TAMPEREEN TEKNILLINEN YLIOPISTO
TAMPERE UNIVERSITY OF TECHNOLOGY

SUNIL BHARAMGOUDAR

DETECTION OF FREE SPACE/OBSTACLES IN FRONT OF THE EGO CAR USING STEREO CAMERA IN URBAN SCENES

Master of Science thesis

Examiner: prof. Jose L. Martinez Lastra
Examiner and topic approved by the Faculty Council of the Faculty of Engineering Sciences
on 3rd Sept 2014

ABSTRACT

SUNIL BHARAMGOUDAR: Detection of Free Space/Obstacles in Front of the Ego Car Using (Stereo) Camera in Urban Scenes
Tampere University of technology
Master of Science Thesis, 78 pages
May 2016
Master's Degree Program in Machine Automation
Major: Factory Automation
Examiner: Professor Jose L. Martinez Lastra

Keywords: ADAS, image processing, ground surface detection

Transportation is one of the most important needs of a modern human being. It sustains (either directly or indirectly) the basic human needs such as food, education and his livelihood. Humans spend considerable amount of time in transit. Transportation is also not without its dangers. An estimated 1.25 million people died in the year 2013 in road traffic accidents[1]. With the advancements in electronics, smarter and faster sensors can help reduce some of these road traffic accidents. Detection of an obstacle is an essential aspect in avoiding collisions. The aim of this thesis report is to address the challenge of road surface detection. The thesis work begins with the implementation of the v-disparity road surface estimations and proposing certain variations that offer subtle advantages. Additionally the free space estimations through 3D occupancy grid maps (OGM) have also been implemented. A novel 'extended' u-disparity OGM is proposed that has certain advantages to the standard OGMs. All these road surface detection algorithms are evaluated with the training datasets prepared by Karlsruhe Institute of Technology.

[1] http://www.who.int/gho/road_safety/mortality/traffic_deaths_number/en/

PREFACE

First and foremost I would like to thank the DLR institute and Transportation Systems' department for giving me the opportunity to do internship and thesis in the field of driver assistance systems. I am extremely grateful to Mr. Paulin Pekezou Fouopi for supervising and helping me in every step of the thesis work including but not limited to doubts, implementations, programming and even career support. I would like to thank my family especially my brother who has supported me during this period. I am particularly thankful for the support from Ms. Anna Nykänen, she has prepared all the paper work to ease my inception into DLR for thesis work. I am also thankful to Prof. Lastra for agreeing to thesis work in Germany and for being its examiner.

Coventry, 23.05.2016

Sunil Bharamgoudar

CONTENTS

1. INTRODUCTION 1
1.1 Motivation 1
1.2 Approach 2
2. LITERATURE REVIEW AND STUDY OF ADAS TECHNOLOGY 5
2.1 Literature Review 5
2.2 ADAS and machine vision 7
2.3 ADAS: Anti-lock Braking System (ABS) 8
2.4 ADAS: Electronic Stability Control (ESC) 9
2.5 ADAS: Vision based 11
2.5.1 The scope 11
2.5.2 Environment perception 11
2.6 Image Acquisition 13
2.6.1 Pinhole camera model 13
2.6.2 Projective Geometry 15
2.6.3 Lens distortion 15
2.6.4 Camera calibration 16
2.7 Stereo computation 17
2.7.1 v-disparity (v-disp) 19
2.7.2 Occupancy Grid Map (OGM) 19
2.8 Image processing 19
2.8.1 Smoothing 19
2.8.2 Image segmentation 21
2.8.3 Line and Plane fitting 23
2.9 A theory of detections 24
2.10 v-disparity approach to ground surface estimation 25
2.10.1 Derivation of equation for road pixels in v-disparity - 27
2.10.2 Detection of road surface in v-disparity map 30
2.10.3 Refined v-disparity post obstacle elimination 36
2.10.4 Near vehicle warning function 40
2.11 Occupancy Grid Map (OGM) free space estimation approach 42
2.11.1 Math involved in generating OGM 43
2.11.2 Column-disparity map 45
2.11.3 Polar OGM 45
2.11.4 OGM Segmentation using Dynamic programming 46
2.11.5 Polar OGM vs u-disparity OGM 51
2.12 Least square plane fit for ground surface 53
3. EVALUATION OF THE GROUND SURFACE DETECTION ALGORITHMS 56
3.1 Dataset 56
3.2 ROC Curves 56

3.2.1 Basics 56
3.2.2 ROC curves for key algorithm parameters 59
3.2.3 ROC for v-disp algorithms 59
3.2.4 ROC for OGM algorithms 61
3.2.5 Comparison v-disp vs OGM algorithms 62
4. CONTRIBUTIONS TO GROUND PLANE DETECTION ALGORITHMS. 64
4.1 Confidence metrics for road surface detection 64
4.2 Novel Extended u-disparity 68
5. CONCLUSION & FUTURE WORK 69
5.1 Conclusion 69
5.2 Future work: 71
6. REFERENCES 73

LIST OF SYMBOLS AND ABBREVIATIONS

OpenCV	Open source Computer Vision
RADAR	Radio Detection and Ranging
LIDAR	Light Detection and Ranging
ABS	Anti-lock Braking System
ESC	Electronic Stability Control
LASER	Light Amplification by Stimulated Emission of Radiation
DLR	Deutsches Zentrum für Luft- und Raumfahrt
KIT	Karlsruhe Institute of Technology
KITTI	Karlsruhe Institute of Technology and Toyota Technical Institute
v-disp	V-disparity
u-disp	U-disparity
OGM	Occupancy Grid Map
TUT	Tampere University of Technology
URL	Uniform Resource Locator
B-splines	Bezier Splines
DAS	Driver Assistance Systems
ADAS	Advanced Driver Assistance Systems
TP	True Positive (pixels)
FP	False Positive (pixels)
TN	True Negative (pixels)
FN	False Negative (pixels)
ROC	Receiver Operating Curve
ECU	Electronic control unit
u	The column number/ the scalar along image x-axis
v	The row number/ the scalar along the image y-axis
d	The disparity value at a particular pixel (pixels)
f_s	The static friction between two surfaces
f_k	The kinetic friction between two surfaces
μ_s	Static friction coefficient
μ_k	Kinetic friction coefficient
n	Normal force from the surface supporting the object (N)
α/f	Focal length of the lens (m)
f_x	Focal length of the lens along image rows (pixels)
f_y	Focal length of the lens along image columns (pixels)
Z	The normal distance of the object from the camera lens (m)
(X,Y,Z)	The coordinates of object in real world (m, m, m)
(u,v)/(x,y)	The coordinates of the object seen in the image in columns and rows respectively (pixels, pixels)
(c_x , c_y)	Camera offset parameters in pixels along columns and rows respectively (pixels, pixels)
d/Δ	Stereo disparity (pixels)
θ	The camera tilt about the horizontal axis (radians)
σ	Standard deviation (pixels)
b	Distance between the stereo left and right cameras (m)
h	height of the camera above the ground frame (m)
$I_{v\Delta}$	v-disparity image

1. INTRODUCTION

Mankind has come a long way since discovering the wheel. In the beginning the carts and chariots were driven by ox and horses. Even then the transportation had a certain degree of autonomy in transit. Then came automobile, it was both a boon and a bane, the speed offered by vehicles is unquestionable but it also demanded utmost attention from the driver. Since then man has persevered diligently to improve the automobile and also the roads they tread on. The advent of electronics has swept the globe with a wave that has brought television, computers, cellphones to the shores far and wide. It is the fusion of electronics and automobiles that this thesis work is willing to bridge.

1.1 Motivation

With the ever increasing world human population, improving economies, personal transport is no longer a luxury. With each passing year, the same tarmac of road is being shared by increasing number of people. According to WHO approximately 1.2M people died in road accidents in 2010. Pedestrians and cyclists are most vulnerable in road accidents since they do not have the protective shell like the automotive do and face nearly full brunt of the collision impact. Advanced Driver Assistance Systems (ADAS) has tremendous potential in reducing the road accidents. Road scene perception is made through a multitude of sensors to reinforce robustness. This perception forms a basis on top of which safety critical functions are built to support, warn and even intervene to keep the traffic constituents safe. The introduction of Electronic Stability Control (ESC) in vehicles has brought about a notable decrease in road accidents [27]. At the Transportation Systems department of the DLR Institute, studies related to ADAS are being conducted on the Vehicle Simulators. These simulators have a cockpit similar to vehicles on the road. Synthetic road scene images are projected in front of the cockpit to simulate various driving scenarios to study the response of the drivers and their interactions with the ADAS technologies in development. Such studies are very important to assure that introduction of new safety features in vehicles work in harmony with the driver rather than causing discord. With the improvements in IT, smartphone and internet connectivity; access to quality entertainment is literally in the grasp of common man. This is both a boon and a bane. Entertainment in leisure is good, but on the road it is dangerous. A distracted person is unfit to walk the streets let alone drive a vehicle with immense momentum; where response time is measured in seconds. In Germany cyclists can be fined if they access their phones even to check the time when waiting for green signal at the intersections. Several studies have concluded that human error is the main cause of road accidents [28]. Considering the previous statement, one cannot

completely eliminate humans from the driving scene; the accountability for autonomous driving accidents is still in debate. There are several systems designed to introduce a safety buffer between the driver and different vehicle functions. Brake-by-wire, steer-by-wire and drive-by-wire are systems that do not translate human commands perfectly into action. In other words, human commands act as input to these systems that take decisions with a priority on vehicle safety. ABS and ESC are two systems that are quintessential examples of such systems and these are elaborated in section 2.3 and 2.4. Although the human commands are altered to some degree in these systems, human input can seldom be rejected. Various regulations prevent handing over control to computers in safety critical functions where human lives are at stake; blaming a software bug is much easier than a human in the court of law. Monotonous development in ADAS alone does not ensure decrease in road accidents, one also has to cater to the appeal of the general public and lawmakers to welcome progressive change and encourage development. A good example is - Continental receiving Automated Driving Testing License for testing autonomous driving in the roads of Nevada State of USA. Accidents are also an expensive affair. Road accidents in 2010 amounted to 32999 deaths, 3.9M injured people and 24M damaged vehicles; the total cost of these unfortunate events amounted to $242 billion [29].

All the above points highlight the importance of ADAS systems and their pursuit in vehicle safety. Ground surface detection forms a critical part of road scene interpretation. It defines the boundaries of drivable surface area for a vehicle. The importance of road surface detection lies in the fact that we can learn about the surface that the vehicle can tread on and also this knowledge assists us to easily and reliably detect obstacles within a scene presented based on the 3D data. All objects that protrude above the detected road surface (above a certain threshold) can be classified as obstacles. This crude object detection can serve as a preprocessing step to limit the search window of vehicle detections. To bring any autonomy into vehicle driving, we should detect the surface we should tread on.

1.2 Approach

The work of this master thesis is concerned with free space/ground plane detection. Throughout this thesis report – ground surface, ground plane, road plane, road surface are synonymously used since there is practically no difference in road or ground as far as their 3D presentation is concerned. Various ground surface and free space detection algorithms have been studied. This study has been restricted to solutions that are not heavy on computational load. This requirement rules out the machine learning algorithms like those that make use of support vector machines and their variants for road surface detection [15]. The road surface detection algorithms based on 3D dimensional data are particularly simple to implement and offer reasonable robustness. This behavior is due to the fact that the external world as seen from the vehicle dashboard can be

modelled to a great degree of simplicity. Theoretically there are numerous solutions available that provide 3D data of a scene. RADAR, LIDAR, ultrasonic, stereo camera sensors are some examples of technologies that have been successfully tried and tested in vehicles to this effect. RADAR and LIDAR offer accurate 3D scene data but the technology is not as affordable as the camera. Ultrasonic technologies rely on the sound signals for calculating the distance to the reflecting surface. The accuracy of the ultrasonic sensors depends on the ability to accurately predict the speed of sound in the environment of operation. Speed of sound depends on the carrier medium (air), its temperature, pressure, etc. Furthermore the dynamic weather conditions and vehicles in motion make the estimation challenging still. In contrast, the sensors LIDAR and SONAR depend on the speed of light. This is a far more stable entity than the speed of sound. There is one drawback with using light for distance calculations and that is the sheer magnitude of speed of light. It takes 67 nanoseconds for light to make a round trip of 10m. This implies that an error of 10% in sensor stopwatch introduces a distance error of 50cm. Using stereo camera the advantage is two-fold; one can generate the 3D data using the stereo computation and have access to the light reflecting from the scene. Thus the focus of the thesis has been on using the 3D data from the stereo camera for road surface detection. Different algorithms using derived data from stereo camera have been studied. These are implemented in the Visual Studio 2008 build environment using OpenCV 2.4.6 library. The algorithms are also analyzed for robustness, accuracy and speed of execution. Part of the research contribution of this thesis lies in this analysis. Furthermore, based on the analysis of these algorithms, certain improvements are suggested that make them more suitable for the challenge in discussion. These novel algorithms are implemented and their results are studied; this forms the other research contribution of the thesis. Due to the nature of this combined analysis and suggested improvements for the ground surface detection algorithms; the implementations of algorithms and study of suggested improvements has been seamlessly integrated into Sections 4.2 & 4.3. Detailed analysis of the ground surface detection algorithms have been carried on benchmark dataset. Their accuracy and speed of execution has been tabulated.

My tasks during the thesis can be broadly classified as follows:

- Study of the road surface detection algorithms.
- Implementation of these algorithms and analyzing their pros and cons
- Suggesting improvements in algorithms wherever possible
- Evaluation of the different algorithms implemented.

The constitution of this thesis in each section is as follows. Section 2 starts with the literature review undertaken to learn the current state of different technologies. Section 2 also presents common ADAS systems and the image processing basics that are used in this thesis. Furthermore, section 2 also details the implementation of the road plane estimation using v-disparity images and occupancy grid maps. Both the existing ap-

proaches have been implemented, analyzed and certain improvements are proposed wherever necessary. Section 3 charts the various evaluation statistics. Section 4 presents the contributions made throughout the thesis work. Section 5 draws the main conclusion from the thesis work and finally Section 6 lists the various references used in this report.

2. LITERATURE REVIEW AND STUDY OF ADAS TECHNOLOGY

2.1 Literature Review

The thesis work started with literature review to get myself acquainted with the current trends in driver assistance systems with a focus on use of camera in this field. Use of dense stereo images to extract road surface, obstacles has been detailed by Florin Oniga and Sergiu Nedevschi [2]. The author generates the 3D points from the images and uses them for two purposes. The first one is to fit a quadratic road surface to these points and isolating structures that deviate significantly from this model and the second one is to compute the density map and noting that - vertical structures concentrate points within the grid cell upon which they are projected. These two estimates are fused to form a complete estimate of the road plane and obstacles.

In [4] the disparity computation yields not only the disparity images but also their variance. This variance along with the ego motion of the vehicle is used for Kalman filtering of disparity images. The filtered disparity images are used to generate the occupancy grid maps. The occupancy grid maps are also subjected to Kalman filtering to mitigate outliers. The authors make use of 3 kinds of grid maps; Cartesian, polar and column disparity. Segmenting these grid maps is carried out using dynamic programming algorithms

Don Murray and Jim Little [5] implement an obstacle detection feature in a mobile robot equipped with stereo cameras. Using the stereo images, disparity images are generated. The highest disparity along each column is assumed to be the obstacle that is closest to the robot in the column. This way a map indicating the nearest obstacle along each row is charted and serves as the obstacle boundary. One thing to be noted is that in our context because the highest disparity along every column will almost always be the road, hence this implementation will be of little use.

The disparity images tend to be poor when the image fails to offer distinct feature to match in right and left images. This is especially true for road pixels which are fairly uniform. To overcome this shortcoming, the authors of [6] prepare a set of candidate lines in v-disparity that can correspond to the road plane, they score these candidate lines with the matching cost of a wide window (in stereo image) at select rows and at the disparity provided by candidate lines. And the line with the least cumulative matching cost is assumed to correspond to the road. It should be noted that as with most if not

all v-disparity based ground plane estimation algorithms, the assumption that majority of pixels in each row correspond to the road - heavily influences the outcome accuracy.

With modern dense disparity images the v-disparity approach outlined in [9] provides good ground plane estimates with little computational footprint. The approach presented relies on the assumption that along each image row, the road pixels form the majority. This assumption tends to be violated at large distances from the ego-vehicle hence the ground plane estimates also tend to be compromised. To account for road surfaces that are not flat or exhibit significant deviation from flat surface, the authors of [7] suggest the use of B-splines to fit the points belonging to road. The authors illustrate the shortcomings of line, envelope, quadratic and cubic curve fitting and state that B-splines supersede these. Also the B-spline fitting is done to points in the world coordinates rather than the v-disparity to better accommodate the points at large distances from the vehicle. Custom scene maps generated and stored offline (similar to google street view) are used to localize the vehicle in [8]. These maps are generated manually and include among others the lane markers, curbs, and GPS location where they are observed. The stereo images in real time are matched to find correspondence to this digital map which ascertains the location, orientation and provides information that is very close to ground trut. The inertial measurement unit tracks the real-time changes in vehicle position assisted with a Kalman filter, while the Digital maps keep the drift in check, similar to several indoor positioning systems in smartphones. A more recent publication [17] addresses the issue of road surface detection with stereo camera data and providing results real time. The author thresholds the u-disparity image to eliminate potential obstacles' pixels in disparity image. This 'filtered' disparity image is used to generate the v-disparity image. Instead of fitting lines or predefined geometric models to the v-disparity, the authors claim that the road surface pixels are most likely to correspond to the maxima along image rows of the v-disparity.

Table 1. *Comparison of different road surface detection algorithms*

Authors	Sensors	Input	Output	Advantages	Disadvantages
Florin Oniga et al. [2]	Stereo camera	Disparity image	Road surface, sidewalks, obstacles.	Detailed classification of traffic participants, Simple implementation	Use of several constant thresholds
Labayrade et al. [9]	Stereo camera	Disparity image	Road surface	Simple and fast road surface detection	Fixed road surface model
H. Badino et al. [4]	Stereo camera	Disparity image	Free space	Good free space detection upto the obstacles	Slow dynamic programming segmentation
Meiqing Wu et al. [17]	Stereo camera	Disparity Image	Road surface	Robust road surface detection with real-time estimates	Use of constant thresholds

2.2 ADAS and machine vision

The flight control systems on board the Fighter jet F-22 Raptor runs on 1.7 million lines of code. The new Boeing 787 Dreamliner which flies with about 300 passengers on board runs on about 6.5 million lines of code. In contrast an automobile, for instance a premium segment car runs on 100 million lines of code [25]. An automobile runs this volume of code on 70-100 microprocessors, which are in turn embedded in Electronic control units (ECU). Figure 1 gives an overview of which automotive functions are dependent on such ECUs. Automobiles have ECUs for control of systems like engine, powertrain, instrumentation, suspension, steering, brakes and infotainment (in-car entertainment). An engine control unit controls the spark plug ignition which burns the fuel in the engine cylinders; a critical function for conversion of fuel to kinetic energy. An engine control units must make sure that an engine does not stall when there is no throttle input from the driver, this speed is called the idle speed. It should be a compromise between minimizing the energy loss during idle and reliably keep the engine running. In modern engines the engine control unit also controls the valves that feed the engine cylinder with fuel. The mixture of air and fuel fed into the cylinder is called charge and depending on the ratio of fuel and air, the charge can be either ‘rich’ or ‘lean’. This is also one of the functions of an engine control unit, to observe a good balance between performance and efficiency. A transmission control unit reads the engine speed and current operating state from various sensors (Wheel speed sensor, Engine speed sensor) and decides the appropriate time to shift gears (in automatic transmission vehicles). Transmission control unit also makes sure that the clutch engages and disengages the engine to the drivetrain in an optimum fashion. It must also make sure that the transmission fluid temperature is within operating temperature range.

There are complex vehicle systems that make sure that the driver is in control even under extreme operating conditions. These systems work in close cooperation with multiple ECUs sharing information, sensor readings to make sure that the vehicle is stable and has traction at all times. Following section introduces two of the most popular and effective ADAS systems Anti-lock Braking Systems (ABS) and Electronic Stability Control (ESC), followed by an overview of vision based ADAS. Thereafter the flow of image information from image acquisition to image processing is detailed within the framework of OpenCV.

