching methods and validity measures for stereo vision," *The International Journal of Robotics Research*, 2001.

[70] M. Z. Brown, D. Burschka, and G. D. Hager, "Advances in computational stereo," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2003.

[71] C. Fookes, A. Maeder, S. Sridharan, and J. Cook, "Multi-spectral stereo image matching using mutual information," in *3D Data Processing, Visualization and Transmission*, 2004.

[72] L. Wang, M. Gong, M. Gong, and R. Yang, "How far can we go with local optimization in real-time stereo matching," in *3D Data Processing, Visualization and Transmission*, 2006.

[73] I. Sarkar and M. Bansal, "A wavelet-based multiresolution approach to solve the stereo correspondence problem using mutual information," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, 2007.

[74] P. Kovesi, "Symmetry and asymmetry from local phase," in *Tenth Australian Joint Conference on Artificial Intelligence*, 1997.

[75] P. Kovesi and P. Kovesi, "Image features from phase congruency," Videre, Tech. Rep., 1995.

[76] Y. Boykov, O. Veksler, and R. Zabih, "Fast approximate energy minimization via graph cuts," *Transactions on Pattern Analysis and Machine Intelligence*, 2001.

[77] V. Kolmogorov and R. Zabih, "What energy functions can be minimized via graph cuts?" in *European Conference on Computer Vision*, 2002.

[78] Y. Boykov and V. Kolmogorov, "An experimental comparison of min-cut/max- flow algorithms for energy minimization in vision," *Transactions on Pattern Analysis and Machine Intelligence*, 2004.

[79] K. jin Yoon, S. Member, and I. S. Kweon, "Adaptive support-weight approach for correspondence search," *Transactions on Pattern Analysis and Machine Intelligence*, 2006.

[80] P. Mordohai, "The self-aware matching measure for stereo," in *International Conference on Computer Vision*, 2009.

[81] C. Premebida, O. Ludwig, and U. Nunes, "Lidar and vision-based pedestrian detection system," *Journal of Field Robotics*, 2009.

[82] F. R. Bertrand Douillard, Dieter Fox, "Laser and vision based outdoor object mapping," in *Robotics: Science and Systems*, 2008.

[83] F. T. Ramos, J. I. Nieto, and H. F. Durrant-Whyte, "Recognising and modelling landmarks to close loops in outdoor slam," in *IEEE International Conference on Robotics and Automation*, 2007.

[84] J. Y. Bouguet, "Camera calibration toolbox for matlab," http://www.vision.caltech.edu/bouguetj/calib$_doc/index.html$.

[85] F. Vasconcelos, J. P. Barreto, and U. Nunes, "A minimal solution for the extrinsic calibration of a camera and a laser rangefinder," *Transactions on Pattern Analysis and Machine Intelligence*, 2011.

[86] L. Grammatikopoulos, G. Karras, and E. Petsa, "An automatic approach for camera calibration from vanishing points," *Journal of Photogrammetry and Remote Sensing*, 2007.

[87] M. E. Antone and S. J. Teller, "Automatic recovery of relative camera rotations for urban scenes," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2000.

[88] J. Kosecka and W. Zhang, "Video compass," in *European Conference on Computer Vision*, 2002.

[89] S. N. Sinha, D. Steedly, and R. Szeliski, "Piecewise planar stereo for image-based rendering," in *International Conference on Computer Vision*, 2009.

[90] G. Baatz, K. Kser, D. M. Chen, R. Grzeszczuk, and M. Pollefeys, "Handling urban location recognition as a 2d homothetic problem," in *European Conference on Computer Vision*, 2010.

[91] P. Denis, J. H. Elder, and F. J. Estrada, "Efficient edge-based methods for estimating manhattan frames in urban imagery," in *European Conference on Computer Vision*, 2008.

[92] G. Schindler and F. Dellaert, "Atlanta world: An expectation maximization framework for simultaneous low-level edge grouping and camera calibration in complex man-made environments," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2004.

[93] S. T. Barnard, "Interpreting perspective images," *Artificial Intelligencegence*, 1983.

[94] C. Rother, "A new approach for vanishing point detection in architectural environments," in *British Machine Vision Conference*, 2000.

[95] F. M. Mirzaei and S. I. Roumeliotis, "Optimal estimation of vanishing points in a manhattan world," in *International Conference on Computer Vision*, 2011.

