Improved Stixels Towards Efficient Traffic-Scene Representations

Noor Haitham Saleem Al-Ani

A Thesis Submitted in Partial Fulfilment of the
Requirements of the Doctor of Philosophy

School of Engineering, Computer and Mathematical Science
Auckland University of Technology
New Zealand

January 2019

Declaration

I hereby declare that this submission is my own work and that, to the best of my knowledge and belief, it contains no material previously published or written by another person (except where explicitly defined in the acknowledgements), nor material which to a substantial extent has been submitted for the award of any other degree or diploma of a university or other institution of higher learning.

Signature of candidate

Date: 23/01/2019

Dedication

To my wife Zinah, my parents Haitham and Nadia, my sister Aya, and little angel (Saleem). It is all for you.

Abstract

Stixels are medium-level data representations used for the development of computer vision modules for self-driving cars. A stixel is a column of stacked space cubes ranging from the road surface to the visual end of an obstacle. A stixel represents object height at a distance. It supports object detection and recognition regardless of their specific appearance.

Stixel calculations are commonly based on binocular vision; these calculations map millions of pixel disparities into a few hundred stixels. Depending on applied stereo vision, this binocular approach is sometimes incapable to deal with low-textured road information or noisy data. The main objective of this work is to evaluate and propose approaches for calculating stixels using different camera configurations and, possibly, also a LiDAR range sensor.

This study also highlights the role of ground manifold modelling for stixel calculations. By using simplifying ground manifold models, calculated stixels may suffer from noise, inconsistency, and false-detection rates for obstacles, especially in challenging datasets. Stixel calculations can be improved with respect to accuracy and robustness by using more adaptive ground manifold approximations. A comparative study of stixel results, obtained for different ground-manifold models, also defines a main contribution of this thesis. We also consider multi-layer stixel calculations.

Comprehensive experiments are performed on two publicly available challenging datasets. We also use a novel way for comparing calculated stixels with ground truth. We compare depth information, as given by extracted stixels, with ground-truth depth, provided by depth measurements using a highly accurate LiDAR range sensor (as available in one of the public datasets). Experimental results also include quantitative evaluations of the trade-off between accuracy and run time. The results show significant improvements for particular ways of calculating stixels.

Keywords: Stixels, ground manifold, v-disparity, y-disparity, monocular, binocular, trinocular, obstacle height, dynamic programming, LiDAR, height segmentation, multi-layer stixels.

Acknowledgements

During the past three years I gratefully acknowledged the support of wonderful people such that I feel really blessed. Without their generous help and encouragement, this PhD thesis would not have been possible. I am greatly thankful to my beloved parents Nadia and Haitham, my wonderful wife Zinah, and my lovely sister Aya, who have always believed in my ability to complete this research and encouraged me even through the most difficult stages with their ever-present love, support and prayers.

My main debt of gratitude is to my supervisor, Professor Reinhard Klette; without his guidance and encouragement, this thesis would never have been accomplished, not even started. Also, I like to thank co-supervisor Dr. Mahdi Rezaei, for his constructive comments and support during this journey.

I like to express my indebtedness to my colleague Johnny Chien for his support during our academic visit at Wuhan University in China, his advice and fruitful discussions which shaped my knowledge on this journey.

I am also thankful to Auckland University of Technology (AUT) for the scholarship provided for the development of this thesis. Their countless support and outstanding environment is highly appreciated. Also, I like to thank Sulaimani Polytechnic University (SPU) in Kurdistan Region, Iraq, for their kind support.

Lastly, I am deeply grateful to all my colleagues at AUT's research centre CeRV and to Ms. Fatina Awediah for kind support. My deepest appreciation go to my colleagues and friends in New Zealand, Iraq, and Malaysia for always being there for me.

Noor Haitham Saleem Al-Ani
Auckland, New Zealand
June 7, 2019

Contents

List of Figures

List of Tables

List of Symbols

Geometry and images

I, Ω	Image and its domain
f	Focal length of left (rectified) camera given in pixels [px]
b	Base length of stereo cameras giving in meter [m]
$\mathbb{R}$	Set of real numbers
$\mathbb{N}$	Set of natural numbers
w, h	Size parameters (width and height) where $w, h \in \mathbb{N}^+$
W, W_p	Window in an image, windows with reference pixel p
λ	Ground depth scalar used for ground plane
F	Variable for road plane
$a_0, \ldots, a_n$	Coefficients
n, m	Natural numbers
$\subset$	Subset relation
$\in$	Belonging to a set
Π	Products
Σ	Summation
N_{cols}, N_{rows}	Number of columns, number of rows of an image
$\mathbf{a}^\top, [.]^\top$	Transpose of a vector or matrix

Camera variables and matrices

$\mathbf{K}_{rec}$	Camera rectification matrix
$\mathbf{R}$	Rotation matrix
$\mathbf{t}$	Translation vector
P_r	3D point in LiDAR coordinates
P_s	3D point in camera coordinates
X, Y, Z	Coordinates in a 3D space $\mathbb{R}^3$
x, y	Real variables; pixel coordinates (x, y) in an image (i.e. column and row)
r	Range data
P	3D LiDAR sparse points in $\mathbb{R}^3$ projected in image plane

Bi- or trinocular variables and operations

$\mathbb{D}$	Set of all possible disparity values
D	Disparity map where $D \in \mathbb{D}^2$
GD	Ground disparity map (ground plane)
d	Disparity value where $d \in \mathbb{D}$ given in pixel [px]
$\mathbf{d}$	Set of quantised disparities
V_y	Value of y-disparity (also common as v-disparity)
Q	Quantisation function
γ	penalty
δ	Distance
Θ	Smoothness function
E	Cost function
ϕ	Warp function
τ	Concatenation of two disparity maps
$\mathcal{M}$	Mapping feature
Γ	Trinocular confidence measure
S_m	Saliency map
R	Range distance map
G_σ	Gaussian filter
p, q	Points in $\mathbb{R}^2$, with coordinates x and y
d_2	Euclidean distance function
κ	Confidence weighting term
ω	Weighting factor

Single-layer stixel variables

b_x	Base points
t_x	Top points
B	Set of base points, e.g. $B_i = \{b_x, \ldots, b_{w-1}\}$
T	Set of top points, e.g. $T_i = \{t_x, \ldots, t_{w-1}\}$
M	Membership map

ϵ	Exponential membership function
C	Cost map
T_S	Total number of stixels
R_S	Number of resulting stixels

Multi-layer stixel variables

$\mathbb{C}$	Set of possible classes
$\mathbb{L}$	Set of possible segments
L	Label of a stixel
s_n	Segment as part of a column labelling
c_n	Class used to represent a segment s_n, it can be either ground, object, or sky
g, o, s	Classes denoted for ground, object, and sky
G, O, S	Costs denoted for ground, object, and sky
$f_n(.)$	Candidate function
$\chi(d_y)$	Variable used as a flag to indicate outliers with disparity value d_y

Chapter 1

Introduction

This introductory chapter informs about motivations, and a review of state-of-the-art work in the area of vision-based driver assistance systems in general, and for stixels, specifically. The chapter also presents current challenges, highlights the planned contributions for this thesis, demonstrates potential spaces for improvements, and outlines the structure of the thesis.

1.1 Background

Robust obstacle segmentation and scene understanding are key tasks for visual sensors (cameras) in autonomous cars for being able to interpret and act within a dynamic environment. Cameras are playing a significant role in autonomous driving; they are capable of providing rich information including distances to obstacles given in traffic scenes.

Vision-based driver assistance systems (VB-DAS) contribute to the current transition process towards autonomous vehicles. Examples of applications of developed technologies are auto-braking systems, evasive steering assistance, and blind spot monitoring [29]. VB-DAS are already widely used, both in academia and industry [30, 50]. Cameras are one type of sensors that are commonly installed in modern cars. In particular, stereo vision contributes to systems that aim at distance measurements, surface modelling, or object detection [82]. There remains ongoing research interest in stereo vision related to VB-DAS, for example for scene analysis [49], feature descriptors [103], optimising learning time [28], or for reducing processing efforts in general [10].

During the development of scene representation techniques, stixels (short for "stick elements") have been introduced in computer graphics in [65].

Stixels turned out to be a useful way for describing 3-dimensional (3D) scenes in computer vision [11], especially in the context of VB-DAS. Stixels are in general compact representations towards subsequent semantic segmentation, thus bridging the gap between low-level representations (e.g. by input images or disparity maps) and high-level representation by semantic labelling; see Fig. 1.1.

A *base-line stixel calculation* [11] cascades multiple independent techniques: mapping disparities into an occupancy grid, ground plane computation, height segmentation (which involves membership-map calculation), and the final stixel extraction.

We briefly define three basic terms used in this study.[1] The *ground manifold* is the estimated surface function for road and adjacent levelled areas; a plane defines the simplest model (i.e. a *ground plane* [55]). In this study we consider different surface functions as models for the ground manifold.

The *ego-vehicle* is the vehicle in which the system is operating in [73]. The *free space* is a region ahead of the ego-vehicle where this vehicle may potentially (i.e. safely) drive in, for example, in the next few seconds [12, 93]. The term ground manifold is more generic than free-space.

Compact representations of disparity maps provide a possible way for semantic segmentation by grouping neighbouring cells in an occupancy grid (e.g. above a $w \times w$ base) which are at about the same depth. A stixel forms a vertical "stick" above such a base [73]. In this case (i.e. occupancy grid on a plane), stixels are rectangular thin columns on the ground plane (on a regular grid) as shown in Fig. 1.2. A stixel maps pixels, which belong to an on-road object at about the same distance to the recording camera, vertically into "columns" [11], sitting on the ground manifold and ideally upper-bounded by the top of an on-road object. See Fig. 1.3 for such a representation in a real world scene.

1.1.1 Medium-level representation by stixels

In 2009 a novel *medium-level* (i.e. between pixel data and semantic segments) representation has been proposed for urban road scenes known as stixels [11].

[1] The use of "we" throughout this thesis is purposeful. It is used to involve the reader with the thesis as recommended by Knuth et al. [53].

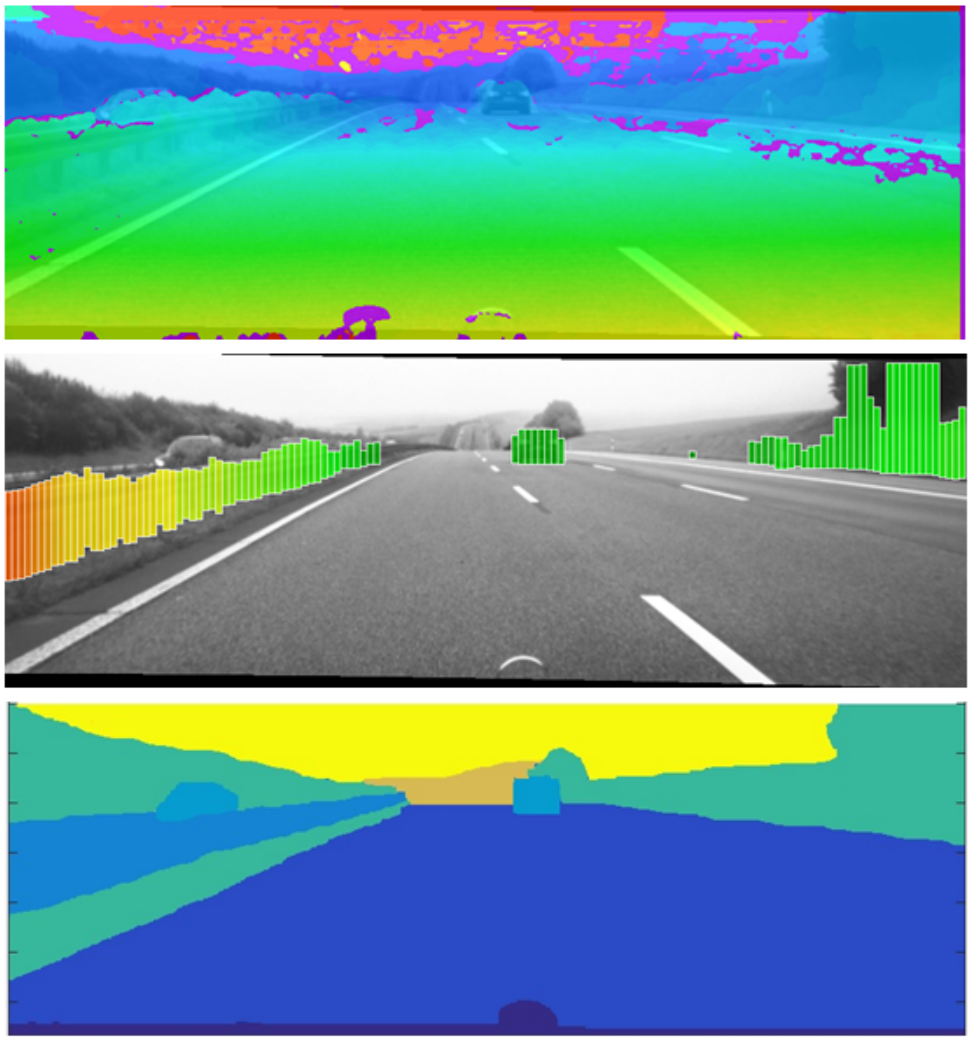

Figure 1.1: Levels of representation. *Top*: A disparity map is a low-level representation. *Middle*: Stixels provide medium-level (i.e. intermediate) information. *Bottom*: High-level representation by semantic segmentation.

A stixel starts on top with a detected upper "end" of an object and ends at the bottom on the ground plane (or ground manifold in general, also addressing non-planar surfaces). Stixels are computed from a disparity map in three

stages:

- **Base point detection.** Base points are identified by locating the boundary of free space in the given image. The boundary is found by first building an occupancy map from range data above an estimated road manifold, then solving for an optimal cut separating free space from the rest of grid cells in the map.
- **Height segmentation.** Foreground pixels are separated from the background, and the upper boundary (i.e. top points) of obstacles, founded on the ground, is detected.
- **Stixel extraction.** Column-wise obstacles are grouped and represented by bounding boxes, and the depth values of pixels in the same group are integrated to form a stixel.

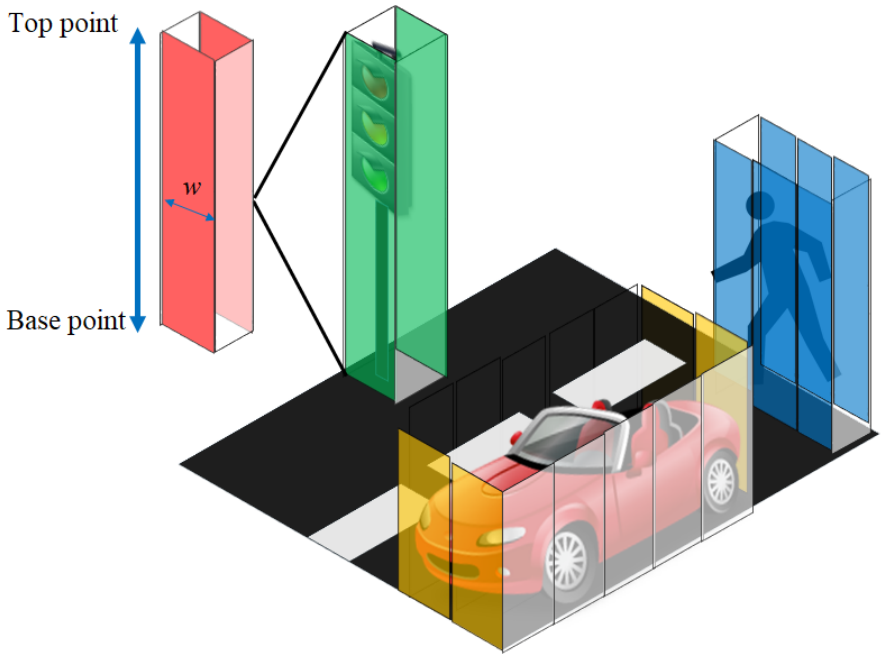

Figure 1.2: Stixels (vertical sticks) describing obstacles: A stixel has a square base, and goes ideally from a ground manifold to the top of an object.

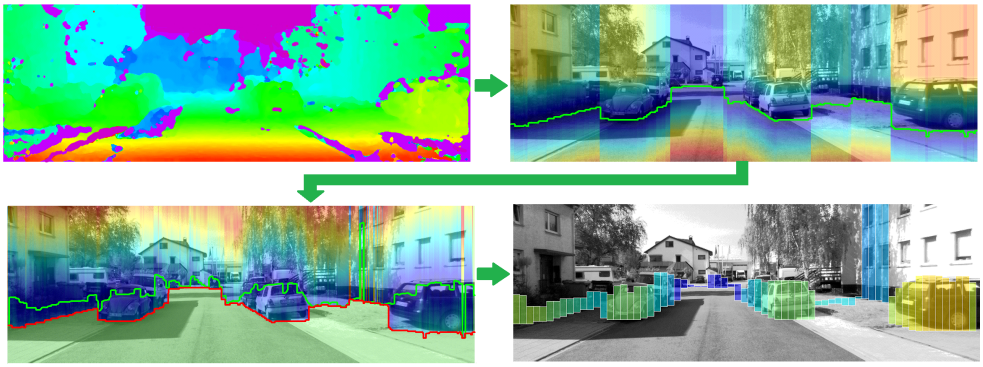

Figure 1.3: Stixel representation of a street scene. *Top-left*: Disparity map (using an SGM-variant for stereo matching) visualised by applying a colour key. *Top-right*: Base-point selection by a minimum cut (shown in green) through a cost image. Regions in deep blue show lower costs which are preferred by a dynamic-programming-based optimiser. *Bottom-left*: Top-point selection by a minimum-cut (shown in green) through a cost image, subject to the base points (shown in red). *Bottom-right*: Extracted stixels.

The stixel representation yields a highly efficient modelling of scene objects in complicated urban traffic environments [88]. Recently, joint stixel representations, combining semantic data and depth, are proposed to integrate both categories in terms of a joint optimised scene model [86].

To construct a "stixel world", multiple independent techniques have to be cascaded. For example, these may include mapping disparities [2] to occupancy grids, ground-plane estimation, stixel-height estimation, and finally stixel extraction. The generated stixels then surround a region in the ground manifold, already introduced above as the free space. The detection of free-space is important for intelligent transportation control [69]. It is also crucial for collision avoidance for the ego-vehicle, or, for example, for assisting a blind pedestrian.

[2] We adopt a *semi-global matching* (SGM) algorithm [45] for disparity calculation.

1.1.2 Ground manifold modelling

Ground manifold estimation can be approached using either monocular or multi-ocular vision [8] and as we mentioned stixels are interested in a region of ground manifold which is free-space. There are also combined monocular-binocular stixel methods; free-space is estimated by using a single camera only, followed by obstacle detection using stereo vision [57]. In order to detect free-space from a single camera, we may employ a time-efficient lane-based free-space detection method [93]. For example, lane detection can be performed by using a Hough transform for straight lines following edge detection; the Hough transform is a basic method for line extraction [22].

Figure 1.4 illustrates possible steps: Cropping of a recorded frame into a defined *region of interest* (ROI), edge detection using the Sobel operator due to its "unbiased" definition, and straight line detection by an application of an optimised Hough transform; the transform is applied recursively, using optimised (Otsu algorithm [70]) threshold values, until a predefined number of lines is found, or the threshold reaches its minimum. Finally, that "dominant" pair of lines with the best correspondence in angular directions is selected for specifying road contours (i.e. the free-space) in such a monocular vision approach.

As illustrated by Fig. 1.4, there remain many spaces which were not properly estimated regarding free-space or possible base-points of obstacles; these deficiencies would yield an early estimation of obstacles.

Currently emerging vehicle test-beds (e.g. equipped with sensors along roads, and vehicle-to-infrastructure communication; see [107, 111, 112] for an example) aim at exact and comparative evaluations of control components designed for driver assistance or driver-less vehicles. Having different options for sensors and ground-manifold models, it is, of course, important to compare efficiencies and possible accuracies of stixel calculations.

Overall, improving and analysing ground manifold and stixel estimation, using either trinocular (see Fig. 1.5 for such a configuration), binocular, or monocular vision [8, 51], became necessary for at least the following four reasons. First, it is important to investigate about their individual benefits and drawbacks, thus helping to improve towards the creation of systems being adaptive for a changing traffic environment. Literature reviews reveal

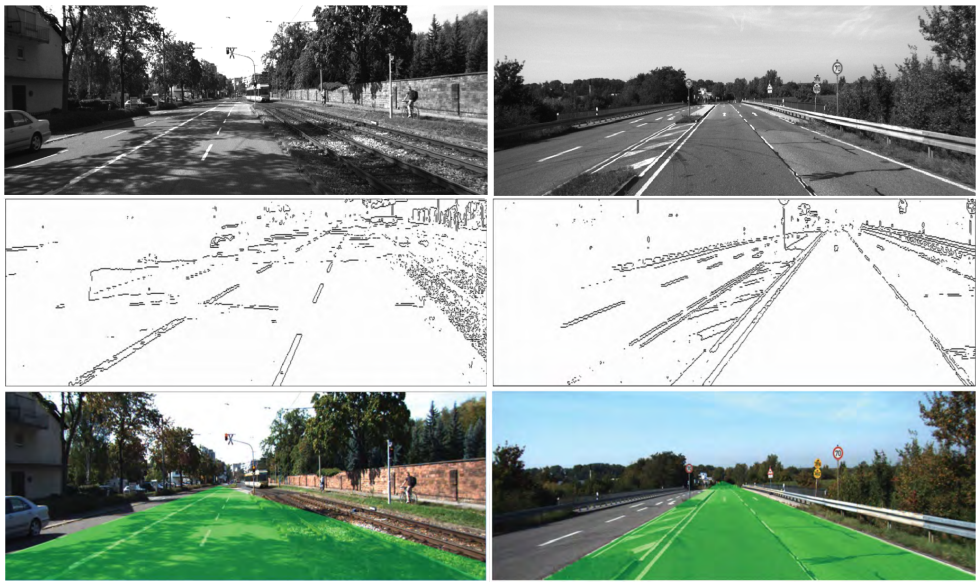

Figure 1.4: Free-space detection using monocular vision, shown for two images of the KITTI [35] *road* dataset. *Top*: Original images (grayscale). *Middle*: Edge detection using Sobel applied on ROI (i.e. "middle rows" of a frame only). *Bottom*: Detected free-space.

that there are limited studies so far on this domain of adaptive system adjustments. Second, mobile devices, or special-purpose-build components to be added to existing cars, are often limited to monocular systems. Third, camera-based control modules (e.g. for lane detection) in currently offered cars are, if existing at all, (still) limited to monocular vision. Fourth, the design of vehicle test-beds (e.g. with sensors along the road, and vehicle-to-infrastructure communication, see [107, 111, 112] for an example) requires a careful selection of appropriate camera systems.

1.2 Motivation

Both ground manifold estimation and stixel segmentation provide options to go towards 3D scene understanding. However, we identified issues for

Figure 1.5: Trinocular vision configuration. Test vehicle, as used while at Wuhan University in October 2016

these techniques when there are challenging weather conditions, road geometry variations, or low-texture information. The common stixel estimation pipeline considers stereo images taken in traffic scenes and offers dis-

parity values which can be used to estimate stixels [73]. The stixel accuracy requires a disparity signal of "good" quality; this quality often decreases in cases of occlusions or texture-less image patches [83]. Unfortunately, these issues are common in traffic scenes, thus more efforts are needed to improve disparity signals, also aiming at more reliable ground-manifold estimation and stixel calculations. Noisy 3D points (i.e. noisy disparities) have a direct negative effect on ground-manifold estimation. Unreliable disparity values need to be identified before they are transformed into 3D space and used for ground-manifold or stixel estimation. Colour-based image analysis, fused into disparity approaches, has been widely deployed in the context of free-space detection to remove false-positives associated with obstacles [83] using auto-label generation from depth. Despite the shown effectiveness, such techniques have a negative impact on stixel segmentation as the performance is significantly degraded when there are low-quality of labels [101].

Furthermore, road-geometry variations, and difficulties in recording these properly (e.g. due to weather conditions or traffic density) has recently led to several studies. Current research, as reported, e.g., in [43, 56, 84, 102], still uses just a ground-plane for modelling the road-surface and establishing stixels; we discuss how this is prone to errors as road-geometry is not always perfectly planar.

On the other hand, current methods from stixel estimation usually deploy membership functions which can identify the height of stixels (foreground versus background pixels). However, challenging lighting conditions can play a disturbing role when portraying foreground disparities in a membership map, causing, e.g., that foreground pixels are considered to be part of the background. To the best of our knowledge, there is very limited research improving the height of stixels since current approaches (published elsewhere) focus on free-space estimation.

Also, techniques found in the literature typically consider only specific sensor models (i.e. binocular vision) for multi-layer stixel segmentation. This is achieved by applying particular stereo matching algorithms [45, 94]. There are some methods focused on designing a reliable stixel representation; since the input of that model suffers from noise then it can still degrade the accuracy of the detected stixels. Difficulties arise when one is trying the same

pipeline as for stereo-based stixel segmentation but applying monocular-guided LiDAR or trinocular data. In order to obtain a more reliable disparity map, the use of multi-ocular stereo vision or of some additional depth sensors appears to be essential (as a result of our studies).

Finally, as observed in the literature, the evaluation of stixels on the KITTI dataset is challenging due to the lack of preceding frames with annotated stixel ground truth [83]. Further investigation reveals that in one dataset the stixel ground truth was provided, they were annotated using a corridor (not the free-space), so further statistical measures are required to perform evaluation.

It is challenging to seek for a versatile model that brings a good flexibility to support extensions when using different optical or range sensors such as binocular, trinocular, monocular and LiDAR. Such challenges motivated our research to develop appropriate models for ground manifolds and for height segmentation towards improved stixel segmentation.

1.3 Contributions

The objective of this research is to design, implement, and evaluate a novel stixel framework that can be applied to build 3D representations of traffic scene sequences including those recorded under challenging conditions. Results depend on a versatile design of ground-manifolds, applicable to monocular, stereo, and multi-ocular optical sensors, optionally also equipped with a LiDAR (especially combined with monocular recording).

In order to meet this goal, we propose a low-cost and accurate ground-manifold framework for reducing false positives in stixel estimation with a reduced number of parameters. The framework treats the cost of a cut for generating a robust lower envelope in v-disparity space (i.e. a row histogram of a disparity map) as optimisation problem. It efficiently seeks the "greatest rising" edge in the v-disparity space for each column while maintaining global smoothness. It deals directly with v-disparities; it attempts to cut through v-disparities to divide those into top and bottom regions.

Moreover, this study also seeks to present an adaptive solution to detect changes in ground manifolds (e.g. with respect to non-planar road geome-

try) by using a polynomial curve fitting algorithm based on the v-disparity space for ground manifold estimation. Hence, we aim to present an extensive analysis of widely used ground-manifold methods to study "how would different linear segments in v-disparity space can help to define a piecewise linear curve?" which defines the road manifold, and later the stixel detection. We decided to obtain road information using a v-disparity map and this is beneficial for two reasons: First, the coordinate space has inherently finite boundaries, which is useful when working with probability densities. Second, it leads to reduced computation time.

This work also fills a previously existing gap between confidence measures and ground manifold techniques by using calibrated collinear trinocular vision systems. Our method takes three conjugate stereo images at the same time to measure the consistency of disparity values by means of a transitivity error in disparity space. Unlike previous stixel estimation methods that are built based on a single disparity map, our proposed method introduces a multi-map fusion technique to obtain more robust stixel calculations. We apply two of our proposed ground-manifold models to detect accurate road manifolds using the v-disparity space which is built based on a confidence map. This supports our endeavour to improve stixel calculations in a single-layer or in a multi-layer format. We consider our method as a modification of baseline multi-layer stixel segmentation which can make stixel segmentation more reliable.

Moreover, we study possible effective solutions to identify the height of stixels. We study the fusion of disparity membership with a saliency map towards an improved height calculation of stixels. By incorporating a saliency map we provide a possible way for identifying visual boundaries of objects in colour images; map calculation involves calculating saliency of each pixel with reference to its neighbourhood in terms of lightness and colour properties.

Various approaches for stixel estimation have been investigated by mainly involving bi- or trinocular vision, since depth can be obtained from stereo cameras at low cost. A failure of disparity estimation on obstacles or low-textured road surfaces still causes concerns [95]. Unstable results caused by challenging imaging conditions (represented by illumination, colour, or

texture) may be resolved by also using sensors (such as LiDAR) which are reliable under such conditions. Incorporating LiDAR adds benefits to autonomous cars as it provides depth information at high accuracy. Some market growth is expected for LiDAR technologies for the next few years. Firms already advertise low-cost LiDAR sensors [108, 109].

As a result, this may lead to improved disparity maps. Yet, LiDAR points are sparse and there must be an optimised interpolation approach that would support us in our endeavour to obtain a dense depth map, and later a dense stixel representation. Therefore, this study proposes monocular stixels guided by LiDAR data for verified stixel positions.

Finally, previous literature states that the evaluation of stixels is challenging due to the lack of preceding frames of annotated road images and stixel ground truth in KITTI datasets. We address those by making use of Velodyne's high-definition 3D laser scanner data provided by the KITTI dataset. We use those range data as a ground-truth reference to evaluate the distance values assigned to the extracted stixels. We calculate a number of statistical measures (e.g. the true positive rate) as major indices to evaluate ground-manifold models on a large dataset of 2,988 frames from a 6D Vision dataset [72] plus 2,306 frames from a KITTI dataset, covering good and bad weather sequences.

1.4 Thesis organisation

The outline of this thesis is as follows:

- Chapter 2, **Related work and basics**, introduces stereo vision systems, the generation of disparities, and v-disparity maps. Also, it gives a basic structure for calculations of first-layer stixels, and specifies different methods for ground manifold and height estimation. We summarise discussions on current difficulties, found while developing and evaluating first-layer or multi-layer stixels. Stixel applications are also described in this chapter.

- Chapter 3, **Ground manifold modelling**, aims at analysing road profiles and proposes two novel ground-manifold models, namely a poly-

nomial curve model and a graph-cut dynamic programming approach, which both are used to detect piecewise defined models in v-disparity space. This chapter assesses the strength of a trinocular vision system which provides a confidence indicator to detect ground-manifolds. Finally, we propose ground-truth evaluation measures as used throughout this study.

- Chapter 4, **Stix-fusion: Data fusion towards efficiency**, starts by introducing a novel method to improve the height of stixels in traffic scenes using a saliency map. The introduced type of saliency maps is then used in an updating membership function to ensure that the foreground pixels are separated from the background. By a further investigation, we found that incorporating sensor information such as LiDAR with monocular vision, adds a great value to improve the height of stixels. Thus, monocular single layer stixels, guided by LiDAR and monocular vision, is also presented in this chapter. The keyword "stix-fusion" in this thesis denotes the data medium used to present stixels since we have employed colour information from camera and LiDAR data to construct stixels.

- Chapter 5, **Improvement of multi-layer stixels**, presents the improvement of stixel segmentation in variant traffic scenes. We increase the robustness of segmented stixels by incorporating trinocular weighted disparities, the transitivity disparity error (TED), and robust mean estimation. Also, we found that a LiDAR sensor allows us to validate the stixel segmentation, hence, by using point-cloud information we derive road-surface information.

- Finally, Chapter 6, **Conclusions and future work**, concludes this thesis and gives directions for future work based on the stixel model.

Chapter 2

Related Work and Basics

Stixel segmentation is a relatively new and promising field for traffic scene understanding which has been actively studied over the past 10 years. In line with this development, this chapter reviews the developed techniques for traffic scenes, also elaborating on related theories, implementations and applications. We discuss work on road surface and stixel extraction, which are both considered to be crucial steps towards stixel calculation.

2.1 Stereo vision

Perceiving a 3-dimensional (3D) appearance of objects, observed with our left-right eyes, is a straight-forward task for a human. Enabling a machine to perform this task is an absolutely non-trivial task due to many constraints. For implementing a stereo camera system (binocular vision) (see Fig 2.1), two cameras are placed next to each other with a fixed distance b referred to as the (base-line) distance between both cameras [90].

In traffic scenes, the representation of stationary objects, projected in a left and a right camera, is different, specifically, the object position, where close objects to the camera-pair are having a larger visual shift. In the depth estimation concept, this difference is referred to as *disparity*.

Modern existing stereo matching algorithms permit to calculate the disparity value d for nearly every pixel of an image to build the real depth Z of corresponding (i.e. in left and right camera images) world points. The parameters of camera systems affect the correspondence and disparity calculation. Based on epipolar geometry (see, e.g., [51]), this correspondence

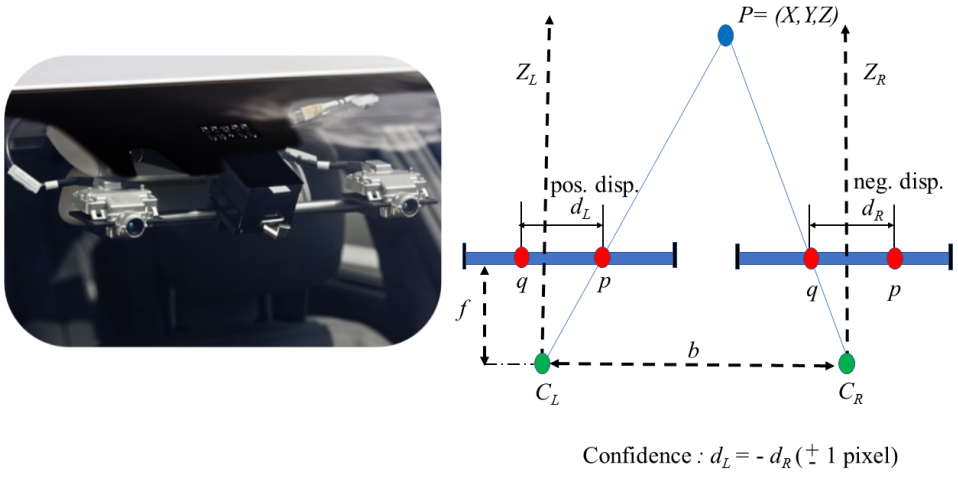

Figure 2.1: Binocular vision system. *Left*: Example of stereo cameras mounted in a test vehicle. *Right*: Simplified diagram of pinhole projection in the stereo case (b denotes the distance between both optical axes).

search is minimised to a 1-dimensional search (along an epipolar line). Moreover, when the provided images are properly calibrated (i.e. provided with intrinsic and extrinsic parameters) and rectified [90], the epipolar line may be across the horizontal image direction only, and a simple pinhole-camera model can be applied. This case is portrayed in Fig. 2.1(right).

Left-to-right stereo matching produces non-negative disparities d_L, and right-to-left stereo matching produces non-positive disparities d_R. If d_L and d_R differ by at most one pixel position, then it is common to take this as a confidence confirmation. In this thesis, we use non-negative disparities d.

The depth for a disparity value can be calculated as follows:

$$Z = f \cdot \frac{b}{d} \tag{2.1}$$

where b is the length of the baseline (connecting the focal points of the two cameras) in world units. Parameter f denotes the unified focal length of the rectified cameras.

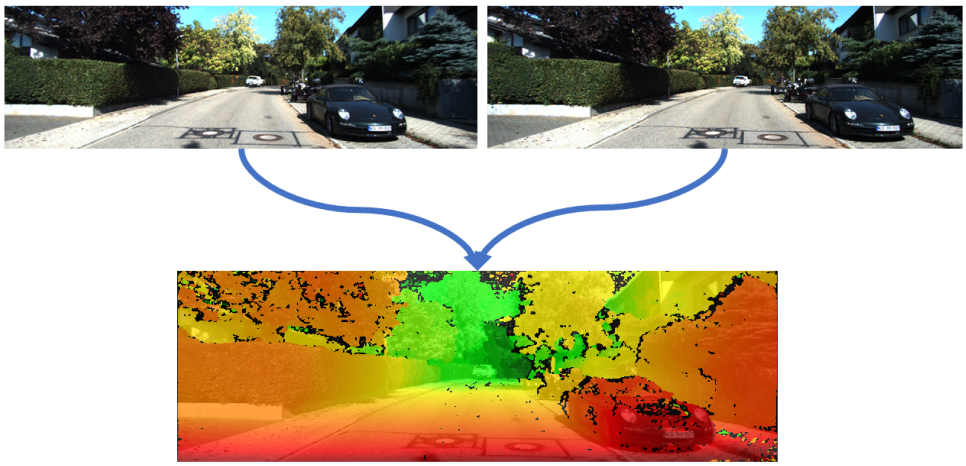

Figure 2.2: *Top*: Stereo pair (left and right image). *Bottom* The dense disparity map is computed by semi-global matching (SGM). The colours encode the disparity values from dark green (far away) to red (close).

Technically, a dense disparity map can be computed by matching all pixels in one image of a stereo pair with their correspondences in the other image as shown in Fig. 2.2. There might be pre-processing applied before the stereo matching step, such as in [38, 64], for enhancing matching outcomes. Results can be filtered by applying confidence measures; see [51] for various stereo-matching confidence measures.