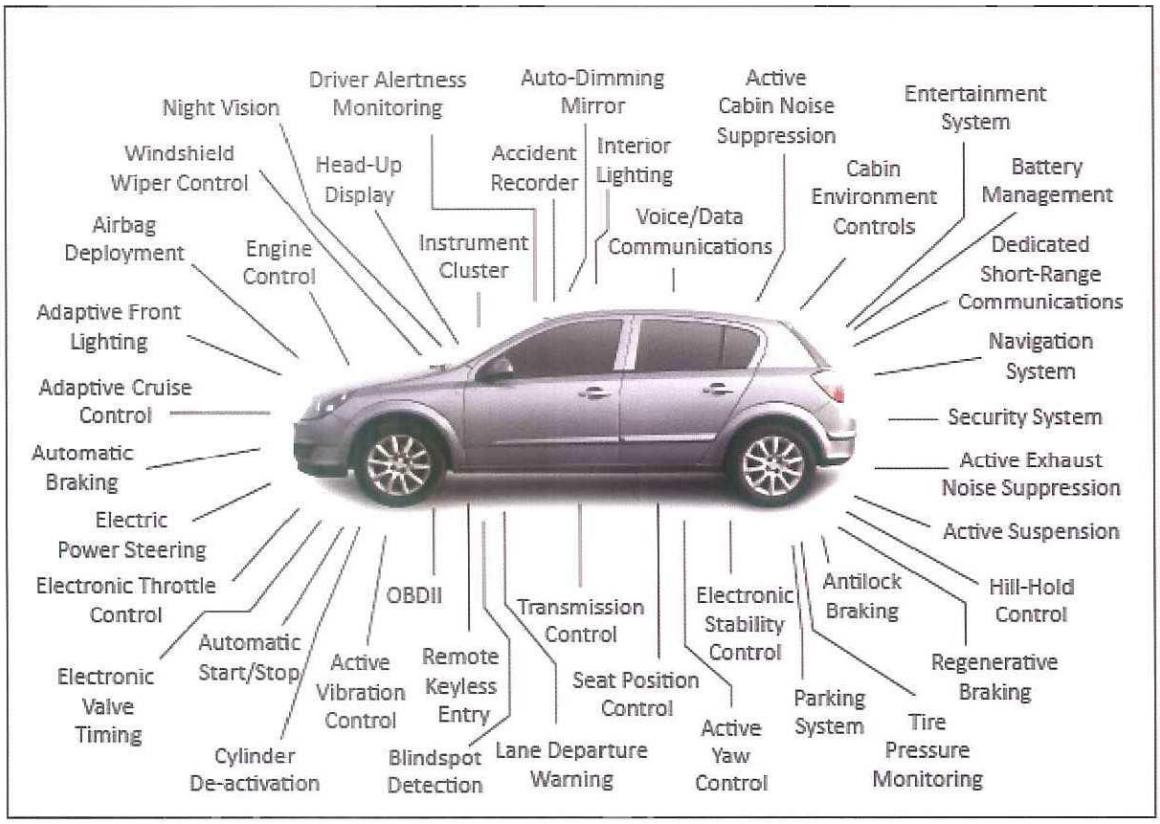

Figure 1. *ECU dependent car functions [26]*

2.3 ADAS: Anti-lock Braking System (ABS)

Humans like all animals respond with impulses to external stimuli. When faced with an imminent head on collision, a human driver stomps on the brakes as hard as he physically can. This force on the brake pedal is so high that it commands the brake calipers to bite the brake discs or drums with enough force to 'lock' the wheels relative to the calipers and the wheels stop rotating immediately. This means that the vehicle skids on the road surfaces until it comes to a halt either from continued skid or from a collision. There are two very important reasons why wheel skid is not a favorable braking strategy. First, we know that the maximum static frictional force is higher than the maximum kinetic frictional force between two surfaces (In Figure 2 f_s refers to static friction and f_k to kinetic). In other words to extract the maximum frictional force from the tire/road surface pair (and hence stop within the shortest distance), we need to keep the wheel at the limit of grip (near the apex of plot in Figure 2). The second reason why skids are not favorable is that during a skid the steering input has very limited effect on the directional control of the vehicle. This implies that the driver is almost at the mercy of the surroundings to halt his car in a safe manner.

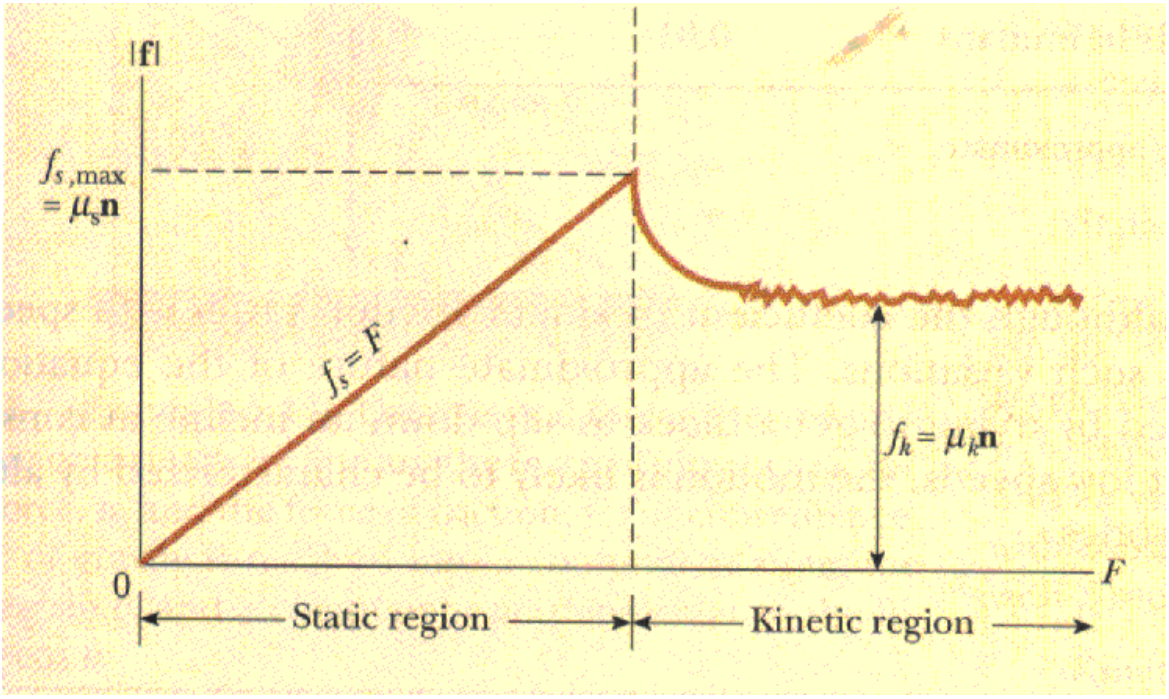

Figure 2. *Static (f_s) and kinetic friction(f_k)*

The function of the ABS system in vehicles is to keep the wheel turning during hard braking so that the driver can have some steering control at the same time to brake the wheels at the limit to extract max frictional force from the road surface and stop the vehicle in shortest possible distance. A standard ABS system reads wheel speed from sensors on each wheel. When it detects a sudden drop in wheel speed from one of the sensors, the system immediately decreases the brake pressure on this wheel so that the wheel starts rotating again at speeds similar to those of the other wheels. Note that even while driving under normal conditions with traction on all 4 wheels, we can observe certain difference in wheel speeds, especially during turning. The ABS systems are designed to accommodate these minor variations.

2.4 ADAS: Electronic Stability Control (ESC)

Figure 3 presents two extreme conditions that are observed when cornering at high speeds. We turn the steering wheel while negotiating a turn. The steering wheel in turn commands the front wheels to turn in the corresponding direction (assisted by the power steering). Under normal driving conditions the turning moment required to keep vehicle in traction on the curve is derived from the 4 wheels. When the front wheels fail to provide the necessary grip to keep the vehicle on curve, they skid and the vehicle understeers.

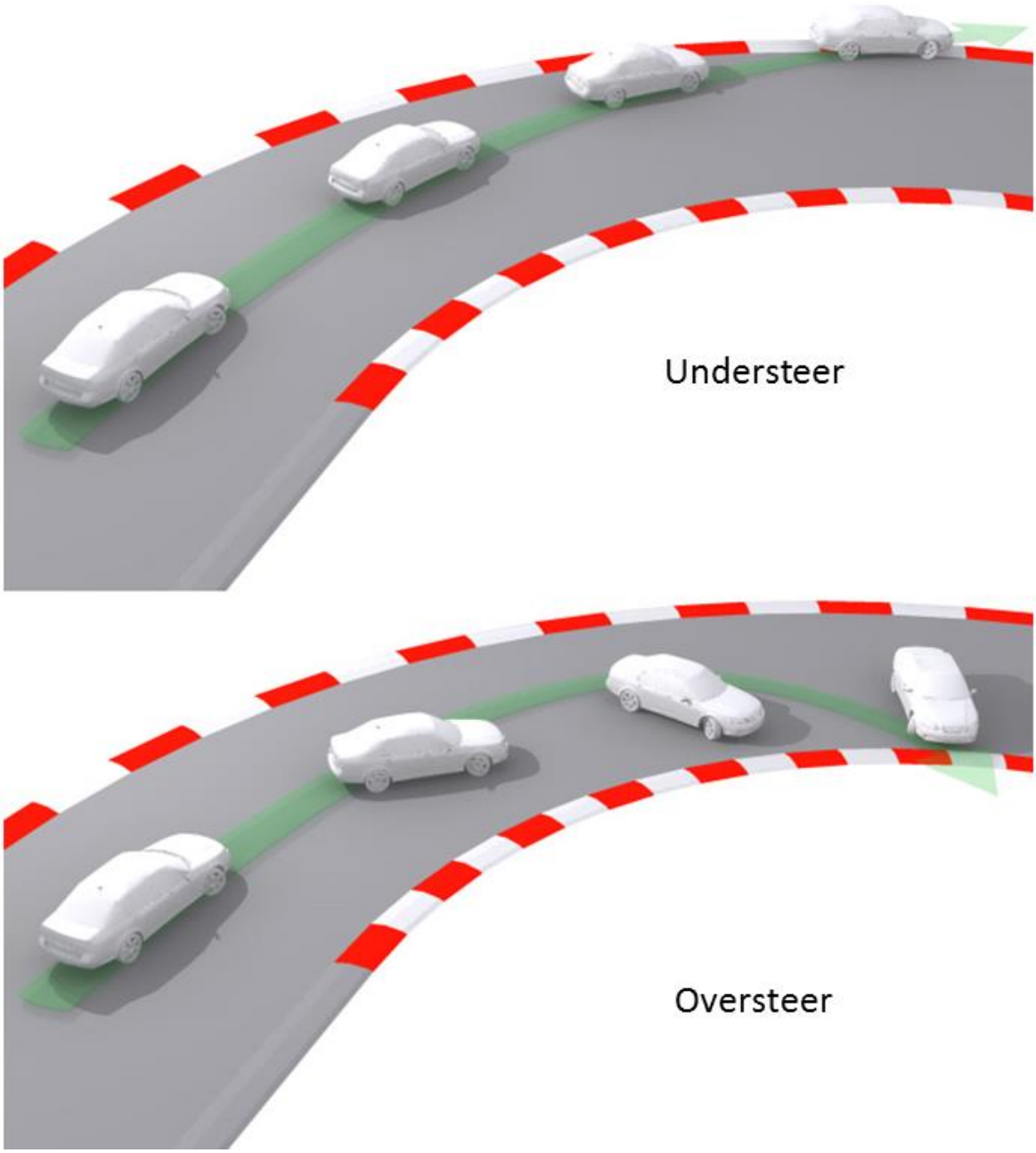

Figure 3. *Understeer and oversteer*

When the rear wheels fail to provide the necessary grip, they skid and the vehicle oversteers. ESC has been designed to obviate the above situations. To detect such situations the ESC reads the driver intention (steering position) and the vehicle actual trajectory (gyroscopes, accelerometers). If the actual vehicle rotation is lower than the driver's intention, understeer is detected and vice versa. To counter understeer, the ESC brakes the right rear wheel to produce the additional moment to negotiate the curve. To counter oversteer, the ESC brakes the front left wheel. Note that in both understeer and oversteer situations the ESC brakes the wheels that still remain in traction.

2.5 ADAS: Vision based

Vision based ADAS systems make use of images as sensory input to offer vehicle safety functions. The following is the description of the scope of these systems and the degree of environment perception they achieve.

2.5.1 The scope

Historically the cameras have been used in cars for lane departure warning and blind spot detection. Nowadays the cameras are used in a much larger capacity. These systems are responsible for sensing the vehicle's surroundings and create a virtual safety net. For instance, when the vehicle senses another vehicle on the adjacent lane, switching to this adjacent lane can be prohibited to avoid collision, particular attention has to be paid to the probability of vehicle detection, probability of its estimated position, velocity and finally the probability of a possible collision in case of lane switch considering the host vehicles kinematic parameters. Such systems are designed to assist the driver and not hamper his will to drive so a compromise between safety and driver's freedom has to be reached. A very crude classification of vision based systems in vehicles can be made on the processing stage of the image data – low level vision (image processing, stereo vision, optical flow), medium level vision (object detections) and high level vision (tracking detected objects and their influences on host vehicle). The vision based ADAS has to function in diverse driving conditions (rainy, sunny, night, tunnels, hairpin turns, within-city, highways traffic jams and all probable combinations of these). Vision based ADAS offers safety and comfort functions like presenting the blind spots for a driver without being overwhelming, augmented vision capabilities during night, fog, snow, rain, etc. The augmented scene can be projected on a display or the windscreen itself. Critical information such as sign boards (speed limits, sharp turns) current driving lane, detected traffic constituents (pedestrians, cyclists, etc) can be projected onto the screen.

2.5.2 Environment perception

A traffic scene can be segmented as – an ego vehicle (host vehicle), ground surface (of which road is a subset), other traffic participants (vehicles, pedestrians, cyclists), traffic signs and barriers. Ego motion describes the absolute kinematics of the vehicle in the real world frame. Vision based environment perception includes computation of the following key parameters.

Distance computation is achieved reasonably well with stereo camera data. But challenging driving conditions like rain, snow, sun glare, etc limit the scope of its effectiveness. To improve robustness the data is supplemented with that from other sources like Laser range finders, RADAR, etc. Scene motion estimation is made for image pixels to

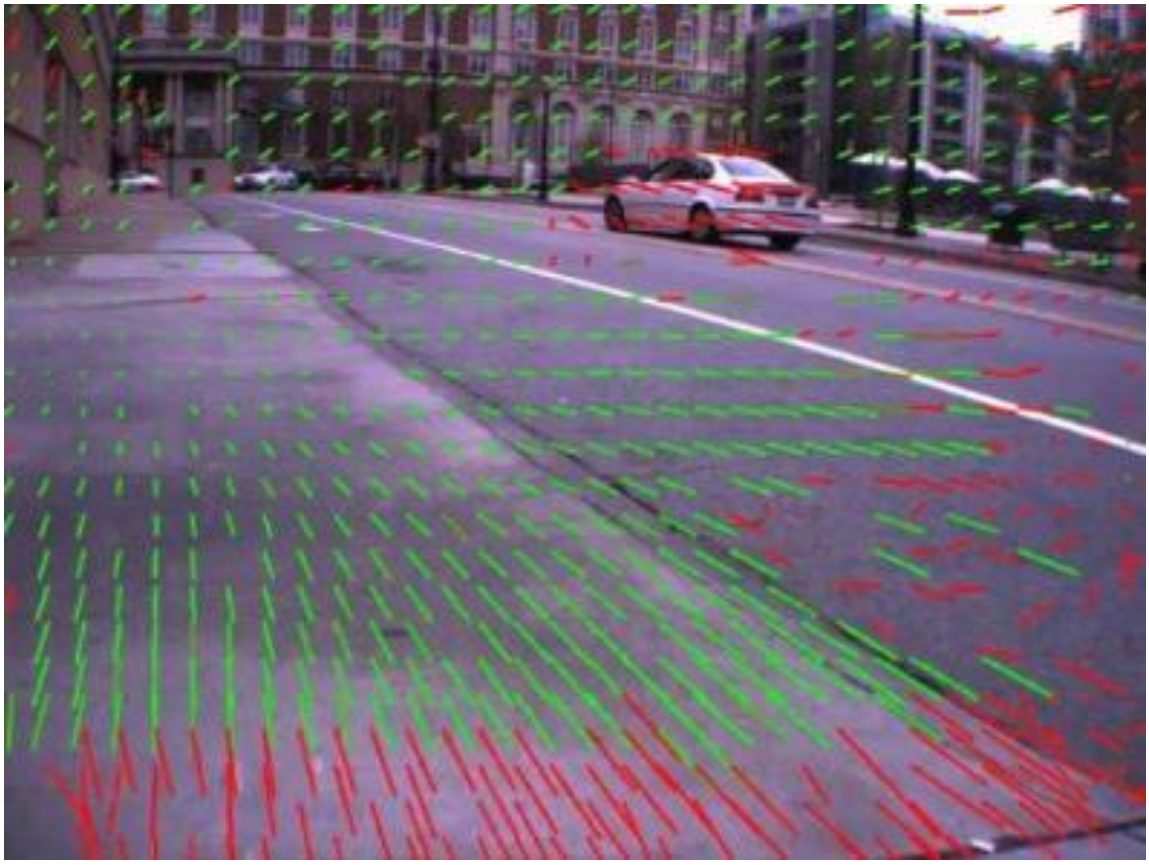

Figure 4. *Optic flow vectors are tangent to the direction of motion of the pixels*

have a better understanding of the dynamic scene flow. Pixel displacement is studied between image frames and the motion estimates are then made for the scene Figure 4. Pixels exhibiting 'similar' motion generally belong to the same objects in scene.

Obstacle detection is based on data from multiple sources like mono camera, stereo camera, LIDAR, RADAR, etc. Traffic constituents like other vehicles, pedestrians, cyclists, etc constitute as obstacles. Obstacles detection can be made either in every frame or the detected obstacles can be tracked over multiple frames. Object state parameters like position, velocity etc are useful in finding the object in subsequent frames and saving computational cost in making new detections in every frame. Vehicle tracking is an important aspect in collision detection and avoidance. Trajectories of the ego vehicle and that of its neighboring entities are used to predict collisions in near future. Tracking of vehicles is a much easier proposition than that for pedestrians; mainly because of the changing stance and hence appearances of walking humans, furthermore high relative velocity between the ego-vehicle and pedestrians generally adds a lot of inter image frame variance in the same pedestrian appearance. This makes tracking pedestrians a more difficult preposition that tracking neighboring vehicles. Detection of surrounding infrastructure (road surface, marked lanes, traffic signs, etc) can also be achieved with vision based systems in vehicles. Road surfaces detection is a key contribution of this thesis work and will be discussed in greater detail in sections that follow. Traffic signs and signboards can be detected and the information contained within extracted to aug-

ment the driver's scene perception. A typical approach to traffic sign detection would be to extract known traffic sign patterns (triangular, polygonal structures) from image frames, extract feature contained within these patterns and compare them with a lookup data base for sign boards to find the meaning.

Complete scene analysis needs the understanding of the current traffic related components, their kinematic parameters and the study of their near future impact with the ego vehicle. An intelligent vehicle speed regulator takes into account the current speed regulations indicated by the signboards, the speed of the vehicles in the immediate vicinity. Modern cars are fitted with futuristic functions like automated parking in mid segment cars. This functionality requires 360 degree surround view, which is made available through ultrasonic and/or close range RADARS. Intelligent headlamp control adjusts the beam of the headlamps based on the traffic scene. This control is introduced to maximize the visible road for the driver while at the same time not dazzling the oncoming vehicles' driver unnecessarily.

2.6 Image Acquisition

The process of capturing the light reflected from a scene and presenting this information as an image can be termed as image acquisition. The light incident on an object is absorbed by the surface of the object itself and the spectrum of light that is not absorbed gets reflected and this reflected light carries a definite spectrum of light that is perceived as the color of the object. The geometry that governs the capture of these rays are important to build suitable camera models, which in turn help us to reconstruct the 3D world once we have the stereo images. One simple but useful model is the pinhole camera model. A pinhole is an imaginary aperture in a plane (pinhole plane) of infinitesimal thickness and zero aperture. The rays are allowed to pass through the plane through this aperture alone. This pinhole model is not sufficient to gather enough light for real camera image and we make use of lenses to gather more light. Unfortunately this leads to a more complex camera model and also introduces distortions in images. All these factors affect the reconstruction of the real world given the stereo camera geometry and hence it is important to study the following topics.