[96] J. C. Bazin, Y. Seo, C. Demonceaux, P. Vasseur, K. Ikeuchi, I. Kweon, and M. Pollefeys, "Globally optimal line clustering and vanishing point estimation in manhattan world," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2012.

[97] N. Lazic, B. J. Frey, and P. Aarabi, "Solving the uncapacitated facility location problem using message passing algorithms." *Journal of Machine Learning Research*, 2010.

[98] N. Lazic, I. E. Givoni, B. J. Frey, and P. Aarabi, "Floss: Facility location for subspace segmentation," in *International Conference on Computer Vision*, 2009.

[99] E. Tretyak, O. Barinova, P. Kohli, and V. Lempitsky, "Geometric image parsing in man-made environments," *International Journal of Computer Vision*, 2012.

[100] I. E. Givoni, C. Chung, and B. J. Frey, "Hierarchical affinity propagation," in *Conference on Uncertainty in Artificial Intelligence*, 2011.

[101] H. Li, "Two-view motion segmentation from linear programming relaxation," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2007.

[102] J. Xiao, J. Wang, P. Tan, and L. Quan, "Joint affinity propagation for multiple view segmentation," in *International Conference on Computer Vision*, 2007.

[103] B. J. Frey and D. Dueck, "Clustering by passing messages between data points," *Science*, 2007.

[104] A. Delong, A. Osokin, H. N. Isack, and Y. Boykov, "Fast approximate energy minimization with label costs," *International Journal of Computer Vision*, 2012.

[105] F. Kschischang, S. Member, B. J. Frey, and H. andrea Loeliger, "Factor graphs and the sum-product algorithm," *IEEE Transactions on Information Theory*, 2001.

[106] R. I. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2004.

[107] R. Klette, N. Kruger, T. Vaudrey, K. Pauwels, M. van Hulle, S. Morales, F. Kandil, R. Haeusler, N. Pugeault, C. Rabe, and M. Lappe, "Performance of correspondence algorithms in vision-based driver assistance using an online image sequence database," *Vehicular Technology, IEEE Transactions on*, 2011.

[108] M. Pollefeys, D. Nistér, J. M. Frahm, A. Akbarzadeh, P. Mordohai, B. Clipp, C. Engels, D. Gallup, S. J. Kim, P. Merrell, C. Salmi, S. Sinha, B. Talton, L. Wang, Q. Yang, H. Stewénius, R. Yang, G. Welch, and H. Towles, "Detailed real-time urban 3d reconstruction from video," *International Journal of Computer Vision*, 2008.

[109] M. A. Fischler and R. C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM*, 1981.

[110] A. Bartoli, "A random sampling strategy for piecewise planar scene segmentation," *Computer Vision and Image Understanding*, 2007.

[111] Y. Furukawa and J. Ponce, "Accurate, dense, and robust multiview stereopsis," *Transactions on Pattern Analysis and Machine Intelligence*, 2010.

[112] M. H. Lin and C. Tomasi, "Surfaces with occlusions from layered stereo," *Transactions on Pattern Analysis and Machine Intelligence*, 2004.

[113] M. Bleyer, C. Rother, and P. Kohli, "Surface stereo with soft segmentation," *IEEE Conference on Computer Vision and Pattern Recognition*, 2010.

[114] S. Baker, R. Szeliski, and P. Anandan, "A layered approach to stereo reconstruction," in *IEEE Conference on Computer Vision and Pattern Recognition*, 1998.

[115] S. Birchfield and C. Tomasi, "Multiway cut for stereo and motion with slanted surfaces," *International Conference on Computer Vision*, 1999.

[116] H. Tao, H. S. Sawhney, and R. Kumar, "A global matching framework for stereo computation," *International Conference on Computer Vision*, 2001.

[117] R. Grompone von Gioi, J. Jakubowicz, J. M. Morel, and G. Randall, "LSD: A Fast Line Segment Detector with a False Detection Control," *Transactions on Pattern Analysis and Machine Intelligence*, 2010.

[118] A. Delong, A. Osokin, H. N. Isack, and Y. Boykov, "Fast approximate energy minimization with label costs," *International Journal of Computer Vision*, 2012.

[119] D. Gallup, J.-M. Frahm, P. Mordohai, and M. Pollefeys, "Variable baseline/resolution stereo," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2008.