After the disparity map $D(x, y)$ is computed, we have to compute the y-disparity (also known as v-disparity) value $V_y(d)$ to identify the road surface [55]. [1] We populate the y-disparity space by accumulating pixels with the same disparity value on a horizontal row of the disparity map. The y-disparity space is a row-based matrix which stores the disparity values for every column from the disparity map. The principle of y-disparity map generation is demonstrated in Fig. 2.3. The values that represent the histogram

[1] Labayrade et al. used (u, v) to denote image coordinates in [55]. As we use (x, y) image coordinates, the name "v-disparity" turns into "y-disparity".

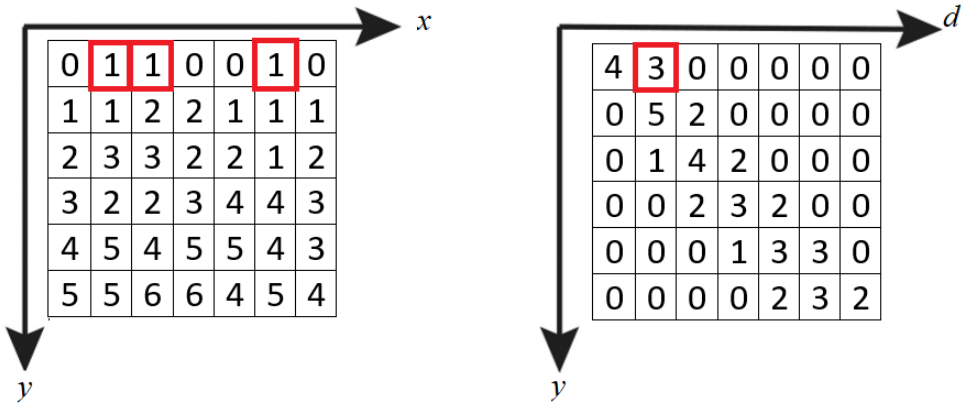

Figure 2.3: y-disparity map generation. *Left*: Example of a disparity map with values $D(x, y)$. *Right*: Corresponding y-disparity map with values $V_y(d)$.

of disparities in one row, shown by the white line in Fig. 2.4, represent "on the right" a lower envelop which can reveal a ground plane if basically a linear segment.

The histogram value of an element in the y-disparity map represents the number of pixels which has the same disparity value in the corresponding row y of the disparity map [8]:

$$V_y(d) = \text{card}\{x : 1 \leq x \leq N_{cols} \wedge D(x, y) = d\} \tag{2.2}$$

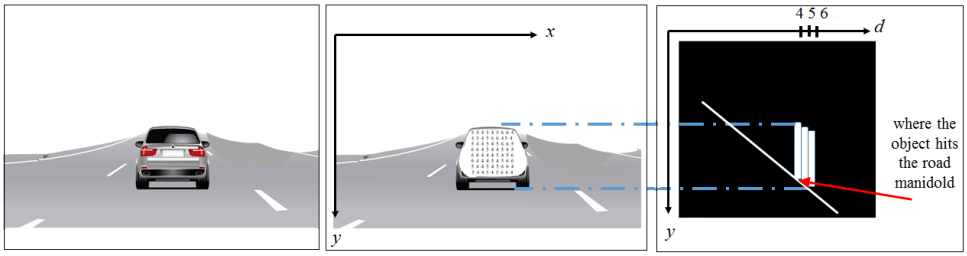

Figure 2.4: : y-disparity map projection. *Left*: Sketch of a traffic scene. *Middle*: Example of disparity values calculated for the vehicle located on the road manifold. *Right*: Corresponding y-disparity map with values $V_y(d)$.

where $0 \leq d \leq d_{\max}$, and $V_y(d)$ represents the value (note: V like "value") from the y-disparity space which accumulates the number of pixels with disparity d from row y in the disparity map. The y-coordinates go downward of disparity image. The next step is to extract the ground manifold from the generated y-disparity map. The ground manifold is identified based on a calculated lower envelop in the y-disparity space. More details will be elaborated in Section 2.2.2.

2.2 First-layer stixels

Stixels are compact representations towards semantic segmentation. Neighbouring cells in an occupancy grid (e.g. above a $w \times w$ base in a top-down view) are at about the same depth; a stixel forms a vertical "stick" above its base [73]. Stixels are thus rectangular thin columns on the ground plane (on a regular grid) as shown in Fig. 1.2. A stixel starts on top with a detected upper "end" of an object and ends at the bottom on the ground plane (or ground manifold in general, also addressing non-planar surfaces). Stixels are computed from a disparity map in three stages: *base point detection*, *height segmentation*, and *stixel extraction*.

In this section we provide an in-depth walk-through for this process following authors of [11] who introduced stixels as a medium-level scene representation.

2.2.1 Base-point detection

The first step of stixel construction is to find the bottom of the closest obstacle, for every column [11, 71]. The search is based on the the free space analysis by means of *occupancy grids* [98], which represent the scene in a top-down view as a 2D discrete map. The use of occupancy grids for free space detection dates back to 2007 [12].

A probabilistic occupancy grid can be built by projecting depth data (or equivalently disparities) along the (bottom-up) Y-axis of the camera onto a plane, and then *binning* the projected data using a 2D histogram. The grid can be defined in either 2D Cartesian coordinates (on the XZ plane) or polar coordinates. In the latter case, the grid shows distribution of pixels in the

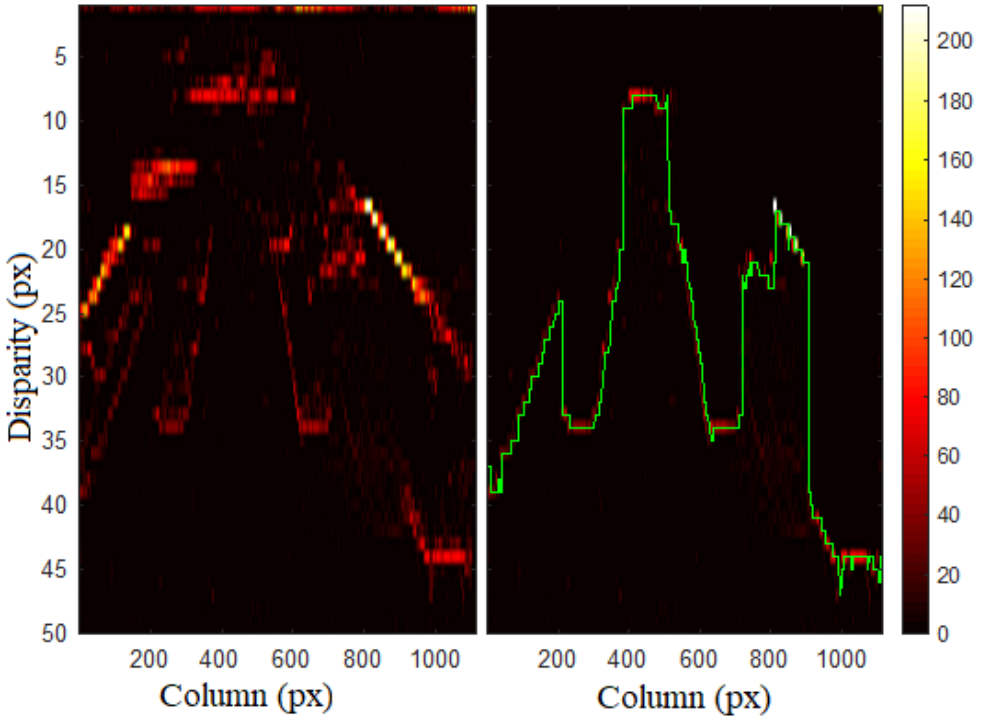

Figure 2.5: Occupancy map showing the distribution of objects above the road surface. *Left:* Computed polar-coordinate occupancy grid. *Right:* After background object removal. The green curve visualises a column-by-column maximum cut found by means of dynamic programming. Note the larger a disparity is, the closer the object to the camera.

column-disparity space, which is also known as a x-disparity map (contrary to the y-disparity map). An example is shown in Fig. 2.5.

To correctly find the free space from an occupancy map, the ground manifold has to be estimated to include only obstacles above the ground to build the grid. Details regarding the estimation of a ground manifold will be discussed in Section 2.2.2.

By means of an occupancy grid, the free space is efficiently found using a graph-cut algorithm. The nearest prominent object is first identified for each column, and the grid cells behind this object are occluded. After removing

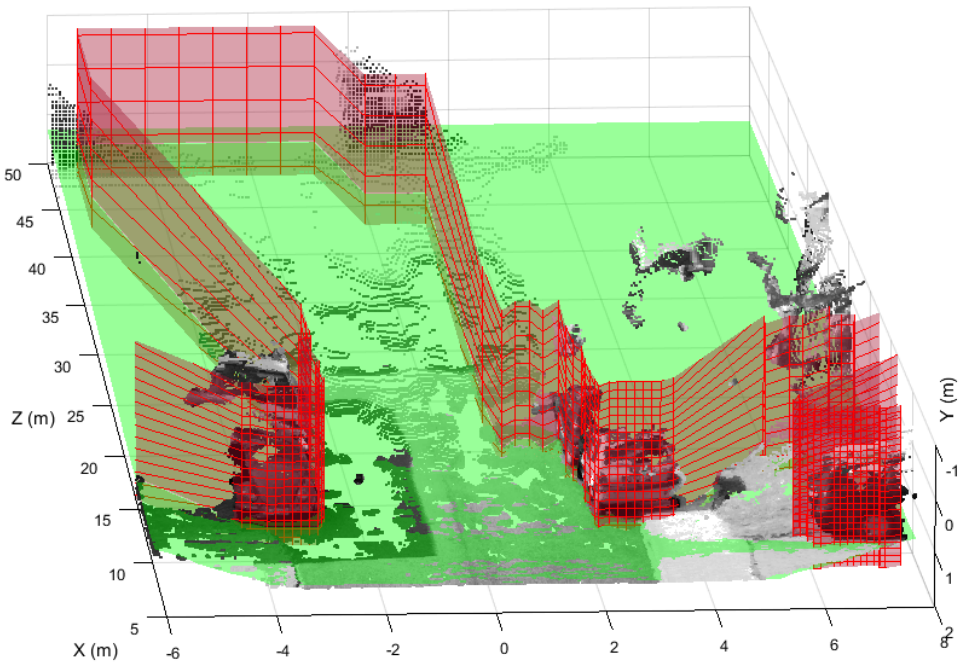

Figure 2.6: Reconstructed 3D points from a disparity image, road manifold (green), and obstacle manifold (red) from an occupancy map.

background objects from the occupancy map, a dynamic programming technique is carried out to locate the maximum-cut through the map that separates free space and the obstacles [12]. For each column x the process decides a disparity d, as illustrated in Fig. 2.5. Back projecting such a cut from the occupancy map into the image's disparity space, and, subsequently, into the Euclidean space defines an obstacle manifold, as rendered in red in Fig. 2.6.

At the end of this stage, a base point is decided for each column of D by locating the intersection of the obstacle manifold with the road manifold (see Fig. 2.6). The per-pixel distances between the road manifold and the obstacle manifold are then computed as a cost function for deciding base points (see Fig. 1.3, for example). The minimum cut through the cost then defines the base points of stixels, as represented as a set of row indices $\{b_1, b_2, \ldots, b_{N_{\text{col}}}\}$ where N_{col} is the number of columns of the image domain, and (x, b_x) denotes the image coordinates of a base point in column x.

2.2.2 Ground manifold

For the monocular case, various monocular features have been exploited as cues for ground manifold estimation such as colour [60, 80], intensity [81], shape [41], boundary [42], or vanishing points [54, 62].

For the stereo-vision case, the ground plane may be estimated by using normal vectors in disparity space [46, 69]. Urban environments feature complex surroundings in which the ground plane is limited by large or just relatively flat obstacles, such as cars or curbs [40]. Ground plane methods rely on single frame measurements, suffer from sensor noise and depth artefacts, which lead to increased deviations from correct estimations.

A temporally filtered ground plane estimation method, using dense disparity images (from stereo vision), is proposed in [40]. Stereo vision supports ground manifold estimation techniques by providing y-disparity analysis, object-related disparity analysis [96], or occupancy-grid generation [11, 16, 71].

In [78], a ground-plane method is proposed using omnidirectional images. Such images support in the discussed application purely vision-based robot navigation in indoor environments; the system employs naive Bayes classifiers for fusing multiple visual cues and features generated from heterogeneous segmentation schemes. The considered schemes maintain separate appearance models, and initiates seeds for floor and obstacle regions.

At this stage the ground manifold can be formed as a disparity map where the disparity of the estimated ground pixels can be stored at pixel locations. A variety of methods have been proposed in literature to obtain ground disparities. Some methods directly work on raw data, such as image intensities, disparities, or 3D points, while others apply data projections to reduce the dimensionality of the raw data. These direct methods and projection-based methods are reviewed in this section.

Plane fitting

In a typical road scene, the ground manifold is the dominating surface that lower-bounds other objects in the scene. In this case, the manifold can be identified by finding the best-fit 3D surface given a set of 3D points.

When the ground manifold is assumed to be flat, the estimation can be approached by means of 3D plane fitting. In the case that the 3D points are derived from a disparity map, the fitting can be done directly in the image-disparity space. This is shown as follows.

Consider a plane $a_0X + a_1Y + a_2Z + a_3 = 0$ in 3D Euclidean space with plane coefficients $a_0, \ldots, a_3 \in \mathbb{R}$. A point (X, Y, Z) in 3D space is mapped onto an image pixel (x, y) following the pinhole model:

$$x = f_x \cdot \frac{X}{Z} + x_c, \quad y = f_y \cdot \frac{Y}{Z} + y_c \tag{2.3}$$

where (f_x, f_y) are the focal lengths, and (x_c, y_c) is the principal point.

Two calibrated and horizontally rectified pinhole cameras introduce a disparity space, where every pixel (x, y) in the (say) left image is mapped to $(x - d, y)$ in the right image via $d \in [0, d_{\text{max}})$, the disparity value bounded by d_{max}. The disparity-to-depth conversion follows (2.1). By first substituting (2.3) into the plane equation, resulting in

$$a_0 \cdot \frac{Z}{f_x}(x - x_c) + a_1 \cdot \frac{Z}{f_y}(y - y_c) + a_2 Z + a_3 = 0 \tag{2.4}$$

and then (2.1) into (2.4) producing

$$a_0 \cdot \frac{x - x_c}{f_x} + a_1 \cdot \frac{y - y_c}{f_y} + a_2 + a_3 \frac{d}{bf_x} = 0 \tag{2.5}$$

the plane in the Euclidean space is now modelled in the image-disparity space as another plane:

$$a'_0 x + a'_1 y + a'_2 d + a'_3 = 0 \tag{2.6}$$

in terms of $a'_0 = (bf_y)a_0$, $a'_1 = (bf_x)a_1$, $a'_2 = f_y a_3$ and $a'_3 = (bf_x f_y)a_2 - (bf_y x_c + bf_x y_c)$. This way the road plane can be found without any need of back-projecting a disparity map into the 3D Euclidean space [23].

Line fitting

When the height of a road manifold does not change significantly along the image's x-axis, the plane model reduces to a line:

$$d = -\frac{a'_1}{a'_2} y - \frac{a'_3}{a'_2} = my + b \tag{2.7}$$

which turns road-manifold estimation into a line-fitting problem of seeking the best-fit line model (m, b).

A computational efficient way to find the best-fit line is to use the histogram of the y-*disparity* or *row-disparity* map that models the distribution of (y, d) in 2D space (as introduced above).

In [47, 55], a Hough transform is used to detect the road manifold in form of a straight line in the y-disparity map. A more efficient and noise-resistant approach is to locate the dominating line following a stochastic process known as *random sample consensus* (RANSAC) [61]. The process first selects two bins randomly from the histogram, and a line hypothesis is solved $(\hat{m}, \hat{b})$. As values in the map define a density distribution, fitness of the hypothesis can be determined by summing up all the entries in $V_y(d)$ (see 2.2) that are considered to be on the line up to a tolerable deviation (i.e. the inliers).

Such a process is repeated for a finite number of iterations and the hypothesis achieving the highest fitness is considered to be the dominating line. As the precision of a line hypothesis is limited by the grid resolution, one may optionally perform a weighted line fitting based on the inliers to further improve the estimation.

2.2.3 Top-point calculation

The height of obstacles, which sit on the ground manifold, is obtained by seeking an ideal segmentation between *foreground* and *background* disparities. The goal of the stage is to find top points $t_1, t_2, \ldots, t_{N_{\text{col}}}$ that together with those base points found at the previous stage define the span of obstacles in a column-wise manner.

In [73] the height-of-obstacle calculation begins with selecting membership votes. Briefly, the membership values rely on the selection of every disparity of each column from the disparity for its member to the foreground obstacle. A membership value can be positive if it does not exceed the maximum distance of the expected obstacle disparity; otherwise it will be negative.

The Boolean membership vote brings the challenge to identify a threshold value for the distance; if this value is too large then all disparities will

be chosen from the foreground membership, and vice-versa. Therefore, the application of Boolean membership in continuous variation is a better alternative with an exponential function of the form

$$M(x,y) = \begin{cases} 2^{1-\delta^2(x,y)} - 1, & \text{if } y < b_x \\ 0, & \text{otherwise} \end{cases} \tag{2.8}$$

where

$$\delta(x,y) = \frac{d_x - D(x,y)}{d_x - Z^{-1}\left(Z(d_x) + \triangle Z\right)} \tag{2.9}$$

with $d_x = D(x, b_x)$ the disparity of an obstacle's base point in column x, Z as the disparity-to-depth conversion function, and $\triangle Z$ as a defined soft constraint range in depth.

The approximated Boolean function is illustrated by Fig. 2.7, and an example of evaluated membership values are shown in Fig. 2.8.

The next step is to decide the boundary between foreground and background votes from the membership function. For this purpose, the cost image is computed as follows:

$$C(x,y) = \begin{cases} \sum_{j=1}^{y-1} M(x,j) - \sum_{j=y}^{b_x} M(x,j), & \text{if } y < b_x \\ \infty, & \text{otherwise} \end{cases} \tag{2.10}$$

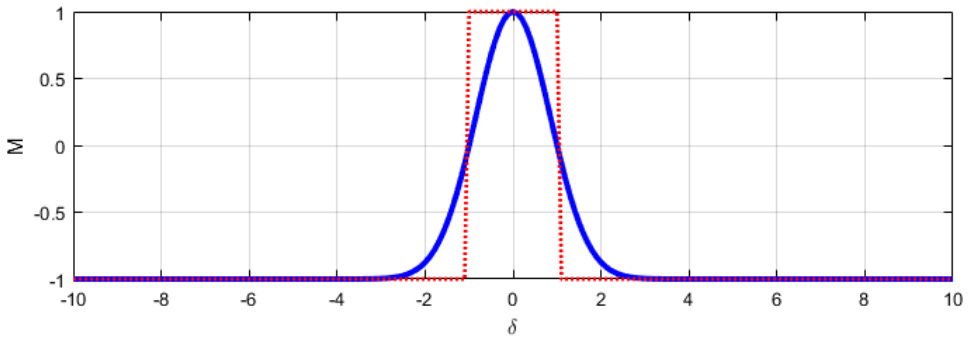

Figure 2.7: Exponential membership function (blue) adopted to approximate the Boolean membership (red). The width of the function is determined by ΔZ in (2.9).

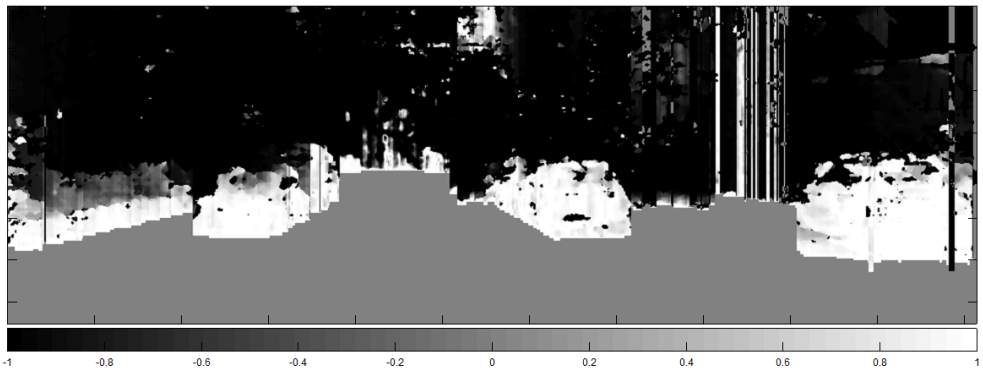

Figure 2.8: Evaluated membership of pixels in background (black) and foreground (white) classes. For pixels below the base points the membership value is undefined (grey).

A minimum cut that divides the cost image into upper and lower parts is then found using, say, a dynamic programming technique, while maintaining a smoothness constraint (see [73] for an example). The cut defines the top points $\{t_1, t_2, \ldots, t_{N_{\text{col}}}\}$.

A visualisation of the cost image used for the height segmentation is illustrated in Fig. 1.3. As can be seen, there are lower costs which show a high likelihood for performing a foreground-background separation.

2.2.4 Stixel extraction

By combining base points $b_1, b_2, \ldots, b_{N_{\text{col}}}$ found in Section 2.2.2 and top-points $t_1, t_2, \ldots, t_{N_{\text{col}}}$ found in Section 2.2.3, the stixels are extracted. A column grouping technique proposed in [11, 76] is carried out. Given $w \in \mathbb{Z}^+$, a predefined width of stixel, every w neighbouring columns are grouped across the whole image, resulting in $\lfloor \frac{N_{\text{col}}}{w} \rfloor$ non-overlapping stixels (see Fig. 2.9).

For the i-th stixel we have a set of w base points $B_i = \{b_{x_i}, b_{x_i+1}, ..., b_{x_i+w-1}\}$ and a set of w top points $T_i = \{t_{x_i}, t_{x_i+1}, ..., t_{x_i+w-1}\}$ where $x_i = (i-1)w+1$. The rectangle spanned from column $x = x_i$ to $x = x_i + w - 1$ and row $y = \min(T_i)$ to $y = \max(B_i)$ defines the scope of a stixel in the image domain. Instead of using only base points' disparities, all the disparities within

the scope are integrated to yield a more robust estimate of the stixel's depth z_i, by means of a histogram-based regression technique proposed in [11].

Stixel detection is also a way of ground manifold estimation; all the base points of stixels can act as interpolation points for ground-obstacle segmentation using geometry data with the aim of improving the accuracy. Besides that, it clearly represents the height of the first obstacle facing the vehicle along each viewing direction. A resulting stixel representation is illustrated in Fig. 1.3. The colours of the stixels encode the distance to the ego-vehicle. Red-scale colours represent closer objects while blue-scale colours represent farther objects.

The accuracy of the extracted stixels is directly affected by the estimated ground manifold. In the following section we provide more details about the ground manifold estimation methods.

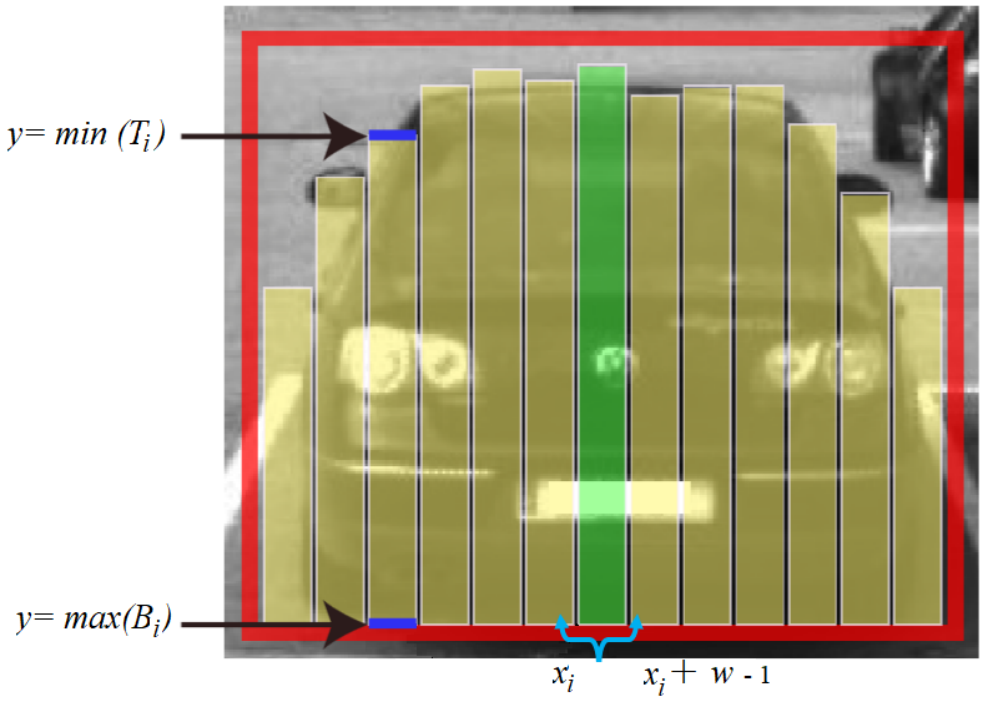

Figure 2.9: Visualisation of extracted stixels

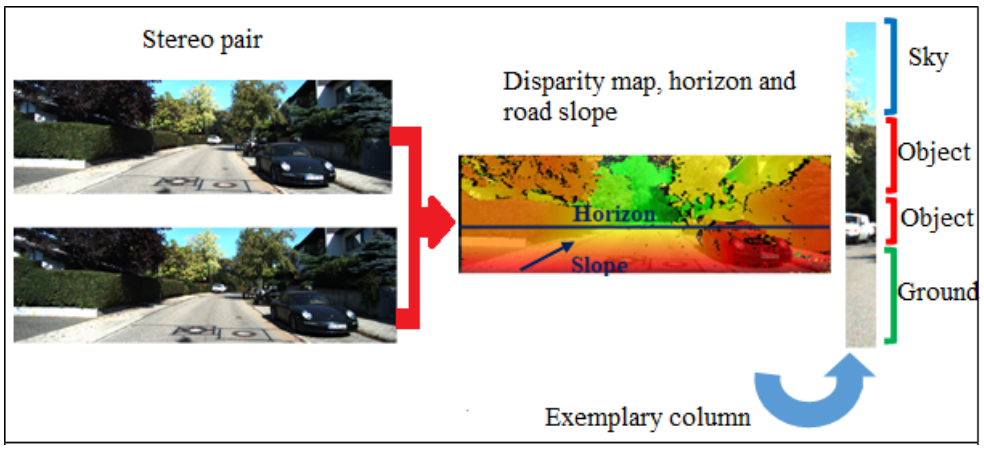

Figure 2.10: Visualisation of data model in multi-layer stixel, considering the dense disparity map as the input, and detecting a certain ground slope and horizon line, exemplary columns will be segmented into multiple stixels and categorised into object, ground and sky categories.

2.3 Multilayer stixels

Single-layer, or *first-layer* stixels are extended by [73] into a model with multiple stixels in each column (see Fig. 2.10) using a unified probabilistic approach. This approach uses dynamic programming applied on y-*disparities* to measure the occurrence of a certain class (i.e. object, sky, or ground) for multiple stixels per column. The extended representation yields a highly efficient modelling of scene objects in urban traffic environments [87]. It is used as a complementary model for various autonomous driving applications such as object tracking, which demonstrate how the velocity of stixels is tracked over a time-stamp [73].

A probabilistic method is carried out to assume that stixels in different columns are independent. For a disparity map D of size $N_{col} \times N_{row}$, each column x defines segments L_x describing classes in $\mathbb{C} = \{g, o, s\}$ (i.e. ground, object, or sky). Let $N_x \leq N_{row}$ be the total number of segments in column x.

Formally,

$$\begin{aligned} L &= \{L_x : 1 \leq x \leq N_{col}\} \in \mathbb{L}, \text{ with} \\ L_x &= \{s_x^n : 1 \leq n \leq N_x\} \end{aligned} \tag{2.11}$$

for each column x, where $\mathbb{L}$ is a set of possible segmentations. A segment s_x^n is represented by

$$s_x^n = \{y_n^b, y_n^t, c_n, f_n(.)\} \tag{2.12}$$

with $1 \leq y_n^t \leq y_n^b \leq N_{row}$ (note: y-coordinates go downward; thus the top coordinate y_n^t is less than, or equal to the bottom coordinate y_n^b), $c_n \in \mathbb{C}$, and function $f_n(.)$ is defined for y, with $y_n^t \leq y \leq y_n^b$, for the disparity of segment s_n at row y. The *ground-based function* is set to be equal to the disparity gradient of the ground surface $f_g(y) = \alpha - (y_{horizon} - y)$. The assumed value for the *sky function* is $f_s(y) = 0$. For an *object function*, we have that $f_o(y) = \mu_n$ (where μ_n is the mean disparity within s_n).

Functions $f_n(.)$ are implemented based on data-terms defined for each function, and they are verified based on prior-terms. This step is treated as a typical *maximum-a-posteriori* (MAP) estimation problem. We aim to find the most probable labelling in $\mathbb{L}$:

$$L^* = \text{argmax}_{L \in \mathbb{L}} \Pr(L|D) \tag{2.13}$$

where $L \in \mathbb{L}$ is an ordered list of N_x adjacent and non-overlapping stixel segments s_n.

2.4 Stixel applications

Although stixel implementation for traffic scene modelling was just introduced less than 10 years ago, there is already a wide range of applications used in the industry [30]. Basically, the stixel was introduced to serve as a middle layer to represent obstacles more efficiently in robotics or the automotive domain. The stixel representation has come as a precise successor of different object representations in 3-dimensional environments [19]. Previous representations in this domain are typically either complicated (not practicable) or very simple to use. For example, the box models are frequently

Figure 2.11: Scene representation using ROADDNA [113].

used to represent vehicle or pedestrian tracking [97, 104], also, to represent distances and object classification [67]. However, box models are being rigid and incapable to track the contour of objects accurately (i.e. occlusion scenarios). Another approach could be defined by polygonal shapes [77], or free-forms [17]. As indicated by [75], these models are considered to be counter-intuitive. Another representation of obstacles is supported by intelligent maps called as roadDNA [113] (see Fig.2.11). The main advantage of these high-definition (HD) maps is to be understandable by machines, enable vehicle to locate themselves, and visualise small details of traffic environment such as traffic signs, poles, and lane geometry. To build such representation, building HD maps requires advance mapping capabilities.

Sophisticated representations introduce a solution for visualising the state of an object, and to track non-stationary objects in a non-explicit grid. A dynamic 4D map is proposed by [36], and dynamic occupancy grids by [21] which demand the usage of massive system resources (e.g. main memory), and they are mathematically complex. Hence, considering that automotive and robotics applications are complex in nature, we consider to use stixels for several reasons. First, they bridge the gap between low-level (pixel based) and high-level (semantic-based) vision. The level of details for a particular object can be chosen freely by adjusting the width of stixels (small width gives more details and vice versa). An example of stixel implementation in the automotive domain is depicted in a virtual 3D view in Fig. 2.12. The colour encodes distances from close (red) to far away (green). An application of using stixel for pedestrian detection in street scenes is proposed

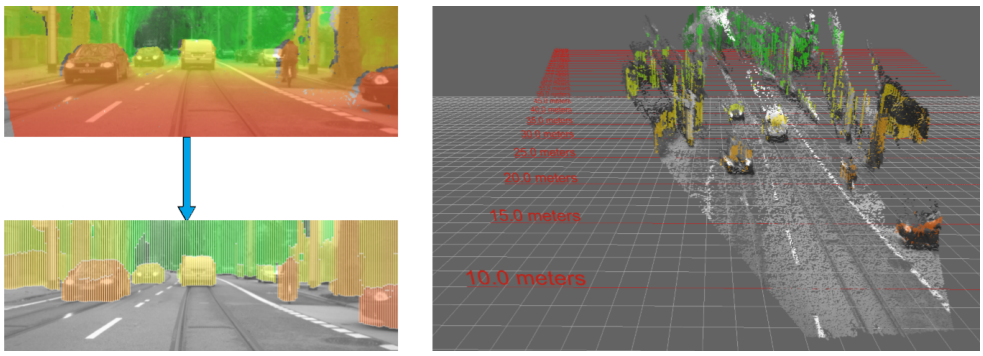

Figure 2.12: Virtual 3D view of stixel representation in the automotive domain [90]. *Left-top*: Disparity map. *Left-bottom*: Resulting stixel representation (multi-layer). *Right*: The measured disparity and stixel representation is provided in a 3D view.

by [14]. The method targeted high-speed performance without a depth map; see Fig. 2.13. They adopted ground-plane detection by matching pixels in left-right images.

The extended representation yields a highly efficient labelling and modelling of scene objects in urban traffic environments [27, 87, 88]. It is used as a complementary model for various autonomous driving applications such as object tracking (see Fig. 2.14, which demonstrates how the velocity of stixels is tracked over a timestamp [63, 76]).

Another application is object detection based on stixels, where stixel properties are employed to reduce classification assumptions; this significantly supports fast object detection [20, 25]. The recognition performance and

Figure 2.13: Pedestrian-based stixel estimation; green lines indicate stixels [14].

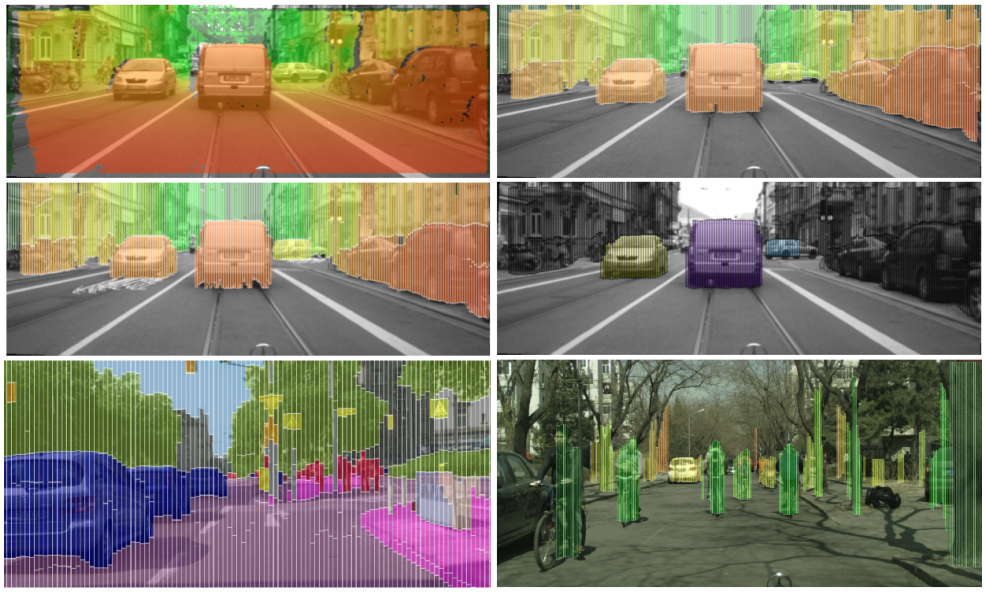

Figure 2.14: Examples of stixel applications. *Top-left*: Disparity map using SGM [33, 45], the colours encode distances to obstacles; red-green is close-far away. *Top-right*: Multi-layer stixels which demonstrate object boundaries [74] *Middle-left*: Dynamic stixels where arrows point to the predicted position of stixels within half-second time intervals [76] *Middle-right*: Multi-class scene representation based on dynamic stixels [27]. The three moving objects are represented in different colours cyan, magenta, and yellow. The static background is shown in black. *Bottom-left* : Cyclists detection using stixel representation [58]. *Bottom-right* : Semantic stixel representation; colour codes represent semantic classes for each given segment [86].

computational efficiency is significantly reduced when a stixel-based filtering scheme is used (see Fig. 2.15). In [25] the prior knowledge about object classes of interest is fused with 3D information given by stixels to accurately focus processing on well-defined local regions; in this way initial object hypotheses will be established.