2.6.1 Pinhole camera model

In this camera model a single ray from any point on the object surface and passes through an imaginary hole of zero diameter on the pinhole plane and is caught on an image plane; refer Figure 5. The size of the image on the image plane is calculated by the formula $x = -f\frac{X}{Z}$; note that the negative sign indicates that the image is inverted, as can be seen from the top frame in Figure 5. A minor rearrangement of the pinhole camera model (bottom frame of Figure 5) can make the math simpler. In this simpler model the rays still reach the hole in the pinhole plane, en route they strike the image

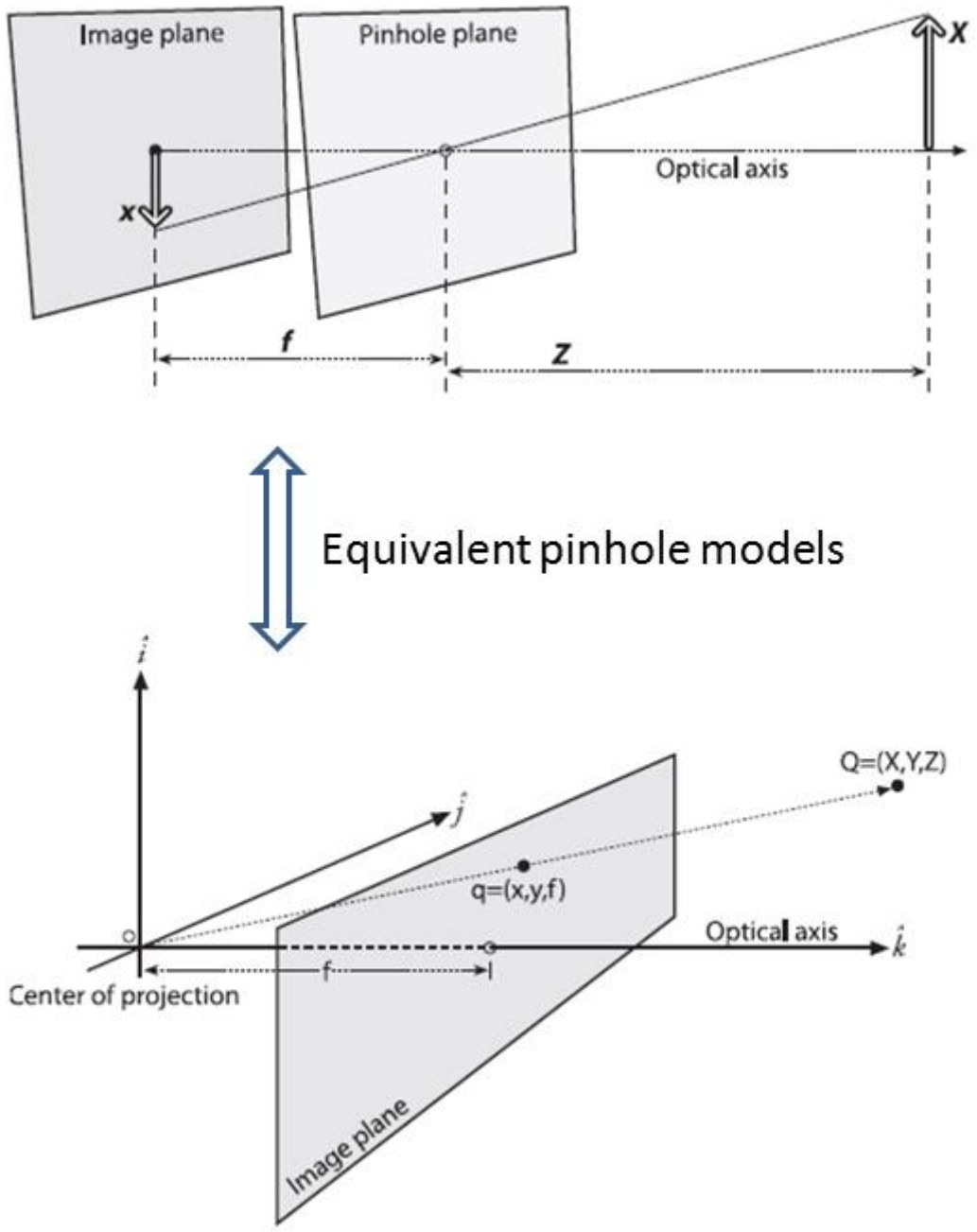

Figure 5. *Equivalent pinhole camera models*

plane such that the image remains upright while the size remains the same as in the previous model.

Principle point is defined as the point where the image plane intersects the optic axis. The center of image plane is usually considered as the origin of frame coordinates. During manufacturing the center of image plane cannot be made to absolutely coincide with the principle point, which implies we need a correction term to accommodate the offset parameters (c_x, c_y). Thus

$$x_{screen} = f_x \frac{X}{Z} + c_x; y_{screen} = f_y \frac{Y}{Z} + c_y \quad (1)$$

2.6.2 Projective Geometry

The relation that maps the points Q_i in the real world coordinates (X, Y, Z) to the points in the image space with the coordinates (x, y) is termed as projective transform. The projection of the points in the physical world into the camera coordinate frame is mathematically expressed below.

$$\begin{bmatrix} x \\ y \\ w \end{bmatrix} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} . \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \quad (2)$$

The parameters c_x, c_y, f_x, f_y are characteristic of the camera and are called camera intrinsic parameters. The matrix in the middle in above equation is therefore called the camera intrinsic matrix. In addition to the above nominal intrinsic parameters there are other undesired parameters that characterize the camera behavior (image acquisition). These parameters will be discussed in the next section.

Consider the point P (X, Y, Z) in Figure 13, where (X, Y, Z) are in reference to the coordinate frame in real space. The dotted line originating from this point meets two optic centres O_l and O_r of the left and right cameras respectively. The image planes of the left and right cameras are presented as parallelograms with normals originating from O_l and forming Z_l for the left camera and vice versa. The point where the dotted line meets the image plane represents the image of the point P in the respective camera image. Through simple geometric transformation equations, the image (u, v) made by point P in the left and right cameras have been presented in [9]. These equations are also presented in this report as equations 18 & 19.

2.6.3 Lens distortion

Although it is possible to mathematically devise a lens that produces no distortion, the lens manufacturing is never perfect. Furthermore to save manufacturing cost, spherical lenses are manufactured instead of the ideal parabolic lens. Also there are errors arising during placement of lens and image sensor. Two main distortions that appear in images due to all these inaccuracies are the radial and tangential distortions. Radial distortions are irregular spacing of image pixels radially about the principal axis. Figure 6(a) gives

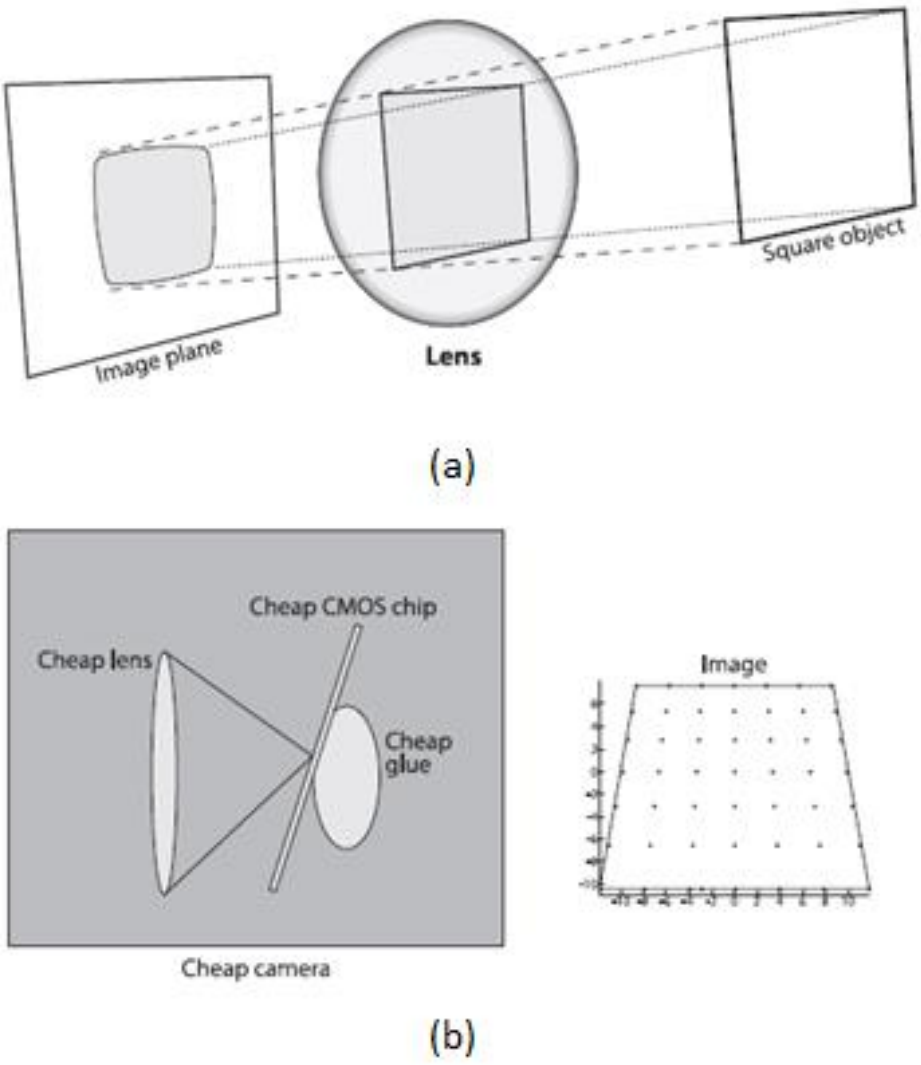

Figure 6. *Lens distortions (a) radial distortion; (b) tangential distortion*

an intuitive presentation of the radial distortion and the reason for this. This arises due to the fact that the refractive power of lens is higher around the edges of the lens than near the principle axes. The radial distortions are typically zero around the center and increase as one moves away from the principle axes. The second form of lens distortion is the tangential distortion. This arises due to the fact that the image sensor surface can never be 'perfectly' normal to the principle axes. Non uniform glue distribution further accentuates the problem as shown in Figure 6(b).

2.6.4 Camera calibration

The above two sections described the camera intrinsic parameters (the focal lengths and offset errors) and the distortion errors. Camera calibration is a process that is carried out to find the parameters that quantify camera intrinsic as well as the distortion behavior. Each object that is in view of the camera field of view can be represented in the carte-

sian frame of reference as 3 translational parameters (T_x, T_y, T_z) and 3 rotational parameters (R_x, R_y, R_z). Hence there are 6 unknown positional parameters for each object view. Furthermore we have 4 unknown camera intrinsic parameters. Usually we use a chessboard from which the corners of the squares are easy to pinpoint by image processing algorithms. By capturing the images of chessboard in various orientations we can ascertain the camera intrinsic parameters using OpenCV functions.

2.7 Stereo computation

A stereo camera is a pair of cameras that share the same image plane and whose optic axes are separated by a fixed distance. The coordinate frames assigned to left and right cameras can be seen in Figure 13. Note the frame assignment to the camera image frames in the figure. The frame assigned to actual images used in OpenCV has a similar orientation but is translated to the top left corner of the image when looking along +ve Z-axis and the axes unit is pixels.

Given the position vector of a point $\vec{P} = [x\ y\ z]^T$ w.r.t the ground frame of reference and assuming zero camera inclination (i.e. $\theta = 0$), we will prove that z is a function of the camera image coordinates of this point in the left and right frames i.e.

$$z = f\left(img_{cl}^{p}, img_{cr}^{p}\right) \tag{3}$$

Where $img_c^p = (u_c^p, v_c^p)$ represents the image of point p in camera c. The subscript *cl* and *cr* refer to the left and right camera respectively.

And (u, v) are the pixel coordinates with u being the column and v being the row.

Assume that the position vector of the left and the right camera frames are given by $\overrightarrow{O_{cl}}$ & $\overrightarrow{O_{cr}}$ with

$$With\ \overrightarrow{O_{cl}} = [O_{clx}\ O_{cly}\ O_{clz}]^T = \left[\frac{-b}{2}\ h\ 0\right]^T \tag{4}$$

$$And\ \overrightarrow{O_{cr}} = [O_{crx}\ O_{cry}\ O_{crz}]^T = \left[\frac{b}{2}\ h\ 0\right]^T \tag{5}$$

The position vector of Point *P* w.r.t the left and right camera frames is given by $\overrightarrow{P_{cl}}$ & $\overrightarrow{P_{cr}}$. Using the triangle law of vectors we can write

$$\overrightarrow{P_{cl}} = \left[P_{clx}\ P_{cly}\ P_{clz}\right]^T = \vec{P} - \overrightarrow{O_{cl}} \tag{6}$$

Using the rule of similar triangles we can write

$$\frac{P_{clx}}{u_{cl}^p} = \frac{P_{cly}}{v_{cl}^p} = \frac{P_{clz} + \alpha}{\alpha} \tag{7}$$

By substitution of equation (6) in (7) we get

$$for\ left\ camera\ \frac{x + \frac{b}{2}}{u_{cl}^p} = \frac{y + h}{v_{cl}^p} = \frac{z + \alpha}{\alpha} \tag{8}$$

$$for\ right\ camera\ \frac{x - \frac{b}{2}}{u_{cr}^p} = \frac{y + h}{v_{cr}^p} = \frac{z + \alpha}{\alpha} \tag{9}$$

Comparing the above two equations we can draw the following conclusions

$$v_{cl}^p = v_{cr}^p \tag{10}$$

$$and\ u_{cl}^p - u_{cr}^p = \Delta = \left[\frac{\alpha}{z + \alpha}\right] b \tag{11}$$

Assuming $\alpha \ll z$ we can write the above equation as

$$\Delta = \left[\frac{\alpha . b}{z}\right] \tag{12}$$

The term Δ is known as disparity which represents the separation of the image of an object in two stereo cameras (usually along the x-axis or the image columns). Note that

in the derivation of the above equation we assume zero camera inclination (i.e. the angle made by the camera about the horizontal axis). Now that we understand disparity images being representative of the 3D scene data; we introduce two images that are derived from disparity images: v-disparity and OGM. Almost all our road surface detection algorithms are based on these images hence it makes sense to introduce them here.

2.7.1 v-disparity (v-disp)

This image is a formed by calculating the disparity histogram along image rows. Since disparity image provides the disparity for all image pixels, the v-disparity image is formed by calculating the disparity histograms along each image rows. Fig. 7 presents a good illustration of v-disparity from a synthetic scene.

2.7.2 Occupancy Grid Map (OGM)

OGMs are grid maps where the vertical surfaces get highlighted. With stereo images we get disparity images from stereo computation. From the disparity images we can calculate the real 3D coordinates of each pixel. We then project this cloud of points onto a flat horizontal grid with cells of certain size. A simple projection is made by calculating the number of points that lie within an imaginary cuboid formed by extending OGM cell vertically in both directions. This number represents the occupancy for the cell. A more complex projection can also be done to smooth the cell occupancy. Figure 31 presents a good illustration of OGM for a scene with an obstacle placed before the camera. Vertical surfaces remain vertical even when the ground surface exhibits tilt about the horizontal axis, which is why OGMs are effective on non-horizontal surfaces as well. This is one of the key reasons that the thesis work considers OGMs for ground surface detection as an alternative to use of v-disparity for the same

2.8 Image processing

Image processing is as the name implies the processing of images to extract either transformed images or relevant parameters of interest. Image is a multi-dimensional matrix with individual elements (also known as pixels) representing the scene as either grayscale intensity or color.

2.8.1 Smoothing

Image smoothing (also known as blurring) is the process of ironing sharp changes in intensity or color of pixels. One of the simplest blurring operations is done by equating the intensity of a pixel to the mean of the intensity of its surrounding pixels.

Figure 7. *Gaussian smoothing [12]*

Gaussian smoothing is one of the most popular smoothing algorithms and involves the convolution of a Gaussian kernel and the image matrix, refer Figure 7. In simple terms it is weighed average of pixel intensity where the weights are defined within the kernel and the size of kernel defines the degree of smoothness. Following is the mathematical representation of convolution.

$$H(x,y) = \sum_{j=0}^{n}\sum_{i=0}^{m} I(x+i-a_i, y+j-a_j)G(i,j) \tag{13}$$

Where $H(x,y)$is the intensity of the resulting image pixel; (m,n) is the size of the kernel; (a_i, a_j) are the anchor coordinates on the kernel. This kernel can be either symmetric or asymmetric about the horizontal and vertical axes. Asymmetric kernels are particularly useful in preserving features that have a known direction of presentation within images. Road lanes present themselves in a certain angle range when viewed from a dashboard, non-uniform smoothing is useful in preserving lane markings in such images [11]. A more complex and computationally intensive smoothing is the edge preserving smoothing. This smoothing is particularly important when we are interested in extracting the geometric structures (lines, edges) from an image and not interested in pixels that exhibit gradual change in intensity. Smoothing images is very beneficial when cluster segmentation is carried out on images. Clusters are blobs in images that have similar presentation in intensity, color or geometric structure. Presence of outliers/noisy pixels within images greatly affects the image clustering algorithms. Image smoothing is helpful in mitigating the effect of noisy pixels in such algorithms.

2.8.2 Image segmentation

Image segmentation is the process of segmenting the image into regions of interest, this is done so as to either transform the image into something more manageable or decrease the size of the data being handled. Thresholding is the simplest method of image segmentation where the segments are made based on the intensity of the pixels. For instance, when we observe pixel intensity from 0-255 in an image we could convert this grayscale image (grayscale images have pixel intensity variation from black to white) into a binary image (binary images have pixels that are either black or white) by setting a threshold say 125 pixels. Pixels having intensity above 125 (gray) can be assumed white and the rest black. Although crude, such algorithms are very useful in machine vision applications in automation where speed of execution is a priority. Counting the number of pellets that lie on a conveyor belt can be achieved by using such segmentation algorithms in conjunction with a clustering algorithm. Segmentation algorithms can be tailored to extract only the pixels of interest. For instance, in the game of tennis players are allowed to challenge the decision of the line judges on where the ball landed. The arbitration is carried out by a machine vision system where the camera tracks the tennis ball on court. The images captured by the camera can be processed with a segmentation algorithm that filters everything apart from yellow pixels from the image. This algorithm is not very robust since there might be yellow colored clothes, advertisements and so on. In reality, the images are processed based on the difference observed in two successive images. Since the tennis ball is the fastest travelling entity on court, the difference in the two images is bound to include the ball pixels in high proportion. To consolidate this detection further, one can segment the difference image at yellow color. Thresholding is based on a threshold which can be either constant or variable. A more sophisticated thresholding technique is using the adaptive threshold where instead of using a fixed threshold to segment whole of image matrix, a threshold value is calculated for each pixel of the image by considering a intensity of pixels within a square window in the neighborhood of the pixel. Such adaptive thresholds perform better when segmenting images that have non uniform noise (for instance non uniform illumination). Figure 8 presents the advantage of adaptive thresholding when segmenting a chessboard image to segment the black checkers from the white. Such thresholding is spatial since the threshold is a function of the position for which the threshold is calculated. Temporal adaptive threshold is calculated using time as a factor.