GPU-based acceleration for stixel calculation is presented by [43]; the used stereo-matcher was based on GPU-acceleration of a dense stereo calcu-

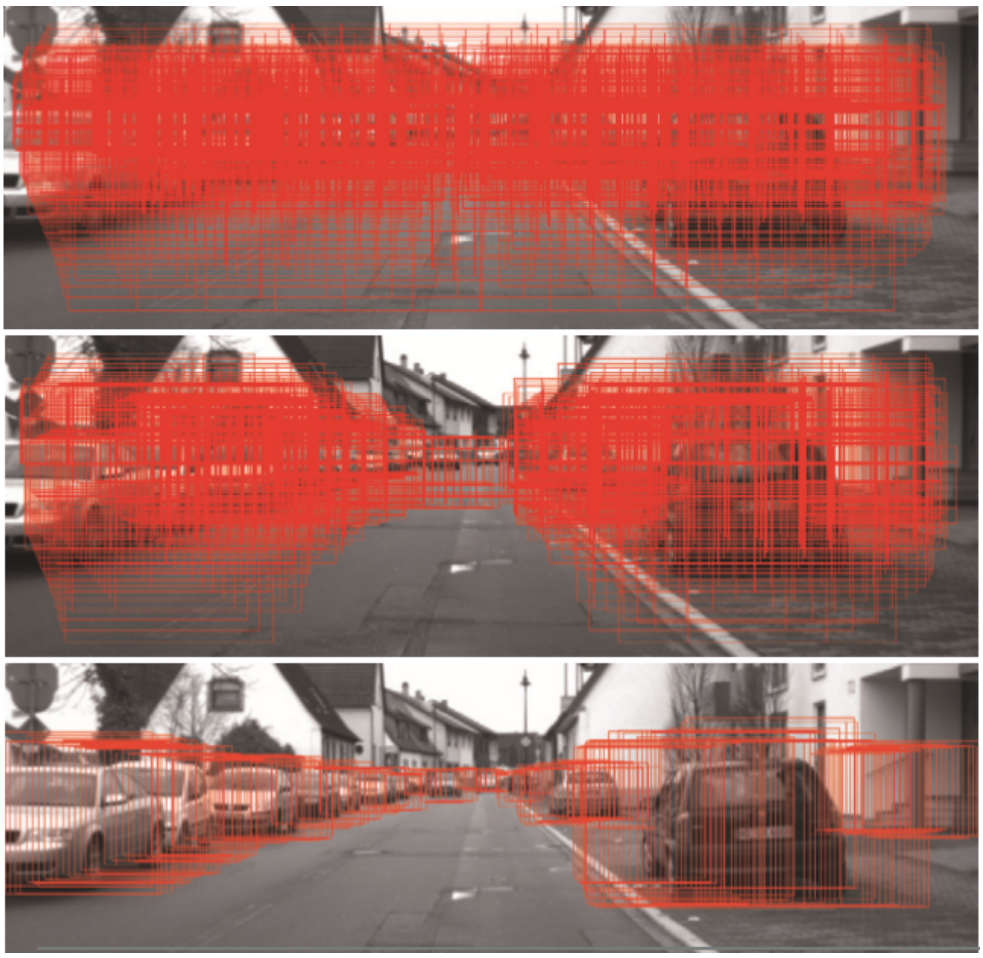

Figure 2.15: ROI captured through different techniques [25]. *Top*: ROI using monocular vision [26]. *Middle*: Stereo-based depth filtering [32]. *Bottom*: Stixel-based filtering [25].

lation using semi-global matching (SGM). The ground manifold is estimated by assuming a ground plane, and estimated by using a line fit in y-disparity space (based on a Hough line transform) to estimate the road surface.

Stereo-based ground-truth for cyclist detection is proposed by [58] where stixels were used to improve the detection process. The large annotated

datasets were used to train and evaluate cyclist detectors using a stereo-proposal-based fast R-CNN. Recently, joint stixel representations, combining semantic data and depth, are proposed to integrate both categories in terms of a joint optimised scene model [44, 86], as shown in Fig. 2.14.

2.5 Discussion

Ground-plane based analysis estimates a plane for road surfaces is still popular to detect road surface. Due to road-geometry variations, and difficulties in recording those properly (e.g. weather, road conditions, or traffic density), there is also ongoing works to improve ground-manifold estimation and stixels calculation [43, 56, 59, 84]. However, most of these models still using ground-plane for detecting road-surface which is prone to errors as road-geometry is not always perfect plane.

Rapid stixel-based analysis enhances stixel extraction by having lower computational costs; authors in [14] proposed a direct stixel computation by changing the parametrisation from the disparity space into a pixel-wise cost volume for speed improvement. In [57], the authors use deep convolutional neural networks for free-space detection using monocular vision, while obstacle detection and stixel calculation is done by using stereo vision.

A fast stixel computation without using depth maps is proposed in [15]. It supports high-speed pedestrian detection (at the speed of 200 fps). In [15] a pixel-wise cost volume is generated which can be used to estimate a ground plane (i.e. the special *linear* case of a ground manifold). This introduces stixel disparities that contribute to the identification of the height of a stixel. Rapid stixel-based methods have some drawbacks. These methods suffer from low depth accuracy, which in turn affects stixel extraction negatively. Besides, it focuses on free-space separately from the obstacles.

Base-line stixel calculation applies *membership-based methods* [11, 73]. Briefly, here a membership function describes how disparity values in one column contribute to the voting, either in favour of an object located on top of the ground manifold, or not. The selection process depends on an exponential membership function rather than background subtraction [73]. Depending merely on disparity map to calculate membership function is a challenging

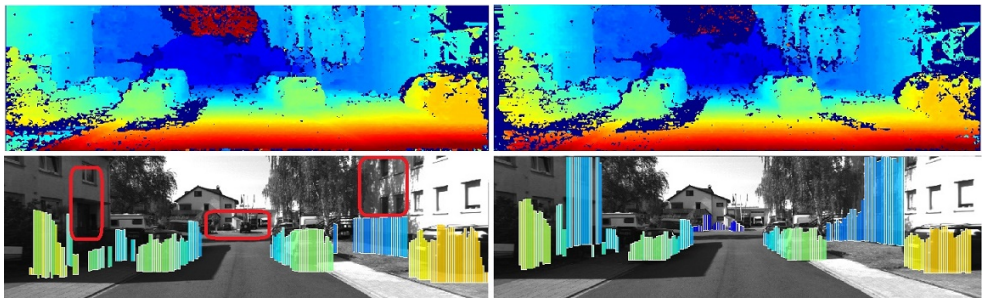

Figure 2.16: Example of a stixel world using the KITTI (binocular versus trinocular) *residential* dataset. *Top-left*: Disparity map. *Top-right*: Synthesized trinocular disparity map. *Bottom-left*: Stixel estimation using occupancy mapping (red rectangles represent the missing stixel information). *Bottom-right*: Proposed stixel estimation using the transitivity disparity error (including recovered stixel information compared to the bottom-left image).

tasks because most of traffic scenes come with a variety of lighting conditions.

Colour fusion models compute stixels by using stereo images (i.e. depth cues) in combination with colour appearance. Such methods have been presented for stixel segmentation [83]; their implementation can be done by using a low-level fusion of depth with image signals or semantic information in free-space detection and stixels. Scharwächter et al. employed pixel classification with random decision forests [87], while in [20] semantic information via object detectors is used for a suitable set of classes. Yet another method has been presented in [83] to improve stixels using low-level appearance models in an on-line self-supervised framework.

In order to recover stixel segmentation, an adopted colour fusion model for free-space detection might not be suitable due to the shortages highlighted in [101].

Stereo confidence-based methods, on the other hand, use confidence estimation within the stereo-matching process to replace spurious disparity matches by interpolating surrounding disparity values at these locations; see [16, 39, 72] for examples. In [72], the authors incorporate three confidence measures, namely the naïve peak-ratio (PKRN), the maximum-likelihood mea-

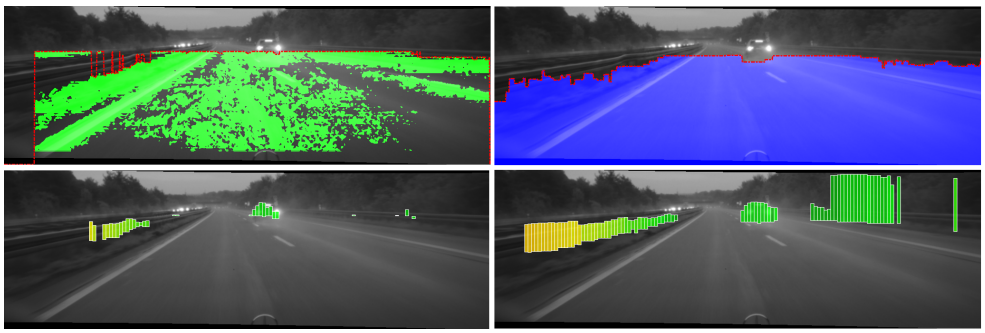

Figure 2.17: Example of a stixel world for a bad-weather situation. *Top-left*: Ground-plane estimated by calculating a linear lower envelop (by using a Hough transform) in y-disparity space; green illustrates the use of a Hough transform. *Top-right*: Dynamic programming is used for ground-manifold estimation as proposed in this study; blue illustrates the use of dynamic programming graph-cut. *Bottom-left*: Stixels for the estimated ground plane. *Bottom-right*: Stixels (via dynamic programming graph-cut) as proposed in this study.

sure (MLM), and local curve (LC) information into stixel representations. The stereo confidence measures use stereo confidence cues based on an extended Bayesian approach. In [39], the authors adopted an ensemble learning classifier to increase accuracy in stereo-error detection. In [16], histogram-sensor models are explored to model on a real-world application using a global formulation of 3D reconstruction through an occupancy grid.

Object detection models, where stixel properties are employed to reduce classification assumptions, significantly support fast object detection [20].

Multiple-stixel segmentation with GPU-based acceleration is proposed by authors in [43] for stixel calculation; the used stereo-matcher was based on GPU-acceleration of a dense stereo calculation using SGM. The ground manifold is estimated by assuming a ground plane, and estimated by using a line fit in y-disparity space (i.e. a Hough line transform) to estimate road surface.

As identified earlier, degradation disparities lead to a lot of false positive ground manifold detections which is interpreted as false-negative single-layer stixels as well (since a stixel is build on the base-point). Figure 2.16,

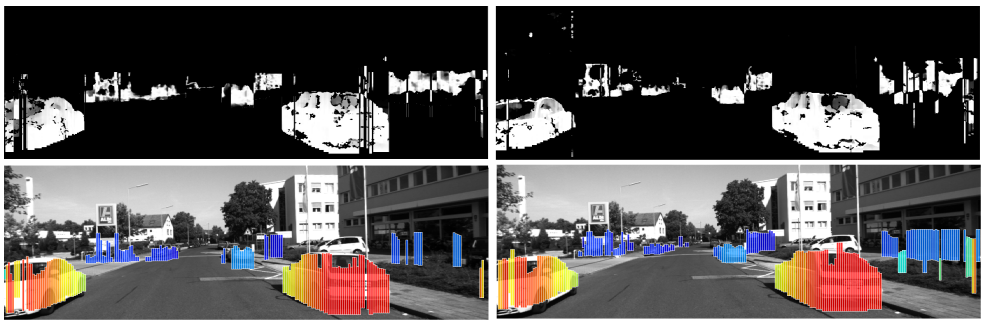

Figure 2.18: Example of stixel's height segmentation. *Top-left*: Membership votes for pixels estimated by using disparity information; foreground votes in white, background votes in black, and neutral votes in gray. *Top-right*: Membership votes estimated by fusing disparity and colour-based saliency information (as proposed in this study). *Bottom-left*: Stixels for the membership vote shown top-left. *Bottom-right*: Stixels for the membership vote shown top-right.

bottom-left, shows an example. Hence, our study aims at producing a confidence indicator for ground function and obstacle detection by employing, e.g., trinocular vision. To this end, this study applies a confidence map, which can vote for consistent disparity values to be undertaken during ground-manifold estimation.

Nevertheless, for optimising stixel representation for challenging datasets we need to efficiently leverage ground manifold information (see Fig. 2.17). In multi-ocular systems, a disparity map is calculated and the corresponding y-*disparity map* [55] is computed by accumulating pixels with the same disparity value on a horizontal row of the disparity map. It is challenging to ensure the monotonicity of the curve detected in y-disparity space.

One solution to this problem is to find a representation that ensures monotonicity of curve detection in y-disparity space. To gain more accuracy and to satisfy our requirements, we consider a dynamic programming approach for finding maximum-length line segments in the y-disparity space, defining a piecewise linear curve as our envelop function.

Traffic scenarios may define a challenge for this base-line stixel calcula-

tion process due to several reasons. Stixels require a disparity signal of good quality, but this often decreases in cases of occlusions or image regions with little texture information [7]. This leads to an immediate negative effect on height detection due to resulting noise in the membership map. Challenging lighting conditions may play a disturbing role when portraying foreground disparities in a membership map, causing, e.g. that foreground pixels are considered to be part of the background.

A *saliency map* [9] is a possible way for identifying visual boundaries of objects in colour images; map calculation involves calculating saliency of each pixel with reference to its neighbourhood in terms of lightness and colour properties [9]. We study the fusion of disparity membership with a saliency map towards improved height calculation of stixels. Figure 2.18

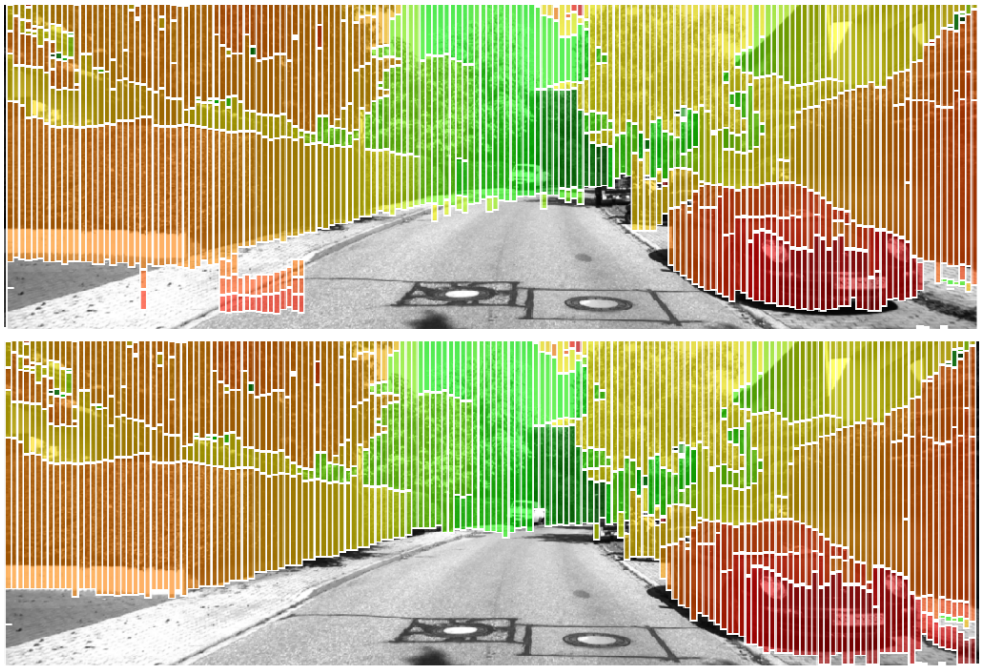

Figure 2.19: Example of multi-layer stixel segmentation. *Top*: Base-line approach using binocular stereo only. *Bottom*: Proposed multi-layer stixel using transitivity disparity analysis

shows results for base-line stixel calculation and a method proposed in this thesis.

The base-line stixel calculation may lead to more erroneous results; see Fig. 2.18, bottom-left. The figure already illustrates that the proposed method improves results in general (without being perfect). Also including a saliency map ensures improved consistency of foreground pixels and also helps to reduce false-positives.

Finally, for multi-layer stixel estimation, each stixel carries important but noisy information, and it is challenging to decide for each stixel to which object it belongs to (see Fig. 2.19). By revising the problem of stixel generation into a segmentation problem, we follow the proposed model defined in [73]. In this study, we extend the concept by paying attention to missing or faulty disparities especially in the sky and ground regions based on a defined confidence map derived from multi-ocular vision or employing LiDAR sensors. The extension is required since we are interested in semantic segmentation, and this step supports the labelling efficiently.

2.6 Summary

The reviewed studies are focused on certain types of camera configurations and road manifolds. For example, binocular vision and plane detection are widely used to estimate stixels, yet some of the studied traffic scenes show clearly non-planar ground manifolds, and are challenging, e.g. reflected in low qualities of disparity signals. Improvement may start with improving the quality of disparity signals by using different sensors, and by empowering the ground-manifold information. Moreover, an efficient and low-cost architecture approach is required for ensuring applicability in mobile systems.

Chapter 3

Ground Manifold Modelling

This chapter highlights the role of ground manifold modelling for the next generation of intelligent autonomous driving. Existing ground manifold methods, using single disparity maps, still suffer from noise and inconsistency, and false-detection rates for obstacles in challenging datasets. We present a novel method for stixel construction using a calibrated collinear trinocular vision system. Furthermore, we aim at improving binocular-based ground manifold estimation; we apply cuts through a given y-disparity matrix along columns to achieve a minimal cost specified by the matrix itself while maintaining desired smoothness; the minimisation is based on the Viterbi algorithm, a dynamic programming technique. - Parts of this chapter is published in [1, 5, 6, 7].

3.1 Impact of ground-manifold models

A ground manifold may be encoded as a disparity map GD where $GD(x, y)$ stores the disparity of the ground at pixel location (x, y). Let D be the disparity map computed by stereo matching, pixel (x, y) is considered to be *above the ground manifold* if $D(x, y) > GD(x, y) + \varepsilon$, where $\varepsilon > 0$ defines a tolerance margin.

A variety of methods has been proposed in literature [6, 7, 8, 12, 55] to obtain map GD. As we have seen in the previous chapter, some ground estimation methods work directly on raw data (i.e. low-level image). Raw data include image intensities, 3D points, and disparities. Other methods apply data projections to reduce the dimensionality of the raw data. In this chapter we see a potential opportunity to obtain ground manifolds based on either one of two models.

The first represents polynomial fitting as a popular data fitting model to be applied in y-disparity space. We are expecting in certain situations that road surfaces are not perfectly planar. Hence, we have addressed earlier the question "to which degree the piecewise linear curve detection in y-disparities can affect stixel detections". Furthermore, the occupancy polar grid or x-disparities (contrary to y-disparities) have been proposed in [11] for constructing stixels. The free-space calculation depends here on a graph-cut applied on the occupancy grid supported by dynamic programming (DP). This structure has several shortcomings highlighted in [76]. Besides,"the polar occupancy grid requires a high computational effort" [73].

Therefore, we propose a method that applies graph-cut to detect piecewise linear curve segments in the y-disparity map. Based on first-order differentiation of the y-disparity map, we applied a novel graph-cut technique and enforce monotonicity to ensure a piecewise linear curve, also detected under challenging circumstances.

There is ongoing current work to improve ground-manifold estimation and stixel calculations elsewhere; see, for example, [43, 56, 84, 85]. However, the used model for ground-manifolds in these efforts is still the ground plane which is prone (as demonstrated by us) to errors as the road geometry is not always perfectly planar.

3.1.1 Curve fitting

Road-manifold fitting methods as presented in Section 2.2.2 can only handle planar road surfaces [95]. For a non-flat road geometry, the y-disparity map shows a curved distribution of pixel disparities. In [55], such a curve is approximated by a piecewise linear function, which is estimated as a lower envelope formed by straight lines corresponding to the k-strongest peaks in the Hough space, with $k \geq 1$ being a chosen parameter.

In more recent work [7, 91], the approximating curve is modelled by a 2^{nd} order polynomial function or 3^{rd} order B-spline. Adopting the polynomial model, the ground manifold estimation problem is solved by finding the coefficients of a polynomial $f(y)$ of degree n that best fits the curve in the y-disparity map:

$$d = f(y) = a_n y^n + a_{n-1} y^{n-1} + \ldots + a_1 y + a_0 \tag{3.1}$$

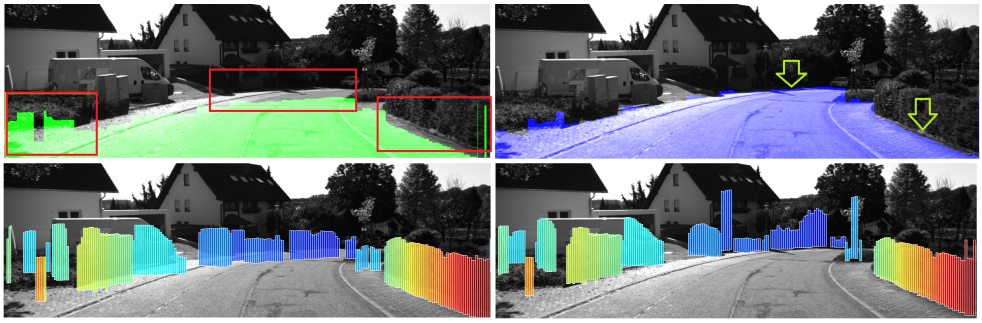

Fig. 3.1: Stixel world. *Top-left*: Ground manifold estimated via occupancy grid (i.e. a plane). *Top-right*: Polynomial fitting (with $n = 2$) of a ground manifold. *Bottom-left*: Stixels via occupancy grid which do not fit to obstacles in terms of distances. *Bottom-right*: Stixels via polynomial fitting (with $n = 2$). Green (see *top-left*) illustrates a ground plane, and purple (see *top-right*) a possibly curved ground manifold (by polynomial fitting).

where $a_0, a_1, \ldots, a_n$ are the coefficients, and the degree $n > 1$ is selected according to accuracy requirements for the algorithm. For impacts of planar versus polynomial ground manifold approximations on stixel calculations, see Fig. 3.1.

Similar to the line-fitting technique, the fitness of a curve is defined by summing up all the curve's containing entries in V [8].

In order to generate the coefficients of the polynomial according to the degree specified, we need to compute a least-square polynomial for a given set of data. Following the least-square principle, we obtain the parameters $a_0, a_1, \ldots, a_n$, which minimise the total square error:

$$E(a_0, a_1, \ldots, a_n) = \sum_{i=1}^{m} [y_i - P(x_i)]^2 \tag{3.2}$$

where $m \geq n$ is the number of samples. The optimal coefficients can be solved linearly.

The generated polynomial $P(x)$ fits given data in general better than just a fitted straight line; and this would impact of detection result (see Fig. 3.1).

The final step, calculating base-points b_1, b_2, ..., $b_{N_{col}}$ of stixels for row y, is now straightforward. Consider a pixel with a given disparity, thus also

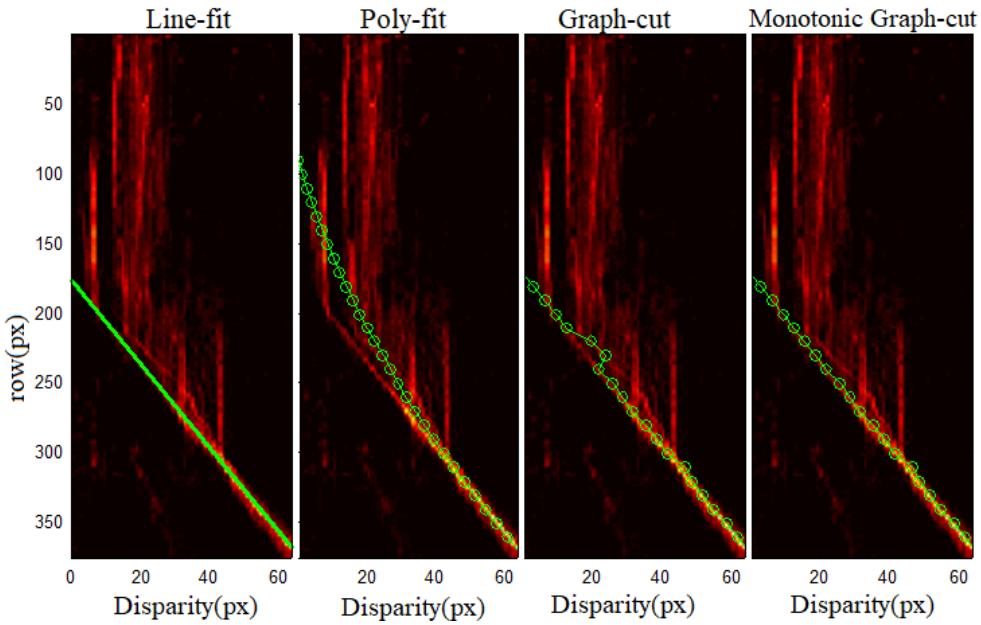

Figure 3.2: Demonstration of y-disparity-based ground-manifold modelling. *First column*: Line fitting. *Second column*: Polynomial-based curve fitting. *Third column*: Graph-cut-based curve fitting. *Fourth column*: Graph-cut-based curve fitting with enforced monotonicity.

with a defined depth, in row y. The projection ray of this pixel intersects the ground manifold at a particular point in *3-dimensional* (3D) camera space.

3.1.2 Dynamic programming and graph cut

Curve models with higher degrees provide flexibility to model a road manifold in y-disparity space. The degree of freedom is still limited by the adopted parametric model (see Fig. 3.2). Furthermore, curve models do not guarantee monotonicity that is often desired, as the depth of a road manifold does in general not increase as the row index goes from y to $y + 1$ (i.e. downward in the image).

Following a discrete formulation, the curve fitting process is essentially a graph cut problem, which aims at finding a set of quantised disparities

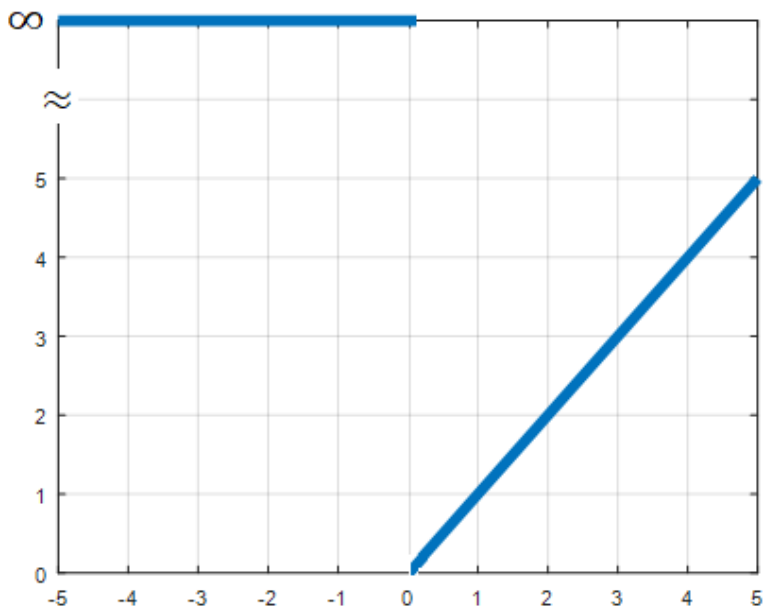

Figure 3.3: L_1 asymmetric Potts model

$\mathbf{d} = \{d_1, d_2, \ldots, d_{N_{\text{col}}}\}$ that minimises a cost function subject to smoothness constraints.

Such a cut $\mathbf{d}$ divides the y-disparity map into left and right parts (see Fig. 3.3).

To find the lower bound of the road manifold, the cost function can be defined by using a first-order derivative V_y of the y-disparity map V (i.e. along row y); see [5]:

$$E(\mathbf{d}) = \sum_{i=1}^{N_{\text{col}}} V_y(y, d_i) + \gamma \sum_{i=2}^{N_{\text{col}}} \Theta(d_{i-1}, d_i) \tag{3.3}$$

where $\gamma \geq 0$ defines a penalty for Θ, the smoothness function.

The value of γ depends on the scale of the data term. To ensure the monotonicity of a cut, the smoothness term can be specified by an asymmetric L_1

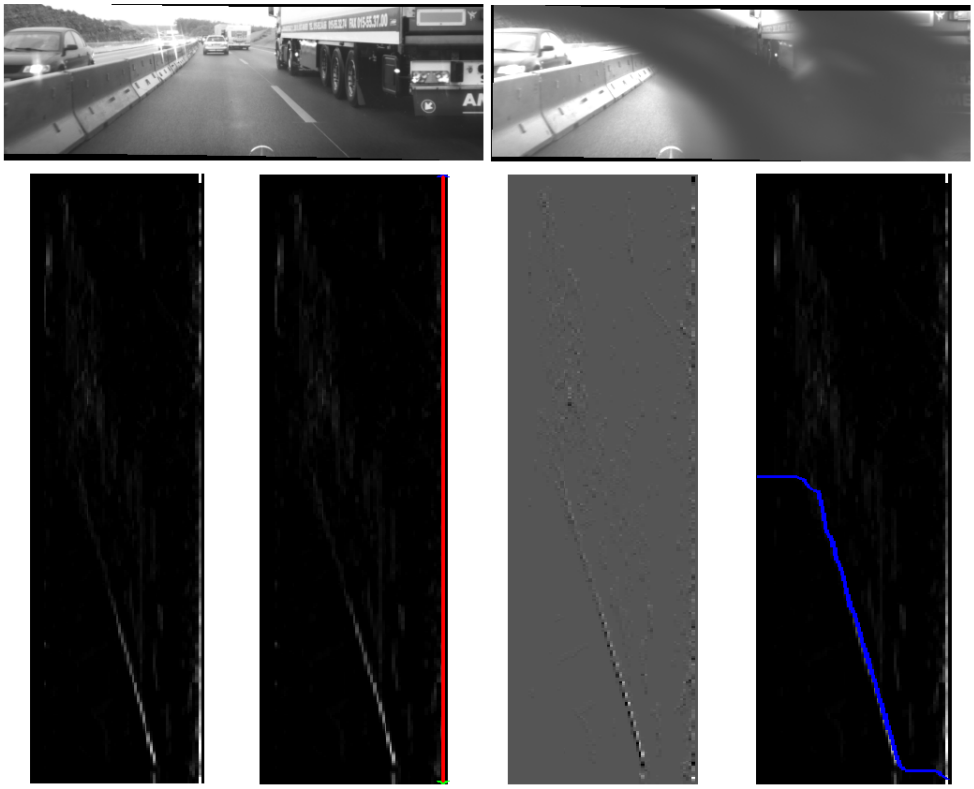

Figure 3.4: Result of piecewise linear approximation for challenging datasets - frame 178 of Seq. 8. *Top*: Stereo pair of 6D vision `bad_weather` dataset. *Bottom, left to right*: y-disparity map; straight line (red) detected on the right-side using Hough line; first-order differentiation of y-disparity image (proposed in this study); curve (blue) detected using dynamic programming graph-cut (our method).

Potts model:

$$\Theta(d_i, d_j) = \begin{cases} \infty, & \text{if } d_i > d_j \\ d_j - d_i, & \text{otherwise} \end{cases} \tag{3.4}$$

The result of the piecewise linear curve detection using graph-cut, compared to line fitting, is provided in Fig. 3.4. The result shows the successful detection if using the proposed graph-cut in a very challenging scenario where ac-

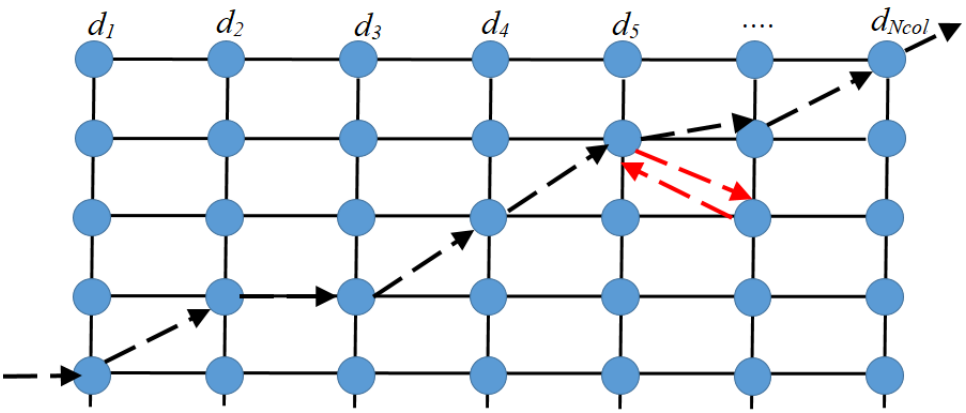

Figure 3.5: Graph-cut mechanism on y-disparity map

tivated window wipers are shown in the right image. Based on dynamic programming, an optimal cut can be obtained using the Viterbi algorithm [100]. An exemplary illustration of the implementation of graph-cut on y-disparity space is provided in Fig. 3.5.

3.2 Extension of trinocular vision

The idea of the y-disparity space can be generalised to a multiocular camera set-up. As disparity spaces, derived from different stereo pairs, are not consistent to each other, the disparities have to be converted first into a universal representation (e.g. by using inverse-depth) as shown in Fig. 3.6. Alternatively, one of the disparity spaces may be chosen as a reference such that all the disparities can be transformed and integrated appropriately.

The trinocular vision or (third-eye technique) can come with different steps as the number of views is increased so all three cameras need to be carefully calibrated with respect to others. Besides, the memory management needs to be more efficient to allow more data and more parallel processing in pipeline while serving client application. It also leads to various types of applications as depicted in Fig. 3.7.

Illustrated by Fig. 3.8, geometric rectifications and synchronisation are

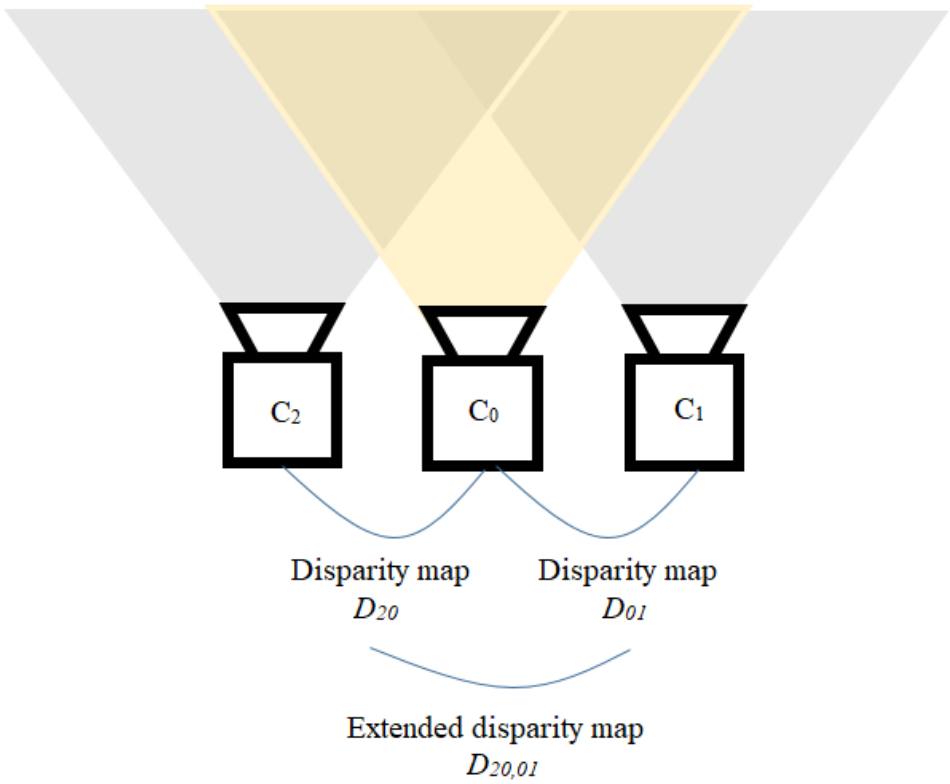

Figure 3.6: Third eye (trinocular) configuration

considered to be challenging steps in trinoncular vision. Rectification is ensured by using parameters calibrated from (say) 75 views with partial occlusions. Note the flickering of lights, which is one good evidence that all three cameras are successfully synchronised by the trigger box. Note also that the epipolar lines are well aligned after rectification.

In [7], a trinocular implementation is proposed for a generalisation of the y-disparity map for three binocular stereo pairs defined by three cameras; Figure 3.9 shows a trinocular data example from the KITTI *road* dataset [35]. Our extension is based on *transitivity error analysis in disparity space* (TED) as introduced in [18]. The approach is briefly described as follows.

Consider left-to-right stereo-pair matching. A disparity map $D : \Omega \rightarrow$

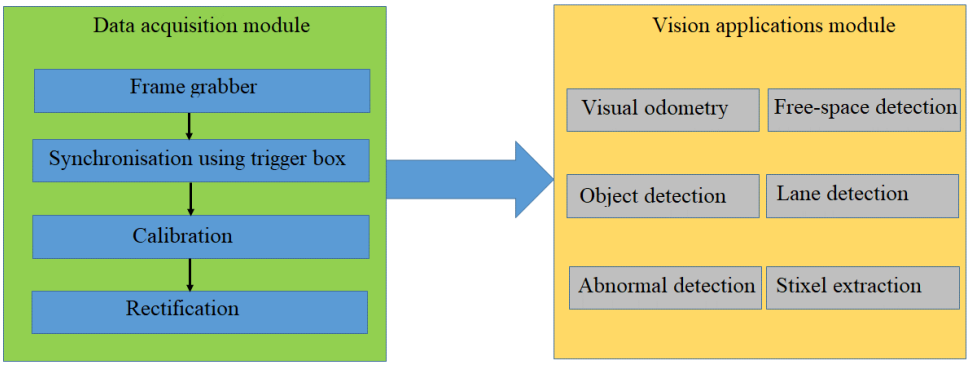

Figure 3.7: Third eye (trinocular) steps and applications

$[0, d_{\max}]$ maps each pixel location $(x, y) \in \Omega$ from the left image domain Ω into a pixel location $(x - D(x, y), y)$ in the right image. A disparity map defines therefore a warping function $\mathcal{M} : \Omega \to \mathbb{R}$ as follows:

$$\phi(\mathcal{M}, D)(x, y) = \mathcal{M}\left(x - D(x, y), y\right) \tag{3.5}$$

Figure 3.8: Synchronised and calibrated sequence - Wuhan dataset

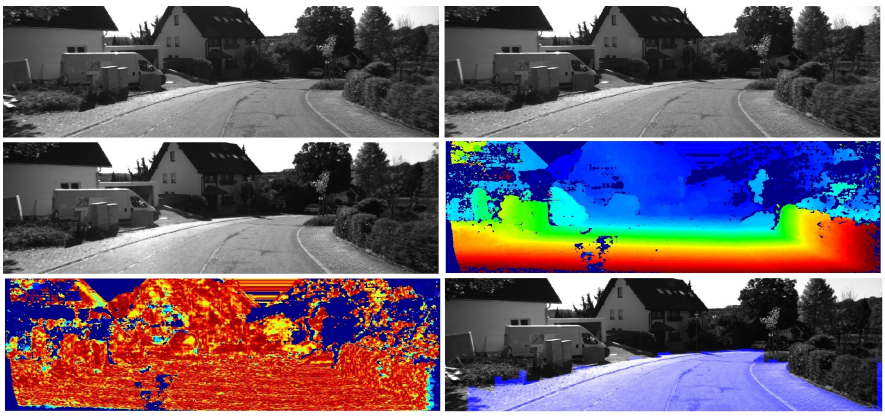

Figure 3.9: Trinocular confidence and free space. *Top row and middle-left*: Trinocular stereo pair from the KITTI *road* dataset. *Middle right*: TED-based disparity. *Bottom left*: Confidence indicator in which red and blue pixels indicate high and low confidence values, respectively. *Bottom right*: Calculated free-space (using y-disparity, confidence map, and polynomial curve fitting).

Given a collinear m-camera configuration, there are $m(m-1)/2$ left-right stereo pairs, for $m \geq 2$. The warping function ϕ can be used to construct the concatenation of any two disparity maps, following

$$\tau(D_{ij}, D_{jk})(x, y) = D_{ij}(x, y) + \phi(D_{jk}, D_{ij})(x, y) \tag{3.6}$$

where $1 \leq i, j, k \leq m$. This concatenation defines the *TED-based disparities* and will improve the coverage of valid disparities as shown in Fig. 3.10.

A *TED-based error measure* can now be defined as

$$d_{ik,ijk}(x, y) = \|\tau(D_{ij}, D_{jk})(x, y) - D_{ik}(x, y)\| \tag{3.7}$$

with respect to camera sequence (i, j, k). Function $d_{ik,ijk}$ measures the difference between an explicitly computed disparity map D_{ik} and the concatenated one, i.e. $\tau(D_{ij}, D_{jk})$.

For $m = 3$ (trinocular case) and for applying TED when building a y-disparity map with respect to a camera pair, say camera pair $(0, 2)$, a *trinocular confidence measure* is defined:

$$\Gamma(x, y) = \frac{1}{1 + \|\tau(D_{01}, D_{12})(x, y) - D_{02}(x, y)\|} \tag{3.8}$$

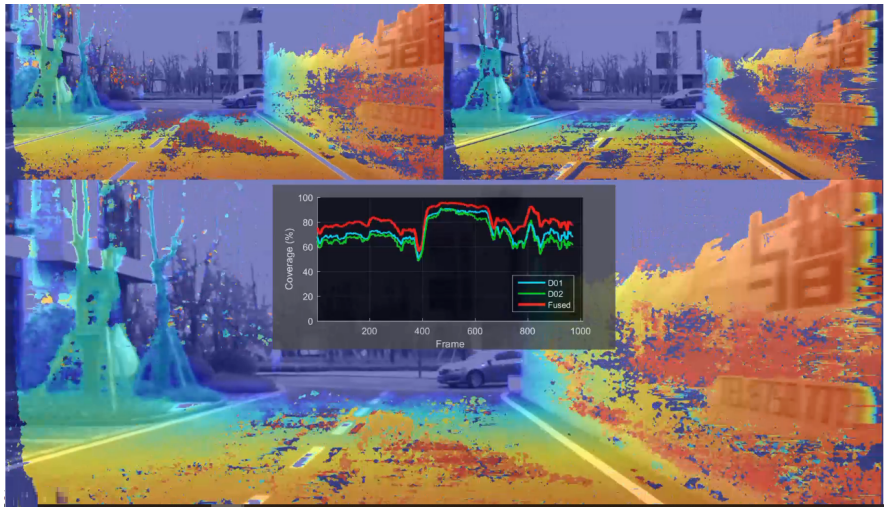

Figure 3.10: Disparity coverage in Wuhan dataset. Disparity map from camera 0 and camera 1 (*top-left*). Disparity map from camera 0 and camera 2 (*top-right*). TED-based disparities (*bottom*).

and a TED-weighted y-disparity map is constructed as follows:

$$V(y,d) = \sum_{1 \le x \le N_{\text{col}} \wedge Q(D_{01}(x,y))=d} \Gamma(x,y) \tag{3.9}$$

Here, elements with higher TED-based confidence become more influential in the weighted y-disparity map, which can then be processed using again the described line fitting, curve fitting, or dynamic programming techniques. When a weighted sum of trinocular confidence values is applied, this will enforce candidates in y-disparity map with higher TED-based confidence to become more prominent; see Fig. 3.11.

3.3 Experimental results

We implemented the stixel construction process for four different disparity-based ground-manifold models as introduced above.[1] The base-line stixel method is implemented by mapping disparities into occupancy grids. Such a

[1]Implementation is in MATLAB R2017A.

scheme suffers from the shortcomings highlighted in [73]. We compare with four models as discussed above: plane-fit and line-fit [23], poly-fit [6], and graph-cut [5]; all are mainly dependent on the y-disparity space. This brings numerous advantages for ground-manifold detection. It is stated in [73] that working with original image coordinates, identified by the y-disparity space, is more practical when including probabilistic densities into the used model.

Using the y-disparity space suppresses additional quantisation artefacts,

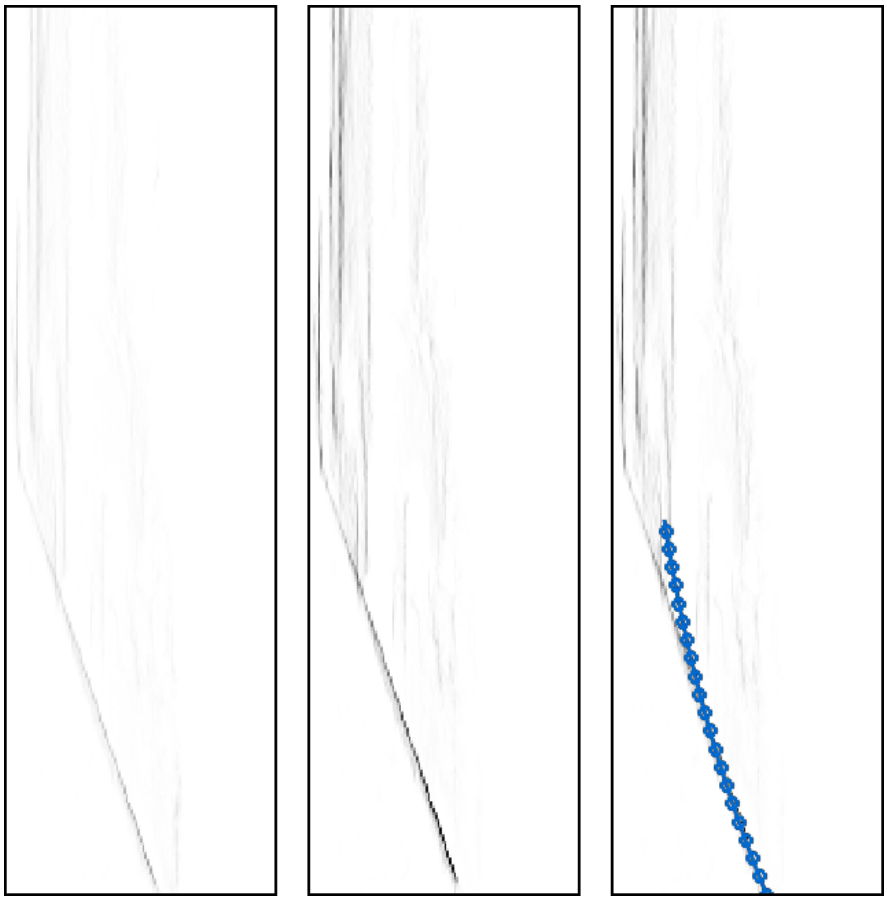

Figure 3.11: Demonstration of y-disparity using trinocular vision. *Left*: Common cardinality-based y-disparity map. *Middle*: Novel TED-based y-disparity map. *Right*: Detected curve using polynomial fitting.

which is an arising problem when mapping measurements in Euclidean space into a grid or voxel space. Line or curve models are (still) dominant when using the y-disparity space for ground-surface estimation [85], thus also (still) dominating current stixel calculations [43, 56, 84, 102].

Following [5], the number of missing stixels is used as an indicator for showing robustness when using the graph-cut approach. In this study we extended the idea of using the graph-cut approach by including one more camera (i.e. a trinocular set-up) utilising the confidence map derived from TED. Furthermore, the experimental evaluation reported in this study is more comprehensive than in [5] by also using LiDAR data and a number of statistical measures (more details later).

The computation of our disparity maps is based on the *Computer Vision System Toolbox* by calling a wrapped semi-global block matcher from the `OpenCV 3.1.0` library. In this section we report about the evaluation of detected stixels when applying one of those listed four ground-manifold models, and also when deciding either for binocular or trinocular recording, tested on $3,861$ frames. The evaluation is done using two widely-adopted datasets in the field, namely Daimler's 6D Vision Dataset [72, 89, 106] and the KITTI Vision Benchmark Suite [35, 110].

3.3.1 Different ground manifold models on 6D vision dataset

We evaluate the performance of stixel extraction for the following four ground manifold models: plane-fitting, line-fitting, polynomial-fitting, and graph-cut. The extracted stixels are verified on binocular stereo-image sequences downloaded from Daimler's 6D Vision website [72].

We applied the verification to all the twelve sequences which consist of $2,988$ 10-bit gray-scale stereo frames. The first six sequences are from the GOOD_WEATHER category, which present fairly good driving conditions with different illuminations, a variety of road views, shades, and colourings. The other six sequences from the BAD_WEATHER category present more challenging conditions such as rain drops, operating wind-shield wipers, and limited visibility as shown in Fig. 3.12.

Previous literature states that the evaluation of stixels is challenging due to the lack of preceding frames of annotated road images and stixel ground

truth [83]. In our work we compare extracted stixels with labelled frames provided by the dataset, and calculate a number of statistical measures. The *positive predictive value* (*PPV*), also known as *precision*, is calculated as

$$PPV = \frac{TP}{TP + FP} \tag{3.10}$$

where *TP* and *FP* denote the numbers of true positives and false positives, respectively. The *true positive rate* (*TPR*), also known as the *recall rate*, is defined as

$$TPR = \frac{TP}{P} \tag{3.11}$$

where $P = TP + FN$ is the number of positive pixels in the ground truth. We also calculated the *accuracy* (*ACC*) following

$$ACC = \frac{TP + TN}{TP + TN + FP + FN} \tag{3.12}$$

where *TN* and *FN* denote the numbers of true negative and false negative pixels.

For those true positive pixels, we further evaluate the deviation of the disparities of the corresponding stixels against the ground truth. The provided ground truth include the top-points, base-points, and disparity value. The root-mean-squares of the errors (RMSE) are also listed in Table 3.2. These results are tabulated in Table 3.1 and Table 3.2, with the best true positive rate and smaller RMSE rate in each sequence marked in bold. It is found that all the models show low positive predictive values, ranging from 0.12 to 0.53. Further investigation reveals that the reason is due to a lot more false-positive. In many cases, a detected stixel is not annotated in the test sequence. Although stixel ground truth was provided, they were annotated using a corridor[2] instead of the free-space, as it was observed during our experiments. An example is shown in Fig. 3.13.

We therefore use the *recall rate* (*TPR*) as the major index to evaluate the ground-manifold models.

The four tested models perform similar for the GOOD_WEATHER category. The best recall rate average is achieved for the graph-cut model, which

[2]The corridor is a subset of the free-space, and it denotes the region where the ego-vehicle is expected to drive in [48, 93, 105].

Table 3.1: Evaluation of stixel extraction (PPV and TPR) on Daimler's 6D-VISION dataset using various ground manifold modelling methods.

Sequence	Plane-fit		Line-fit		Poly-fit		Graph-cut	
	PPV	TPR	PPV	TPR	PPV	TPR	PPV	TPR
Seq. 1	0.44	**0.69**	0.42	0.63	0.41	0.63	0.40	0.64
Seq. 2	0.12	0.62	0.14	0.74	0.13	**0.74**	0.14	0.74
Seq. 3	0.47	0.74	0.50	0.74	0.49	0.68	0.50	**0.74**
Seq. 4	0.47	0.89	0.51	0.91	0.53	0.91	0.52	**0.91**
Seq. 5	0.22	0.94	0.23	**0.95**	0.23	0.89	0.23	0.92
Seq. 6	0.34	0.94	0.37	**0.95**	0.37	0.90	0.37	0.95
Average	0.34	0.80	0.36	0.82	0.36	0.79	0.36	**0.82**
Seq. 7	0.28	**0.47**	0.28	0.43	0.27	0.43	0.29	0.46
Seq. 8	0.23	0.80	0.24	0.81	0.25	0.82	0.26	**0.83**
Seq. 9	0.23	0.41	0.23	0.26	0.22	0.32	0.26	**0.44**
Seq. 10	0.26	0.76	0.25	0.65	0.28	0.78	0.28	**0.84**
Seq. 11	0.28	0.76	0.31	0.74	0.31	0.78	0.32	**0.81**
Seq. 12	0.27	0.58	0.25	0.34	0.28	0.50	0.29	**0.63**
Average	0.26	0.63	0.26	0.54	0.27	0.60	0.28	**0.67**

is just 2% better than the worst case - the plane-fit model. An overall accuracy around 0.86 is consistently found among all models, and the *RMSE* in disparities is between 2.18 to 2.28 pixels.

In the BAD_WEATHER category (as shown in Fig. 3.14 and Fig. 3.15), however, distinctive results are found. In five out of six tested sequences, the graph-cut model achieves the best recall rate, which is 30% better than the worst rates in some extreme cases (Sequences 10 and 12); the graph-cut model is here followed by the poly-fit, plane-fit, and line-fit models. As shown in Figure 3.14, the resulting ground manifold using graph-cut in y-disparity is robust enough with the existing of wind-shield wipers. In general it is observed that the ground manifold cannot be effectively modelled by the line-fit method due to severely corrupted disparity maps under bad weather conditions.

An overall accuracy of about 0.88 is consistently found among all the

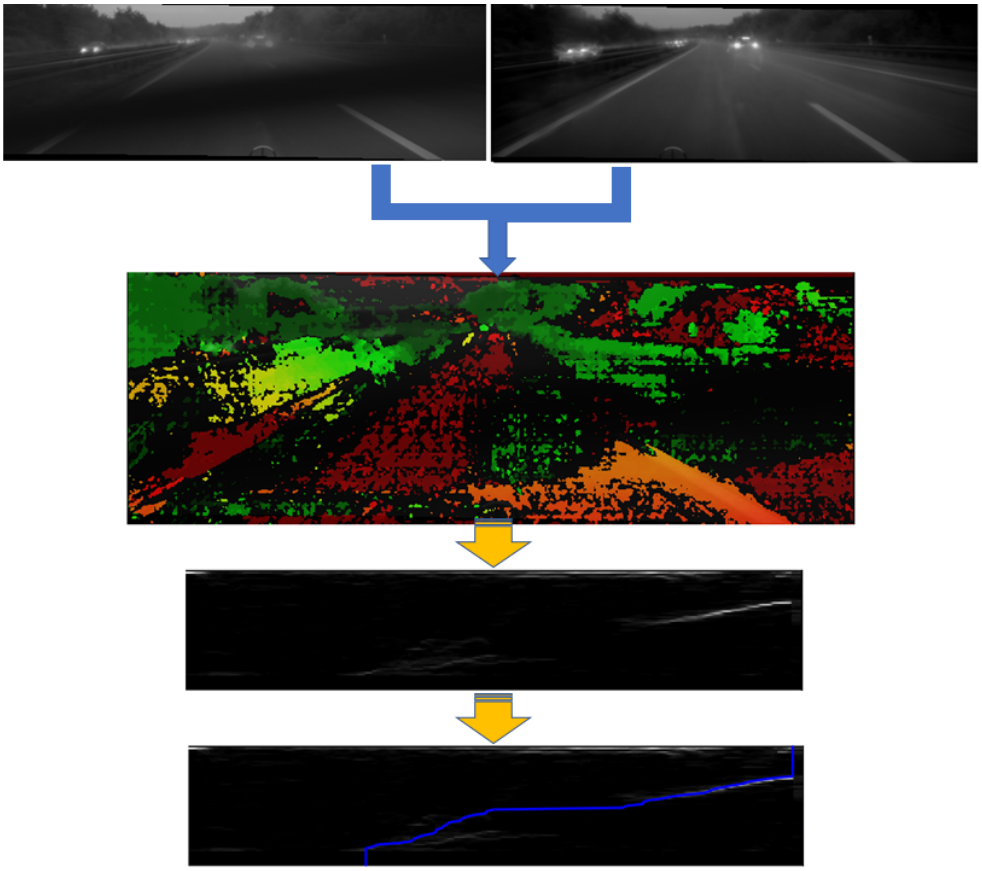

Figure 3.12: Demonstration of using graph-cut approach in challenging representation of y-disparity map. *Top*: stereo pair, and *middle*: noisy disparity map. *Bottom*: "complex" representation of y-disparity map and corresponding piecewise linear detected (blue-line) using our novel graph-cut approach.

models, and the *RMSE* in disparities is between 3.60 to 3.54 pixels.

We also profiled the run-time for each model and show the average processing time per frame in Table 3.3. The line-fit, poly-fit, and graph-cut models show similar computational time costs with a difference of not more than 5 milliseconds. The poly-fit yields the fastest approach for GOOD_WEATHER because it is insensitive to slope changes which widely exist in Sequence 1

Table 3.2: Evaluation of stixel extraction (ACC and RMSE) using various ground manifold modelling on the Daimler 6D-VISION dataset.

Sequence	Plane-fit		Line-fit		Poly-fit		Graph-cut	
	ACC	RMSE	ACC	RMSE	ACC	RMSE	ACC	RMSE
Seq. 1	0.92	1.66	0.92	1.52	0.92	1.51	0.92	**1.49**
Seq. 2	0.82	2.33	0.82	2.03	0.81	**1.99**	0.82	2.05
Seq. 3	0.82	**2.45**	0.83	2.46	0.83	2.69	0.83	2.53
Seq. 4	0.89	3.06	0.91	3.03	0.91	**3.02**	0.91	3.05
Seq. 5	0.80	2.20	0.80	**2.15**	0.81	2.40	0.81	2.18
Seq. 6	0.84	1.99	0.86	1.85	0.86	1.91	0.86	**1.85**
Average	0.85	2.28	0.86	**2.18**	0.86	2.25	0.86	2.19
Seq. 7	0.89	3.36	0.90	**3.36**	0.89	3.44	0.90	3.41
Seq. 8	0.87	4.12	0.88	3.93	0.89	4.02	0.89	**3.92**
Seq. 9	0.88	3.86	0.90	3.70	0.90	3.66	0.89	**3.58**
Seq. 10	0.81	2.90	0.82	2.91	0.82	2.82	0.82	**2.82**
Seq. 11	0.83	4.62	0.85	4.22	0.85	**4.19**	0.85	4.22
Seq. 12	0.91	3.62	0.92	3.54	0.92	3.53	0.91	**3.29**
Average	0.87	3.75	0.88	3.60	0.88	3.61	0.88	**3.54**

Table 3.3: Run-time profiling for stixel extraction using various ground-manifold models on the Daimler 6D-VISION dataset

Category	Plane-fit	Line-fit	Poly-fit	Graph-cut
GOOD_WEATHER	0.356 s	0.327 s	**0.326** s	0.332 s
BAD_WEATHER	0.452 s	**0.411** s	0.418 s	0.418 s

(see Fig. 3.13). The plane-fit model is found to be most time consuming due to the iterative RANSAC process over a large amount of 3D data.

In terms of efficiency and detection rate, the performance is measured by the number of false-positive stixel detections per frame. This is done by the following processes:

(1) Generate the stixel map which forms the stixel distances as shown in Fig. 3.15.

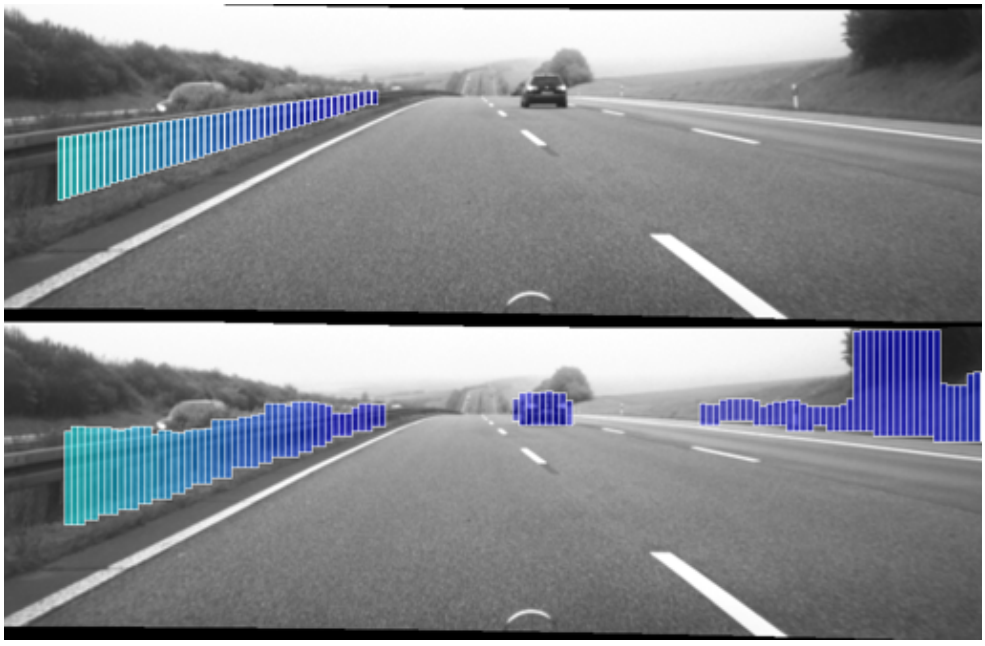

Figure 3.13: Annotated ground truth (top) and extracted stixels (bottom) of the first frame of Sequence 1 from the 6D Vision dataset. The ramp on the right and the car are not annotated by the ground truth but detected by stixel implementation (poly-fit).

(2) Given integer $w > 0$ as a predefined width of stixels, neighbouring w columns are grouped across the whole image, resulting in $\lfloor T_S = \frac{N_{\text{col}}}{w} \rfloor$ non-overlapping stixels in the whole frame. Hence, the comparison of the total number of stixels T_S and of the number of resulting stixels R_{S_i} in frame i form the error measurement using the root-mean-square error computed by:

$$E_{RMS}(n) = \sqrt{\frac{\sum_{i=1}^{n} \left(T_S - R_{S_i}\right)^2}{n}} \tag{3.13}$$

where n represents the number of frames.

Errors for the three methods are plotted in Figs. 3.16 and 3.17. Seq. 8 and Seq. 9 measurements show that the stixel estimation using graph-cut has a smaller number of missing stixels compared to the line fitting technique or

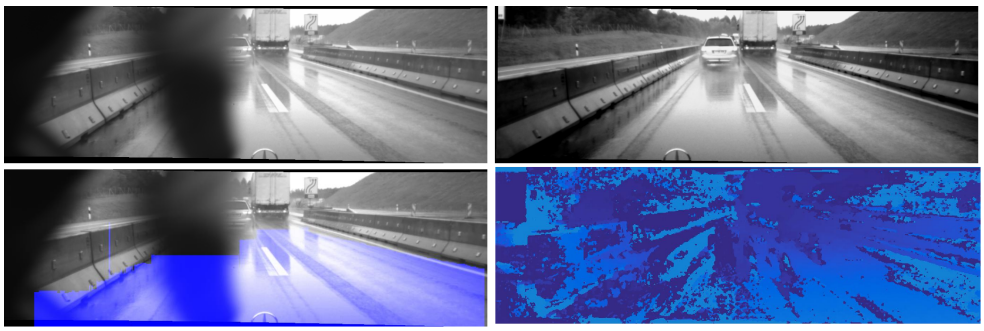

Figure 3.14: *Top-left*: Left image with window wiper. *Top-right*: Frame number 142 of Sequence 11 (bad weather) from the 6D Vision dataset. *Bottom-left*: The ground-manifold detection using binocular graph-cut. *Bottom-right*: Disparity map for this challenging scene.

the plane mapping. Figure 3.15, for example, shows the accuracy of the proposed method to detect the piecewise linear curve in the y-disparity image. Confirmed by the results obtained from Fig. 3.4 and by visual evaluation, the line fitting method "favours" the line on the right side in most cases and fails to detect the ground manifold. This scenario was repeated several times in Seq. 8. This identifies the line fitting method a being weak to resist bad weather situations. Furthermore, it has been found that the processing time for calculation free-space is significantly reduced when graph-cut is taken into account. In summary, experimental analysis illustrates improved robustness of the proposed graph-cut method across the chosen dataset. Challenging weather, or low-texture road surfaces are in particular cases where our method is more robust.

3.3.2 Comprehensive evaluation on KITTI dataset

We evaluate the quality of stixels not only for the selected four ground-manifold models, but also for binocular versus trinocular recording, using the trinocular data provided on the KITTI Vision Benchmark Suite [35]. Regarding previously stated challenges in evaluating stixels using the KITTI dataset [83] it is agreed that ground truth evaluation is a laborious work. Besides, the annotated ground truth provided by this dataset for road might be

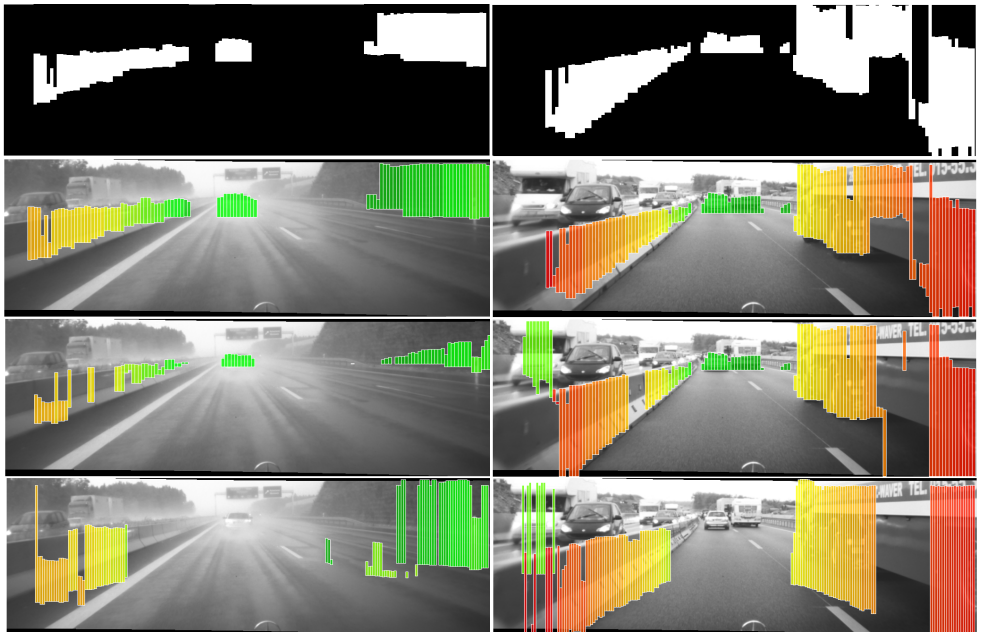

Figure 3.15: Qualitative results using the 6D vision `bad_weather` dataset *First row*: Stixel map. *Second row*: Stixels estimated using graph-cut. *Third row*: Stixels estimated using a line-fitting estimation. *Fourth row*: Stixels estimated using plane mapping.

unsuitable for stixel detection purposes. For example, as shown in Fig. 3.18, the corridor was annotated instead of road manifold [31]. There is a high possibility of false positive error for stixel detection if these labels (384 frames) is prepared for training, learning features, or performing classification purposes. Because the end of road detection area (i.e. base-points) indicates the beginning of stixels.

We address those by making use of the Velodyne high-definition 3D laser scanner data provided by the KITTI dataset. We use those range data as a ground-truth reference to evaluate the distance values assigned to the extracted stixels. This comprises of several processes:

1. Generate a disparity map from extracted stixels. The map contains valid disparities only for pixels belonging to a stixel. The map is then

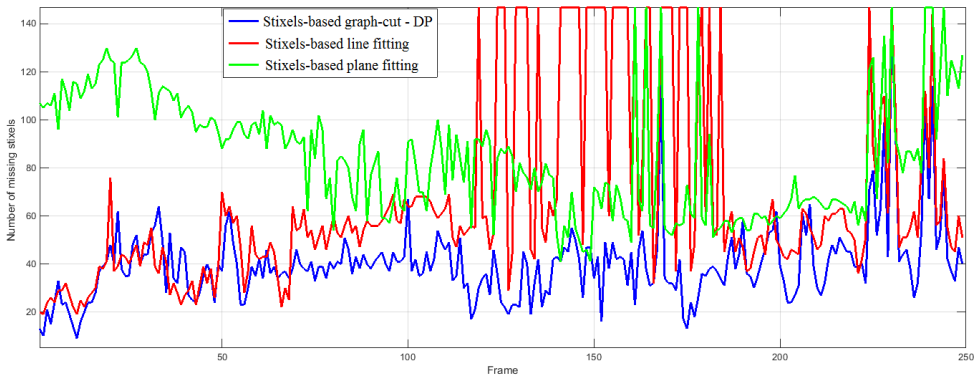

Figure 3.16: Error rates illustrate the number of missing stixels using the 6D vision `bad_weather` dataset -Seq. 8.

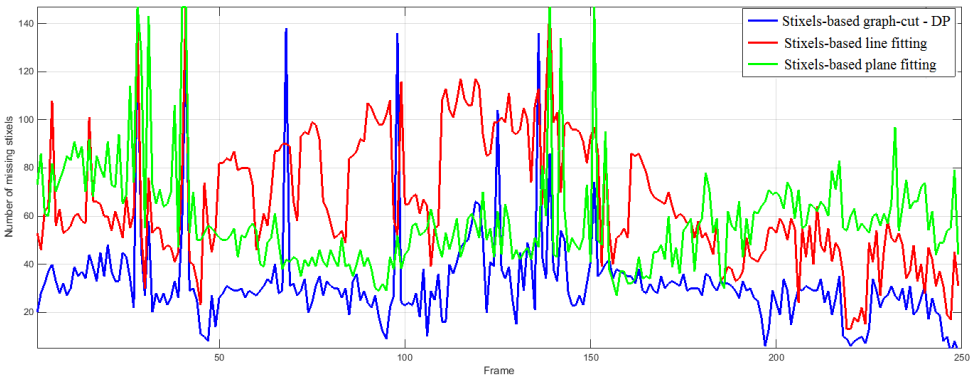

Figure 3.17: The error rates illustrate the number of missing stixels using the 6D vision `bad_weather` dataset - Seq. 9.

converted into a depth map following Eq. (2.1).

2. Project LiDAR points into image coordinates. Figure 3.19 shows some example LiDAR point projections. The projections associate a subset of LiDAR range data to the extracted stixels.

3. For each associated LiDAR point, its depth is compared with the stixel depth map. The signed difference is then used to evaluate the perfor-

mance of the stixel extraction process.

4. As the extracted stixels are in rectangular shape with reduced spatial resolution, it is often found close to edges of a stixel that background LiDAR points are wrongly assigned to a stixel. To exclude such outliers from the evaluation, we ensure a zero-mean for the error distribution of each model. Then, we discard LiDAR points that are outside the interval $[-0.5\sigma, +0.5\sigma]$ of all the range data associated to the same stixel before calculating the mode (note: not the mean) and the standard deviation.

We selected 873 trinocular stereo frames from the ROAD, RESIDENTIAL, and CITY categories, which include cars, cyclists, pedestrians, trees, and traffic signals. The test sequences are listed in Table 3.4, also called for short A , B, and C in the following tables. Qualitative results are listed in Table 3.5 using

Table 3.4: Selected test sequences from the KITTI dataset

Category	Tag	Sequence	Frames
ROAD	A	`2011_09_26_drive_0032`	390
RESIDENTIAL	B	`2011_09_26_drive_0035`	137
CITY	C	`2011_09_26_drive_0091`	346

a binocular configuration. We also use frames captured by the third camera to conduct additional tests on binocular versus trinocular stixels Table 3.6.

Figure 3.18: Ground truth annotation for ROAD in KITTI.

Bold numbers indicate the best case per group, and coloured numbers are the best case over all the seven models. Note that the plane-fit model is not of relevance here. As illustrated, a negative value means that laser points are in front of the stixels. Instead of using mean, we have used mode as an indicator to interpret signed distance error. Furthermore, as there are many non-flat objects present in the scene, and many background points are covered by the extracted stixels, we expect to see large standard deviation values.

For the ROAD sequence, the trinocular line-fit model achieves the lowest rate of a LiDAR-stixel error of -5.3cm, which is 55.5% better than the worst case yielded by the plane-fit model -11.6 cm. The main reason for this achievement is due to open-road scenarios which normally correspond closely to a straight-line in y-disparity space supplemented by the confidence measure using TED. This is slightly different compared to trinocular poly and graph-cut which achieve -6.5 and -6.2cm respectively.

In the RESIDENTIAL sequence, the used data show cars parked on the side of the road, houses, and road junctions. Based on the experiments, more obstacles (impacting the y-disparity map) make identifying a curve (using line-fitting or poly-fitting) more complicated. For this sequence, the trinocular graph-cut model has superior performance with a lowest mean LiDAR-

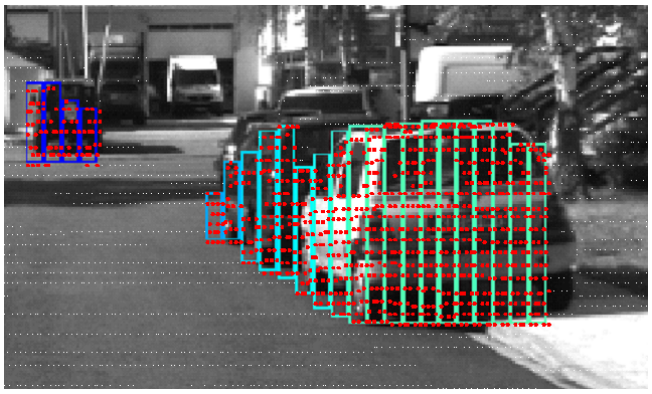

Figure 3.19: Extracted stixels (colour-coded by depth) and LiDAR points marked by white and red dots. Points hitting any extracted stixel are shown in red and used to evaluate the accuracy of the extraction process.

Table 3.5: LiDAR-based qualitative evaluation [cm] of ground manifold modelling using KITTI dataset (binocular configuration).

Sequence	Binocular stereo							
	Plane-fit		Line-fit		Poly-fit		Graph-cut	
	Mode	Std.dev.	Mode	Std. dev.	Mode	Std. dev.	Mode	Std. dev.
A	-11.6	49.9	**-6.6**	54.9	-10.2	53.0	-9.5	54.2
B	-14.4	54.1	-11.8	53.5	-12.5	54.1	**-10.9**	52.0
C	-5.1	47.4	-4.0	48.7	-3.5	50.2	-3.8	50.5

Table 3.6: LiDAR-based qualitative evaluation [cm] of ground manifold modelling using KITTI dataset (Trinocular configuration).

Sequence	Trinocular stereo					
	Line-fit		Poly-fit		Graph-cut	
	Mode	Std.dev.	Mode	Std. dev.	Mode	Std. dev.
A	-5.3	56.5	-6.5	**55.5**	-6.2	55.7
B	-12.9	53.8	13.0	53.7	-10.2	**52.9**
C	-4.2	**48.5**	**-3.8**	49.6	-3.9	50.1

stixel error of -10.2 cm. The disparity map relatively suffers from low-depth in this dataset due to lighting conditions accompanied with many pedestrians and buildings in the scenes. The performance of graph-cut is better suited for cases where there are irregular changes in a piecewise linear curve.

On the other hand, the binocular poly-fitting model provides the lowest mean LiDAR-stixel error of -3.5 cm for the CITY sequence as there are a number of non-flat objects in this sequence. This defines only a slight difference compared to the other techniques.