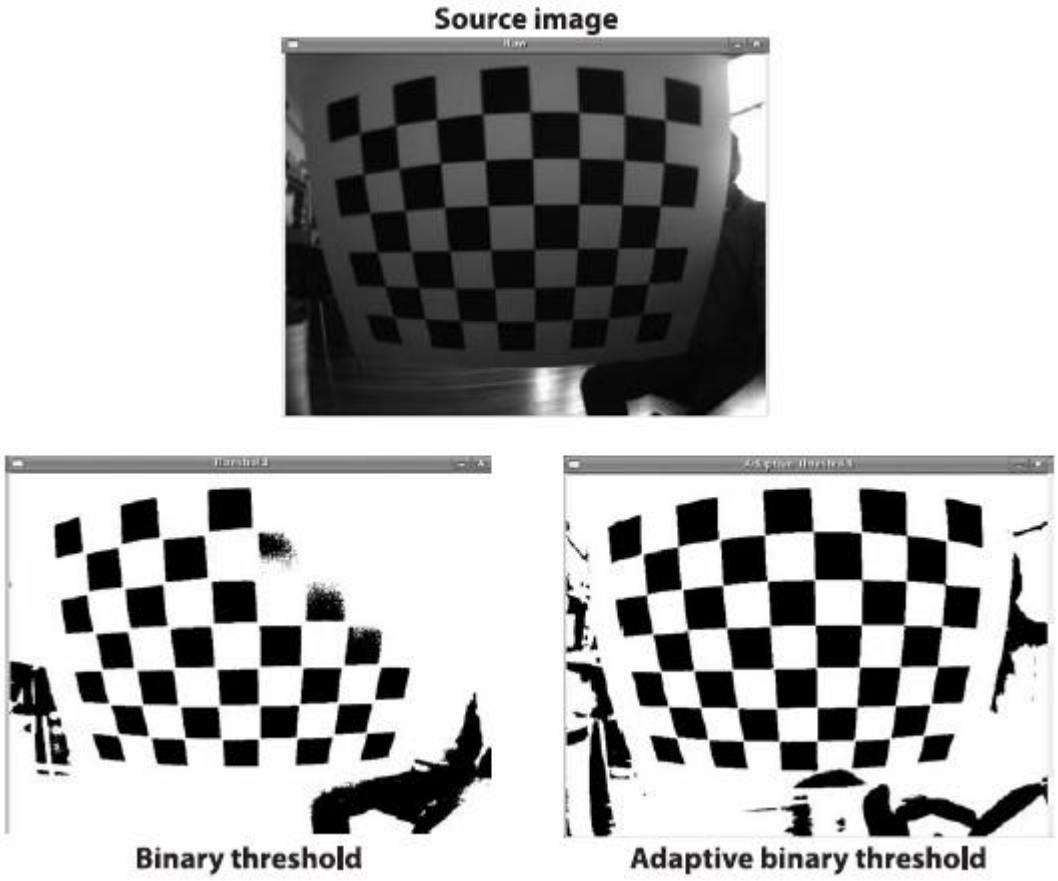

Figure 8. *Binary and Adaptive thresholding [12]*

Figure 9. *Image gradient [12]*

Opposite of image smoothing is highlighting intensity/color variation within image. Such transformation is termed image gradient. Image gradient is also an inbuilt function in OpenCV [12]. It is a convolution of certain gradient kernels and the image matrix.

Figure 9 presents the gradient of an image, notice how the edges that are have sharp intensity gradient across them get highlighted in the gradient image. Image gradient is a powerful tool in object detection because most of the objects have a silhouette that presents itself with a sharp intensity gradient and using Image gradient we can extract this

silhouette effectively. Furthermore gradient kernels can be designed to exhibit sensitivity to gradients along particular directions, such sensitivity is very useful when identifying lines that are known to present themselves at certain nominal angles. For precise angular sensitivity, larger kernels must be used, although this adds computational costs to the convolution operation. Canny filter is another image processing tool that highlights edges in images. OpenCV also has functions for Canny filter [12]. In this thesis work it will be used to highlight the line features from v-disparity images.

2.8.3 Line and Plane fitting

Line and plane fitting as the name suggests is the process of fitting lines and planes to cluster of points. One of most popular and effective line fitting algorithm in image processing is the Hough transform. The algorithm generates candidate lines with different slopes and intercepts and scores them based on how well they fit the point cluster. OpenCV has inbuilt functions that fit lines to point clusters using Hough transform. Let us assume that the point (x_0, y_0) in an image lies on family of lines described by the inclination and intercept (θ, ρ). Figure 10 presents the geometry behind Hough transform.

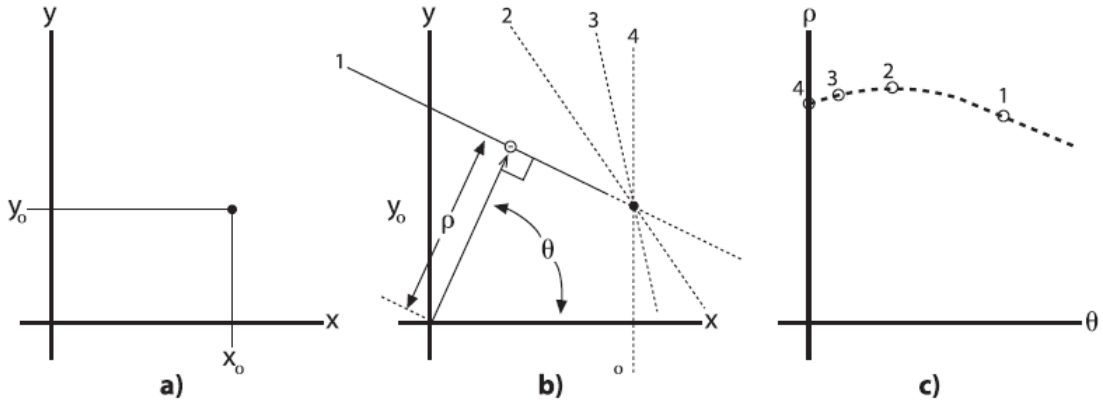

Figure 10. *Hough transform basics. (a) Point in image space (x_0, y_0); (b) locus of lines that pass through point (x_0, y_0); (c) locus of line inclination and intercept that pass through point (x_0, y_0)*

This implies that each point in an image traces a curve in the slope-intercept plot (θ, ρ). All points on this curve in (θ, ρ) describe a unique line in image space that passes through (x_0, y_0). Hence multiple points in image space trace multiple curves in the (θ, ρ) space. If all the points in the image space (x, y) are collinear, then all curves in the (θ, ρ) space meet at a unique point. For a cluster of image pixels, when we sum the (θ, ρ) plots corresponding to individual image pixels, the maxima observed in (θ, ρ) plots correspond to the lines that best 'fit' the image pixels.

OpenCV does not have an inbuilt plane fitting function. Given a set of points $[x_i \ y_i \ z_i]_{i=1:m}$. We assume that the points are related by a linear equation and lie on a

plane having equation $z = ax + by + c$. We find the parameters of the plane that minimize the sum of squared errors between z_i and the plane value $ax_i + by_i + c$. Note that we are not minimizing the normal Euclidian distance of the point from the estimated plane, rather considering the difference along z-axis as error and subsequently minimizing the sum of such squared errors. From [23] we note that this is a simple problem of matrix calculation and one inversion.

$$Ax = B \tag{14}$$

$$Ax = B \tag{14}$$

Where

$$A = \begin{bmatrix} \sum_{i=1}^{m} {x_i}^2 & \sum_{i=1}^{m} x_i y_i & \sum_{i=1}^{m} x_i \\ \sum_{i=1}^{m} x_i y_i & \sum_{i=1}^{m} y_i^2 & \sum_{i=1}^{m} y_i \\ \sum_{i=1}^{m} x_i & \sum_{i=1}^{m} y_i & m \end{bmatrix};\ B = \begin{bmatrix} \sum_{i=1}^{m} x_i z_i \\ \sum_{i=1}^{m} y_i z_i \\ \sum_{i=1}^{m} z_i \end{bmatrix} \ \&\ x = \begin{bmatrix} a \\ b \\ c \end{bmatrix} \tag{15}$$

2.9 A theory of detections

Detections are usually based on a simplified or abstract representation of the object to be detected. For example when detecting lanes one assumes that lanes are elongated bright structures on a dark background [11]; what follows is the search to find pixels in Image that fit this representation. This simplification of the detection process has a catch – the detections of the object are only as good as the abstract model we assume. The higher the detail in abstract model, the more robust the estimation is. But higher detail in abstract model comes will mean more detailed comparison between the model and the unknown object to be classified and subsequently more computational load and slower detections. Hence one must always make a compromise between the detail of the model and the computational load.

The definition of the abstract model also depends on the constraints of the system. For instance in cancer diagnosis we are in no rush to get the results, the robustness of the

model and hence the detection accuracy is stressed. One might argue that in road surface detection, accuracy is just as important, but we do compromise with the detection accuracy because we get road detections real time and by tracking the objects over time (and hence between frames) we have a stronger prognosis of the road surface. We do not always have such luxury with medical diagnosis; for instance patient's exposure to X-rays is limited since it harms the immune system. The classification of road surface algorithms, that are most relevant in choosing the camera, is whether we should to detect road surfaces with mono camera or stereo camera images. The accuracy of estimation is dependent on the data with which the estimation is made. A more accurate estimation can almost always be made with more relevant information. With stereo camera images we get the depth information in addition to the pixel intensity/color of the objects, information that is critical to detection algorithms that are based on topographic abstract road surface models.

2.10 v-disparity approach to ground surface estimation

One abstract model of the road that provides real time detections is the assumption that the road is flat. This model implies that the road points along an image row have the same depth and hence the disparity. So indirectly we search for pixels along rows that share a particular disparity. In reality the road pixels along an image row do not necessarily have the same depth; instead the disparity of road pixels along a row can be assumed a Gaussian distribution about a certain disparity. A disparity histogram along the rows of the image highlights the Gaussian distribution. The disparity histogram for all the image rows can efficiently be represented in a v-disparity.

Labayrade [9] first presented this concept describing the use of v-disparity images to model ground plane. u-disparity [10] and v-disparity represent the disparity histograms along columns and rows of a disparity image. The column number of u-disparity image corresponds to the column of the disparity image along which the histogram is calculated; the row number of the v-disparity corresponds to the row number of the disparity image along which the histogram is calculated. In both u-disparity and v-disparity images the intensity represents the number of pixels that share this disparity (i.e. the strength of the histogram) along the column and row respectively. Ideally the roads are assumed flat and horizontal and the vehicles as flat vertical structures perpendicular to the road surface. This highly abstract model of the surroundings has the mathematical implication that the road surface has constant disparity along the rows and the obstacles surfaces have constant disparity throughout. Hence ideally –

- The points sharing a disparity along a row should correspond to either roads or obstacles.
- The points sharing the disparity along columns are exclusively obstacles.

The above two postulates can be put to effective use with the help of v-disparity and u-disparity as shall be illustrated in this report. What follows is the construction of v-disparity image as detailed in [9]. Let the disparity image generated from stereo images be represented by I_{Δ}. Let the v-disparity image be represented by $I_{v_{\Delta}}$. The intensity value at the position (i,j) for an image Img is represented as $Img(i,j)$; where i corresponds to the image column while j corresponds to the image row. The v-disparity is calculated as follows.

$$I_{v_{\Delta}}(i,j) = \sum_{k=1}^{I_{\Delta}\ columns} F(i,j,k) \tag{16}$$

Where the function F is defined as

$$\begin{aligned} F(i,j,k) &= 1\ if\ \{\, I_{\Delta}(k,j) = i \,\} \\ &= 0\ otherwise \end{aligned} \tag{17}$$

The v-disparity images are generated from the disparity images. The intensity of the points in this image is equal (or proportional when scaled) to the number of pixels that share this disparity in the same row. Hu and Uchimaru [10] further these concepts to generate u-disparity maps similar to the v-disparity. Figure 12 presents an illustration for u and v-disparity.

The window in top left shows a disparity map of a corridor with intermittent cavities in both side walls. On the corridor floor a solid sphere, cone and a rectangular block are placed with increasing depth from the camera. Using this disparity image the u-disparity and v-disparity images are presented in the top right and bottom left windows respectively. Neglecting the roof of the corridor, this disparity map is a good model of the actual driving situations. The floor can be analogous to the road plane in front of the ego-vehicle. The walls on either sides of the corridor can be building, rail guards or vehicles in adjacent lanes. The objects on the floor are analogous to obstacles in the path of the vehicle. The cavities can be the cross roads to the ego-lane. These features have complex representation in the 3D world making it difficult to hypothesize their presence. Here lies the advantage of the u-disparity and the v-disparity. The road/ground plane presents itself as a lower bound to the v-disparity map. With the exception of roads that have adjacent railway tracks (with elevation lower than that of road), adjacent footpaths (with elevation lower than that of road), this assumption is true for most of the real world situations. The obstacles' surfaces perpendicular to the ground plane represent a

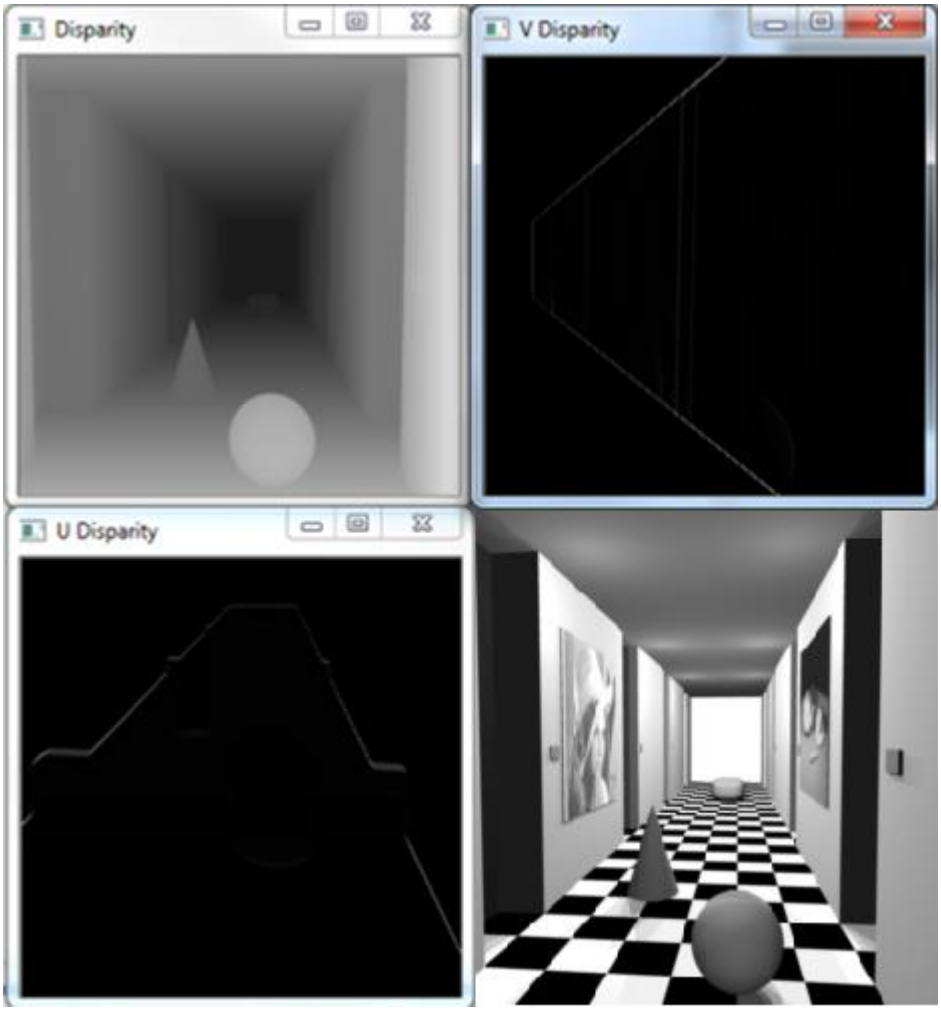

Figure 11. *Illustration of u-disparity (bottom left) and v-disparity (top-right)generated from the disparity image (top-left); real image (bottom right)*

collection of points vertical in the v-disparity image. Intersection between the road plane in v-disparity and the extrapolated obstacle pixels gives us the contact point of the

obstacle to the ground plane. The distance of the ego-vehicle from this obstacle can be calculated with this contact point location. The side walls present themselves as horizontal lines in u-disparity image. These side walls are analogous to vehicles (especially trucks with containers that have vertical surfaces) in adjacent lanes in real world frames. It has been proved in this report that for low camera pitch angles a flat horizontal road presents itself in the v-disparity image as a straight line. Hence we fit a straight line to the v-disparity image to find the road/ground plane.

2.10.1 Derivation of equation for road pixels in v-disparity -

The equations that relate image pixels (u, v) to their corresponding location in the world frame (X, Y, Z) are derived with certain assumptions in [9]. We present these equations below –

Figure 12.

$$u = \frac{\alpha X + u_0\,(Y + h)\sin\theta + u_0\,Z\cos\theta - ((\alpha\varepsilon_i\,b)/2)}{(Y + h)\sin\theta + Zcos\,\theta} \quad (18)$$

$$v = \frac{(Y + h)(\alpha\cos\theta + v_0\sin\theta) + (v_0\cos\theta - \alpha\sin\theta)Z}{(Y + h)\sin\theta + Zcos\,\theta} \quad (19)$$

$$\varepsilon_i = -1\ for\ left\ stereo\ images\ and + 1\ for\ right. \quad (20)$$

$$d = \frac{\alpha b}{(Y + h)\sin\theta + Z\cos\theta} \quad (21)$$

Standard assumptions in derivation of equations 18 & 19 are –

- Stereo image planes are parallel and at the same height w.r.t the world coordinate frame
- Camera 'roll' and 'yaw' angles w.r.t the ground frame is zero.
- Camera focal length and sensor pixel density is same along the horizontal and vertical axes

The parameters above correspond to frame attributes indicated in Figure 13. The 3 coordinate frames are represented by R_a- road frame ,R_{cr} - right camera frame & R_{cl} - left camera frame.

θ represents the angle between optic axes of cameras and horizontal (Pitch angle). h is the height of the camera from the ground surface. b (stereo basis) is the distance between the stereo cameras.

The image coordinates of the projection of the optical center will be denoted by (u_0, v_0). Camera focal length expressed in pixels as α.

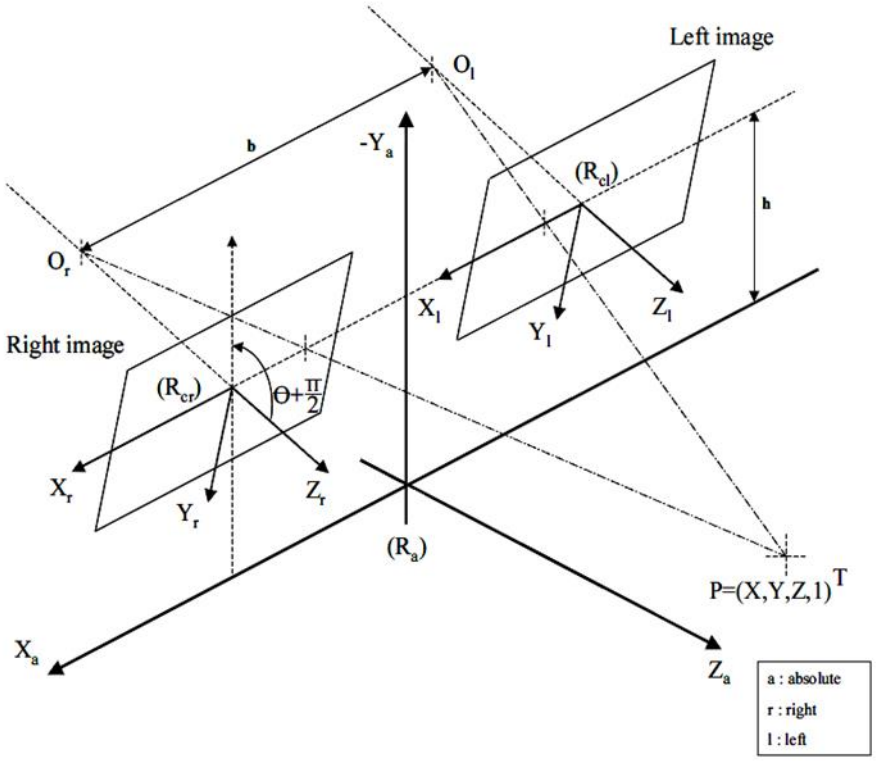

Figure 13. *Coordinate frame assignment and relevant parameters*

Assuming the road is flat i.e. Y=0 greatly simplifies the above equations. The partial derivative of row v against the Z coordinate can be derived as –

$$\frac{\partial(v)}{\partial(Z)} = \frac{-h\alpha}{(h\sin\theta + Z\cos\theta)^2} \tag{22}$$

Also since disparity is a function of depth Z as described in equation (21), we can write the partial derivative of disparity w.r.t depth Z as –

$$\frac{\partial(d)}{\partial(Z)} = \frac{-\alpha b\cos\theta}{[(Y+h)\sin\theta + Z\cos\theta]^2} \tag{23}$$

Using this equation we find the partial derivative of depth ‘Z’ w.r.t ‘disparity’ –

$$\frac{\partial(v)}{\partial(disparity)} = \frac{\partial(v)}{\partial(Z)} \cdot \frac{\partial(Z)}{\partial(disparity)} = \frac{-h\alpha}{(h\sin\theta + Z\cos\theta)^2} \cdot \frac{[(Y+h)\sin\theta + Z\cos\theta]^2}{-\alpha b\cos\theta} \quad (24)$$

For datasets that have cameras on vehicle at zero inclination i.e. $\theta = 0$. The above equation (24) is simplified to –

$$\frac{\partial(v)}{\partial(disparity)} = \frac{h}{b} \quad (25)$$

In other words the set of points in the v-disparity image that correspond to road pixels appear as a line with constant slope. This is true for certain datasets like the one provided by KIT [14], [15].Therefore detecting the ground plane is equivalent to detecting a line in the v-disparity map.