In addition to the statistics for the LiDAR-stixel error, we also calculate the improvement by the use of the third camera applying TED-weighted y-disparities (see Section 3.2) as input for ground-manifold modelling.

As illustrated in Table 3.5, the trinocular graph-cut approach covers more valid disparities compared to others, and appears to be insensitive to weather changes. It outperforms the trinocular polynomial or line-fit methods re-

Table 3.7: Improvement rate with trinocular ground manifold modelling using KITTI dataset

Sequence	Line-fit		Poly-fit		Graph-cut	
	Mode	Improve	Mode	Improve	Mode	Improve
A	-5.3	19.7%	-6.5	36.3%	-6.2	34.8%
B	-12.9	-10.2%	-13.0	-4.0%	-10.2	6.4%
C	-4.2	-5.0%	-3.8	-8.6%	-3.9	-2.6%

Table 3.8: Average number of stixels extracted per frame in the tested KITTI sequences

Sequence	Binocular stereo				Trinocular stereo		
	Plane-fit	Line-fit	Poly-fit	Graph-cut	Line-fit	Poly-fit	Graph-cut
A	32.6	32.2	33.8	34.2	35.0	**35.3**	34.8
B	**69.1**	27.3	24.3	29.1	29.7	26.7	28.7
C	66.9	71.0	69.5	70.7	**71.7**	71.0	70.6

garding robustness. The improvement rate is obvious for the ROAD and RESIDENTIAL sequences when using the graph-cut model. We notice that using trinocular cameras, the performance of poly-fit and line-fit decreases for RESIDENTIAL. This occurs because disparity values fluctuate roughly at the end of the data sequence (Frame 100 and onwards) because of having a roundabout in the shown scenes. There are some values missing between $D01$ and $D12$ and this is reflected in values $\Gamma(x, y)$ since they are derived from these maps. The graph-cut model pays more attention to the disparity values, and using a penalisation scheme is thus still able to recover the most relevant values compared to the ground manifold. The graph-cut model yields the highest improvement for ROAD and RESIDENTIAL with the trinocular configuration, and it still has promising results. This shows that, with such an extension, we can have a robust ground-manifold detection, resulting in accurate stixel estimation.

Finally, we summarise in Table 3.8 the average number of stixels extracted per frame using binocular and trinocular vision-based ground man-

ifold models. As shown, the binocular plane-fit performs best on the RESIDENTIAL sequence with an average of 69.1% stixels detected. On the ROAD sequence, the trinocular polynomial-fit method yields the best result with an average of 35.3% stixels detected. The line-fit model achieved the best result on the CITY sequence with an average of 71.7% stixels detected per frame.

3.4 Summary

This chapter presented an in-depth analysis for binocular and trinocular vision-based stixel calculations using four ground-manifold models across two challenging datasets. For a comprehensive comparison, we provided an insight into the accuracy of extracted stixels on long-run sequences (for a total of $3,861$ frames); we also provided a brief run-time profiling to illustrate the performance of these models. The main objective of the reported research was to present an analysis on adopting a low-cost architecture (ground-manifold estimation method) for reducing false-positives in stixel estimations. Also, we extended the graph-cut model for a trinocular configuration which yields obvious and robust improvements compared to other models.

In our analysis we covered the number of cameras required and the road profile for obtaining accurate stixels. Experiments show for the binocular case that the graph-cut model (using dynamic programming) presents a promising technique to ensure accuracy of stixels for 6D vision and KITTI datasets. The number of true-positives is large when the graph-cut model is used as a minimisation method for calculating a y-disparity cut; see results for the 6D vision dataset for the GOOD_WEATHER as well as the BAD_WEATHER categories. As illustrated, the polynomial-fit model shows the fastest run-time for GOOD_WEATHER, while the line-fit model achieves the fastest run-time for BAD_WEATHER.

In order to evaluate the effects for the KITTI dataset, a comprehensive study was conducted not only for comparing ground-manifold models but also bi- versus trinocular recording. Results show that the number of generated stixels highly increases when using trinocular line fitting for ROAD sequences, and binocular poly-fitting for CITY sequences; finally, trinocular graph-cut proved to be the best alternative on RESIDENTIAL sequences.

Having especially challenging scenes in mind, altogether we recommend the trinocular graph-cut approach.

Chapter 4

Stix-Fusion: Data Fusion Towards Efficiency

This chapter informs about two methods to improve stixel accuracy. The first one is a stereo-based method towards robust identification of obstacle heights using stixels. The second method is a monocular-based approach but guided by LiDAR sensors to enhance obstacle representation using stixels. The aim is to ensure a very low false detection rate for obstacle detection in video data recorded in vehicles. Basically, the given frames are either segmented into ground manifold or obstacles (approximated by stixels). To robustly estimate the obstacle height, our methods improve the height of stixels by fusing colour information, represented by a saliency map, with disparity information, represented by a membership map. We also show more efficient and robust 3D point representations, even if only integrating monocular vision into the LiDAR-based approach for generating monocular stixels. - Parts of this chapter is published in [4, 2].

4.1 Height segmentation improvement

The main idea is to reduce background-versus-foreground errors by distinguishing the foreground from the background more efficiently, for reducing false-positive stixels per image.

4.1.1 Saliency map calculation

For identifying salient objects, a saliency map S_m for input image I is established. Following [9], a scalar image I is subjected to a Gaussian filter G_σ at the first stage for removing fine texture details as well as artefacts or noise (σ needs to be chosen according to given data); I_μ represents the mean in a

small neighbourhood:

$$S_m(x,y) =\mid I_\mu(x,y) - I \star G_\sigma(x,y) \mid \tag{4.1}$$

In the case of colour images $\mathbf{I}$, we choose to apply G_σ in the *Lab* colour space for individual L, a, and b channels. The final saliency map is then computed by combining channel-specific saliency maps using the pixel-wise Euclidean norm as follows:

$$S_m(x,y) = \|\mathbf{I}_\mu(x,y) - \mathbf{I} \star \mathbf{G}_\sigma(x,y)\|_2 \tag{4.2}$$

where $\mathbf{I}_\mu$ is the mean image feature vector, and $\|.\|_2$ is the L_2 norm. For the sake of simplicity, we consider the Otsu binarization algorithm for the final step of region segmentation. See Fig. 4.1, second row, for an example. An Illustration of proposed fusion toward stixel height segmentation is provided in Fig. 4.2.

4.1.2 Membership vote calculation

The height of an obstacle, which "sits" on the ground manifold, is obtained by seeking an ideal separation between foreground and background disparities. We briefly recall the method as defined in [11].

The estimation of top-points t_1, t_2, ..., $t_{N_{\text{cols}}}$ of stixels, for row y, begins with selecting membership votes for pixels ("Does a pixel belongs to background or foreground?"). Afterwards, a *benefit image* is produced and used for an approximate calculation of those top-points [73].

The membership values rely on the selection of a disparity in a column with respect to the other disparities in this column for defining, or not defining a foreground obstacle. The membership value is positive if it does not exceed the maximum distance of the expected obstacle disparity; otherwise negative. [11] suggests the following exponential membership function:

$$M_d(x,y) = 2^{1-\varepsilon(x,y)} - 1 \tag{4.3}$$

where

$$\varepsilon(x,y) = \left[\frac{D(x,y) - d_x}{\Delta D_x}\right]^2 \tag{4.4}$$

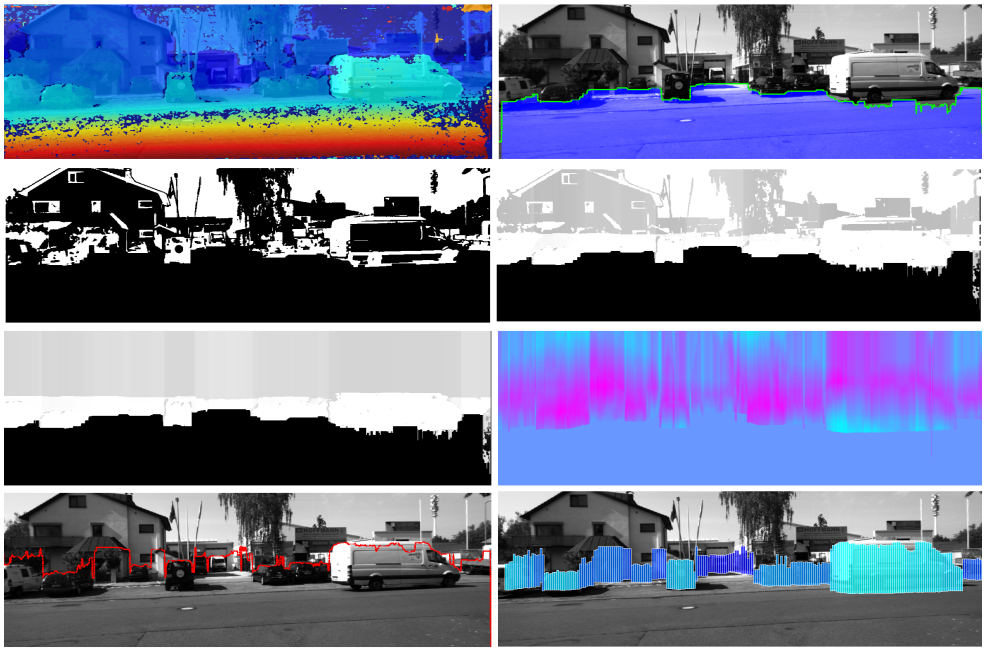

Fig. 4.1: Stixel world for an example of the KITTI `residential` dataset. *1st row, left*: Disparity map. *1st row, right*: Calculated area of ground manifold. *2nd row, left*: Salient region segmentation. *2nd row, right*: Base-points, visualised for salient regions. *3rd row, left*: Proposed fused membership; white for foreground and gray for background. *3rd row, right*: Benefit image. *4th row, left*: Height segmentation. Red dots are used for detected top points. *4th row, right*: Extracted stixels of 'substantial' height above ground manifold.

where d_x is the disparity at base point b_x (i.e. the intersection point with ground manifold in column x), and ΔD_x is a computed parameter which determines the difference between the disparity obtained from the ground manifold vector and the disparity corresponding to the depth value (see Fig. 4.3). We improve height estimation by extending the disparity membership of [73] by introducing fused membership votes as follows:

$$M_f(x, y) = M_d(x, y) + \lambda \cdot S_m(x, y) \tag{4.5}$$

Values of $\lambda \geq 0$ do have impacts on results, so we use $\lambda = 1$.

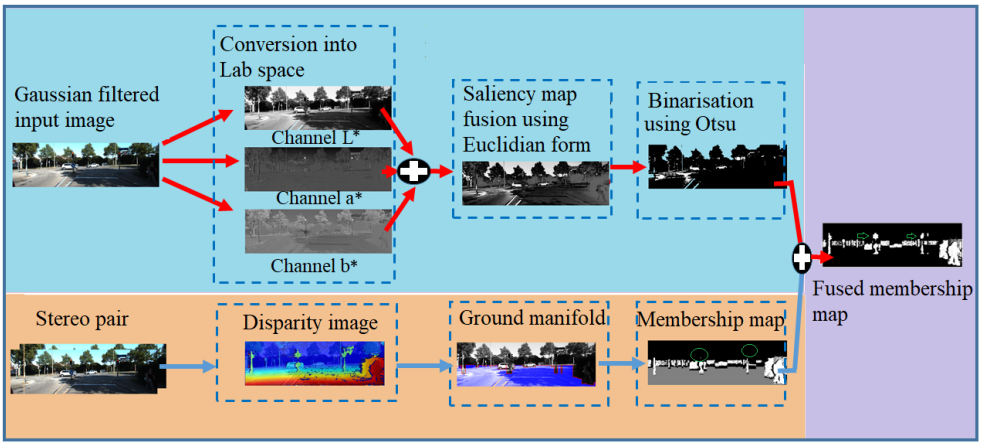

Figure 4.2: Illustration of proposed fusion towards more accurate membership votes. Green circles show missing height of object, while the green arrows depict corrected height of stixels due to fused membership votes.

Figure 4.3: Illustration of membership map.

4.1.3 Benefit image calculation

Following the base-line method, (now modified) membership values generate the *benefit image*:

$$C_{x,y} = \Sigma_{i=y}^{b_x} M_f(x,i) - \Sigma_{i=1}^{y-1} M_f(x,i) \tag{4.6}$$

(Note that Row 1 is the topmost row in the image.) The given equation indicates that when C_{max} occurs, there is a high likelihood for a *height cut* (i.e. top-point t_x identified) via foreground and background separation (column-wise maximum). This height cut is performed for each column x at row y_{max}

Figure 4.4: Benefit image, where bright pixels present perfect cut between background and foreground.

where the maximum value occurs. Each column supposed be reach at row y_{max} when positive membership votes lie below (foreground) and the most negative membership values lie above y_{max}.

In other words, the benefit image is used for calculating top-pixels (x, y_{max}) which maximise the benefit. See Fig. 4.1, 3rd row, right. The colour encoding indicates the height costs where pink colour indicate high possibility of segmentation to separate foreground from background.

4.2 Monocular single stixels: LiDAR guided

Incorporating remote sensing (e.g., LiDAR) adds benefits to autonomous cars if it provides depth information at high accuracy. A high market growth is expected for LiDAR technologies for the next few years. Firms already advertise low-cost LiDAR sensors [108, 109]. Various approaches for stixel estimation have been investigated by mainly involving bi- or trinocular vision, since depth can be obtained from stereo cameras at low cost. A failure of disparity estimation on obstacle or low-textured road surfaces still causes concerns [95]. Unstable results caused by challenging imaging conditions (represented by illumination, colour, or texture) may be resolved by also using sensors (such as LiDAR) which are reliable under such conditions. As a result, this may lead to improved disparity maps. Yet, LiDAR points are sparse and there must be an optimised interpolation approach that would support us in our endeavour to obtain a dense depth map (see Fig. 4.5), and

later a dense stixel representation. This research proposes monocular stixels guided by LiDAR data for verified stixel positions.

The proposed method aims at including LiDAR data in the estimation of monocular stixels. To do this, we first extract a disparity map from a distance map assuming a hypothetical second camera at an assumed base distance b (see also Fig. 4.6). Then we do the following:

- Project 3D LiDAR points into the 2D image plane (point projection). The results improved by discarding points outside the camera plane (noisy points) and the remaining points are sorted according to their position in pixel units, in order to speed up the search process.
- Construct a dense distance map from sparse LiDAR points using colour and texture information.
- Convert the distance map into a disparity map based on the camera matrix (we simply use focal length f and base distance b as reported for the used KITTI data.

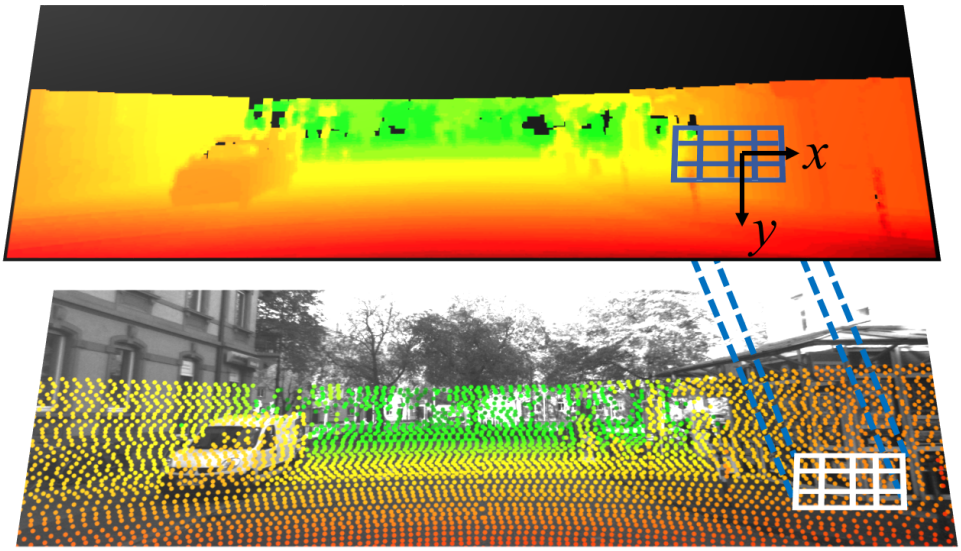

Figure 4.5: *Top*: Dense disparity map (supporting a dense depth map). *Bottom*: Sparse 3D points measured by a LiDAR sensor.

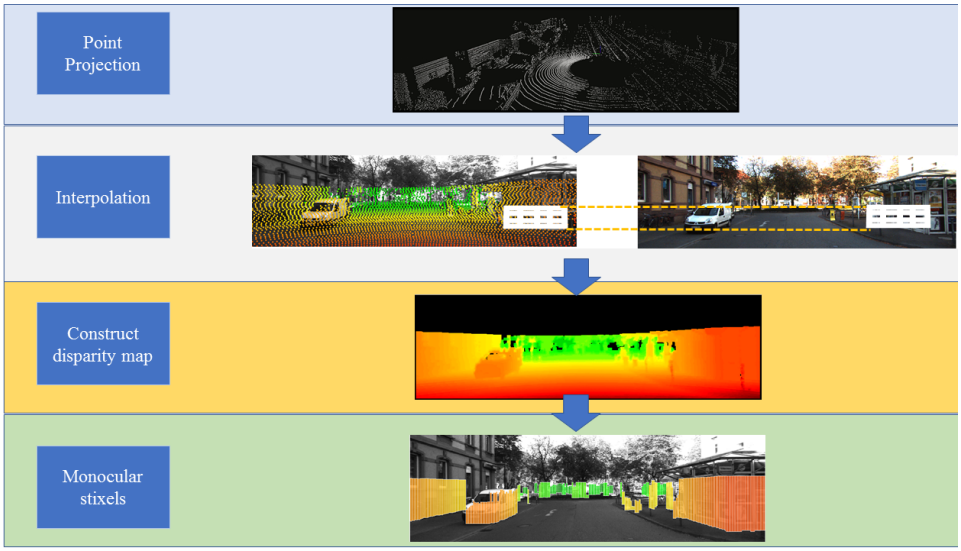

Figure 4.6: Illustration of the proposed steps using monocular guided LiDAR.

- Construct stixels based on this "monocular disparity" map following the common procedure as for binocular vision.

4.2.1 Point-projection phase

The provided calibrated data (images and LiDAR points) in the KITTI dataset are the input used to obtain the dense map. This study uses the spatial relationship between 3D points projected into the image plane to construct a dense map. As described by [35], the Velodyne HDL-64E S2 is employed in the KITTI dataset which has 0.09 degree angular resolution and 2 cm distance accuracy. It is efficient and able to collect around 1.3 million points/second. Scans are stored as floating points with $[x, y, z]$ coordinates in which x, y, z represent forward, to the left, and upward directions, respectively, using:

$$\mathbf{K}_{\text{velo}}^{\text{cam}} = [\mathbf{R}_{\text{velo}}^{\text{cam}} | \mathbf{t}_{\text{velo}}^{\text{cam}}] \tag{4.7}$$

where $\mathbf{R}_{\text{velo}}^{\text{cam}} \in \mathbb{R}^{3\times3}$ is the rotation matrix, and $\mathbf{t}_{\text{velo}}^{\text{cam}} \in \mathbb{R}^{3\times1}$ is the translation vector, in both cases of a Velodyne sensor into the camera pose.

Detailed information regarding LiDAR and camera calibration, data alignment, and the calibration matrices can be found in [35], and intrinsic and extrinsic parameters are given in [34]. A 3D point in LiDAR coordinates P_r = $[x, y, z, 1]^\top$ is projected into a point in camera coordinates P_s = $[x, y, z, 1]^\top$, based on:

$$P_s = \mathbf{K}_{\text{velo}}^{\text{cam}} P_r \tag{4.8}$$

Every point P_s is then rectified to match the image plane using a rectification matrix $\mathbf{K}_{\text{rec}}$:

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \mathbf{K}_{\text{rec}} P_s \tag{4.9}$$

Considering the projected LiDAR points in pixel coordinates (x, y), as given by (4.9), some operations are performed prior to the interpolation stage (described in the following section). The points outside the camera plane are discarded, and the remaining points are sorted according to their position in pixel units, in order to speed-up the search process. Finally, the points are rearranged into a new space that combines the coordinates in pixel units (x, y) and the range r, such that a point P is represented by P = $[x, y, r]^\top$.

4.2.2 Interpolation phase

The point clouds, provided by LiDAR, are sparse and in some cases noisy, and thus an interpolation method is required to derive a smooth (filtered) "dense" distance map. The interpolation process carried out is a combination of methods proposed by [24, 79] which both focused on colour and texture information. Basically, these methods present a solution to sparse data by merging 3D points with information from RGB images. The assumption is based on the idea that pixels in a connected region, having similar texture in the camera image in their neighbourhood, will have identical depth values.

Furthermore, [79] generates a [0-255] normalised depth map image from LiDAR data. This situation does not work for us since the stixel calculation requires a real depth, not a [0-255] normalised depth map. Points $P = [x, y, r]^\top$ represent a calibrated set of 3D sparse LiDAR points projected into

the image plane, as described in the previous phase. In order to derive the distance (or range) map R at a given position (x, y), we can calculate this map by a weighted fusion of the range values r_k of the sparse points P in a window W_p centred at position $p = (x, y)$, as follows:

$$R(p) = \sum_{k \in W_p} \omega_k \cdot r_k \tag{4.10}$$

The window W_p is of size 11×11 in our experiments.

Even for the fixed-size window, the number k of points in W_p varies and depends on the 3D-cloud's sparsity. A similar mechanism is applied to a bilateral filter when interpolating low-resolution images; each weight ω_k is computed by two factors:

- a pixel distance function $d_2(p, q)$ (here: assumed to be the Euclidean distance in pixel units) between the window's central point $p = (x, y)$ and the considered k points $q = (i, j)$ within the window W_p, and
- a confidence weighting term $\kappa(r)$ which is determined as a function of the measured distance r. In some cases (e.g., uncertainty in sensor data), $\kappa(r)$ decreases linearly corresponding to the range value, penalising 3D points in direct relation to their distance from the LiDAR. The $\kappa(r)$ values are normalised by the maximum range value r_k in W_p; see [24, 79].

Hence, the 2D spatial neighbourhood filter is re-written as:

$$R(p) = \sum_{q \in W_p} d_2(p, q) \cdot \kappa(r_q) \cdot r_q \tag{4.11}$$

From the distance map, the calculation of the disparity map is as follows:

$$D(p) = f \cdot \frac{b}{R(p)} \tag{4.12}$$

where f is the focal length and b is the (assumed) camera baseline; here we use $b = 0.54$ m as in the KITTI data.

To construct a single layer stixel, we adopt the process outlined in [1, 11] but use the disparity map D that is derived from monocular vision (see

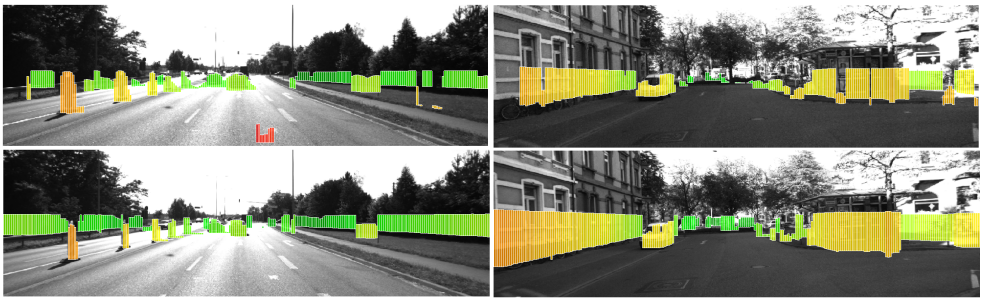

Figure 4.7: Single-layer stixel estimation. *Top row*: Estimation using a disparity map resulting from stereo-matching (binocular). *Bottom row*: Estimation using a depth map incorporating a LiDAR sensor (monocular).

Fig. 4.7. The disparity map D would be more accurate since we are fusing multiple sensors' information. As detailed in [5], an optimisation approach was proposed to minimise the cost of a cut in y-disparity space to identify a piecewise linear curve. Following a discrete formulation, the curve fitting process is essentially a graph-cut problem, which aims at finding a set of quantised disparities $\mathbf{d} = \{d_1, d_2, \ldots, d_{N_{\text{col}}}\}$ that minimises a cost function subject to smoothness constraints. Such a cut $\mathbf{d}$ divides the y-disparity map (row-wise) into left and right parts. To find the lower bound of the road manifold, the cost function (i.e., error or energy E) can be defined by using a first-order derivative V_y of the y-disparity map V (i.e., along row y) [5]:

$$E(\mathbf{d}) = \sum_{i=1}^{N_{\text{col}}} V_y(y, d_i) + \gamma \sum_{i=2}^{N_{\text{col}}} \Theta(d_{i-1}, d_i) \tag{4.13}$$

where $\gamma \geq 0$ defines a penalty for Θ, the smoothness function, and $V(y, d)$ represents the number of pixels sharing the same disparity of d in the y-th row of the disparity map D derived from LiDAR in 4.12. The value of γ depends on the scale of the data term. To ensure the monotonicity of a cut, the smoothness term can be specified by an asymmetric L_1 Potts model (more details in [5]).

4.3 Experimental results

The accuracy of the proposed methods reported in this chapter (saliency map fusion and monocular plus LiDAR fusion) will be assessed against base-line stixel reported in [11]. The ground model used for the two proposed methods in this chapter is based on graph-cut dynamic programming which is proposed and reported in the previous chapter and also reported in [7]. The main reason behind this selection is due to successful and significant results achieved in various datasets.

The datasets included for the evaluation in this chapter is KITTI dataset and relevant information provided in Table 4.1. The KITTI dataset [35], that includes diverse collections of traffic scenes, was used for the experiment. Totally 1,231 stereo images from the *residential*, and *road* datasets which show cars, pedestrians, trees, and traffic signals were selected. We aimed at having a wide diversity of challenging traffic situations including different lighting conditions, different road views, shades, and colourings. A SGM stereo matcher was adopted for disparity calculation [45], without any further pre- or post-processing of disparity values. SGM matchers prove to be dominated in computer-vision-based driver assistance systems, and this is because of its robust capabilities to perform 3D reconstruction [73].

For evaluation purposes, stixel-LiDAR depth was used as ground truth, as suggested in [7]. All stixels, in every frame, are evaluated individually. This comprises of several processes as discussed in [7]. We address current obstacles to evaluate extracted stixels in KITTI dataset which include the lack of preceding frames of annotated road images, or lack of stixel ground truth. Besides, it is hard to evaluate the quality of the 3D reconstruction based on manually observed disparity images. Furthermore, the accuracy needs to

Table 4.1: Selected test sequences from the KITTI dataset

Category	Sequence	Tag	Frames
ROAD (OPEN-ROAD)	2011_09_26_drive_0015	A	297
RESIDENTIAL (URBAN)	2011_09_26_drive_0035	B	131
RESIDENTIAL (SUBURB)	2011_09_26_drive_0036	C	803

be evaluated and verified in terms of reference robust sensors to ensure the safety and quality demand by driver assistance systems.

Hence, our objective is to automate this process to enable a more accurate verification and validation on a large dataset covering diverse traffic scene sequences.

Since 3D laser scanners are accurate as reference sensors, we utilise the Velodyne LiDAR data obtained by a 3D laser scanner which are publicly available [35]. All stixels in every frame are evaluated individually. To find the efficiency of our proposed stixels fusion methods in terms of distance error, we compare the performance of Velodyne LiDAR data with base-line, saliency map fusion and monocular plus LiDAR fusion stixels. This comprises of several processes:

1. Generate a stixel map which forms the stixel distances as shown in Fig. 4.8.

2. Project LiDAR points (X_j, Y_j, Z_j) into image coordinates (u_j, v_j). Such an example LiDAR point projection is illustrated in Fig. 4.8. The projections are used to build a LiDAR-stixel correspondence function β_{ij}, where $\beta_{ij} = 1$ if LiDAR point j hits stixel i, otherwise $\beta_{ij} = 0$.

3. The degree of correspondence of these images verifies the accuracy of the estimated stixels. Hence, the comparison of LiDAR depths with corresponding stixel depths form the error measurement using the root-mean-square computed by

$$\text{RMSE} = \sqrt{\frac{\sum_{i=1}^{N_{\text{stx}}} \sum_{j=1}^{N_{\text{pts}}} \beta_{ij} (z_i - Z_j)^2}{N_{\text{hit}}}} \tag{4.14}$$

where N_{stx} and N_{pts} represent the number of stixels and LiDAR points, respectively, and N_{hit} is the number of non-zero elements in β.

It is worth to mention that stixel map generated in monocular plus LiDAR approach is formed based on the interpolated LiDAR points. When we obtained the dense "monocular disparity" it is convolved from sparse points and information from RGB images. Hence, this convolved disparity map (or so-called monocular plus LiDAR) is used to build stixel map and for

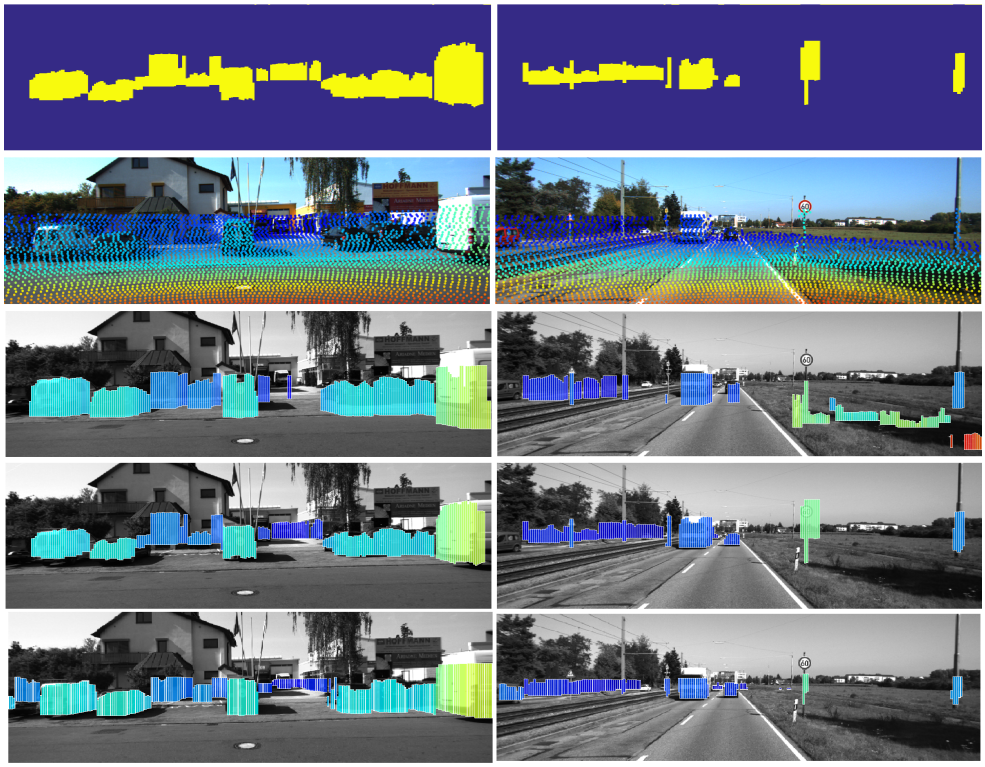

Figure 4.8: Qualitative results using KITTI `residential` (first column) or `road` (second column) data. *1st row*: Stixel map. *2nd row*: LiDAR projections. *3rd row*: Stixels estimated using base-line method. *4th row*: Stixels estimated using proposed saliency fusion. *5th row*: Stixels estimated using proposed LiDAR and monocular.

evaluation process we do not compare LiDAR against LiDAR. Instead, we obtain the stixel map from the interpolated LiDAR points and we compare it against the raw sparse points.

Mean distance differences are summarised in Table 4.2 and run-time profiling is also provided in Table 4.3. Error rates are plotted in Fig. 4.9, Fig. 4.10, and Fig. 4.11 for road and residential data for the three methods across 1,231 frames. As shown in these figures the error rate was significantly reduced using the two proposed methods compared to base-line.

By visual evaluation, the base-line method has some limitations on identify foreground objects independently in the membership map. This was reflected on the speed sign pole and the van height shown in Figure 4.8. This problem occurred several times in both tested datasets with different type of objects. This shows the resulting stixels, using fused membership votes, are more accurate than for the original membership votes. The height was clearly improved by the fused membership using saliency map (see Fig. 4.8). On the other hand, the monocular plus LiDAR was providing a good advantage for a long-run sequences and low false-positive rate using the data provided in KITTI dataset.

Data provided for road and residential are different in terms of road structure. In open-road we noticed there are a plenty of moving vehicles with very limited number of pedestrians. While in residential itself also differs since category C (suburb) provides long-sequence so we expect to see some structure that represent outside of city. The urban was recorded for a small sequences and we noticed also there are a number of pedestrians available in this category.

As we can observe from Fig. 4.10 the saliency fusion was providing accurate results for residential (urban) from frame 1-100. It outperforms the other two methods (base-line and monocular plus LiDAR). Still the base-

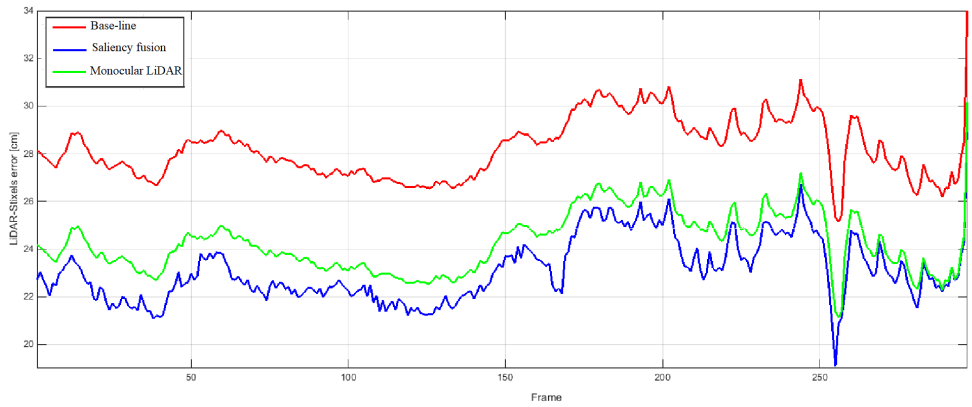

Figure 4.9: Error rates representing differences of distances between LiDAR data and stixels for road data - open road (A).

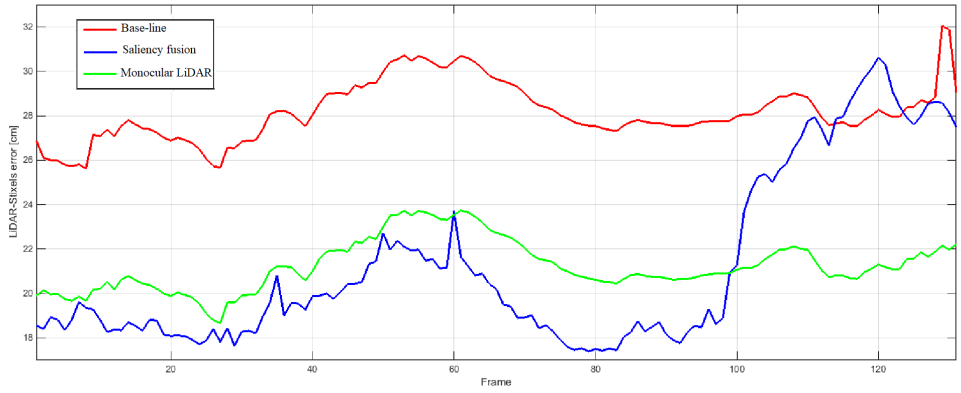

Figure 4.10: Error rates representing differences of distances between LiDAR data and stixels for residential data - urban (B).

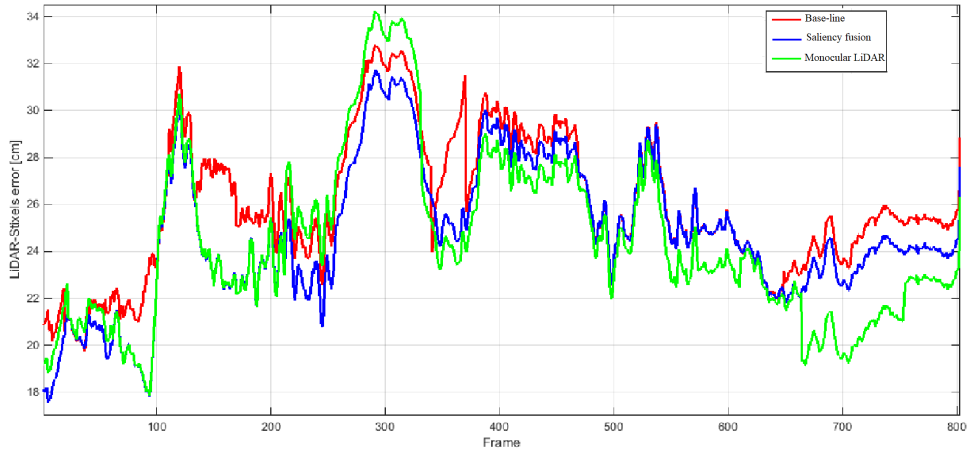

Figure 4.11: Error rates representing differences of distances between LiDAR data and stixels for residential data - suburb (C).

line method shows high false-positive stixel detection and also low-accuracy rate. This returns to many factors including the road surface which do not fit to obstacles in terms of distances. It is very obvious that in category B the saliency map improved around 18.07 % compared to the base-line method. The mean error rate between stixel map and LiDAR for saliency map is 23.12

cm, still achieving good result although at the end of the frame the error rate was increasing. The main problem as investigated and looking into the frame 100-131 (end of category B) it is obvious that shiny reflection on road accompanied with road junction which exist at the end of the sequence plays a major role because there was a number of missing stixels. Still with these circumstances the saliency fused membership map is showing improvements compared to base-line method. A feature tracking could have added an advantage for saliency map. However, the monocular plus LiDAR still having a low error rates for frame sequences mentioned above.