2.10.2 Detection of road surface in v-disparity map

The generated v-disparity images are treated with a Canny filter and then fed into the Hough transform. OpenCV Hough converts the given intensity image into a binary image with all non-zero pixels represented as 1 and then finds the lines within such image. Since we are interested in only the lines that correspond to pixels which high intensity (high intensity corresponds to disparity shared by a larger fraction of row pixels) we use Canny filter to eliminate majority of the low intensity pixels. It has been proved that for a flat horizontal road, the road pixels constitute a straight line in v-disparity [9]. Hence we detect lines in v-disparity to find the road pixels in images. OpenCV function for progressive probabilistic Hough Transform HoughLinesP [12] is used to generate possible candidates for lines in v-disparity that could represent the ground surface. A score is assigned to each candidate line. This score is the summation of intensity of v-disparity pixels lying directly on the candidate line. The line with the maximum score is elected to best represent the ground plane. Disparity of the image pixels is compared with the 'expected' disparity of the road pixels at the pixel location. If the difference in these two disparities is within a fixed tolerance, then the pixel is highlighted as a road. Figure 14 presents the first attempt at road surface estimate. Note the ground plane is highlighted with negated disparity (if it is estimated that the pixel belongs to road, the disparity of the pixel is inverted). Thus stark contrast in the left frame of Figure 14 represents the road/obstacle boundary.

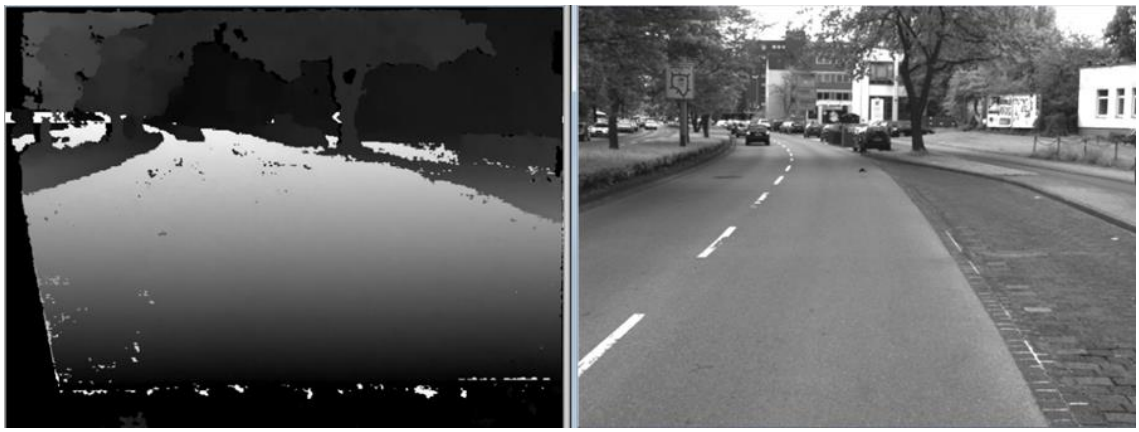

Figure 14. *Road plane estimate inverted in disparity image (left)*

Figure 15. *Poly line fitting of v-disparity (left) and road surface highlighted in yellow in the right image.*

The roads that are not flat will not generate a straight line in v-disparity. To accommodate the non-linear v-disparity cloud of points that result from such roads a poly line fit to v-disparity image is employed; this gives a better representation of the road than the single line fit [9]. Hence the Probabilistic Hough transform was fed sections of the v-disparity rather than the whole image. The results of this modification are presented in Figure 15. Note that the Hough transform is not obligated to generate line that span the entire width of the window. This is the reasoning for presence of some gaps in the ground plane estimated in Figure 15.

For an image row, the road pixels are almost always the farthest points (from the camera) when compared to other pixels in this row. Mathematically this translates to them having the lowest disparity.

Figure 16. *poly line fitting to minrow v-disparity (left)and colored road estimate*

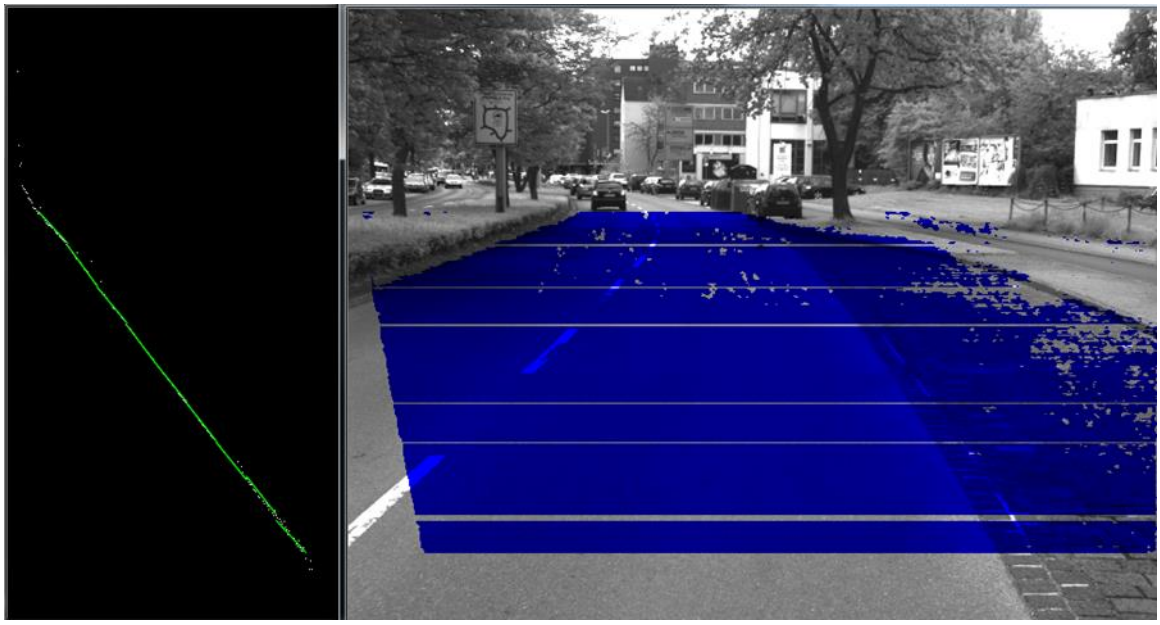

Figure 17. *Extrapolated poly line fitting to minrow v-disparity (left) and road surface estimate highlighted in blue on the right*

Thus one can assume that the points with the lowest non-zero disparity along each row of v-disparity correspond to the road pixels. With this assumption the v-disparity points with lowest disparity along each row are isolated and then fed into the probabilistic polyline Hough transform. Figure 16 presents the road plane estimated with this assumption. The image formed by accumulating the minimum disparity along each row will be referred to as the ‘minrow’ within this report.

The Hough lines detected in each segment shown in Figure 16 were further extrapolated to fill the gaps seen in the road plane estimated. Figure 17 presents the extrapolated lines and the corresponding road plane estimated. Note that there are still some gaps

visible in the estimated road plane due discontinuities persisting between the polylines extrapolated (one such discontinuity can be seen at the last Hough line in the minrow image).

An alternate approach to the road estimation is by obviating the use of Hough transform entirely. The minimum disparity along each row (referred to as minrow) is used as a nominal measure of the disparity for road pixels along that particular row. Hence an alternate representation of road pixels in v-disparity will be the pixels that correspond to the lowest nonzero disparity along each row. The middle frame in Figure 18 shows one such representation. The disparity of the road pixels is allowed a tolerance. This approach is faster than the above approach since we no longer use Hough transform in the estimation. Figure 18 presents the minrow estimation without use of Hough transform. With this assumption the ground surface is a locus of points in v-disparity that hold the least non-zero disparity along each row. Disparity increases with the point elevation in 3D space according to equations in [9]. Since obstacles/vehicles must always be on top on the road surface, they do have higher disparity compared to pixels in the same row and hence get eliminated in the middle frame of Figure 18 leaving behind the points that do correspond to roads. Note that a constant horizon row is assumed in subsequent report to limit the road surface estimate.

Roads which are not horizontal or flat, roads that have some banking angle do not have a constant disparity along the image rows. Instead the disparity of pixels along each row is spread across a finite bandwidth. The above minrow disparity approach fails miserably in such scenarios. Figure 19 presents an illustration where such behavior is observed. The left frame presents the v-disparity, the middle frame presents the minrow disparity and the right frame presents the estimate. All points below the fixed horizon row are assumed to correspond to road surface in the middle frame of Figure 19.

To accommodate such presentations of the road plane we can devise an adaptive tolerance to accommodate the road pixels. Another approach to handle such scenarios is to eliminate points in v-disparity that do not belong to the road plane; the points that belong to vehicles, buildings, trees, other traffic participants, and claim that the remaining points must belong to the road surface. We know that majority of such points protrude from the slanted line (or band in case of banked roads) representation of the road plane in v-disparity. We make a fair assumption that every point other than the road appear as near vertical clouds of points in the v-disparity image. Hence partial derivative of the v-disparity w.r.t the x-axis highlights these undesired points. We subtract these points from the absolute derivative and eliminate majority of the non-road points in v-disparity. Figure 20 presents an illustration of this approach.

On the left is the image of $\frac{(\partial(v-disparity))}{\partial u}$ and on the right is the image of $\frac{(\partial(v-disparity))}{\partial v}$. After subtraction of the left image from the absolute derivate we get the

image shown in left of Figure 21. This image still has some points scattered above the slanted line representation. We can discard the points that are outside a fixed tolerance around the nominal slanted line representation. This filtered v-disparity is used to generate road plane estimates. Figure 22 shows the same banked image as Figure 19. It is clear that this partial derivative approach provides better road plane estimates on banked roads.

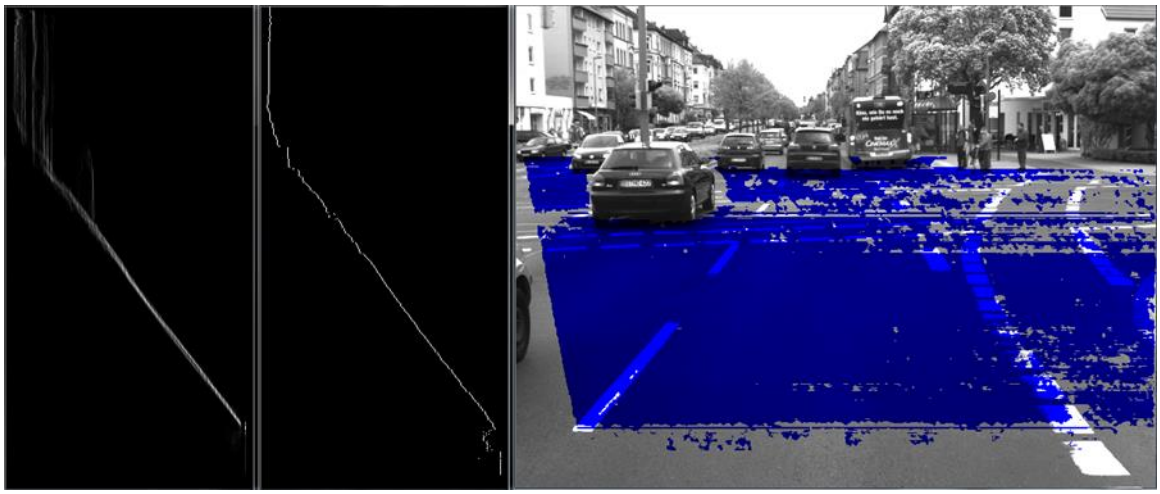

Figure 18. *Road plane estimation without Hough transform*

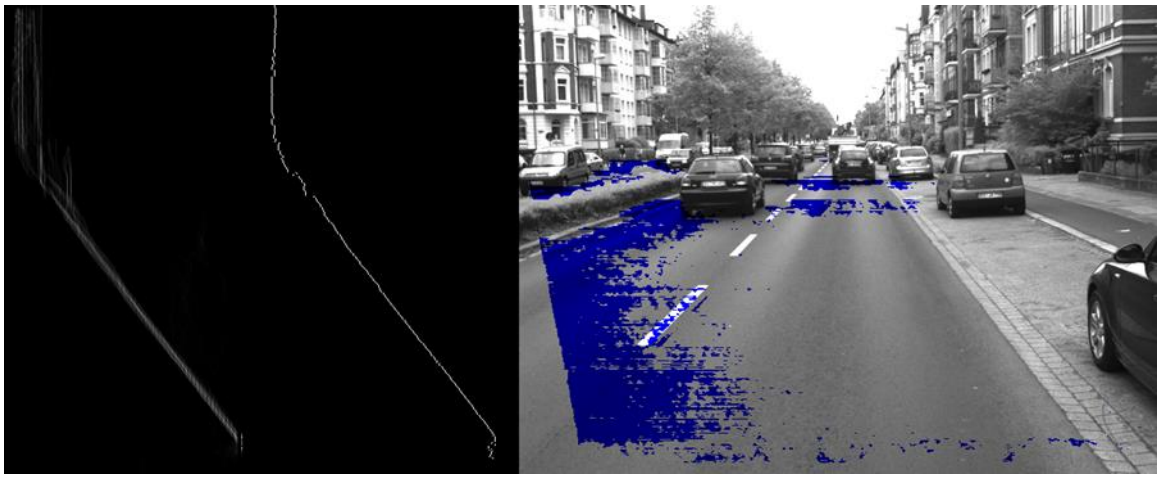

Figure 19. *Failure of minrow approach in roads that are not flat.*

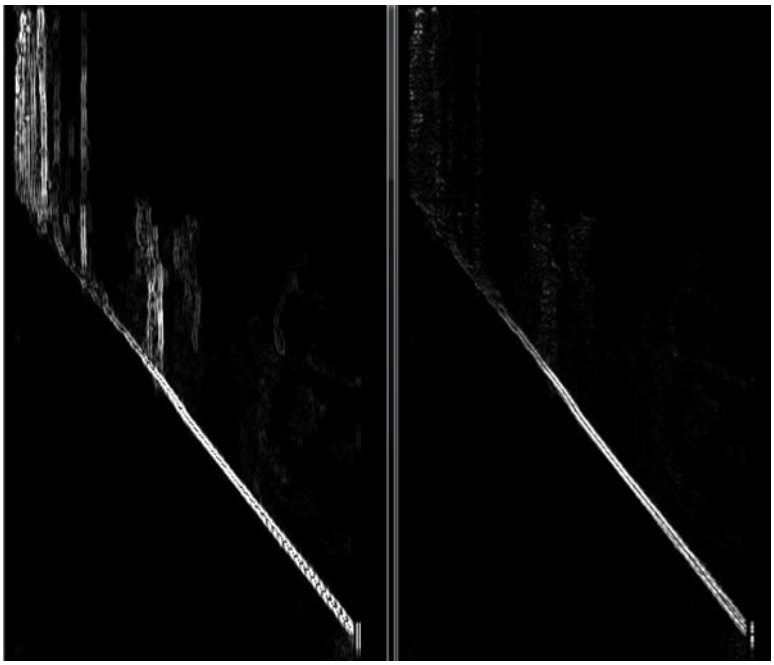

Figure 20. *Partial derivative of v-disparity w.r.t x on the left(highlights vertical structures) and w.r.t y on the right*

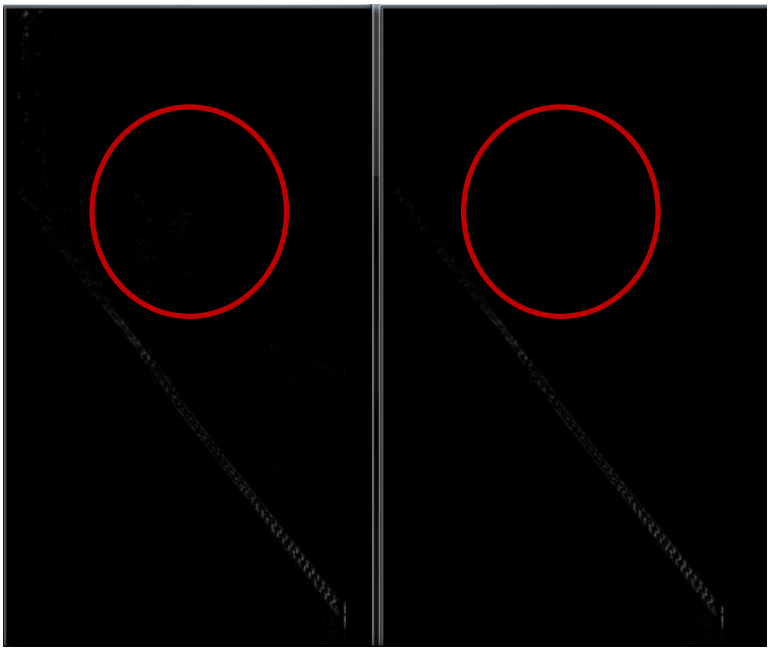

Figure 21. *Separation of partial derivative w.r.t x from absolute derivative of v-disparity (left) and after eliminating outliers (right). Red circle shows the elimination of outliers.*

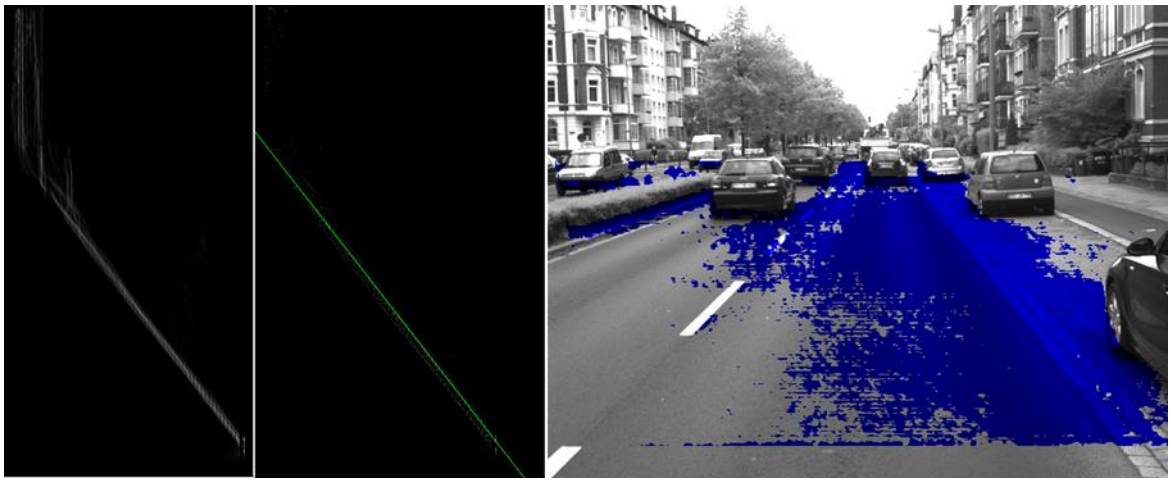

Figure 22. *Improved estimation of road surface with partial derivative separation*

2.10.3 Refined v-disparity post obstacle elimination

The state of ground plane estimates implemented so far has been summarized in Figure 23. These estimates are based on the disparity images corresponding to the stereo pair of the scene presented. Two of the above 3 algorithms (Top and bottom frame Figure 23) assume that roads in front of the ego vehicle are flat. This assumption is not always true. A recent publication [17] eliminates obstacles by thresholding u-disparity. Figure 24 presents a GUI with trackbar to set the obstacles' threshold in u-disparity. Figure 24 reflects changes when the threshold is updated so that a nominal threshold for obstacle elimination can be selected easily. Once the obstacles are eliminated as mentioned above, the disparity map is updated to discard pixels belonging to obstacles. This new disparity map is used to generate the v-disparity. Bottom left frame of Figure 24 and Figure 25 present such updated v-disparity. Notice that vertical could of points that are characteristic of the obstacles are significantly suppressed. Furthermore, [17] also presents an approach for estimation of horizon. It is based on the assumption that the v-disparity curve rises only up to the horizon, Figure 25 presents the horizon thus estimated as a green dot. To see how the ground plane estimation algorithms (whose results are shown in Figure 23) would fare with this updated disparity map, we plot the estimates using the original and updated disparity maps in Figure 26. The improvements in ground plane estimates can be observed around the obstacles themselves.