The pattern for open-road scenario (category A) looks similar with an improvement has been made using saliency map. The saliency map shows a successful identification of stixel height as provided in Fig. 4.8 (second column). The improvement rate recorded for saliency map is 24.6% and for monocular plus LiDAR is 19.63% compared to base-line stixel method (see Fig. 4.9). As observed the LiDAR up-sampling method had to match pixels with sufficient LiDAR points in some situation when there is a shadow extra processing is required (i.e. increasing the window size W_p for up-sampling). On the other hand, there is very limited false-alarm stixel detected using monocular plus LiDAR and monocular plus LiDAR was recorded as fastest method among the other two as identified in Table 4.3 but the height still limited by given range data.

The potential of monocular plus LiDAR was more revealed using long-run sequences. As identified during the experiments there are good outcome obtained by using such a sensor for constructing stixels in category (C). This was also beneficial in terms of data processing. The monocular LiDAR has achieved the fastest method in all datasets. The accuracy rate was very close between monocular LiDAR and saliency map with a difference around 1.21%. The improvement rate achieved for monocular plus LiDAR for 803 frames is 5.97% compared to the base-line stixel method (see Fig. 4.11).

Overall, to optimise the balance between accuracy and processing time when generating stixels then monocular plus LiDAR is recommended for such condition. The monocular plus LiDAR presents the robustness for a given point-cloud under long-run sequences. The accuracy and height of objects was essential for any obstacle detection method. Therefore, saliency

Table 4.2: Mean differences of distances between stixel maps and LiDAR data (in cm)

Category	A	B	C
Stixel calculation using base-line method	28.22	28.16	25.98
Use of saliency map fusion method	**23.12**	**21.23**	24.79
Use of LiDAR and monocular fusion	24.27	22.63	**24.43**

Table 4.3: Run-time profiling for single-layer stixels calculation on KITTI dataset.

Category	Monocular LiDAR	Base-line	Saliency fusion
A	1.16s	7.26s	14.38s
B	1.16s	7.33s	14.49s
C	1.18s	7.19s	12.38s

map considered to be the foremost stixel detection option especially to be used for identifying stixel heights. The processing time shows this method spent a massive amount of time to process stixels per frame, however, such incidents can be overcome especially most of test vehicles are provided with FPGAs.

4.4 Summary

This chapter proposed two methods for robust height calculation of stixels using a fused-data strategy. The main advantage is to produce a low-cost architecture for reducing false-positives in stixel estimations. The proposed methods have been compared to the original base-line method. Experiments show that the error rate was reduced (by 24.6% on the evaluated data) by the fused membership. Also, the error rate was decreased by 5.97% by fusing range and optical data represented by monocular plus LiDAR.

The results demonstrate the potential of this novel method towards more accurate obstacle surface detection and accurate height segmentation of traffic scenes.

Chapter 5

Improvement of Multi-layer Stixels

The integration of a reliable confidence map for multi-layer stixels segmentation can support the selection of prior and data terms; our confidence map uses a calibrated collinear trinocular vision model. It is generated from three conjugate synchronised stereo images for evaluating the consistency of disparity values. The evaluation measure is referred to as transitivity error in disparity space. Multi-layer stixels are commonly generated from a single disparity map which make them merely dependent on the applied stereo matcher. A multi-map fusion is proposed to achieve more reliable stixels segmentation for disparity values. This chapter informs about a solution to multi-layer stixels model based on the extension of multi-ocular framework and monocular plus LiDAR proposed in Chapter 3 and 4, respectively. - Parts of this chapter is published in [3, 2].

5.1 Multi-layer stixel model

Unfortunately, issues being a challenge for stereo matchers are widely existing in traffic scenes, thus more efforts are required to improve disparity maps, also aiming at more reliable stixel segmentation (with minimising the number of required parameters) for forming semantic objects. Each stixel carries important but noisy information, and it is challenging to decide for each stixel to which object it belongs to (see Fig. 5.1). The multi-layer model improves the precision of the *stixel world*. Every column in a given disparity image D is segmented into multiple vertical segments (see Fig. 5.2). Every segment is assigned to one of the classes *sky*, *ground*, or *object*, using an integrated model of *data-term* and *prior-term*.

By revising the problem of stixel generation into a segmentation problem, we follow the proposed model defined in [73]. In this work we extend

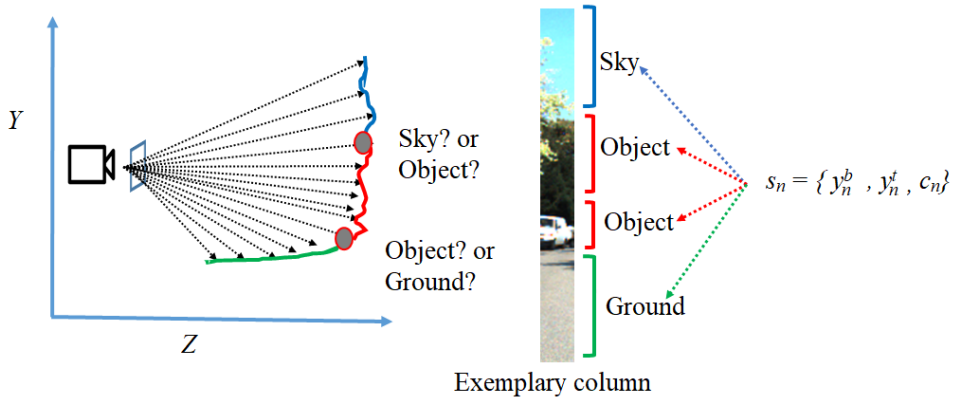

Figure 5.1: Multi-layer segmentation concept.

the concept by paying attention to missing or faulty disparities especially in the sky and ground regions based on a defined confidence map derived from multi-ocular vision. The extension is required since we are interested in semantic segmentation, and this step supports the labelling efficiently.

In Eq. (2.2), each element in the disparity map is considered equally. In this work we propose to use a weighted sum of our trinocular confidence values:

$$V_y(d) = \sum_{1 \le u \le N_{\text{col}} \wedge \text{int}(\delta(u,v))=d} \Gamma(u,v) \tag{5.1}$$

Here, elements with higher TED-based confidence become more influential. A result of confidence map and trinocular multi-layer stixels is provided in Fig. 5.3.

The proposed functions $f_n(.)$ satisfy the following properties:

- Ground-based stixels are generated based on a proposed graph-cut method, supported by the confidence map to enable a robust detection of ground. This enables us to determine the *ground function* $f_g(.)$ (note: "g" instead of number $n = 1$) without depending on direct camera parameters only (i.e. height and tilt). More details below.

- The assumed value for the *sky function* is $f_s(y) = 0$ (note: "s" instead of number $n = N_x$), for all y with $1 \le y \le y_s^t$.

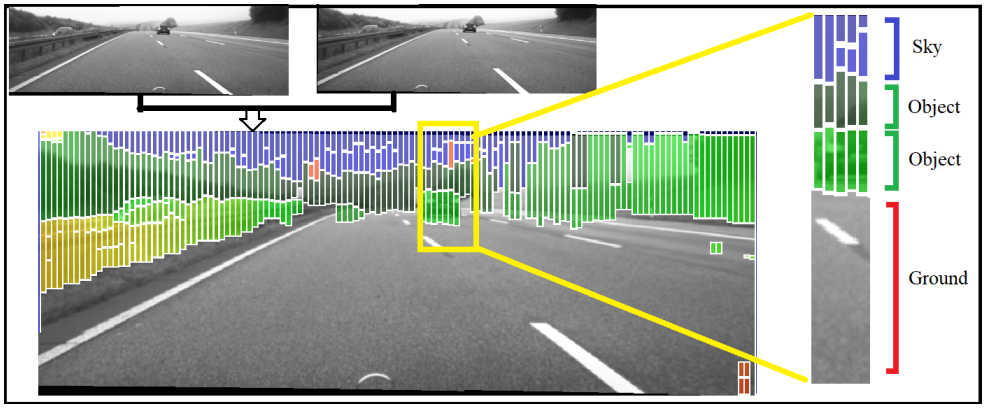

Figure 5.2: Overview of multi-layer stixel segmentation.

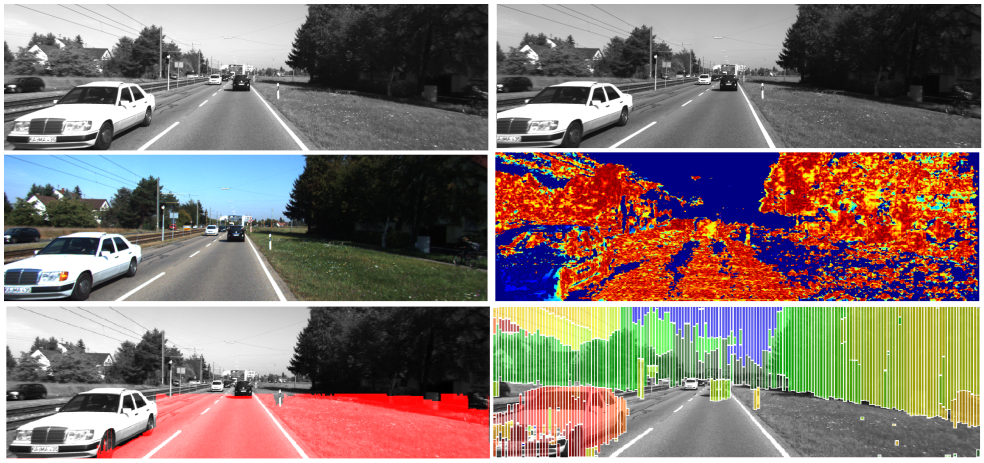

Figure 5.3: Trinocular confidence, ground manifold and multi-layer stixel. *Top row and middle row left*: Trinocular stereo pair from the ROAD dataset on KITTI. *Middle right*: TED-based confidence measure where red and blue pixels indicate high and low confidence values, respectively. *Bottom middle*: Calculated ground-manifold (using y-disparity, graph-cut, and confidence map). *Bottom right*: Multi-layer stixels segmentation using the confidence map.

- For an *object function*, we have that $f_o(y) = \mu_n$ (note: "o" instead of a number n between 1 and N_x) where μ_n is the mean disparity within s_n. We extend this function to enable transitivity error analysis to be used for valid disparity coverage.

Functions $f_n(.)$ are implemented based on data-terms defined for each function, and they are verified based on prior-terms. This step is treated as a typical *maximum-a-posteriori* (MAP) estimation problem. We aim to find the most probable labelling in $\mathbb{L}$:

$$L^* = \text{argmax}_{L \in \mathbb{L}} \Pr(L|D) \tag{5.2}$$

5.1.1 Data and prior terms

As a key contribution, we integrate the valid disparities, provided in Γ of Eq. (5.1), to be used as a verified model for a valid disparity map. In order to solve the L^* segmentation problem more efficiently, we follow [73] by applying the Bayes' theorem. This allows us to write the posterior probability in Eq. (5.2) now as follows:

$$\Pr(L|\Gamma) \approx \Pr(\Gamma|L) \cdot \Pr(L) \tag{5.3}$$

The term $\Pr(L|\Gamma)$ embodies the result of the product of the conditional probability of Γ given L (data-term) and the prior probability $\Pr(L)$ of L.

Considering the data-term part, this implementation uses individual measurements $d_{x,y} \in \Gamma$ as maturely independent. Furthermore, the TED-weighted y-disparity map is used as an input for this model. All data within Γ_x is assumed to be independent from all labels L_x, with $x \neq \overline{x}$. As a result we obtain:

$$\Pr(L|\Gamma) \approx \prod_{x=1}^{N_{col}} \Pr(\Gamma_x|L_x) \cdot \Pr(L_x) \tag{5.4}$$

In order to obtain $f_n(y)$, the data-term model uses the *conditional probability density* in $\Pr(\Gamma_x|L_x)$ to rate the likelihood of disparity (depth), given labelling L:

$$\Pr(\Gamma_x|L_x) = \prod_{n=1}^{N_x} \prod_{y=y_n^b}^{y_n^b} \Pr(d_y|s_n, y) \tag{5.5}$$

where

$$\Pr(d_y|s_n, y) = \begin{cases} \Pr_\Gamma(d_y|s_n, y) \cdot (1 - p^{c_n}), \ \chi(d_y) = 1 \\ p^{c_n}, \ \text{otherwise} \end{cases} \tag{5.6}$$

The term $\Pr_\Gamma(d_y|s_n, y)$ represents the probability for a given disparity value d_y at row y given segment s_n. The symbol χ is used as a flag to indicate if the disparity value are within outliers or not. This means when $\chi(d_y) = 1$ if the disparity d_y at row y is valid (i.e. $0 \leq d_y \leq 64$) and 0 otherwise. For example, the symbol p^g (p^{c_n}) is the probability to observe a ground given that d_y is invalid. It is worth to mention that data-term model is a mixture model of uniform distribution which is used to observe outliers and Gaussian distribution that used to rate affinity of d_y to s_n [74].

The prior-term, encoded in $\Pr(L_x)$, models the real-world constrains. Compared to the data-term $\Pr(\Gamma_x|L_x)$, it does not include direct dependencies to the input data. It is interpreted as a set of cost-tables (i.e. a likelihood instead of actual probabilities).

The prior-terms are evaluated based on a vertical probability shift, from each segment to the segment above: $\Pr(s_n|s_{n-1})$. The prior evaluation is segment-wise for favouring some configurations of segmented neighbourhoods:

$$\Pr(L_x) = \Pr(s_1) \cdot \prod_{n=2}^{N_x} \Pr(s_n|s_{n-1}) \tag{5.7}$$

5.1.2 Computational feasibility using dynamic programming

As we noticed, there are a lot of computational operations required for multi-layer stixels and to classify each segment to a corresponding class. In more detail, to seek optimal segmentation of a column in disparity map to be include ground, object, and sky a dynamic programming technique is performed. We revised the base-line definition of dynamic programming in multi-layer stixel by implementing ground function based on our graph-cut dynamic programming algorithm which modifies the existing assumption of the ground surface is presume to be linear up to horizon (see Fig. 5.4).

As happens with very similar scenarios (e.g. stereo matching), the dynamic programming [13] is performed to subdivide one massive problem

into smaller pieces and store the solution in a memory structure to be partially retrieved upon sub-problem occurred again. This will support our endeavour to calculate multi-layer stixel with efficient use of memory. In order to say a problem is solved by dynamic programming, the problem need to adhere to certain criteria which are:

- optimal substructure solution: as mentioned earlier a major problem must be decomposed into subtasks. Each of those subtasks can be also subsequently decomposed or solved optimally. Hence, when a recursive definition of problem is provided, a minimisation cost need to be calculated rather than maximum posteriori.
- discrete solution: for ensuring dynamic programming is fit for our purpose, the definition of stixel segment $s_n = \{y_n^b, y_n^t, c, f_n(.)\}$ needs to be in discrete format.

The points mentioned above need to be further explained. To ensure optimal substructure solution, we simplify our description to illustrate computation of minimum cost by denoting a table entry C^y at a row y that represent the minimum aggregated cost for three cases: ground, object, and sky:

$$C^y = \{G^y, O^y, S^y\} \tag{5.8}$$

Each case match-up the segmentation ending on a stixel of the corresponding class $\mathbb{C} = \{g, o, s\}$. The stixel at the end of the segmentation linked with each minimum cost is given as:

$$\begin{aligned} g_b^t &= \{y^b, y^t, g\} \\ o_b^t &= \{y^b, y^t, o\} \\ s_b^t &= \{y^b, y^t, s\} \end{aligned} \tag{5.9}$$

For example, a given entry O^{15} it represents a segmentation that is type of object with top point row coordinate 15. Moreover, we can also rate different constellation of neighbouring segment by denoting c as model (priori) cost. For example $c(O^{25}, s_{26}^{30})$ which means the model costs for integration of segmentation ending with type object at top-point 25 next to another sky segment from 26 to 30. It could be also expressed as single segment, for example, $c(g_0^{12})$ which means the model costs for first segment is marked as

ground g_0^{12}. Referring to Eq. (5.8), at the beginning, we need to calculate aggregated cost C for a segmentation of length one. This can be achieved by:

$$\begin{aligned} C^0 &= \{G^0, O^0, S^0\} \\ &= \{g_0^0 + c(g_0^0), o_0^0 + c(o_0^0), s_0^0 + c(s_0^0)\} \end{aligned} \tag{5.10}$$

Hence, if we want to determine the segmentation from 0 to 1 then it can either be a combined segment or utilise the previous solution (this achieves our purpose for dynamic programming) such that:

$$C^1 = \{G^1, O^1, S^1\} \tag{5.11}$$

where:

$$G^1 = min \begin{cases} g_0^1 + c(g_0^1) \\ g_1^1 + c(g_1^1, G^0) + G^0 \\ g_1^1 + c(g_1^1, O^0) + O^0 \\ g_1^1 + c(g_1^1, S^0) + S^0 \end{cases} \qquad O^1 = min \begin{cases} o_0^1 + c(o_0^1) \\ o_1^1 + c(o_1^1, G^0) + G^0 \\ o_1^1 + c(o_1^1, O^0) + O^0 \\ o_1^1 + c(o_1^1, S^0) + S^0 \end{cases}$$

$$S^1 = min \begin{cases} s_0^1 + c(s_0^1) \\ s_1^1 + c(s_1^1, G^0) + G^0 \\ s_1^1 + c(s_1^1, O^0) + O^0 \\ s_1^1 + c(s_1^1, S^0) + S^0 \end{cases} \tag{5.12}$$

Following this solution, the entry of C^2 can be decomposed to (if it's not combined with previous segments):

$$C^2 = \{G^2, O^2, S^2\} \tag{5.13}$$

where:

$$
\begin{aligned}
G^2 = min \begin{cases} g_0^2 + c(g_0^2) \\ g_1^2 + c(g_1^2, G^0) + G^0 \\ g_1^2 + c(g_1^2, O^0) + O^0 \\ g_1^2 + c(g_1^2, S^0) + S^0 \\ g_2^2 + c(g_2^2, G^1) + G^1 \\ g_2^2 + c(g_2^2, O^1) + O^1 \\ g_2^2 + c(g_2^2, S^1) + S^1 \end{cases} \\
O^2 = min \begin{cases} o_0^2 + c(o_0^2) \\ o_1^2 + c(o_1^2, G^0) + G^0 \\ o_1^2 + c(o_1^2, O^0) + O^0 \\ o_1^2 + c(o_1^2, S^0) + S^0 \\ o_2^2 + c(o_2^2, G^1) + G^1 \\ o_2^2 + c(o_2^2, O^1) + O^1 \\ o_2^2 + c(o_2^2, S^1) + S^1 \end{cases} \\
S^2 = min \begin{cases} s_0^2 + c(s_0^2) \\ s_1^2 + c(s_1^2, G^0) + G^0 \\ s_1^2 + c(s_1^2, O^0) + O^0 \\ s_1^2 + c(s_1^2, S^0) + S^0 \\ s_2^2 + c(s_2^2, G^1) + G^1 \\ s_2^2 + c(s_2^2, O^1) + O^1 \\ s_2^2 + c(s_2^2, S^1) + S^1 \end{cases}
\end{aligned}
\tag{5.14}
$$

As we can see, there is a recursive scheme followed for each entry. Thus, we can generalise the calculation of aggregate cost in Eq. (5.8) to be elaborated in:

$$
\begin{aligned}
C^y &= \{G^y, O^y, S^y\} \\
G^y &= min \begin{cases} g_0^y + c(g_0^y) \\ g_1^y + c(g_1^y, G^0) + G^0 \\ g_1^y + c(g_1^y, O^0) + O^0 \\ g_1^y + c(g_1^y, S^0) + S^0 \\ \dots \\ g_y^y + c(g_y^y, G^{y-1}) + G^{y-1} \\ g_y^y + c(g_y^y, O^{y-1}) + O^{y-1} \\ g_y^y + c(g_y^y, S^{y-1}) + S^{y-1} \end{cases}
\end{aligned}
\tag{5.15}
$$

An exemplary path is provided for G^y in Fig. 5.4. We just show G^y so

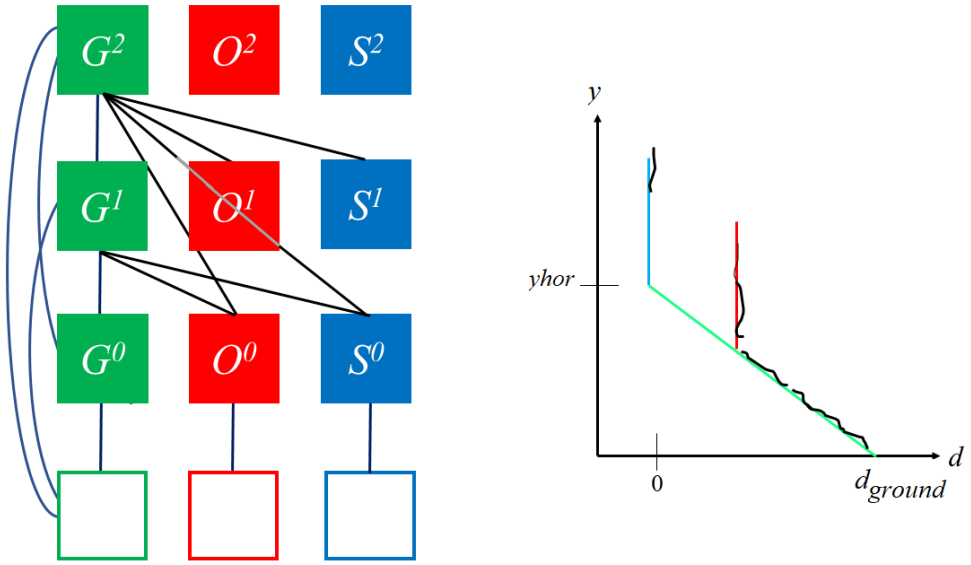

Figure 5.4: Illustration of exemplary y-disparity and cost table calculation. *Left:* explain all assumed paths for the ground notion up to $y = 2$. *Right:* fictitious y-disparity space which illustrate how dynamic programming is optimally performed to cut segments. The black points show how real data might be exist.

the other terms O^y and S^y can follow the same scheme accordingly. The total result for the three cases in cost table are provided in Fig. 5.5. In order to start the backtracking procedure to seek the optimal final path for segmentation, then C^{h-1} need to be calculated. In this context, when we arrive at $C^{h-1}_{min} = \min(G^{h-1}, O^{h-1}, S^{h-1})$, that means that the process of optimal backtracking will be initialised (see 5.1.3 for more details).

It is worth to mention that we can work on negative log-likelihood rather than direct probability values (i.e. all multiplications operations are substituted by summations). This is beneficial for many reasons. First, it provides more efficient implementation. Second, it reduces the malformed input affects to the execution of the scheme [37]. The scheme means the data and prior-term calculations so the problem remains untouched while optimising it. Hence, in order to reflect on the cost table mentioned above the notation

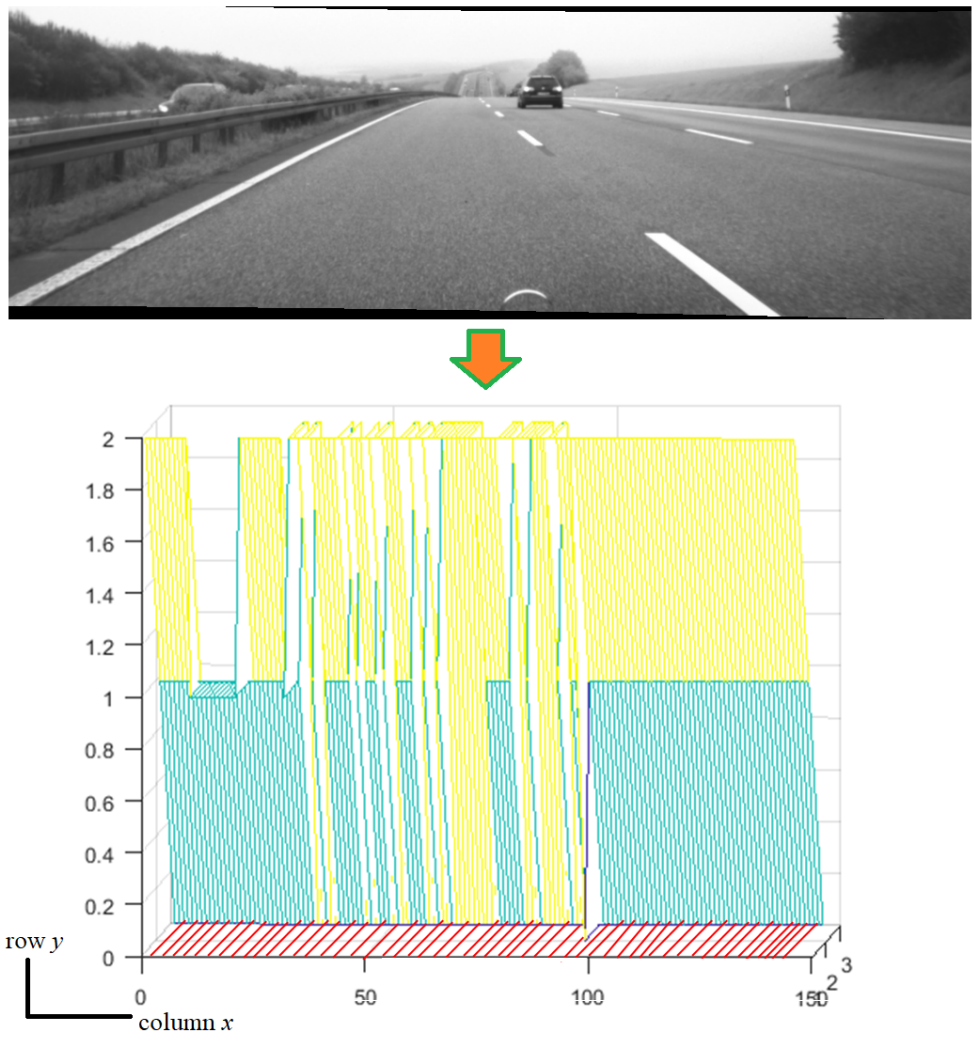

Figure 5.5: Illustration of cost table. *Top:* Sample of LEFT image. *Bottom:* Cost table which shows cost for each case *ground (red), object (green), sky (yellow).*

o_a^b is corresponding to the sum of all data costs related to object hypothesis which is $-\sum_{y=a}^{b} \log \Pr(d_y|s_n, y)$. Using the same scheme applied to model costs movement from one class label to another [i.e. $c(o_a^b, O^{a-1})$] which can be represented in $-\log \Pr(s_n|s_{n-1})$.

Finally, as we want to utilise dynamic programming in multi-layer stixel segmentation. The provided data needed to be in discrete format. By revisiting the $s_n = \{y_n^b, y_n^t, c_n, f_n(.)\}$, we can ensure that $y_n^b, y_n^t, c_n \in \{0, \ldots, h-1\} \subset \mathbb{N}$. However, $f_n(.)$ is identified based on disparity space (data-term) which $d \in \mathbb{R}$. This reminds us about a similar problem that occurred in stereo matching when implemented using dynamic programming [68, 92, 99]. According to the working scheme in stereo matching, we will be able to discretise the optimisation space to integer numbers by setting disparity steps [74]. There is a challenge in multi-layer stixels where we can not assess every function f_n explicitly. In this case, the processing time will be higher [73]. Therefore, we are not incorporating the optimisation step into f_n. We have followed the proposed scheme, identified in base-line multi-layer stixel [73], by performing an explicit check on a well-selected function candidate. However, we will seek to incorporate confidence measures obtained from trinocular vision towards improving the accuracy of f_n as will be discussed in 5.1.4.

5.1.3 Optimal path using backtracking

As we noticed the cost table is considered to be an essential component of dynamic programming. The dimension of a cost table is identical to image height (since we want to see position of cuts) and number of candidate labels [74, 90]. Hence, the evaluation and computation of cost table is implemented on $h \times 3$ (size of cost table). We can optimise this scheme by creating another table can be called as "marker table" (dimension is identical to cost table) which can be used to backtracking step and lead to more efficiency (see Fig. 5.6). The marker table include indices that can represent candidate labels (type) and image row coordinates. Moreover, the indices can connect different entries of the cost table. This will be beneficial to show sequence of possible path optimisation for a segment. In this context, to seek the best cut for a column, the marker and cost tables are calculated simultaneously. Afterwards, the backtracking process is carried out from the top of cost table through all elements to the bottom and this will show how optimum path is extracted.

The mechanism to extract the optimum path is portrayed in Fig. 5.6. Basically, an example of 9 pixel height is provided to show how optimum path

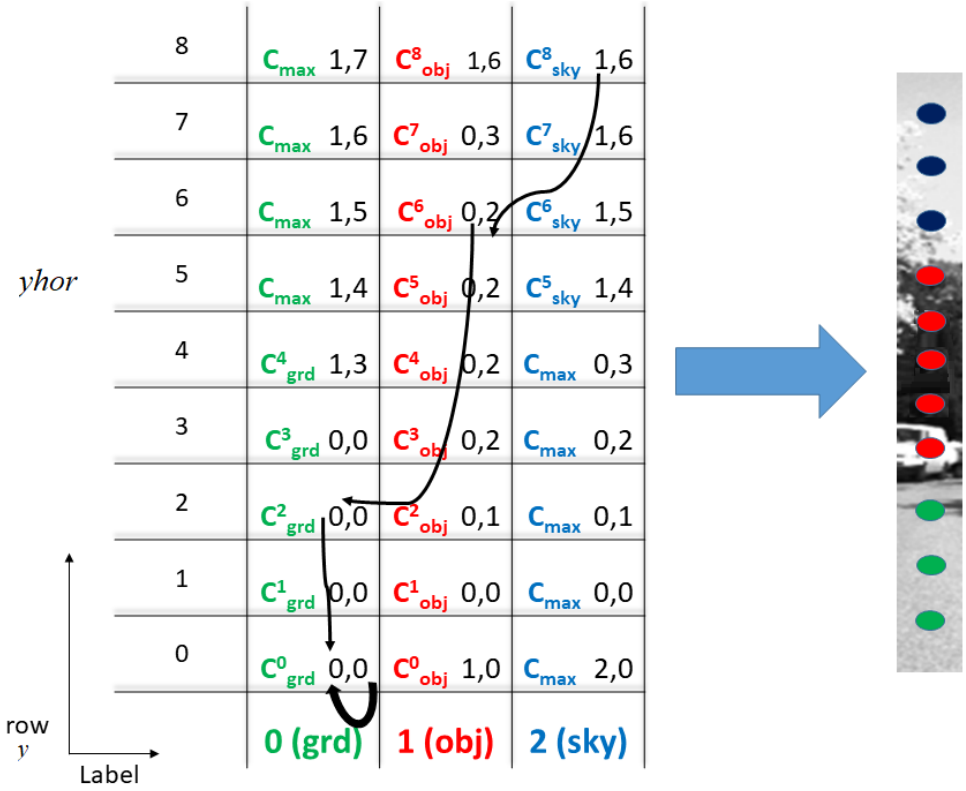

Figure 5.6: Illustration of cost and marker table for a sample image with 9 (virtual) pixels height. The coloured alphabet C represent the minimum cost calculated for this cell. The row and column showing as pair indices represent the type of label and top-point coordinate of that predecessor.

is extracted. For both tables (marker and cost) it consists of two elements as an input: *element A* which represents the minimum cost (shown as colour alphabet C) that could be identified for that cell, and *element B* pair of indices represent information about preceding segment (two digits separated by coma). As we can see in Fig. 5.6, the *element B* providing a rich details about how to traverse through the cost table. It include the preceding label type (0 for ground, 1 for object, 2 for sky) separated by a coma the second index refers to the row that indicate top-point coordinate y^t.

As mentioned earlier, when the minimum cost is reached to C_{min}^{h-1} a backtracking step will be performed. In this example the minimum costs select C_8^{sky}, so the segment will be marked as sky. The row and column showing as pair indices can be translated as the preceding segment information[1] which means the label type and top-point coordinate y^t. As shown in our example, the indices points at object segment starting at row 6. Accordingly, the other segments can be calculated in the same scheme until it reaches the bottom row of cost table ending up with self-loop. In some cases it is helpful to indicate the position of $yhor$ (i.e. horizon line), in which any sky cost comes under this value need to be marked as max or infinity because it is impossible to have sky segment positioned under ground segment. The same scenario is used when ground segment comes above the $yhor$.

5.1.4 Increasing the robustness

The issue of how to use the particular candidate function $f_n(.)$ to collect the data term costs for segment s_n remained unanswered in the previous sections.

Assuming that the used measurement model is a composite of Gaussian distributions only, the most favourable candidate would be the mean disparity value of all disparities within segment s_n. The reason behind choosing the mean is that it can produce the minimum variance [73].

The mean can be efficiently computed by using the following formula, see [73]:

$$\begin{aligned} \mathrm{sum}[y_0] &= \textstyle\sum_{y=1}^{y_0} \chi(d_y) \cdot d_y \\ \mathrm{valid}[y_0] &= \textstyle\sum_{y=1}^{y_0} \chi(d_y) \end{aligned} \tag{5.16}$$

Using these sums, the mean disparity within a range of y^b to y^t, and thus a constant value for candidate function $f_n(.)$, is determined by

$$f_n := \mathrm{mean}(y^b, y^t) = \frac{\mathrm{sum}[y^b] - \mathrm{sum}[y^{t-1}]}{\mathrm{valid}[y^b] - \mathrm{valid}[y^{t-1}]} \tag{5.17}$$

Experiments show that the mean is influenced by inliers and outliers which contribute to the mean cost equally. Furthermore, as pointed out in

[1] The preceding segment means the one which indicates the current costs.

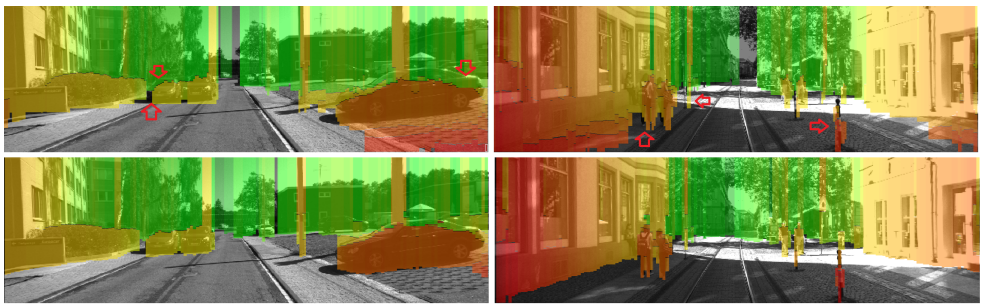

Figure 5.7: Multi-layer stixel maps tested on KITTI data using depth obtained from LiDAR. *Top row*: Red arrows depict current multi-layer stixel problems (displacements in the representation). *Bottom row*: Examples when using the proposed solution.

previous studies [51], disparity values are prone to difficult weather conditions which weakens the confidence. As a result, we chose to incorporate the TED-weighted y-disparity map. This serves as a robust alternative to a base-line multi-layer stixel approach as this does not require any complex re-organisation of the disparity data such as ordering. For obtaining an accurate estimation of f_n, we proposed to use TED-weighted d_y as follows:

$$f_n = \frac{\sum_{y=y_n^t}^{y_n^b} \varepsilon_y \cdot d_y}{\sum_{y=y_n^t}^{y_n^b} \varepsilon_y} \tag{5.18}$$

where

$$\varepsilon_y = \frac{\chi(d_y)}{1 + |d_y - \mu|} \tag{5.19}$$

5.2 Monocular multi-layer stixel: LiDAR guided

An essential step in multi-layer stixel estimation is estimating the road surface. As presented in [73], the road surface can be estimated directly from camera parameters, however, this scheme might be infeasible when the provided dataset is missing some information about camera parameters (i.e., tilt angle). As shown in Fig. 5.7, the monocular stixel (first-row) is supposed to be improved but there are a still a lot of false positives. These

affect obstacle representation and road surface estimation. Using a point cloud we need to estimate the road manifold from which we can derive a 3D rotation and translation matrix. Usually, converting a world coordinate $P_w = [X_w, Y_w, Z_w]^\top$ into an image plane coordinate requires geometric information such as:

$$\varepsilon \cdot [x, y, 1]^\top = \mathbf{K}[\mathbf{R} \,|\, \mathbf{t}][X_w, Y_w, Z_w]^\top \tag{5.20}$$

where x, y represent the image plane coordinates and ε is a depth scalar for depth values > 0. We can remove $[\mathbf{R} \,|\, \mathbf{t}]$ from the above equation since the rotation matrix is an identity matrix and the translation vector is all zeros. We can solve the above unknown parameters by applying matrix inversion:

$$\frac{1}{\varepsilon}[X_w, Y_w, Z_w]^\top = \mathbf{K^{-1}}[x, y, 1]^\top \tag{5.21}$$

We can represent the left side in (5.21) by a variable F. This equation can be then re-written as:

$$F = \mathbf{K^{-1}}[x, y, 1]^\top \tag{5.22}$$

To identify a pixel with world coordinates X_w, Y_w, Z_w, we need

$$[X_w, Y_w, Z_w]^\top = F \cdot \varepsilon \tag{5.23}$$

Then, we can use *M-estimator sample consensus* (MSAC) which fits a plane to a cloud of points. The fitting process is applied only on inlier points that have a maximum tolerable distance to the plane. The model can be verified in the road plane equation to estimate the road plane coefficients:

$$a_0 X_w + a_1 Y_w + a_2 Z_w + a_3 = 0 \tag{5.24}$$

So far we just estimate the road plane coefficients. In order to find a known world coordinate location in the depth map image, we use

$$F = \frac{1}{\lambda}[X_w, Y_w, Z_w]^\top \tag{5.25}$$

where λ is the ground depth scalar to be calculated. That means

$$[X_w, Y_w, Z_w]^\top = [\lambda F_1, \lambda F_2, \lambda F_3]^\top \tag{5.26}$$

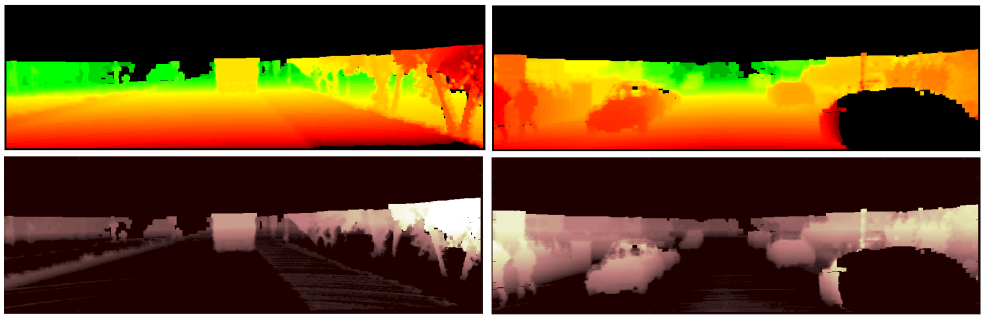

Figure 5.8: *Top*: Disparity map using monocular vision plus LiDAR. *Bottom* Ground disparity.

By substituting Eq. (5.26) in Eq. (5.24), this results in:

$$a_0\lambda F_1 + a_1\lambda F_2 + a_2\lambda F_3 + a_3 = 0 \tag{5.27}$$

The value of λ can be found by

$$\lambda = \frac{-a_3}{a_0 F_1 + a_1 F_2 + a_2 F_3} \tag{5.28}$$

Finally, the ground disparity can be calculated:

$$\mathrm{GD} = f \cdot \frac{b}{\lambda F_3} \tag{5.29}$$

The generated ground disparity is demonstrated in Fig 5.8. For multi-layer stixel construction, ground-based stixels are generated based on a ground disparity map GD. This enables us to determine the *ground function* $f_g(\cdot)$ (note: "g" instead of number $n = 1$) and identify the road surface using a single camera after resolving the displacement issue.

5.3 Experimental results

The quality of the proposed monocular and trinocular multi-layer stixel calculation is evaluated using LiDAR data provided on the KITTI Vision Benchmark Suite [35]. Results of the proposed methods are compared to results

of the base-line stixel detection method [73] and multi-layer based ground plane (by Hough transform in y-disparity space) similar case provided by [43]. For naming convention, we give a short-cut for the four tested methods as following: *Binocular base-line (BB)*, *Binocular ground plane (BGP)*, *Trinocular graph-cut (TGC)*, and *Monocular LiDAR (ML)*.

The aim of the experiments is to analyse the accuracy of multi-layer stixels rather than the processing time (which is optimised for in-car applications by parallel hardware such as FPGAs).

We aimed at having a wide diversity of challenging traffic situations, including different lighting conditions, different road views, shades, and colourings. We selected $1,501$ trinocular stereo frames from the ROAD, CAMPUS ,RESIDENTIAL, and CITY categories, which contain cars, cyclists, pedestrians, trees, and traffic signals. The test sequences are listed in Table 5.1. For bi- and trinocular vision, we use uniformly the SGM stereo matcher for disparity calculation [45], without any further pre- or post-processing of disparity values.

Regarding previously stated challenges for evaluating stixels while using the KITTI dataset [83], we address those by following [7], i.e. by making use of the Velodyne high-definition 3D laser scanner data also provided in the KITTI dataset. In short, we use those range data as a ground-truth reference to evaluate the distance values assigned to extracted stixels.

Quantitative results are listed in Table 5.2 using a binocular, monocular or trinocular configuration. Figure 5.9 illustrates qualitatively achieved results (also shown in Fig 5.10, Fig. 5.11, Fig. 5.12, Fig. 5.13, Fig. 5.14, Fig. 5.15).

Table 5.1: Selected test sequences from the KITTI dataset

Category	Sequence	Tag	Frames
ROAD (OPEN-ROAD)	`2011_09_26_drive_0015`	A	297
ROAD (HIGHWAY)	`2011_09_26_drive_0032`	B	390
RESIDENTIAL (URBAN)	`2011_09_26_drive_0035`	C	131
CAMPUS	`2011_09_26_drive_0038`	D	110
CITY (BUSY)	`2011_09_26_drive_0091`	E	340
CITY (SHADOW)	`2011_09_26_drive_0106`	F	233

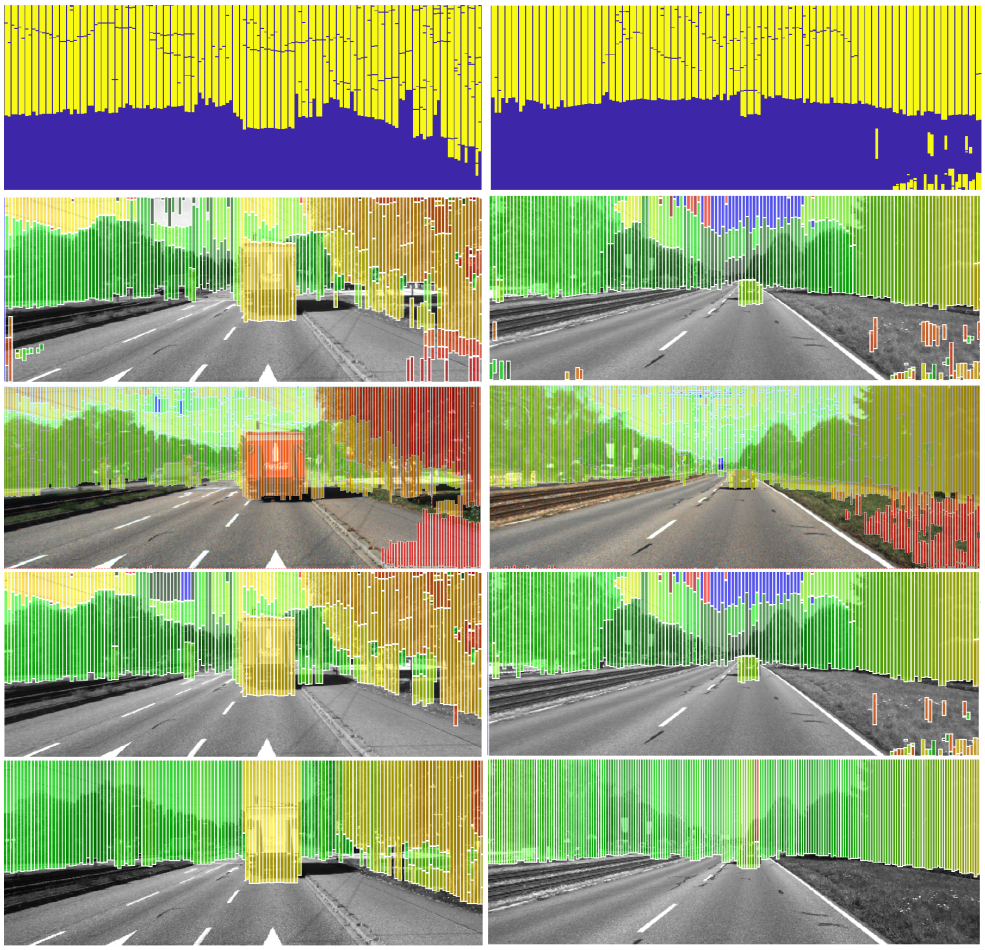

Figure 5.9: Quantitative results using KITTI data; first column RESIDENTIAL, and second column ROAD. *First row*: Stixel maps. *Second row*: Base-line multi-layer stixels (BB) where ground manifold is estimated by using known camera parameters. *Third row*: Multi-layer stixels estimated using binocular ground plane (BGP). *Fourth row*: Multi-layer stixels estimated using trinocular graph cut (TGC). *Fifth row*: Multi-layer stixels estimated using monocular plus LiDAR (ML).

The scenes provided in CITY or RESIDENTIAL data differ from ROAD data

Table 5.2: LiDAR-based qualitative evaluation of different camera configuration using KITTI data

Sequence	BB		BGP		TGC		ML	
	μ	σ	μ	σ	μ	σ	μ	σ
A	24.0	1.25	24.0	1.36	**23.8**	1.14	23.9	1.35
B	25.8	1.93	25.6	1.91	24.3	1.89	**23.8**	1.76
C	20.9	1.14	21.0	1.12	20.5	1.16	**19.6**	2.95
D	18.3	0.73	18.1	0.74	18.3	0.73	**17.5**	0.85
E	18.4	2.91	**18.2**	2.87	18.4	2.91	20.4	2.46
F	19.1	2.63	18.9	2.36	**18.8**	2.16	20.4	27.4

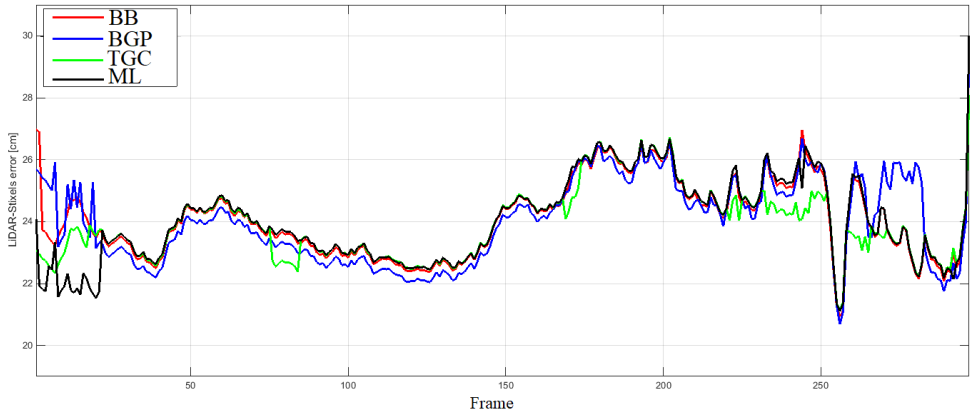

Figure 5.10: Error rates illustrate the mean of LiDAR-stixel distance error (in cm) for the four approaches - category A

by also showing pedestrians, cyclists, and more buildings. Furthermore, the sky class was limited in the two mentioned datasets due to the shown urban scenes. As shown in Fig. 5.10, and Fig. 5.15, the error rate is the lowest when using the TGC model in two categories ROAD(open-road) and CITY(shadow), while ML model achieve the lowest error rate in RESIDENTIAL (urban), ROAD(highway), and CAMPUS (see Fig. 5.11, Fig. 5.12, and Fig. 5.13). BGP outperforms other methods in CITY category (busy) as shown in Fig. 5.14.

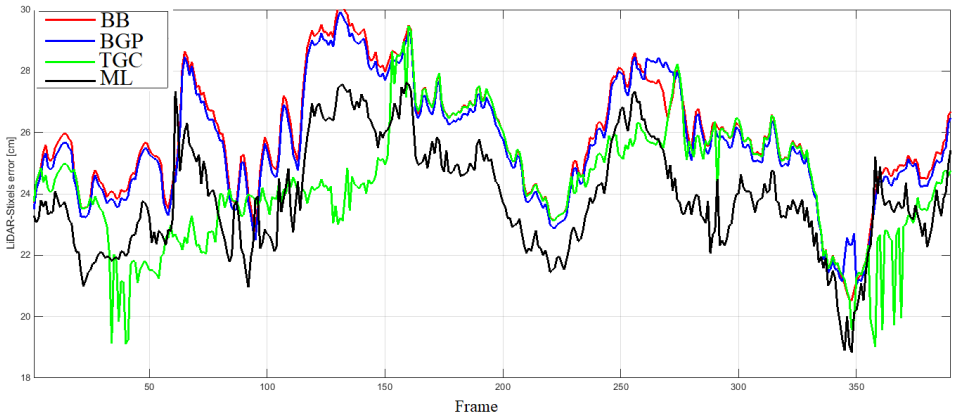

Figure 5.11: Error rates illustrate the mean of LiDAR-stixel distance error (in cm) for the four approaches - B category

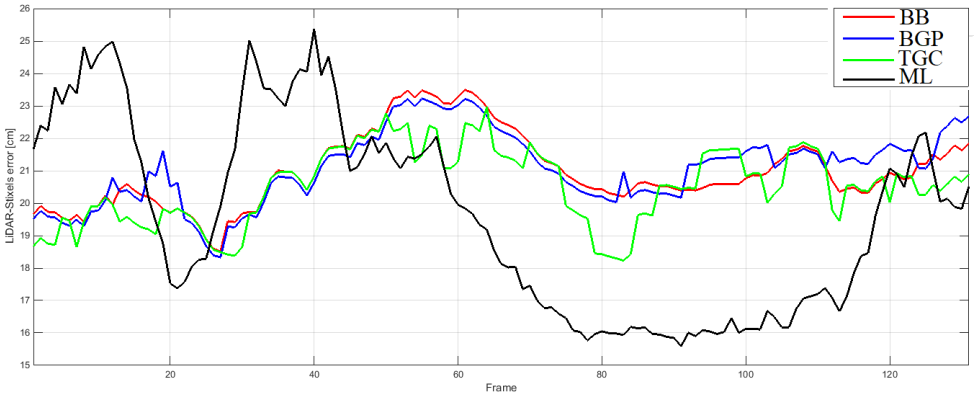

Figure 5.12: Error rates illustrate the mean of LiDAR-stixel distance error (in cm) for the four approaches - category C

In the RESIDENTIAL sequence, the monocular LiDAR model achieves the lowest mean of a LiDAR-stixel error of 19.6 cm, which is a 6.2% decrease compared to the original method with 20.9 cm. The contents include cars parked on side of road, houses, and road junctions. The three methods (BB, BGP, and TGC) faced difficulties to identify the stixel at the road junction which exist from frame 90 - 110. The tested vehicle position forms a major

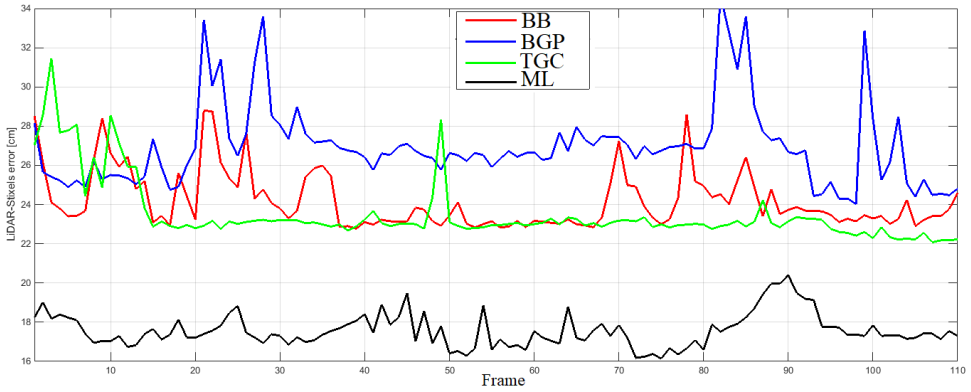

Figure 5.13: Error rates illustrate the mean of LiDAR-stixel distance error (in cm) for the four approaches - category D

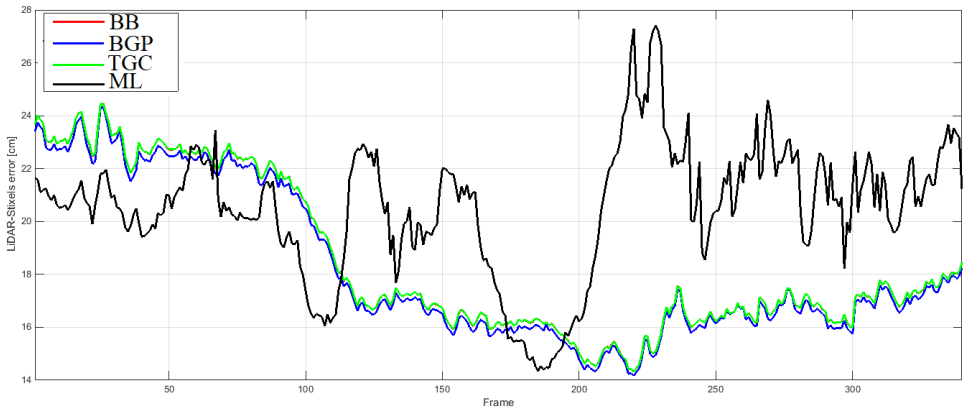

Figure 5.14: Error rates illustrate the mean of LiDAR-stixel distance error (in cm) for the four approaches - category E

challenge for the stereo matchers and using LiDAR sensor can give more details. However, sun-strike started to appear at the end of the sequence still form a challenge for such sensors. As we can observe from Fig. 5.12 and Fig. 5.16, the TGC method shows a robustness in such scenario (see Fig. 2.19 for comparison with baseline).

Yet, the ML method did not perform as good as TGC with respect to

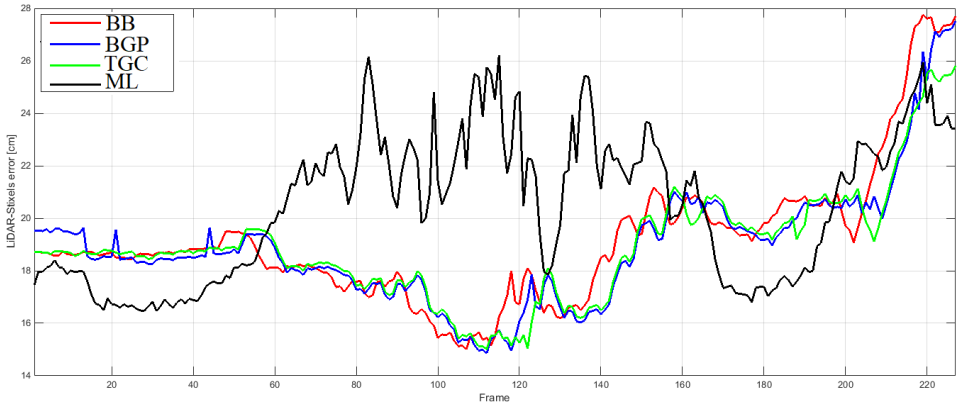

Figure 5.15: Error rates illustrate the mean of LiDAR-stixel distance error (in cm) for the four approaches - category F

Table 5.3: Average number of stixels extracted per frame in the tested KITTI sequences

Sequence	BB	BGP	TGC	ML
A	1,142	1,102	567	213
B	1,196	242	970	256
C	1,204	213	902	306
D	643	996	630	319
E	569	975	567	281
F	1,178	280	624	316

open road scenarios as represented in category A (open road). The main reason for this change is due to having valid disparities covered, and confidence dealt fine with missing texture especially in the road surface. Results show that TGC covers more valid disparities compared to others, and slightly outperforms others, also regarding a smaller rate of false alarms as shown in Fig. 5.9.

The situation changed for the highway sequences (category B) where the mean of a LiDAR-stixel error yields close error rates between ML and TGC. In such datasets with vehicles are moving the error percentage of ML method

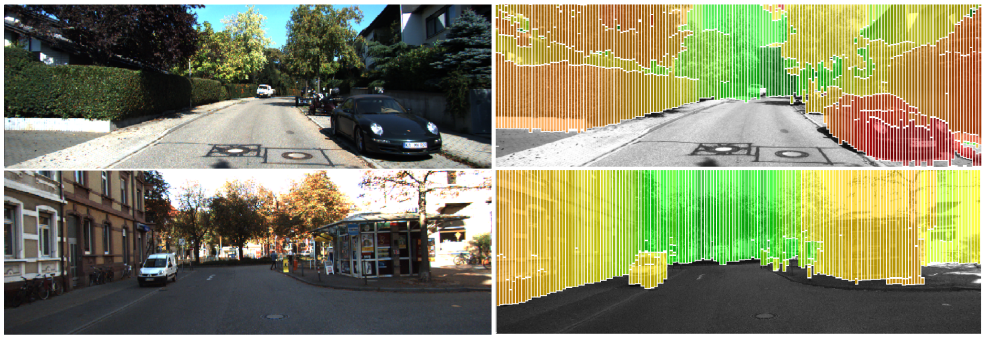

Figure 5.16: *First-column*: original images. *Top-right*: result of TGC implementation. *Bottom-right*: result of ML implementation

is 8.04%, comparatively similar to the base-line stixel model (BB). However, the accuracy rate is dropped significantly from frame number 90-150. In these frames outgoing and incoming lanes started to merge, and there is no guardrail installed plus the open-road scenarios occurred again at the end of the sequences. This scenario enable TGC to outperform ML in these frames

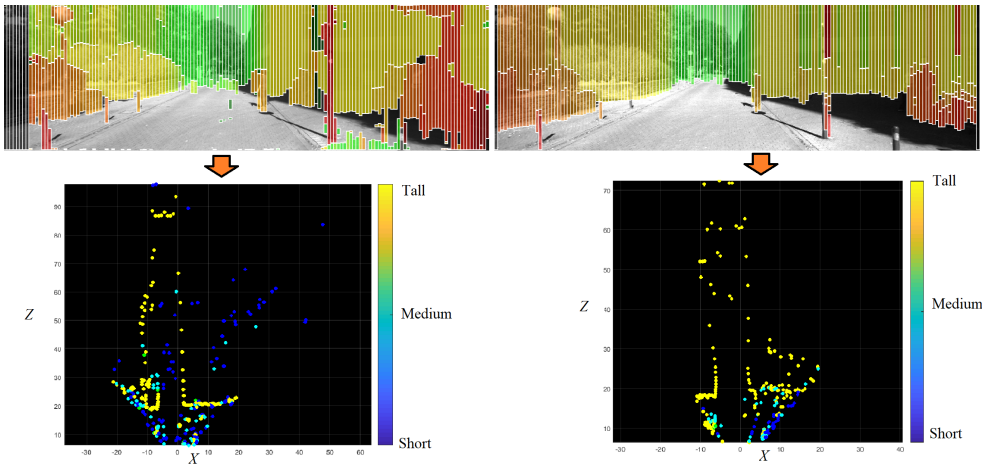

Figure 5.17: Multi-layer stixels (*Top*) and corresponding inverse perspective mapping (*Bottom*). *Left*: Trinocular graph-cut approach. *Right*: Monocular + LiDAR approach

Table 5.4: LiDAR-based quantitative evaluation based on mean distance error [cm] of stixels using KITTI dataset (binocular and monocular configuration).

Sequence	Single layer stixels				Multi-layer stixels			
	Binocular		Monocular		Binocular		Monocular	
	μ	σ	μ	σ	μ	σ	μ	σ
A	28.2	1.25	24.2	1.25	24.0	1.25	23.9	1.35
B	26.0	1.95	26.1	1.94	25.8	1.93	23.8	1.76
C	28.0	1.31	21.2	1.15	29.8	1.14	19.6	2.95

particularly. In campus category (D), the gap is very obvious between ML and the other tested methods. The main strength of ML is the ability to estimate road surface more accurately than others using plane detection.

In both city categories, the ML method shows a higher number of false-positive errors and high-error rate. This is due missing LiDAR points especially when objects become closer to the sensors and thus the uncertainty will be higher(see Fig. 5.8). Through the observation of results, ML method demonstrates an improved robustness across various categories (see Fig. 5.16). Low texture and robust road surface detection are in particular cases where ML method is more robust for multi-layer stixel segmentation. Taking into account the run-time profile across different camera configurations. The ML method outperforms binocular and trinocular vision across all categories (see Table 5.5). The processing time was highly reduced and in some sequences ML maintain the accuracy as well. The binocular multi-layer stixels used for run-time profiling is reported by BB.

Apart from the above mentioned measures, the number of segmented stixels are still low when using the ML method (see Table 5.3) compared to TGC. Based on the experiment analysis, the class segmentation (separating objects from ground) is performed successfully, but the sky class is always missing. This is due to the fact that LiDAR range does not cover such area.

Furthermore, the small stixels positioned above the horizon line are merged with other neighbouring stixels and that was clear when projecting these stixels to inverse perspective mapping (see Fig. 5.17). In this figure, we can

see a few objects' height are spanning and hiding a little details above the horizon line.

To sum-up, we provide more extensive analysis by comparing single-layer and multi-layer stixels using monocular and binocular vision. We plot errors calculated by mean distance differences for three categories A, B, and C as shown in Fig. 5.18 for road and residential data. The number of errors is highly reduced when using the proposed monocular stixel approach (see Table 5.4). Figure 5.18, for example, illustrates the accuracy of the proposed monocular plus LiDAR method (multi-layer and single-layer) for challenging obstacle detection conditions.

Resulting stixels, using monocular+LiDAR multi-layer, are more accurate than the original binocular ones. By visual evaluation, the original method has some limitations in identifying road surfaces and objects independently in the disparity map.

This problem occurred several times in tested categories. Thus, the LiDAR points play an essential role in providing an accurate disparity map, and their use also minimises the processing time required to generate the disparity map (see Table 5.5). This indicates an optimised balance between disparity-map accuracy and processing time in terms of stixel estimation. The limited number of LiDAR points acquired by the sensor define a limitation in our method.

As we can observe from Fig. 5.18, the distance error in (category A) between proposed monocular and conventional binocular multi-layer was very close; this also occurred in open-road scenarios with shadows.

5.4 Summary

This chapter presented a robust model for multi-layer stixel segmentation. The stixels, constructed using TED-based disparities provided by trinocular vision, were found to represent a better accuracy compared to conventional binocular ones, especially when also using graph-cut ground-manifold approximation. The proposed method using trinocular multi-layer stixels demonstrates the significance of confidence maps, which can vote for valid and consistent disparity values. We also provided a monocular plus LiDAR ap-

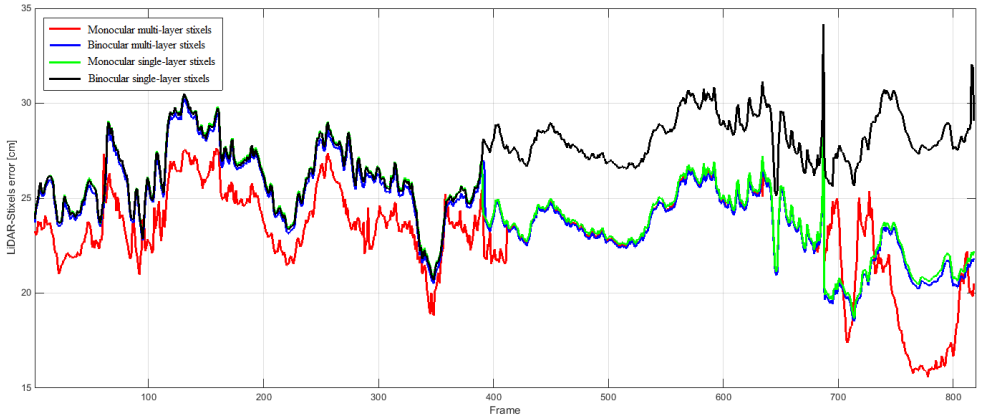

Figure 5.18: Error rates illustrate the mean of LiDAR-stixel distance error (in cm) for the four approaches. The frame ranges [1-390], [391-688], and [689-820] represent categories B, A and C, respectively.

Table 5.5: Run-time profiling for multi-layer stixels calculation on KITTI dataset.

Category	Monocular (ML)	Binocular (BB)	Trinocular (TGC)
A	23.3s	27.3s	27.1s
B	22.6s	26.9s	26.7s
C	22.7s	25.7s	26.6s
D	21.5s	24.1s	23.5s
E	21.6s	23.4s	23.1s
F	22.4s	25.3s	24.4s

proach to feed the multi-layer stixels with robust and accurate dense depth map. Experimental analysis results were obtained based on 1,501 frames, including various traffic scenes and conditions such as campus, road, city, and residential data from KITTI. An important advantage of our work is to demonstrate an improvement towards multi-layer stixel with accurate stixel representation and reduce the processing time required (with maintaining accuracy).

Chapter 6

Conclusions and Future Work

The application of driver assisting or driver-less systems are numerous with different techniques applied for traffic safety. Looking into the vision based application of driver assistance, there are several methods applied to represent data more efficiently and accurately. A 3D representation model called "stixel world" initiated by the automotive industry is used to leverage disparity values to robust and more understandable classes. Motivated by the unsolved issues represented by ground plane models, height of stixels, challenging scenarios in multi-layer stixel, and lack of stixel evaluation measurements, this thesis contributes to development of a comprehensive model that can be used in the development of real-time advance vision based driver assistance system. Throughout the previous chapters we provided a successful modification and improvement towards more accurate and real-time stixels detection using a sensor fusion technique.

This chapter reviews and summarises the main findings of this thesis and provides directions for future work in stixel model.

6.1 Conclusions

Within the scope identified in the thesis we presented a versatile framework that can improve the "stixel world" model. We believe that "stixel world" is still a young scheme that introduced less than 10 years ago. There are some limitations identified and worked on to improve this model. In *chapter one* we focused on related work in this domain which motivate us to carry out this work. We found that the issue of annotated labels of stixel ground truth can form a main obstacle towards evaluation and training purposes. It

looks the annotation is a laborious task and it suffers from lack of accurate statistical measurements due to confusion between corridor and free-space. Moreover, the base-line approach is cascading occupancy grid and ground plane detection and for this purpose we noticed there are numerous scenarios available where road surface detection task was challenging due to bad weather or road geometry. We also noticed unsolved problems in determining the height of stixel which was missing in current related work. Furthermore, due to noisy disparity map obtained from stereo matchers this lead us to think about integrating a confidence map or fusing sensor data. All of the mentioned issues motivated us to further investigate and explore the current context of stixel world based on related work. Although the construction of "stixel world" was not an easy mission due to industry related model and there were limited or none implementation of such a model at the beginning of this journey.

The related work and basic theory was introduced in *chapter two* which discuss works on road surface and stixel extraction, both considered to be crucial steps towards stixel calculation. The findings from this chapter are summarised by seeing stixel model are focused on certain type of camera configuration and road manifold. To be specific, binocular vision and plane detection are widely used to estimate stixels, yet some of traffic scenes are non-planer and challenging and this reflects the quality of disparity signal. Hence, plans towards improving the quality of disparity signal by using different sensors and empowering the ground manifold information. We also found "stixel world" applications are increasing and on high demand and thus, a successful improvement for this model will advance these applications.

In this context, *chapter three* studies the first aspect of stixel model represented by improving ground manifold. We noticed many challenging traffic scenes that require an adaptive ground manifold model to produce base-points. Also, we expect ground surface are not planner all the times so we have proposed to implement polynomial curve detection model in y-disparity map to detect piecewise linear curve. During the experiments we noticed such a model have successfully outperforms other selected model when the ground manifold was not plane (as found in 6D vision dataset).

On the other hand, challenging weather conditions have been studied carefully to see a solution to incomplete piecewise linear curve. Sometimes this scenarios happened as y-disparity shows noisy and missing data due to stereo pair mismatch (i.e. because of wind shield wiper). Thus, an efficient and low-cost architecture approach is proposed for ensuring the monotonicity in the detected curve under challenging weather conditions. The "graph-cut" is the proposed novel method to be applied in y-disparity map and harness smoothness term to ensure the monotonicity in the detected curve. Also, we extended the proposed models (polynomial and graph-cut) models for a trinocular configuration which yields obvious and robust improvements compared to binocular configuration.

As a result, the number of true-positives is large when the graph-cut model is used as a minimisation method for calculating a y-disparity cut. Results also indicate that the number of generated stixels highly increases when using trinocular line fitting for ROAD sequences, and binocular poly-fitting for CITY sequences; finally, trinocular graph-cut proved to be the best alternative on RESIDENTIAL sequences. Having especially challenging scenes in mind, altogether we recommend the trinocular graph-cut approach.

Furthermore, an accurate and sufficient identification of stixels' height is necessary to give a clue of identified objects. In *chapter four* we addressed this gap since it was not much covered in related work. In this regard we proposed two methods dedicated for this purpose one is using colour information represented by saliency map and monocular plus LiDAR. The fused map using saliency and membership maps show sufficient information about the detected top-points of stixels. The monocular plus LiDAR data demonstrate a good alternative in long-run sequences. Thus, the monocular plus LiDAR can be used a reliable real-time model for stixels detection. Experimental results have supported the idea of this improvement and compared to Velodyne HD LiDAR. For a comparison with CNN-based methods, annotated ground truth would be required to illustrate the exact location of each stixel.

Finally, improved multi-layer stixel segmentation methods are proposed in *chapter five* to leverage stixel segments. The stixels, built using TED-based disparities provided by trinocular vision, were found to represent a better ac-

curacy compared to conventional binocular ones, especially when also using graph-cut ground-manifold approximation. The second model presented in this chapter is using LiDAR data as an input for multi-layer stixels with robust and accurate dense depth map. Experimental analysis results were obtained based on a large datasets include scenes recorded in CAMPUS, ROAD, CITY, AND RESIDENTIAL environments. A significant advantage of our work is to present an improvement towards multi-layer stixel with accurate stixel representation at the same time reducing the processing time required.

6.2 Future work

The current theme provided for stixels in this thesis is mainly focused on vision-based driver assistance systems. However, there are many other applications where stixels can be applied such as marine navigation. Hence, a direction for future work can take several aspects based on the required application.

As the direct implementation of deep learning approach is infeasible due to lack of annotated ground truth, we recommend to apply such a learning method in semantic segmentation where classes provided from semantic can verify the geometric stixels with using additional sensors. Such a verification might be helpful for multi-layer stixel where these classes can be directly linked to multiple segments. As we noticed in most experiments visual odometry can support more accurate stixels because it can determine the ego motion and this would add more information about the reconstructed scene which greatly improve stixels position.

Appendix A

Appendix A: Comparative Study on Free-space Detection

A.1 Overview

Autonomous on-road vehicles or vision-based driver assistance benefit from free-space analysis. This section evaluates the accuracy of free-space detection in stereo and monocular vision on KITTI benchmark data. Such an evaluation of low-level computer vision algorithms is, for example, also necessary as free-space analysis is recently becoming an important module for designing vehicle test-beds. The content of this section is defined by comparing a designed monocular algorithm with a selected binocular algorithm on long sequences of images, taken for different road profiles and lighting conditions.

Furthermore, this section extends the detection of a lower envelop in a y-disparity image, usually done by estimating a straight line, by using polynomial curve fitting. To satisfy these needs, we compare the accuracy of free-space estimation algorithms using either monocular or binocular (i.e. stereo) vision using available ground truth. See Fig. A.1.

A.2 Free-space based on Binocular and Monocular Vision

Our system for free-space detection using monocular vision starts with cropping a recorded frame into a defined ROI (region of interest) for faster calculation, as shown in Fig. A.1.

For edge detection we apply the simple Sobel operator due to its "unbiased" definition. See Fig. A.1. Straight lines are now detected by an applica-

tion of an optimised Hough transform. The transform is applied recursively, using optimised (Otsu algorithm) threshold values until a predefined number of lines is found, or the threshold reaches its minimum. Finally, that "dominant" pair of lines with the best correspondence in angular directions is selected for specifying road contours (i.e. the free-space) in this monocular vision approach. This defines a novel but computational simple method in the wide spectrum of considered lane detection approaches [93]. See Fig. A.1.