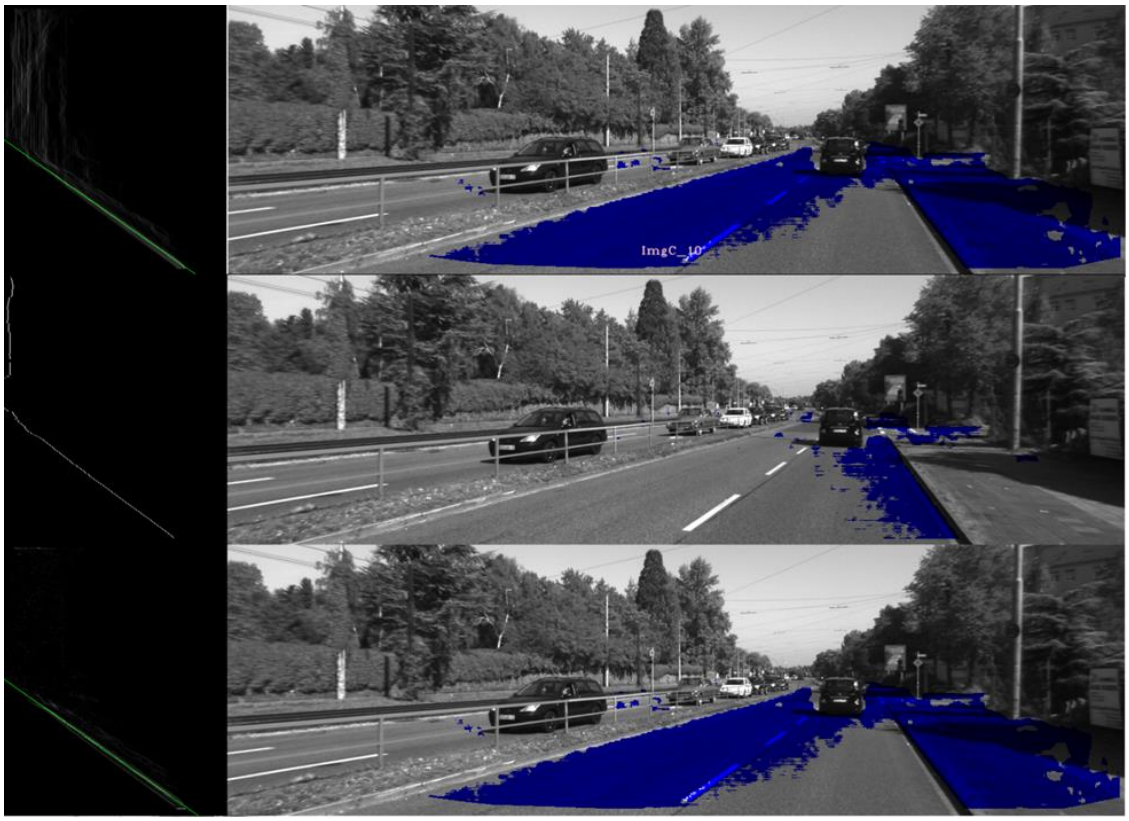

Figure 23. *The 3 ground plane estimate algorithms. Top frame - direct line fit to v-disparity, middle frame – minrow disparity estimate, bottom frame – line fit to filtered v-disparity.*

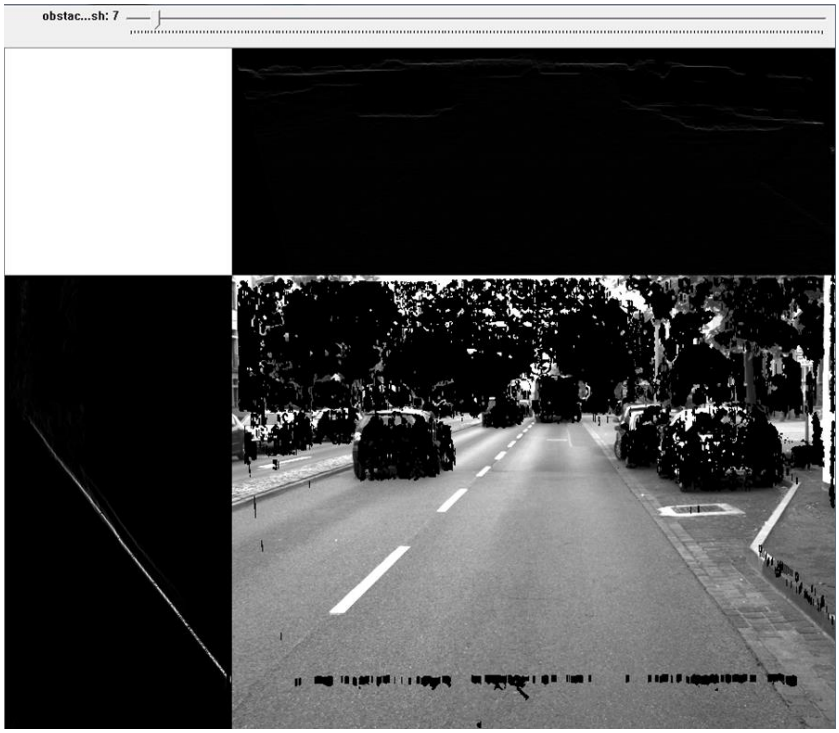

Figure 24. *The OpenCV GUI to set the obstacle threshold (trackbar at the top of frame) in u-disparity(below the trackbar to the right) and visualize change in the v-disparity (bottom left) and the real image (bottom right)*

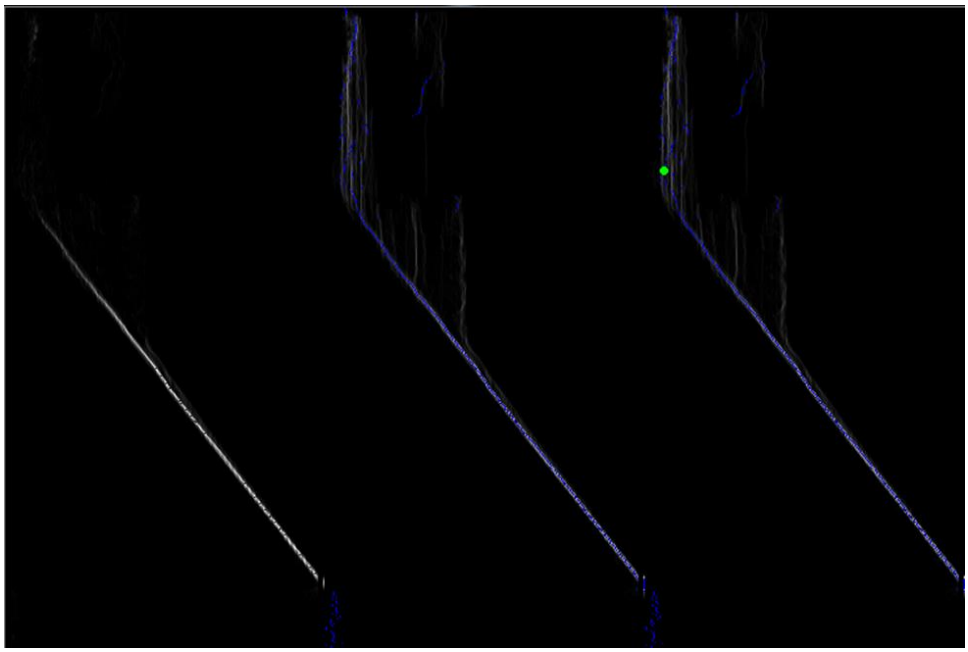

Figure 25. *Left frame presents the v-disparity after the obstacles are eliminated in disparity image. Middle frame presents the max intensity along each row in v-disparity as blue points. Right frame presents the horizon as the green dot. In all three frames, the ground profile as dictated in [17] is colored blue*

While the estimates with original disparity image tend to classify vehicle bumpers as ground plane, the estimates with the updated disparity map (post obstacle elimination) are much more disciplined around vehicles. Instead of assuming any particular road model (flat, quadratic or spline) the authors in [17] claim that once the obstacles are eliminated in disparity images, the maximum number of points that share a particular disparity belong to the road. Such points have the highest intensity in v-disparity and the author calls them the Initial Ground Profile (IGP). Figure 25 above presents the points corresponding to max intensity along each row (IGP) in v-disparity image as blue dots.

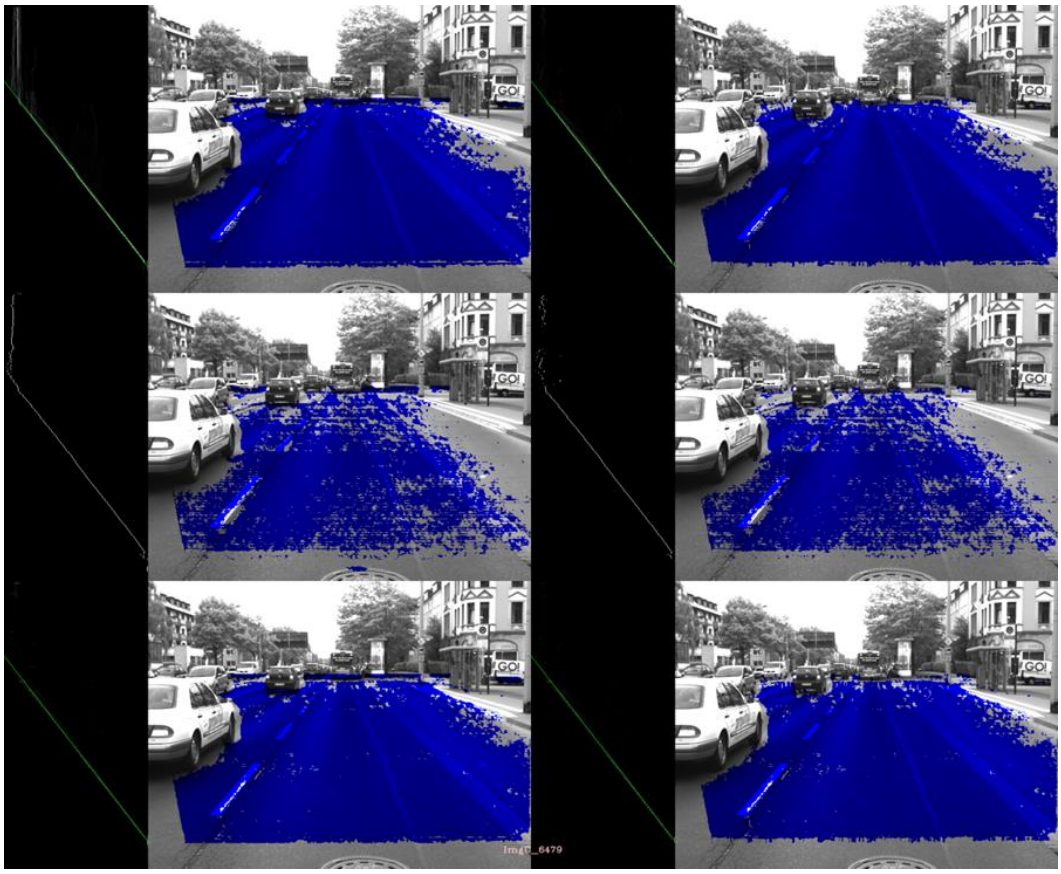

Figure 26. *The 3 ground plane estimates in the left half of image used the original disparity image for estimation. The 3 ground plane estimates in the right half of the image used the disparity images where the pixels corresponding to obstacles had been discarded.*

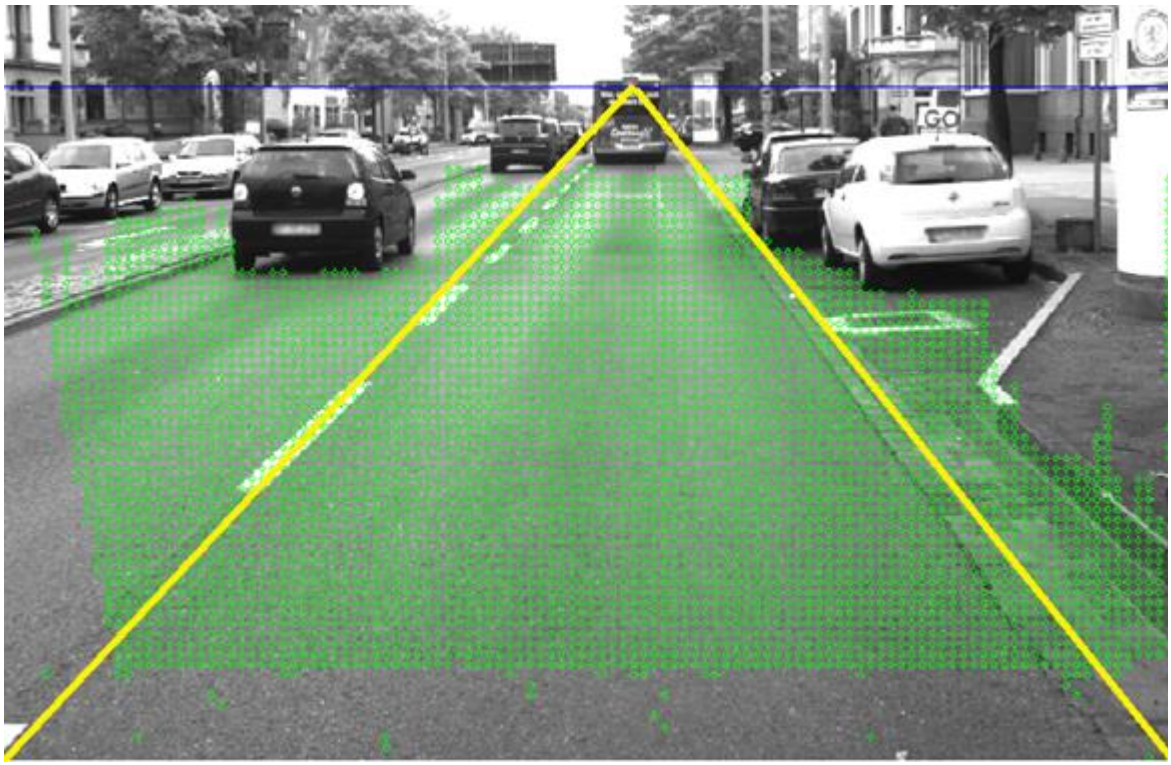

Figure 27. *Near vehicle triangle window (yellow) in front of ego vehicle to trigger warning. The road surface estimation is colored green..*

Figure 28.

2.10.4 Near vehicle warning function

An additional function that can be designed with the disparity data is to detect the closest obstacle (vehicle) in front of the ego-vehicle. This detection can serve to discard the ground plane detection in the situation where the distance to the front vehicle is too less. In such cases the ground plane will not be the most predominant part in the image and therefore violate the v-disparity assumption. A triangular window is spread in front of the ego vehicle and the population of road pixels in this window is studied. If the percentage of the road pixels in this window is below a threshold, a warning message is triggered. Figure 28 presents one such window in green in front of ego vehicle.

Figure 29 presents the warning on a real image. Note that since we have defined road estimate confidences (will be discussed in section 4.1) to be valid only when the max intensity along the v-disparity row is higher than a threshold, fewer rows have confidences defined in this figure. This leads to a lower average confidence for the entire estimate. Since the vehicle search window in Figure 28 is fixed, on a curved road the vehicle in front of the ego vehicle will be detected much later than when it were to approach head on. A more effective window will be one that tracks the ego-vehicle trajectory (by tracking the steering wheel and assuming the driver is not so aggressive that the vehicle skid is significant) and adapts the lane window accordingly. A more compact representation of road surface estimate can be made by highlighting the entire estimate with a single color (rather than a color for each row as represented in Figure 50 & Figure 51). Estimates represented in Figure 30 serve this purpose. One additional feature to be implemented in our road surface estimates is the filtering of the maximum depth for road surface estimation where the confidence is greater than a threshold. The reason for filtering the maximum depth is to avoid random loss of free space due to uncertainty of the disparity map and to have a smoother variation of the associated free space.

Figure 29. *Near vehicle warning, also note the lower number of confidence rows*

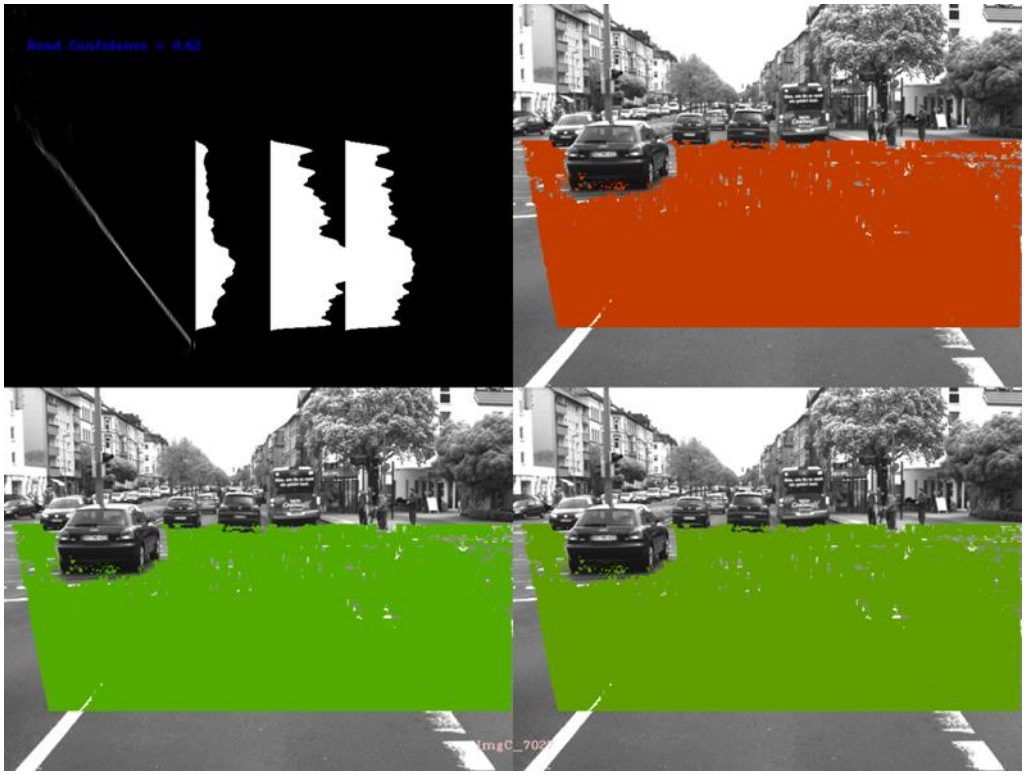

Figure 30. *Unified confidence representations with filtered max estimation depth*

2.11 Occupancy Grid Map (OGM) free space estimation approach

v-disparity approach prophesizes the ground surface as pixels that share similar disparity. The estimations with this approach are good as long as the pixels along rows have similar disparities. Since disparity is a function of the depth Z and height Y as seen in equation (21)), we can say that as long as the road surface plane is parallel to the x-axis, the v-disparity approach produces good results with relatively low computational footprint. This makes the v-disparity approach in freeway environment a very attractive option. Furthermore the v-disparity approach is one of the most simple and robust algorithms for road surface detection.

In the urban environment however the assumptions that are made to realize v-disparity approach can often fail. The free space estimation approach looks at the same problem of road surface estimation from a different perspective. It prophesizes that road surface up to an obstacle can be classified as free space. Clearly the accuracy and validity of this model depends on how well one can detect the obstacles. Obstacles and road surfaces have complementing characteristics in real world and retain this trait in image space. For instance the road surfaces are usually horizontal, while the obstacles usually present themselves as vertical surfaces (assuming flat obstacle rear); subsequently while road pixels share disparity along the image rows, the obstacles share disparity along the image columns. When we take the case of a banking road or a road that has a twist about the z-axis, we can easily visualize that the road pixels along image rows do not have the same elevation Y and depth Z and hence different disparity. Clearly the v-disparity approach is at a disadvantage here. But when you look at the definition of obstacles in such scenarios particularly the obstacles' vertical surface will remain vertical even on the twisted road. This implies that limiting the road surface up to the obstacles is a better approach than to rely on v-disparity approach.

Note that we had already carried out a step of "crude obstacle separation" in the v-disparity approach to ground surface estimation as suggested in [17]. The beauty of this approach is that the v-disparity image is generated after elimination of obstacles' pixels from disparity image. In short it is a fusion of free space estimation as carried out by OGM and the v-disparity approach as detailed in [9].

So far we have avoided the transformation of image points to the 3D world coordinates. There have been publications [18], [5], [19], [4] where the authors suggest the use of the 3D world coordinates of image pixels (using especially the Z axis which stands for the real depth of obstacle from the ego vehicle) to generate the so called Occupancy Grid Maps (OGM). A generic occupancy grid map has been detailed in Figure 31. And in laymen terms OGM is like a 'top view' of a 3D scene. Note the convention that the space limited by the segmentation (black pixels) is considered free space (colored white) while that beyond the black pixels as unknown and is rendered unknown (gray).

Note also the fact that the OGM x and y axes correspond to real world coordinates. And the OGM is based solely on the view presented in Figure 31(a) which corresponds to a conical 3D world projection. Each cell in OGM corresponds to a distinct space in 3D world coordinates. Another simple way of looking at OGMs is that they are the projection of the cloud of 3D points that are seen by a camera on a flat horizontal surface. The impression made on each point in this flat horizontal surface is proportional to the number of 3D world points (seen by the camera) that exist directly above the surface. Note that we consider only the points that are seen by the camera when building the OGM, because our perception of the world is made possible solely by (and hence limited to) the camera images. Furthermore due to the perspective effect, the density of the 3D points captured by camera images is higher for entities closer to the camera than those farther. This nature of the world information capture by camera subsequently makes the information represented on the OGMs non uniform (since the closer cells in OGM are influenced by more 3D points and the farther cells are influenced by fewer 3D points).

Daimler has also published some papers [21], [8] where they construct abstract world representation called 'Stixel World' by extracting meaningful data from the Occupancy grid map (OGM). The 3D world data is squeezed onto a 2D plane (the OGM). The occupancy grid consists of discrete cells that do not intersect with one another. The OGM cells have intensity proportional to the number of 3D points that exist within the imaginary region formed by extending the OGM cells both vertically upwards and downwards. The area of the cells although constant in itself, can map to non-uniform regions in 3D space.

2.11.1 Math involved in generating OGM

A measurement (m_k) is a vector defined as $[u\ v\ d]^T$ where u, v are the image pixel columns and rows respectively and d corresponds to the disparity associated to this pixel. This measurement originates from a ray of light after reflecting on a point in real world at $p_k = (x\ y\ z)^T$.

Solving simultaneous equations 19 & 21 we get –

$$Y = \frac{b}{d}[\,(v - v_0)\cos\theta + \alpha\sin\theta\,] - h \tag{26}$$

$$Z = \frac{b}{d}[\,(v_0 - v)\sin\theta + \alpha\cos\,\theta] \tag{27}$$

And back substituting 26 & 27 into 18 yields –

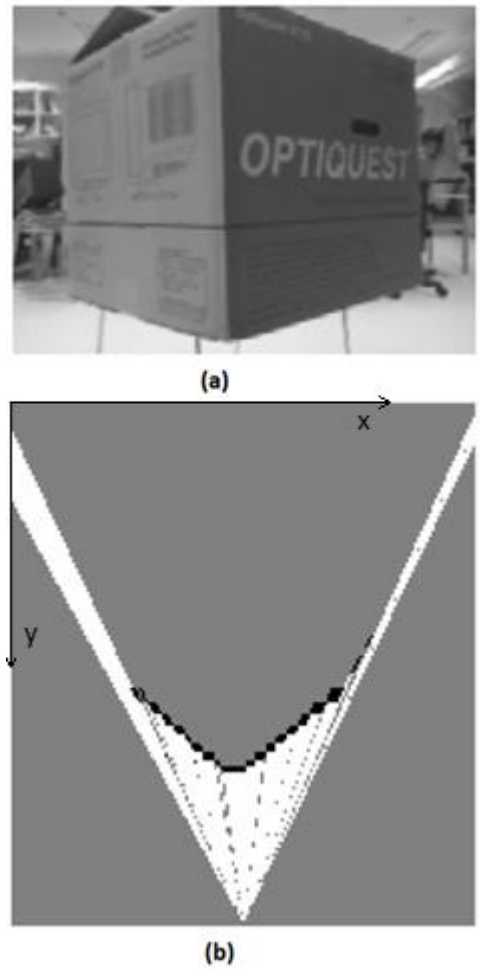

Figure 31. *OGM illustration [5]; (a) obstacle on surface, (b) corresponding Cartesian OGM*

$$X = \frac{b[\,2(u-u_0)-d\,]}{2d} \tag{28}$$

A Gaussian function of vector error δ_k resulting from measurement m_k is defined as follows –

$$G_{m_k}(\delta_k) = \frac{1}{2\pi^{3/2}|C_k|}\exp\left[-\frac{1}{2}{\delta_k}^T {C_k}^{-1}\delta_k\right] \tag{29}$$

The Occupancy likelihood (as the name implies) is a number that stores the current occupancy status for a particular cell in OGM. It is denoted by D(i,j). Each image pixel influences occupancy in the entire OGM, strong occupancy likelihood in some cells

while weak in others. The occupancy likelihood for the entire disparity image is the summation of the occupancy likelihood resulting from each pixel. In other words, each pixel of disparity map produces an occupancy likelihood image, and we add all such images to arrive at the OGM for the entire disparity image. The following equation highlights this concept.

$$D(i,j) = \sum_{k=1}^{rows.cols} L_{ij}(m_k) \tag{30}$$

2.11.2 Column-disparity map

In the column disparity OGM columns correspond to the columns of the disparity image, the grid rows correspond to the pixel disparity. This grid has the same axes layout as the u-disparity map. The occupancy likelihood arising from a single pixel is a Gaussian of error vector. The elements of this vector are - the difference in column, 0 & difference in disparity as indicated in the square bracket below. Every pixel measurement m_k produces the occupancy likelihood for every cell of the OGM. The occupancy likelihood at (i,j) is calculated as –

$$L_{ij}(m_k) = G_{m_k}([u_{ij} - u, 0, d_{ij} - d]^T) \tag{31}$$

The intensity of each OGM pixels/cell (each pixel corresponds to a cell) is given by $D(i,j)$ as in equation (30). The intensity in OGM referes to the occupancy likehood, in other words if there is an obstacle in a location in 3D space, the corresponding cell in the OGM reflects its presence through high intensity. Figure 33 (a) presents the column-disparity OGM (bottom) and its resemblance to u-disparity (middle).

2.11.3 Polar OGM

In [4] the author argues that the use of real distance in contrast to the disparity for generating OGM is preferable, since the OGM with disparity as row (or y-axis) does not provide an intuitive representation of the free space. This inconvenience arises due to the fact that the disparity is non-linearly dependent on the real distance as can be seen with equation 21. Figure 33(b) presents an illustration where the same scene is represented with real image, u-disparity, col-disparity OGM and polar OGM (ordered from

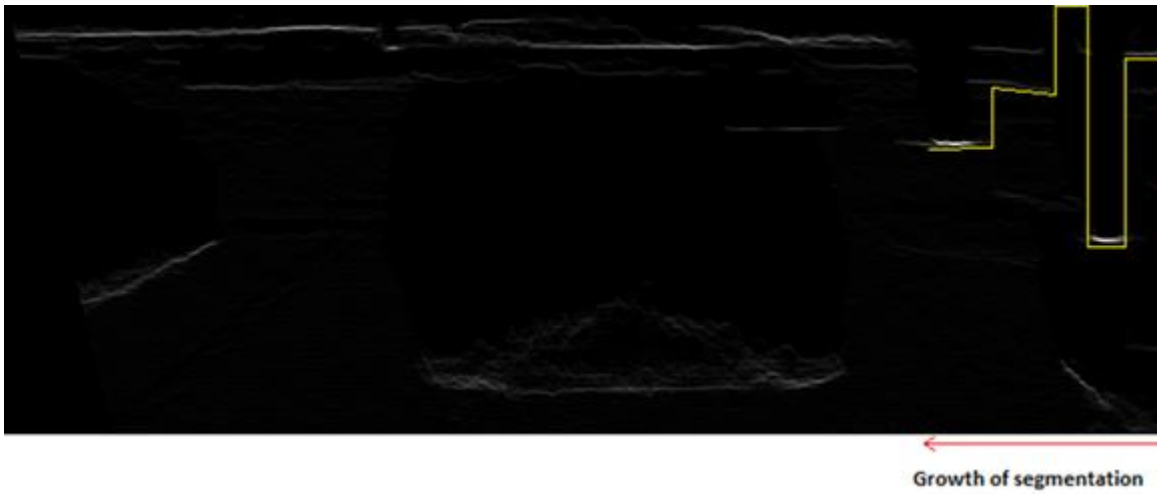

Figure 32. *A stage in dynamic programming. Yellow line indicates the optimal segmentation path from the current column to the last column*

top to bottom). In Figure 33(b) notice that although the polar OGM represents occupancy with real distance along rows (or y-axis); it suffers from lack of data along certain rows. This is attributed to the fact that there are points in the real world which do not have corresponding pixels in the image. The authors in [2] suggest filling data from neighboring cells into these empty rows. Since we are transforming data from the disparity space to the real space, the computations involved in Polar OGM will be more than that involved in column-disparity OGM and u-disparity-OGM. Mathematically this transformation has be expressed in the following equations.

Polar OGM is generated with the following likelihood function –

$$L_{ij}(m_k) = G_{m_k}(\left[u_{ij} - u, 0, d'_{ij} - d\right]^T) \tag{32}$$

Where $d'_{ij}$is the disparity corresponding to the depth j of the OGM cell. This can be calculated from equation 21 while assuming Y=0. The Polar OGM x-axis is the same as the columns of the image. The y-axis of the OGM corresponds to the real distance of objects w.r.t the ego-vehicle.

2.11.4 OGM Segmentation using Dynamic programming

Once the OGM is generated as described above, we need to segment it so as to estimate free space. Although we could select the maximum intensity pixels along each column as done in [19], the authors in [4] suggest the use of ‘dynamic programming’ to segment the image. The book - Applied Mathematical Programming [22] details the dynamic programming through an intuitive illustration and subsequent implementation.

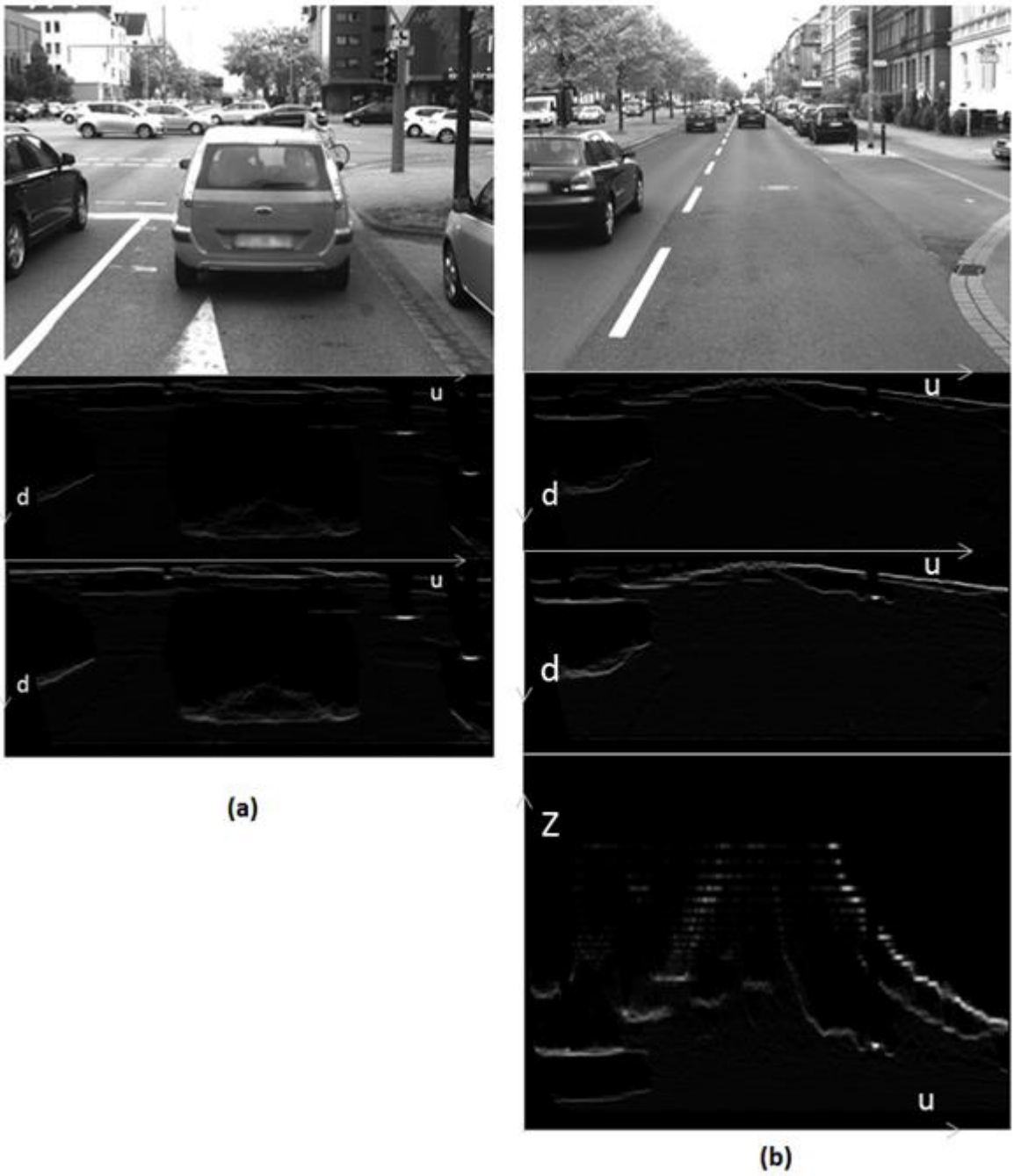

Figure 33. *(a) Similarity between u-disparity OGM (middle) and column-disparity OGM (bottom) (b) Perspective change in Polar OGM(bottom) in contrast to u-disp (2nd from top) and col-disp (3rd from top)*

The objective of the dynamic programming is to segment the OGM while minimizing a cost function. This cost function is dependent on the intensity of the pixels in OGM and a spatial continuity factor which penalizes a jump in depth (in other words the cost penalizes jumps in rows while segmenting the OGM).

There are 3 important features of dynamic programming –

Figure 34. *Polar OGM top right and dynamic segmentation in blue (bottom right)*

Stages: The image (or OGM) segmentation problem is broken down into different stages. The segmentation grows from right end of image and concludes when it meets the left end (this choice is strictly arbitrary). In the image segmentation problem we assume the 'stage' as the number of columns that have been segmented. Figure 32 illustrates a stage in segmentation by dynamic programming.

States: The states reflect the information required to infer the consequences of a decision made at this stage. In our case, this information includes the optimal paths starting at any fixed row in the current stage (column). This information is updated along every column of the segmentation growth.

Recursive optimization: This is the loop that cycles through all the columns of the image and generates the state data along each stage. At the end of this loop we end up with a set of optimal paths starting at each row of the first column. We segment the image from the row in the column that carries the lowest cost. In the recursive loop, we update the cost vector (state) of the segmentation which indicates the cost to segment the image starting at a fixed initial row.

$$Segmentation\ State = S(column, row)$$

Recursive optimization loop –

1. $\forall\ column = n - 1{:}\,1 \rightarrow calculate\ \ S(column, row)$
2. $where\ S(i,j) = \min_{k=1:rows}[cost(\ (i,j),(i+1,k)) + S(i+1,k)]$
3. $\&\ cost(\ (x_1, y_1),(x_2, y_2)\ )\ calculates\ the\ cost\ incurred\ as\ in$ [4]

Figure 34 presents the segmentation for the polar OGM. The intensity in OGM is proportional to the number of pixels that 'share' or lie in the vicinity of a particular disparity (in case of u-disparity & col-disparity OGM) or depth (in case of polar OGM). The

cost function for dynamic programming depends inversely on this intensity of the OGM pixel. This implies that the dynamic programming is more likely to segment the OGM through higher intensity pixel than lower intensity pixels along the same column provided all other conditions are the same. Figure 35 presents the situation where the yellow windows correspond to the building pixels in real image and their corresponding OGM spread. The red window corresponds to the vehicle pixels and their corresponding spread in OGM. The Blue pixels correspond to the segmentation achieved with dynamic programming. Note that the segmentation indicated in Figure 36 prophesizes free space until the building ignoring the vehicle immediately before. To overcome this shortcoming it is suggested in [21] to discard pixels in OGM after the first maxima above a particular threshold along each column.

Although this additional step does tend to alleviate the problem observed in Figure 35, it renders the dynamic programming redundant to some degree. The purpose of the dynamic programming was to define segmentation for each column against all available possibilities. During background subtraction the first obstacle along every column is prophesized to be the first maxima along column that has intensity above a certain threshold. Pixels beyond this row are cleared. A good assumption is that these local maxima pixels in OGM correspond to the obstacles that limit free space. This further saves the processing time without considerable loss in estimate quality.

One additional problem is selection of a threshold for the first maxima along each column of OGM in background subtraction. Nearer obstacles have a larger perspective appearance both in real image and the disparity image and since OGM intensity is proportional to the number of pixels sharing a distance/disparity, these obstacles tend make a stronger impression in OGM than farther obstacles. This implies that a larger column threshold would suit the nearer obstacles and vice versa. A constant maxima threshold for every column in background subtraction thus makes little sense. Also since along each column of OGM we can expect multiple ‘spikes’ corresponding to multiple vertical structures, this is a multilevel thresholding problem.

Urban road environment includes trees, traffic sign posts, landmark boards etc. Since these boards present themselves in vertical plane, the pixels corresponding to these boards have nearly the same distance and hence similar disparity. Figure 34 presents an instance where the overhead signpost presents itself in the polar OGM and consequently limits the free space to the false extent. A more meaningful definition of obstacle is one that defines vertical structures that protrude from the ground surface as obstacles. Similar observations have been made in other instance where traffic sign posts, trees, etc are present. To overcome this nuisance, we limit the rows of the disparity map which are used to derive the OGM. In our case a manual threshold has been set beyond which the disparity pixels influence OGM.

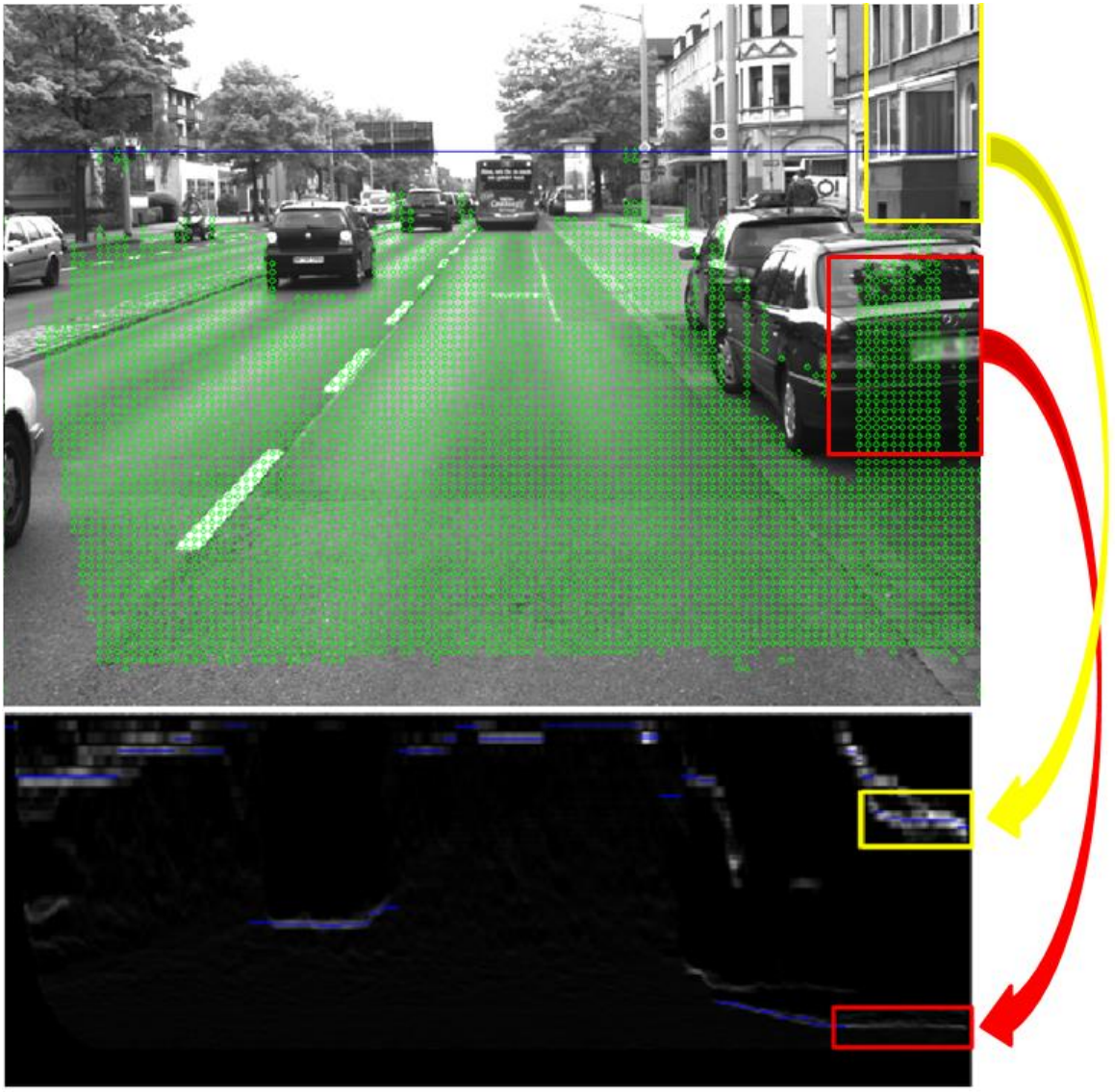

Figure 35. *Failure of dynamic programming to limit Free space upto pixels in red window. Instead the free space (green) is limited by building pixels in yellow window*

Note that the ground plane estimate in Figure 34 above is highlighted as blue circles. This representation of estimate does retain underlying pixels and helps in judging whether certain key obstacles have been completely enveloped/ignored by the estimation algorithm.