Our system for free-space detection based on binocular vision uses polynomial curve fitting for identifying the lower envelop in y-disparity space, thus modifying the considered 3rd-order B-spline approximation studied in [91]. Stereo-vision traffic images with multiple marked, or unmarked lanes are available on the KITTI website [35], with a resolution of 1242×375 pixels. We use those for testing. See Fig. A.2 for an example.

Semi-global matching is a commonly used stereo matching algorithm for driver assistance systems. Our method for stereo-reconstruction uses *rapid semi-global matching* on a CPU (rSGM) [94]. The strength behind rSGM is that it implements low- and high-level parallelism without the need for explicit synchronisation.

A dense disparity map can be computed by matching all pixels in one image of a stereo pair with their correspondences in the other image; see Fig. A.2. After the disparity map is computed, we have to compute the y-disparity space. The corresponding y-disparity map is computed by accumulating the pixels with the same disparity value on a horizontal row of the disparity map.

Figure A.1: Comparison of free-space detection results. *Left*: Original image. *Middle*: Free-space using monocular detection, as estimated in this paper. *Right*: Free-space using binocular vision, as used in this paper. Green is used for the monocular, and purple for the binocular vision case.

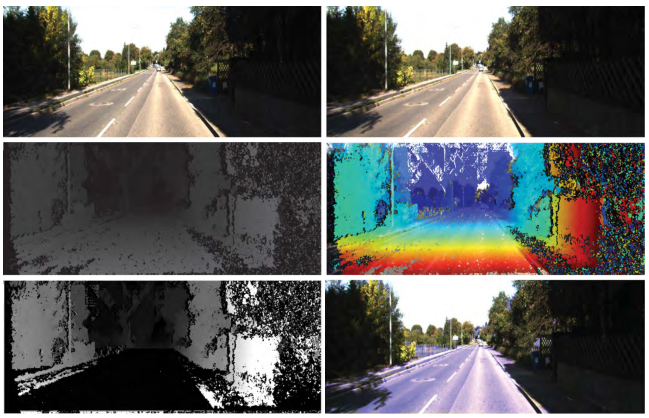

Figure A.2: Free-space detection in binocular vision. *Top row*: Stereo pair of the KITTI dataset (urban, multiple-lanes, marked). *Middle left*: Disparity map using rSGM, visualized visualised with a gray-level key. *Middle right*: Disparity map visualised with a colour key. *Bottom left*: Free-space (black ground plane) approximated based on disparity. *Bottom right*: Free-space visualised in original scene.

The y-disparity space is a row-based matrix which stores the disparity values for every column from the disparity map.

A.3 Results

In this section, we evaluate the accuracy of free-space detection using either the monocular or binocular (common ground-plane, and polynomial) approximated ground manifold instead. In both cases we use the same stereo matcher. Our local processing platform is a standard PC with an Intel Core i7-6700 Processor, 8M Cache, and 32 GB of RAM.

Both methods were tested using a dataset of 184 stereo pairs from KITTI [31]. For selecting our dataset, we aimed at having a wide diversity of challenging traffic situations including different lighting conditions, different road views, shades, and colourings.

We adopt the pixel-based measurements as employed for the KITTI dataset [31] with the proposed and implemented four error measures: precision, recall,

accuracy, and the F-measure (the harmonic mean of recall and precision) calculated for the *birds-eye view* (BEV) projection of a given image; see Fig. A.4, A.5. This can be implemented by calculate the followings:

$$\text{Precision} = \frac{TP}{TP + FP} \tag{A.1}$$

$$\text{Recall} = \frac{TP}{TP + FN} \tag{A.2}$$

$$\text{F} - \text{measure} = \frac{2(\text{Precision} \cdot \text{Recall})}{\text{Precision} + \text{Recall}} \tag{A.3}$$

$$\text{Accuracy} = \frac{TP + TN}{TP + FP + TN + FN} \tag{A.4}$$

The definition of these criteria for free-space detection is also summarised in Table A.1.

Two sets of comparative results are presented in Fig. A.4, A.5. As also shown in Fig. A.3, and certainly not surprising, the free-space based on monocular vision failed to separate obstacles clearly from the road surface, and it deteriorates when lane marks disappear. The performance of the monocular free-space is likely to degrade on unmarked roads. Compared to that, the

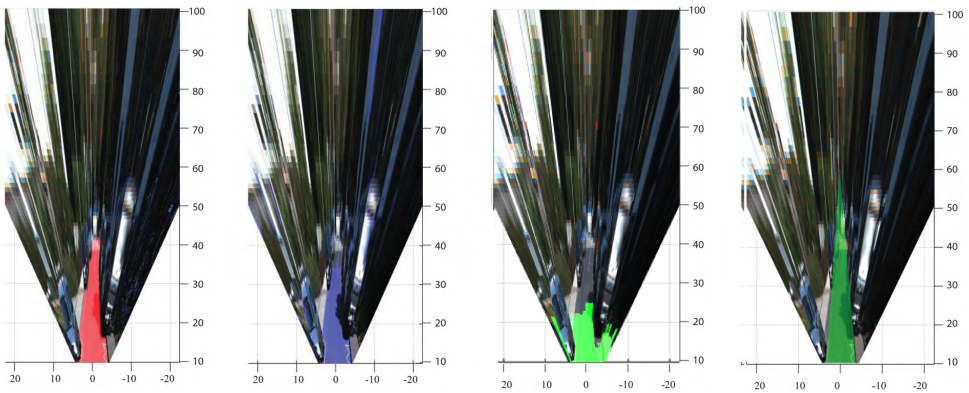

Figure A.3: Classification results for challenging UU road area image using bird's-eye view. *First-column*: Ground truth image. *Second-column*: Polynomial based free-space. *Third-column*: Ground plane free-space. *Fourth-column*: Monocular free-space.

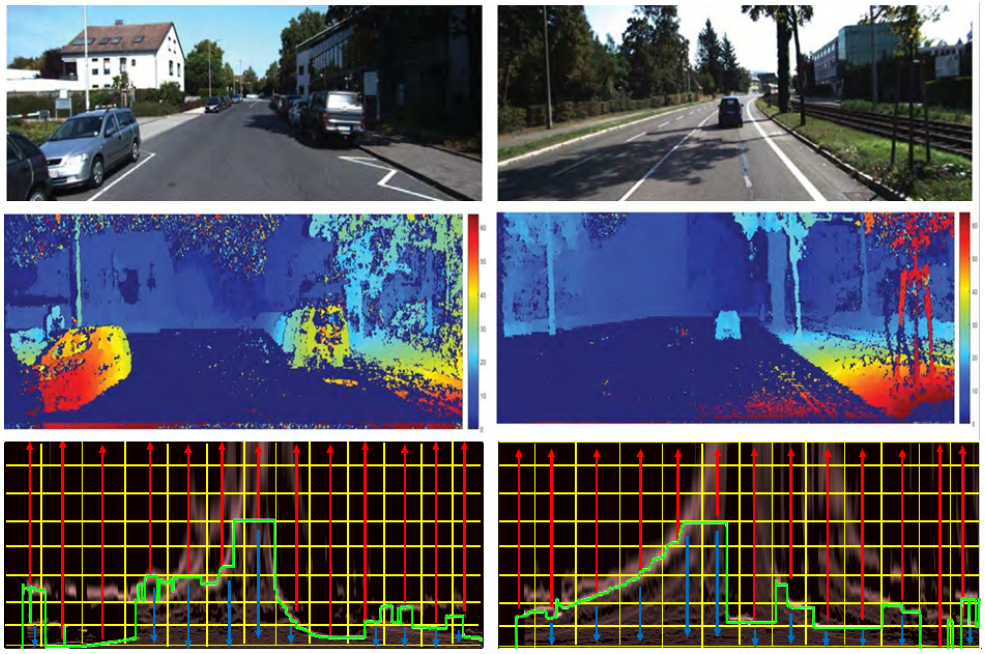

Figure A.4: Ground disparity and occupancy grid results on KITTI datasets. *First row*: An original image. *Second row*: visualisation of polynomial ground manifold. *Third row*: visualisation of occupancy grid after background subtraction.

Table A.1: Definition of criterion

		Ground truth	
		Occupied space	Free-space
Result	Occupied space	*TN*	*FN*
	Free-space	*FP*	*TP*

free-space based on polynomial improves in general. Although the process was more complex to provide the free-space, the stereo-vision-based methods clearly outperform monocular free-space. See Table A.2.

The stereo-vision-based (poly and plane) free-space still suffers from some noise especially for shady road conditions which lead to false detections; this is also visible in the example shown in Fig. A.5. Disparity-based free space

is sensitive to stereo matching errors within low-textured road surfaces or low-contrast obstacles. Under such more difficult circumstances for stereo vision, the performance of monocular free-space was actually very close to that of stereo-vision-based free-space. See Table A.3.

Monocular free-space generates in general many false negatives, espe-

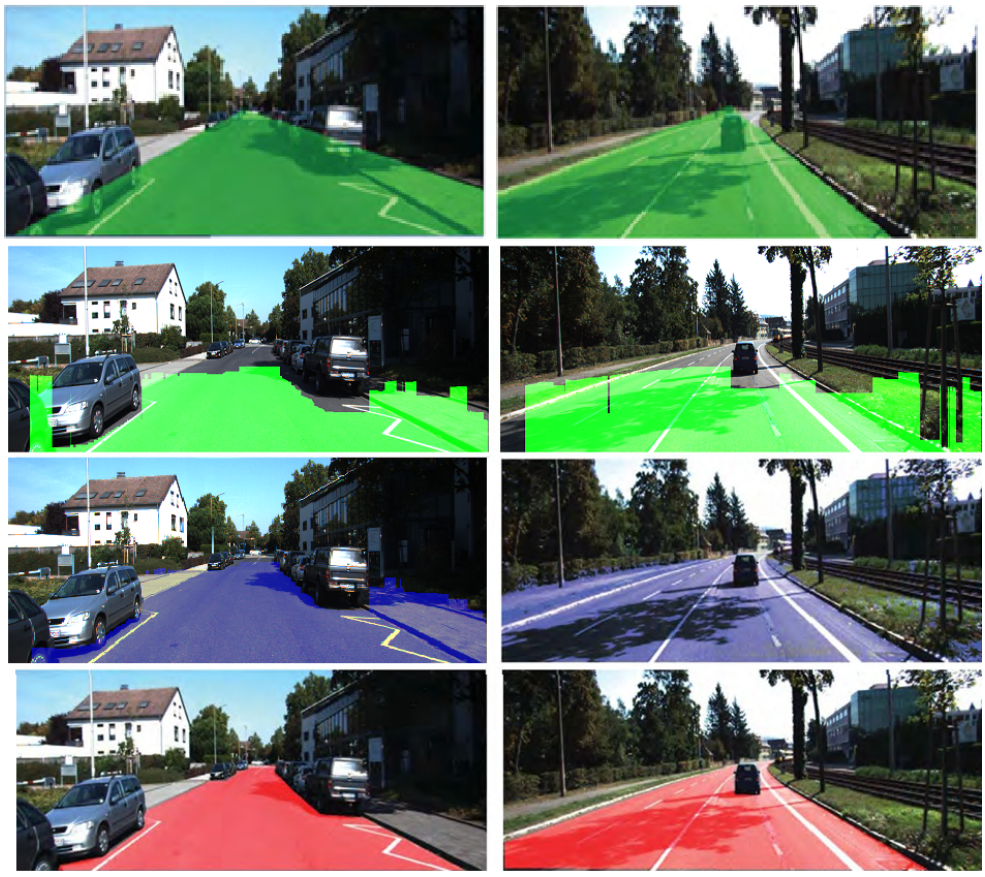

Figure A.5: Qualitative results of free-space on KITTI datasets. *First row*: Monocular free-space detection for challenging UU, and UMM road. *Second row*: Ground plane for challenging UU, and UMM road. *Third row*: Polynomial ground manifold for challenging UU, and UMM road. *Fourth row*: KITTI-provided ground truth for challenging UU, and UMM road.

Table A.2: Results for urban unmarked lanes

	Accuracy	Recall	F-measure	Precision
Plane-based	0.68	0.73	0.31	0.43
Polynomial-based	0.76	0.85	0.47	0.61
Monocular free-space	0.65	0.65	0.49	0.39

cially on unmarked roads, as shown in Table A.2, with a reduced difference in precision between both approaches. Having in mind that the free-space is the navigable area for vehicles, accuracy is the more important factor for driving safety. As expected, stereo vision can guarantee safety better than monocular vision (for the considered methods), but monocular vision also provides solutions which might be sufficient for a given context.

On the other hand, the use of a ground plane results in a lower detection rate and a higher rate of false positives in urban unmarked areas due to many obstacles, while the polynomial fitting method outperforms with a smaller rate of false alarms. Under such more difficult circumstances, containing strong noise, the method can consequently also lead to false alarms. For urban marked multiple-lane situations, the polynomial fitting method shows relatively close behaviour to the plane-based detection. See Table A.3.

We noticed that the border line of the ground plane may occasionally be far away from the object, and this leads to an "early" detection of stixels although they are not apparent as obstacles in the occupancy grid. False rendering also occurs because of digitisation errors when mapping measurements into grid or voxel spaces. Furthermore, as experiments show, the occupancy-grid mapping required more computation time for triangulation or projection, estimated in average around 2.58 s per frame. The use of y-disparity requires less computation time which was on average around 1.6 s

Table A.3: Result for multiple marked lanes

	Accuracy	Recall	F-measure	Precision
Plane-based	0.66	0.52	0.41	0.34
Polynomial-based	0.69	0.70	0.52	0.41
Monocular free-space	0.65	0.68	0.40	0.41

per frame. Having in mind that the ground plane is the navigable area for vehicles, accuracy is the most important factor for safe driving. As shown, polynomial ground-manifold fitting can lead to a more promising safety than just considering a plane (i.e. occupancy grid mapping) for the considered methods.

Bibliography

Co-Authored References

[1] Saleem, N. H. , Chien, H.-J.,Rezaei, M. and Klette, R.: Effects of ground manifold modelling on the accuracy of stixel calculations. In *IEEE Transaction Intelligent Transportation System*, DOI: 10.1109/TITS.2018.2879429, 2018.

[2] Saleem, N. H., Griffin, A., and Klette, R.: Monocular stixels: A LIDAR-guided approach. In Proc. *Int. Conf. Image Vision Computing New Zealand (IVCNZ)*, 2018. Acceptance rate: 56% (oral)

[3] Saleem, N. H., F. Huang, and Klette, R.: Use of a confidence map towards improved multi-layer stixel segmentation. In Proc. *IEEE Conf. Advanced Video and Signal-based Surveillance*, 2018 Acceptance rate: 33% (poster)

[4] Saleem, N. H.,Huang, F., and Klette, R.: Data fusion for efficient height calculation of stixels. In Proc. *IEEE Conf. Control, Automation, Robotics*, DOI: 10.1109/ICCAR.2018.8384699, 2018. Acceptance rate: 40% (oral - best presentation award)

[5] Saleem, N. H., Chien, H.-J., and Klette, R.: Stixel optimization: Representing challenging on-road scenes. In Proc. *Int. Conf. Image Vision Computing New Zealand (IVCNZ)*, DOI: 10.1109/IVCNZ.2017.8402446, 2017. Acceptance:56% (keynote talk).

[6] Saleem, N. H., Rezaei, M., and Klette, R.: Extending the stixel world using polynomial ground manifold approximation. In Proc. *IEEE Conf. Mechatronics Machine Vision Practice*, 526–531, 2017. Acceptance rate: 40% (oral)

[7] Saleem, N. H., Chien, H.-J., Rezaei, M. , and Klette, R.: Improved stixel estimation based on transitivity analysis in disparity space. In *Computer Analysis Images Patterns*. LNCS 10424, 28–40, 2017. Acceptance rate: 36% (oral - best student paper award)

[8] Saleem, N. H. and Klette, R.: Accuracy of free-space detection for stereo versus monocular vision. In Proc. Image Vision Computing New Zealand, 48–53, 2016. Acceptance rate: 59% (poster)

Non-Authored References

[9] Achanta, R., Hemami, S., Estrada, F., and Susstrunk, S.: Frequency-tuned salient map region detection. In Proc. *Int. Conf. Computer Vision Pattern Recognition*, 1597-1604, 2009.

[10] Anders, J. , Mefenza, M., Bobda, C., Yonga, F., Aklah, Z., and Gunn, K.: A hardware/software prototyping system for driving assistance investigations. In *J. Real-Time Image Processing* 11(3), 559–569, 2016.

[11] Badino, H., Franke, U., and Pfeiffer, D.: The stixel world - A compact medium level representation of the 3D-world. In Proc. *German Conf. on Pattern Recognition* LNCS, 5748: 51–60, 2009.

[12] Badino, H., Franke, U., and Mester, R.: Free space computation using stochastic occupancy grids and dynamic programming. In Proc. *Workshop on Dynamical Vision, ICCV*, 2007.

[13] Bellman, R.: The theory of Dynamic Programming. In *Bulletin of the American Mathematical Society*, 503–515, 1954

[14] Benenson, R., Mathias, M., Timofte, R., and Van Gool, L.: Fast stixels estimation for fast pedestrian detection. In Proc. *Europ. Conf. Computer Vision*, 11–20, 2012.

[15] Benenson, R., Timofte, R., and Van Gool, L.: Stixels estimation without depth map computation. In Proc. *Int. Conf. Computer Vision Workshops* , 2010–2017, 2011.

[16] Brandao, M., Ferreira, R., Hashimoto, K., Takanishi, A., and Santos-Victor, J.: On stereo confidence measures for global methods, evaluation, new model and integration into occupancy grids. In Proc. *Pattern Analysis Machine Intelligence*, 116–128, 2016.

[17] Brox, T., Rosenhahn, B., and Weickert, J.: Three-dimensional shape knowledge for joint image segmentation and pose estimation. In Proc.*German Association for Pattern Recognition* , 109–116, 2005.

[18] Chien, H.-J., Geng, H., and Klette, R.: Improved visual odometry based on transitivity error in disparity space: A third-eye approach. In Proc. *Image and Vision Computing New Zealand*, 72–77, 2014.

[19] Cordts, M., Rehfeld, T., Schneider, L., Pfeiffer, D., Enzweiler, M., Roth, S.,Pollefeys M. and Franke, U.: The stixel world: A Medium-level Representation of Traffic Scenes. In *Image and Vision Computing*, 68:40–52, 2017.

[20] Cordts, M., Schneider, L., Enzweiler, M., Franke, U., and Roth, S.: Object-level priors for stixel generation. In Proc. *German Conf. on Pattern Recognition*, 172–183, 2014.

[21] Danescu, R., Oniga, F., Nedevschi, S.: Particle grid tracking system for stereovision based environment perception. In Proc.*IEEE Intelligent Vehicles Symp.*, 2010.

[22] Davies, E.: Machine Vision: Theory, Algorithms, Practicalities. In *Morgan Kaufmann*, 2004.

[23] Dhiman, A., Chien, H.-J., and Klette, R.: Road surface distress detection in disparity space. In Proc. *Int. Conf. Image Vision Computing New Zealand (IVCNZ)*, DOI: 10.1109/IVCNZ.2017.8402459, 2017.

[24] Dolson, J. , Baek, J., Plagemann, C. , and Thrun, S.: Upsampling range data in dynamic environments. In Proc. *Computer Vision Pattern Recognition*, 2010.

[25] Enzweiler, M., Hummel, M., Pfeiffer, D., and Franke, U.: Efficient stixel-based object recognition. In Proc.*Intelligent Vehicles Symposium*, 1066–1071, 2012.

[26] Enzweiler, M. and Gavrila, D. M.: Monocular pedestrian detection: Survey and experiments. *IEEE Pattern Analysis and Machine Intelligence* 31(12):2179–2195, 2009.

[27] Erbs, F., Schwarz, B., and Franke, U: Stixmentation-Probabilistic Stixel based Traffic Scene Labeling. In Proc. *British Machine Vision Conference* , 1–12, 2012.

[28] Farabet, C., Couprie, C., Najman, L., and Lecun, Y.: Learning hierarchical features for scene labeling. In *IEEE Trans. Pattern Analysis Machine Intelligence*, 35(8), 1915–1929, 2013.

[29] Franke, U., Rabe, C., Gehrig, S.K.: Avoidance based on Space-Time Image Analysis. In *J. Information Technology*, 49(1), 25–32, 2007.

[30] Franke, U., Pfeiffer, D., Rabe, C., Knoeppel, C., Enzweiler, M., Stein, F. , and Herrtwich, R.G.: Making Bertha See. In Proc. *International Conference on Computer Vision Workshops* 214–221, 2013.

[31] Fritsch, J., Kuehnl, T., and Geiger, A.: A New Performance Measure and Evaluation Benchmark for Road Detection Algorithms. In Proc. *International Conference on Intelligent Transportation Systems*, 2013.

[32] Gavrila, D. M., and Munder, S.: Multi-Cue pedestrian detection and tracking from a moving vehicle. In *Int. Journal Computer Vision*, 73(1):41–59, 2017.

[33] Gehrig, S. , Eberli, F., and Meyer, T.: A real-time low-power stereo vision engine using semi-global matching. In Proc. *Int. Conf. Computer Vision*, 134-143, 2009.

[34] Geiger, A., Lenz, P. , Stiller, C., and Urtasun, R.: Vision meets robotics: The KITTI dataset. *Int. J. Robotics Research*, (32)11, 1231–1237, 2013.

[35] Geiger, A., Lenz, P., and Urtasun, R.: Are we ready for autonomous driving? The KITTI vision benchmark suite. In Proc. *Computer Vision Pattern Recognition*, 3354–3361, 2012.

[36] Gindele, T., Brechtel, S., Schröder, J., Dillmann, R.: Bayesian occupancy grid filter for dynamic environments using prior map knowledge. In Proc.*IEEE Intelligent Vehicles Symp.*, 2009.

[37] Gribonval, R.: Should Penalized Least Squares Regression be Interpreted as Maximum A Posteriori Estimation? In *IEEE Transactions on Signal Processing*, 2405–2410, 2011.

[38] Guan, S. and Klette, R.: Belief-propagation on edge images for stereo analysis of image sequences. In Proc. *Int. Workshop Robot Vision*, LNCS 4931: 291–302. 2008.

[39] Haeusler. R., Nair, R., and Köndermann, D.: Ensemble learning for confidence measures in stereo vision. In Proc. *Computer Vision Pattern Recognition*,305–312, 2013.

[40] Harms, H., Rehder, E., and Lauer, M. : Grid map based free-space estimation using stereovision. In Proc. *1st Workshop on Environment Perception for Automated On-road Vehicles - IEEE Intelligent Vehicles Symp.*, 2015.

[41] He, Z., Wu, T., Xiao, Z., and He, H.: Robust road detection from a single image using road shape prior. In Proc. *Int. Conf. Image Processing*, 2757–2761, 2013.

[42] He, Y., Wang, H., and Zhang, B.: Color-based road detection in urban traffic scenes. In Proc. *IEEE Trans. Intelligent Transportation Systems*, 309–318, https://doi.org/10.1109/TITS.2004.838221 , 2004.

[43] Hernandez, D., Espinosa, A., Moure, J., Vázquez, D., López, A.: GPU-accelerated real-time stixel computation. In Proc. *Conf. Applications Computer Vision*, 1054–1062, 2017.

[44] Hernandez-Juarez, D., Schneider, L., Espinosa, A., Vzquez, D., Lpez, A. M., Franke, U., Pollefys M., and Moure, J. C.: Slanted Stixels: Representing San Francisco's Steepest Streets. In Proc.*British Machine Vision Conference*, 1–12, 2017.

[45] Hirschmüller, H.: Stereo processing by semiglobal matching and mutual information. In *IEEE Trans. Pattern Analysis Machine Intelligence*, 30: 328–341, 2008.

[46] Hu, Z. and Uchimura K. U-v-disparity: An Efficient algorithm for stereovision based scene analysis. In Proc.*3D Digital Imaging and Modeling*, 48–54, 2005.

[47] Iloie, A., Giosan, I., and Nedevschi, S.: UV disparity based obstacle detection and pedestrian classification in urban traffic scenarios. In Proc. *Intelligent Computer Communication Processing*, 119–125, 2014.

[48] Jiang, R., Klette, R., Vaudrey, T., Wang, S.: Corridor detection and tracking for vision-based driver assistance system. In *Pattern Recognition and Artificial Intelligence*, 25(2), 253–272, 2011.

[49] Kaaniche, K., Demonceaux, C., and Vasseur, P.: Analysis of low-altitude aerial sequences for road traffic diagnosis using graph partitioning and Markov hierarchical models. In Proc. *Int. Multi-Conf. Systems Signals Devices*, 656–661, 2016.

[50] Klette, R.: Vision-based driver assistance. In Wiley Encyclopaedia Electrical Electronics Engineering. John Wiley & Sons, 1–15, 2015.

[51] Klette, R.: Concise Computer Vision. Springer, London, 2014.

[52] Keller, C., Dang, T., Fritz, H., Joos, A., Rabe, C., and Gavrila, D.: Active pedestrian safety by automatic braking and evasive steering. In Proc. *Intelligent Transportation Systems*, 1292–1304, 2011.

[53] Knuth, D., Larrabee, T. L., Roberts, P., M.: *Mathematical Writing*, Mathematical Association of America, ISBN: 088385063X, 1989.

[54] Kong, H., Audibert, J. Y., and Ponce, J.: General road detection from a single image. In *IEEE Trans. Image Processing* 19(8), 2211–2220, 2010.

[55] Labayrade, R., Aubert, D., and Tarel, J.: Real time obstacle detection in stereovision on non flat road geometry through v-disparity representation. In Proc. *IEEE Intelligent Vehicle Symp.*, 646–651, 2002.

[56] Lee, S., Suhur, J. K., Jung, H. G.: Improvement of Stixel Segmentation Using Additive Image Domain Features and Genetic Algorithm-based Optimization. In *Trans. Korean Society Automotive Engineers*, 565–574, 2015.

[57] Levi, D., Garnett, N., and Fetaya, E.: StixelNet: A deep convolutional network for obstacle detection and road segmentation. In Proc. *British Machine Vision Conference*, 1:12, 2015.

[58] Li, X., Flohr, F., Yang, Y.,Xiong, H., Braun, M., Pan, S., Li, K., Gavrila, D. M.: A New Benchmark for Vison-Based Cyclist Detection. In Proc.*IEEE Intelligent Vehicles Symp.*, 2016.

[59] Lim, B., Woo, T., Kim H.: Integration of Vehicle Detection and Distance Estimation using Stereo Vision for Real-Time AEB System. In Proc. *Vehicle Technology and Intelligent Transport Sys.*, 2017.

[60] Lu, K., Li, J., An, X., and He, H.: A hierarchical approach for road detection. In Proc. *Int. Conf. Robotics Automation*, 517–522, https://doi.org/10.1109/ICRA.2014.6906904, 2014.

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

[62] Miksik, O.: Rapid vanishing point estimation for general road detection. In Proc. *Int. Conf. Robotics Automation*, 4844–4849, https://doi.org/10.1109/ICRA.2012.6225206, 2012.

[63] Morales, N., Morell, A., Toledo, J., and Acosta, L.: Fast object motion estimation based on dynamic stixels. In *Sensors*, 16(8), 1182–1191, 2016.

[64] Morales, S., Vaudrey, T., and Klette, R.: Robustness evaluation of stereo algorithms on long stereo sequences. In Proc. IEEE Conf. *Intelligent Vehicles*, 347–352, 2009.

[65] Montani, C. and Scopigno, R.:Rendering volumetric data using STICKS representation scheme. In *ACM SIGGRAPH Computer Graphics*, 24(5): 87–93, 1990.

[66] Neumann, L., Vanholme, B., Gressmann, M., Bachmann, A.,Kahlke, L. , and Schule, F.: Free Space Detection: A Corner Stone of Automated Driving. In Proc. *Int. Conf. Intelligent Transportation Systems*, 1280–1285, 2015.

[67] Oh, S. I., Kang, H. B.: Object detection and classification by decision-level fusion for intelligent vehicle systems. In *Sensors*, 17(1), 207, 2017.

[68] Ohta, Y., and Kanade, T.: Stereo by intra- and inter-scanline search using dynamic programming. In *IEEE Transactions on Pattern Analysis and Machine Intelligence* , 7(1):139–154, 1985.

[69] Onoguchi, K., N. Takeda, and M. Watanabe: Obstacle location estimation using planar projection stereopsis method. In *Systems Computers Japan* , 32(14): 67–76, 2001.

[70] Otsu, N.: A threshold selection method from gray-level histograms. In *IEEE Trans. Systems Man Cybernetics*,9:62–66, 1979.

[71] Perrollaz, M., Yoder, J.D., Negre, A., Spalanzani, A., Laugier, C.: A visibility-based approach for occupancy grid computation in disparity space. In Proc. *IEEE Transaction Intelligent Transportation System*, 13:1383–1393, 2012.

[72] Pfeiffer, D., Gehrig, S., and Schneider, N.: Exploiting the power of stereo confidences. In Proc. *Conf. Computer Vision Pattern Recognition*, 297–304, 2013.

[73] Pfeiffer, D.: The Stixel World. Doctoral Thesis, Humboldt Universität Berlin, 2014.

[74] Pfeiffer, D. and Franke, U.: Towards a global optimal multi-layer stixel representation of dense 3D data. In Proc. *British Machine Vision Conference*, 51–62, 2011.

[75] Pfeiffer, D. and Franke, U.: Modeling Dynamic 3D Environments by Means of The Stixel World. In *Intelligent Transportation System*, 24–36, 2010.

[76] Pfeiffer, D. and Franke, U.: Efficient representation of traffic scenes by means of dynamic stixels. In Proc. *Intelligent Vehicles Symp.*, 217–224, 2010.

[77] Pong, H. and Cham, T.: Object detection using a cascade of 3D models. In Proc.*Computer Vision and Pattern Recognition*, 808–811, 2006.

[78] Posada, L., Narayanan, K. , Hoffmann, F. , Bertram, T.: Detecting free-space and obstacles in omnidirectional images. In Proc. *Intelligent Robotics and Applications*, 610–619, 2011.

[79] Premebida, C., Garrote, L., Asvadi, A., Ribeiro, A. P., and Nunes, U.: High-resolution LIDAR-based depth mapping using bilateral filter. In Proc. *Computer Vision Pattern Recognition*, 2016.

[80] Rahman, A. , Verma, B., and Stockwell, D.: An hierarchical approach towards road image segmentation. In Proc. *Int. Joint Conf. Neural Networks*, 1–8. https://doi.org/10.1109/IJCNN.2012.6252403, 2012.

[81] Rasmussen, C.: Combining laser range, color, and texture cues for autonomous road following. In *IEEE Int. Conf. Robotics and Automation* , https://doi.org/10.1109/ROBOT.2002.1014439, 2002.

[82] Rezaei, M., and Klette, R.: Computer vision for driver assistance. In Computational Imaging and Vision, 1–18, 2017.

[83] Sanberg, W. P., Dubbelman, G., and deWith, P. H. N.: Color-based free-space segmentation using online disparity-supervised learning. In Proc. *Intelligent Transportation Systems*, 906–912, 2015.

[84] Seo, J., Oh, C., and Sohn, K.: Segment-based free space estimation using plane normal vector in disparity space. In Proc. *Connected Vehicles Expo*, 144–149, 2015.

[85] Sivaraman, S. and Trivedi, M.: Looking at vehicles on the road: A Survey of vision-based vehicle detection, tracking, and behavior analysis. *IEEE Trans. Intelligent Transportation Systems*, 1773–1795, 2013.

[86] Schneider, L., Cordts, M., Rehfeld, T., Pfeiffer, D., Enzweiler, M., Franke, U., Pollefeys, M., and Roth, S.: Semantic stixels: Depth is not enough. In Proc. *Intelligent Vehicles Symp.*, 110–117, 2016.

[87] Scharwächter, T., and Franke, U.: Low-level fusion of color, texture and depth for robust road scene understanding. In Proc. *Intelligent Vehicles Symp.*, 599–604, 2015.

[88] Scharwächter, T., Enzweiler, M., Franke, U., and Roth S.: Stixmantics: A medium-level model for real-time semantic scene understanding. In Proc. *European Conference on Computer Vision*, LNCS, 8693 (5): 533–548, 2014.

[89] Scharwächter, T: Stixel-Based Target Existence Estimation under Adverse Conditions. In Proc. *German Conf. on Pattern Recognition*, Springer, Berlin, Heidelberg, 225–230, 2013.

[90] Scharwachter, T.: Stereo Scene Analysis under Adverse Conditions. Master Thesis, Aachen University, 2013.

[91] Schauwecker, K., Morales, S., Hermann, S. , and Klette, R.: A comparative study of stereo-matching algorithms for road-modeling in the presence of windscreen wipers. In Proc. *IEEE Intelligent Vehicles Symp.* 7–12, 2011.

[92] Scharstein, D. and Szeliski, R.: Middlebury online stereo evaluation. `http://vision.middlebury.edu/stereo`. 2002.

[93] Shin, B.-S., Xu, Z., and Klette, R.: Visual lane analysis and higher-order tasks: A concise review. In *Machine Vision Applications*, 25(6): 1519–1547, 2014.

[94] Spangenberg, R., Langner, T., Adfeldt, S., and Rojas, R.: Large scale semi-global matching on the CPU. In Proc. *IEEE Intelligent Vehicles Symp.*, 195–201, 2014.

[95] Suhur, J., and Jung, H.: Dense stereo-based robust vertical road profile estimation using Hough transform and dynamic programming. In *IEEE Trans. Intelligent Transportation Systems* 1528–1536, 2015.

[96] Suhr, J. and Jung, H.: Noise-resilient road surface and free space estimation using dense stereo. In Proc. *IEEE Intelligent Vehicles Symp.*, 1–5, https://doi.org/10.1109/IVS.2013.6629511 , 2013.

[97] Tao, J., and Klette, R. Tracking of 2D or 3D Irregular Movement by a Family of Unscented Kalman Filters. In *Inform. and Commun. Convergence Engineering*, 10:307–314, 2012.

[98] Thrun, S., and Bü, A.: Integrating grid-based and topological maps for mobile robot navigation. In Proc. *Nat. Conf. on Artificial Intelligence*, 2:944–950, 1996.

[99] Veksler, O.: Stereo correspondence by dynamic programming on a tree. In Proc. *Computer Vision and Pattern Recognition*, 384–390, 2005.

[100] Viterbi, A. J.: Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. In *IEEE Trans. Information Theory*, 13(2): 260–269, 1967.

[101] Wang, W., Wang, N., Wu, X., You, S., and Neumann, U.: Self-paced cross-modality transfer learning for efficient road segmentation. In Proc. *Int. Conf. Robotics and Automation*, 1394–1401, 2017.

[102] Wieszok, Z., and Aouf, N. and Kechagias-Stamatis, O., and Chermak, L.: Stixel based scene understanding for autonomous vehicles. In Proc. *International Conference on Networking, Sensing and Control*, 43–48, 2017.

[103] Wu, J., Cui, Z., Sheng, V. S., Zhao, P., Su, D., and Gong, S.: A comparative study of SIFT and its variants. In *Measurement Science Review*, 13(3), 122–131, 2013.

[104] Yang, Z., and Pun-Cheng, L.: Vehicle Detection in Intelligent Transportation Systems and its Applications Under Varying Environments: A Review. In *Image and Vision Computing*, 143–154, 2018.

[105] Ziegler, J., Bender, P., Schreiber, M., Lategahn, H., Strauss, T., Stiller, C., Dang T. et al.: Making Bertha Drive-An Autonomous Journey on a Historic Route. In *Intell. Transport. Syst. Mag.*, 6(2), 8–20, 2014.

[106] 6D Vision Ground Truth Stixel Dataset `http://www.6d-vision.com/aktuelle-forschung/stixel-world`

[107] China Daily: China opens first test field for autonomous vehicles `http://www.chinadaily.com.cn/m/beijing/zhongguancun/2018-02/14/content_35754499.htm`, retrieved November 26, 2018.

[108] Edelstein, S.: Velodyne just cut the price of its most popular Lidar sensor in half. www.thedrive.com/tech/17297/, retrieved September 01, 2018.

[109] LEE, T. Why experts believe cheaper, better lidar is right around the corner https://arstechnica.com/cars/2018/01/driving-around-without-a-driver-lidar-technology-explained/, retrieved September 01, 2018.

[110] The KITTI Vision Benchmark Suite http://www.cvlibs.net/datasets/kitti/, 2015.

[111] The Northland Transport Technology Testbed. www.n3t.kiwi , 2016.

[112] PTV Compass Blog. Test field Autonomous Driving: In May the first vehicles will be rolling in Karlsruhe. http://compass.ptvgroup.com/2018/03/test-field-autonomous-driving\\-in-may-the-first-vehicles-will-be-rolling-in-karlsruhe/\\?lang=en, retrieved November 26, 2018.

[113] TOMTOM HD Map with Road DNA. https://www.tomtom.com/automotive/automotive-solutions/automated-driving/hd-map-roaddna/, retrieved July 02, 2016.

Index