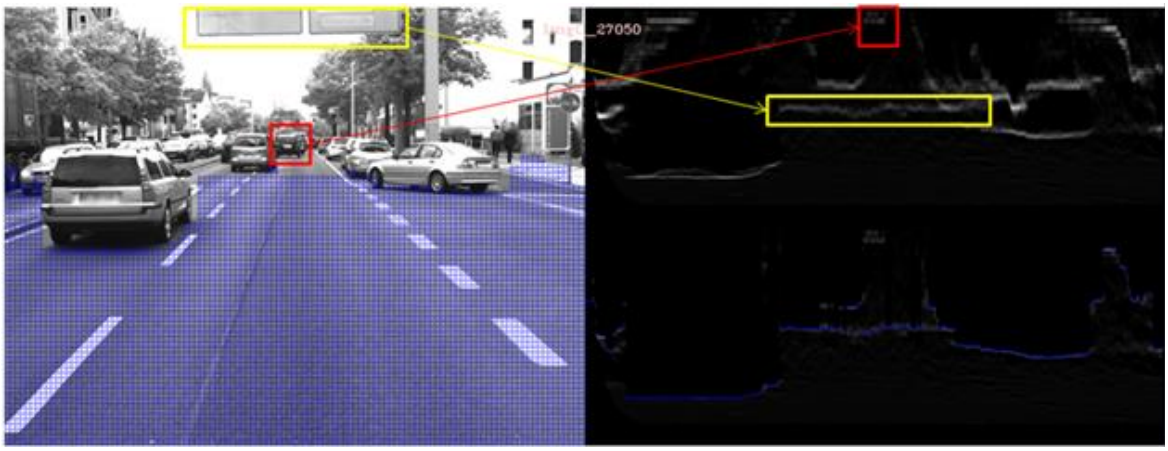

Figure 36. *Presence of flat structures above the ground surface (yellow windows) results in false free space estimation. Red window highlights the car pixels that should be limiting the front free space*

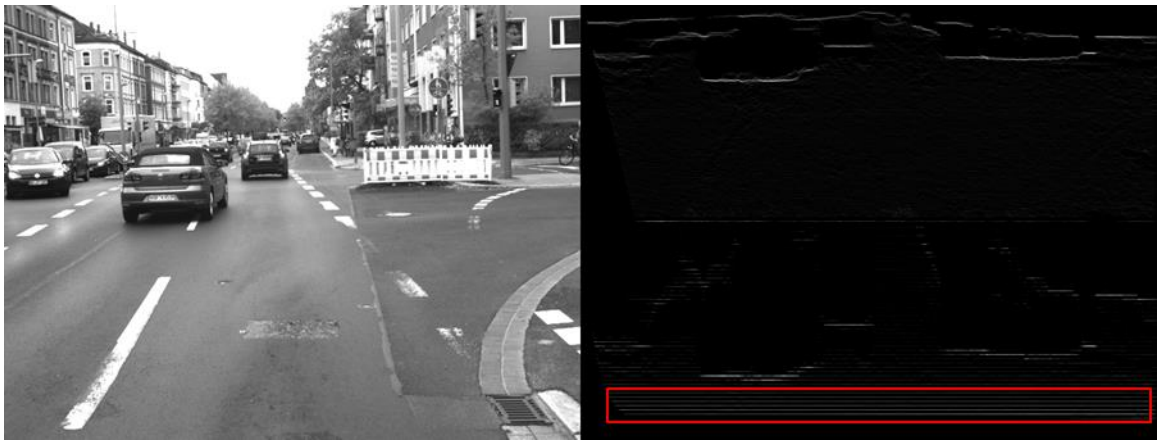

Figure 37. *Top right is the row limited u-disparity; bottom right is the u_Z map. Note the presence of blank between rows and high intensity pixels for large rows (in red window).*

2.11.5 Polar OGM vs u-disparity OGM

Although the Polar OGM is more disciplined around the obstacles, it suffers from poor data resolution for distant points. Furthermore the generation and subsequent use of polar OGM puts a larger load on processor than u-disparity. We have already presented an extended version of u-disparity that accounts for the shortcomings of both polar OGM and the u-disparity.

Another faster alternative to polar OGM is a scaled version of u-disparity. The pixels in u-disparity are scaled (along their rows or y-axis) from disparity to their depth in real world coordinates. This is the same coordinate frame representation as the polar OGM. Figure 37 presents the generation of u_Z from u-disparity for a particular road presentation.

Each row of u_Z map corresponds to 20cm real world length. While the columns in u_Z map correspond to the image column. Notice that in the u_Z map in Figure 37 above that at large row values, the intensity is comparable to that of the obstacles (which are the some of the brightest clusters). This is due to the nonlinear nature of the equation relating disparity and the real distance equation 21. This is illustrated in Figure 38 below. This table is generated by considering a flat horizontal road surface and generating the occupancy values at depths of 20cm each. The introduction of such pseudo noise (highlighted in red box in Figure 37) in u_Z map complicates the process of ground plane estimation. As previously discussed, the obstacle detection depends on identifying the first maxima above a certain threshold (background suppression).

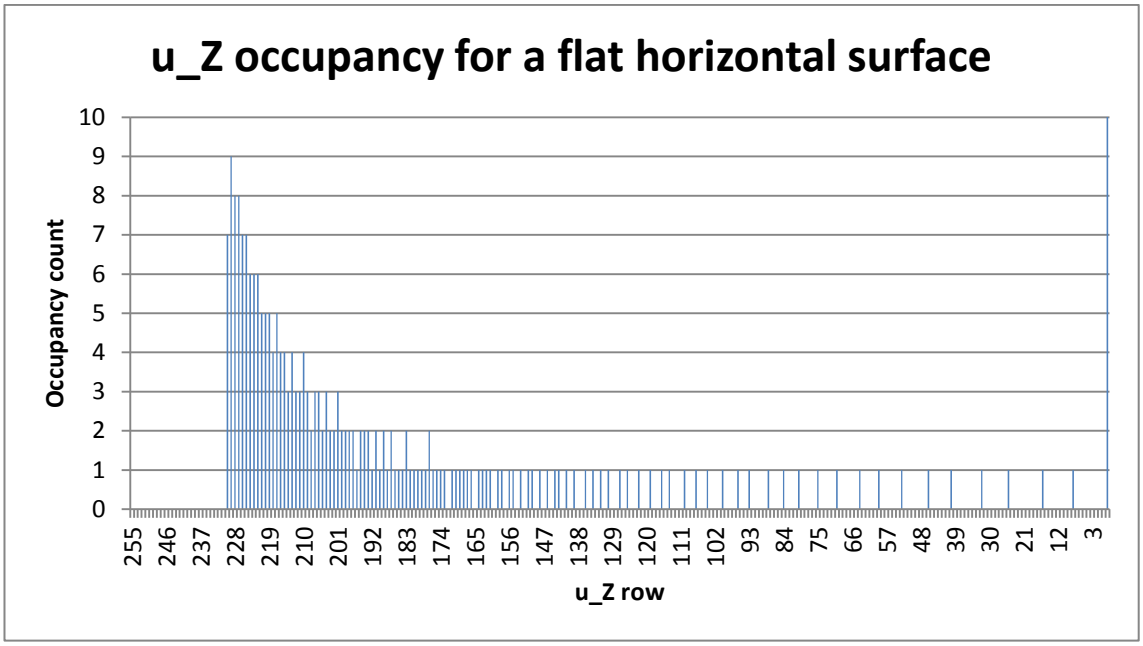

Figure 38. *OGM intensity plot for a column of flat horizontal surface without obstacles*

Furthermore the dynamic programming prefers to segment the image favoring higher intensity. Clearly the pseudo noise will hinder both these approaches. Ideally the obstacles present themselves in u-disparity as a cluster of high intensity pixels (roughly arranged along a line) with noise on one side and void on the other (due to occlusion). The size of the void depends on the vertical surface area of the obstacle. This feature has been used to devise a filter that can eliminate the noise appearing in u_Z map due to road pixels. Figure 39 presents the result of filtering of the u_Z with such a filter.

Although the filter works well in situations where rich disparity maps are available, it is counterproductive in situations where poor disparity maps are available. This is for example true for rainy days when the road reflects considerable light and the disparity images rendered for such frames is lacking information in large empty ‘voids’.

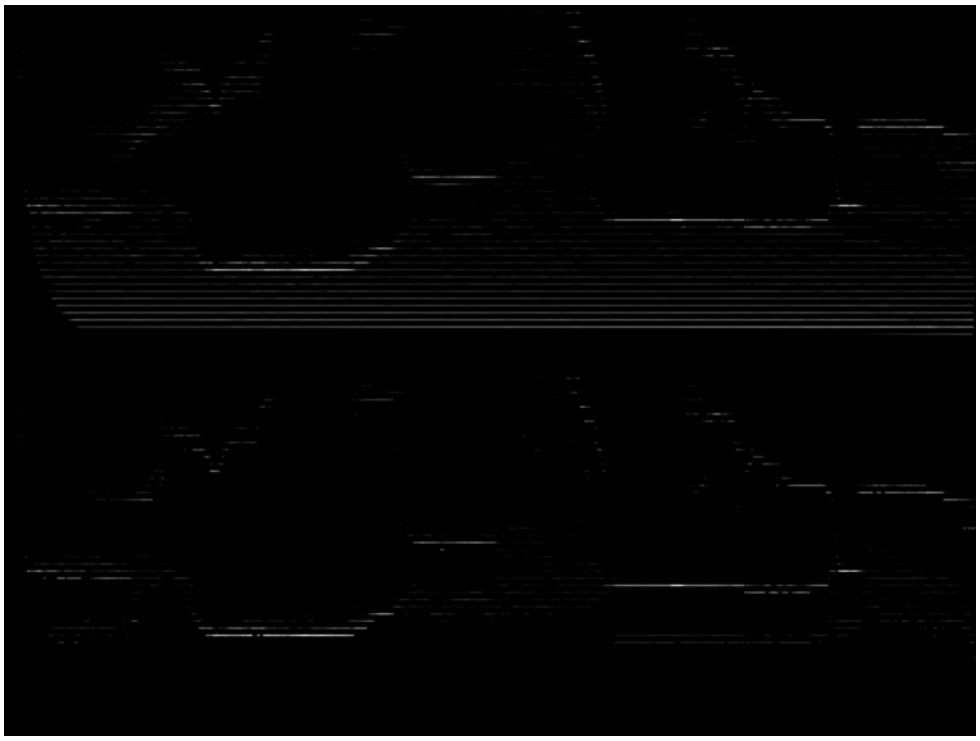

Figure 39. *u_Z map on top and filtered u_Z in the bottom. Note the the noisy pixels corresponding to the road have been eliminated*

Another approach is to interpolate the data between the blank rows present in u_Z map. In [2] the author suggests to copy the data from the nearest neighboring cell. We on the other hand decided to stack Gaussian kernels along the rows with valid data such that the 'spillover' from the kernels filled the empty rows. Figure 40 presents the filled u_Z map.

2.12 Least square plane fit for ground surface

In [2] the author fits a quadratic function to the 3D cloud of points hoping that road surface curvature changes in one direction. In [7] the authors fit a spline line to the 3D point cloud claiming the road surface curvature can change both ways and hence spline is a better fit. The advantage of a road surface model fit lies in the portability of the estimate. It is faster and efficient to pass this model as a parameter to higher level ADAS functions that make use of this road surface. Furthermore tracking of the road surface is much simpler with a road model rather than the whole point cloud.

In both the cases the authors transform points into the 3D space where the error increases with transformation (in [2] the error is a function of the 3D coordinates).

We prove that points that lie on a plane in 3D space $[x\ y\ z]$ will have corresponding points in image space $[u\ v\ d]$ conforming to the plane equation.

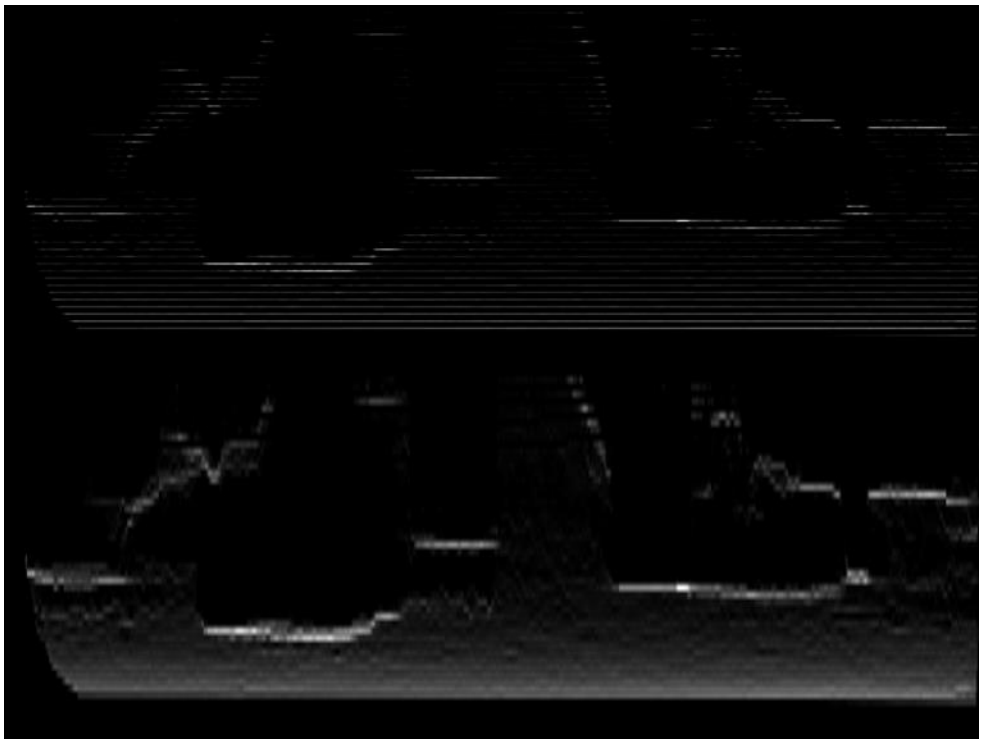

Figure 40. *original u_Z map(Top); Gaussian smoothed u_Z map(bottom)*

2.13

Assume that 3D points $[x\ y\ z]$ are constrained by the plane equation –

$$px + qy + rx + s = 0 \tag{33}$$

Substituting for the terms x, y, z from equation 26, 27, 28 we get –

$$p\left[\frac{b(2(u-u_0)-d)}{2d}\right] + q\left[\frac{b(\,(v-v_0)\cos\theta + \alpha\sin\theta\,)}{d} - h\right] + r\left[\frac{b\left((v_0-v)\sin\theta + \alpha\cos\theta\right)}{d}\right] + s = 0 \tag{34}$$

Rearranging the terms in above equation gives –

$$\begin{aligned}[2pb]\,u + \ [2qb\cos\theta - 2rb\sin\theta]\,v + \ [2 - pb - 2qh]\,d \\ + [2qb\alpha\sin\theta + 2rbv_0\sin\theta + 2rb\alpha\cos\theta \\ - 2qbv_0\cos\theta - 2pbu_0]\end{aligned} \tag{35}$$

The parameters in brackets in the above equation are either intrinsic or extrinsic camera parameters and can be assumed constant. Leading to the simplified equation –

$$p'u + \ q'v + \ r'd + \ s' = 0 \tag{36}$$

Plane fitting of 3D points involves –

- Generating the 3D point cloud $[x\ y\ z]$ from $[u\ v\ d]$
- Fitting plane to this point cloud and getting the plane coefficients $[p\ q\ r\ s]$
- Declaring 3D points$[x\ y\ z]$ that lie within a threshold distance from the plane defined by parameters$[p\ q\ r\ s]$, as belonging to plane/road and vice versa.
- Back projecting these points in $[u\ v\ d]$ space to highlight the plane/road surface for visualization.

Since we have proved that a points belonging to plane in 3D space correspond to points belonging to plane in image space, we can make the process simpler and much faster as follows –

- Fit the points in image space $[u\ v\ d]$ and get the parameters $[p'q'\ r's'\ ]$
- Declare points in image space $[u\ v\ d]$, that lie within a certain image space distance from the plane defined by $[p'q'r's'\ ]$, as belonging to plane/road and vice versa.

Solving the linear matrix equation in Section 2.8.3 and getting plane parameters is straightforward. Note that the least sum of square error fit is like an ‘average’ plane for the point cloud. The fit is definitely affected by outliers like obstacles, buildings and other traffic participants’ pixels. Hence we preprocess the disparity image to remove pixels other than the road surface with the crude obstacle separation. Crude obstacle separation had been already implemented in our application as detailed in [17], and we just used the function here to refine the disparity input image to get the least sum of squared error fit.

3. EVALUATION OF THE GROUND SURFACE DETECTION ALGORITHMS

3.1 Dataset

Vehicles move at speed above 100 km/hr on highways. This necessitates use of real time algorithms providing robust results. The evaluation of algorithms has to be performed on a set of images (including stereo images, the corresponding disparity images and the ground truth) taken from a test vehicle. We had two available sources for such data; the DLR dataset which provided us with the stereo images and dense disparity images but no ground truth images, the KITTI dataset which offered the stereo images and the ground truth but no dense disparity images. Our road plane detection required the dense disparity images. The KITTI dataset does not provide these dense disparity images. Hence we looked for an algorithm that could generate the dense disparity images from the KIT stereo images. Fortunately we found the library 'libelas' by Andreas Geiger who is one of the people responsible for the KITTI benchmark development. An example C++ project can be downloaded from their website[2], the application generates disparity images for some example stereo images. This application has been modified to generate the disparity images for the KITTI dataset stereo images. One key feature of this library was that it accepted only the '.pgm' files while the KIT stereo images were both '.png' files. Fortunately the IrfanView image viewer has a function to batch rename files, and the issue was resolved. The code was modified to target the KIT left and right images and subsequent execution generated the disparity image for all dataset. Figure 42 presents some of the disparity images for the KIT stereo images generated with libelas. The KIT dataset images have a much lower percentage of points that constitute the road plane. In other words, the image has much larger field of view which makes the v-disparity image much more diffuse. The cloud of points that corresponds to the road plane is not as distinct/focused as observed in the v-disparity images for the DLR dataset. The existing algorithms are modified to work well with the KITTI dataset.

3.2 ROC Curves

3.2.1 Basics

Since we have implemented several algorithms for road surface detection algorithms, it makes sense to evaluate and compare the performance of these algorithms. The compar-

[2] http://www.cvlibs.net/software/libelas/

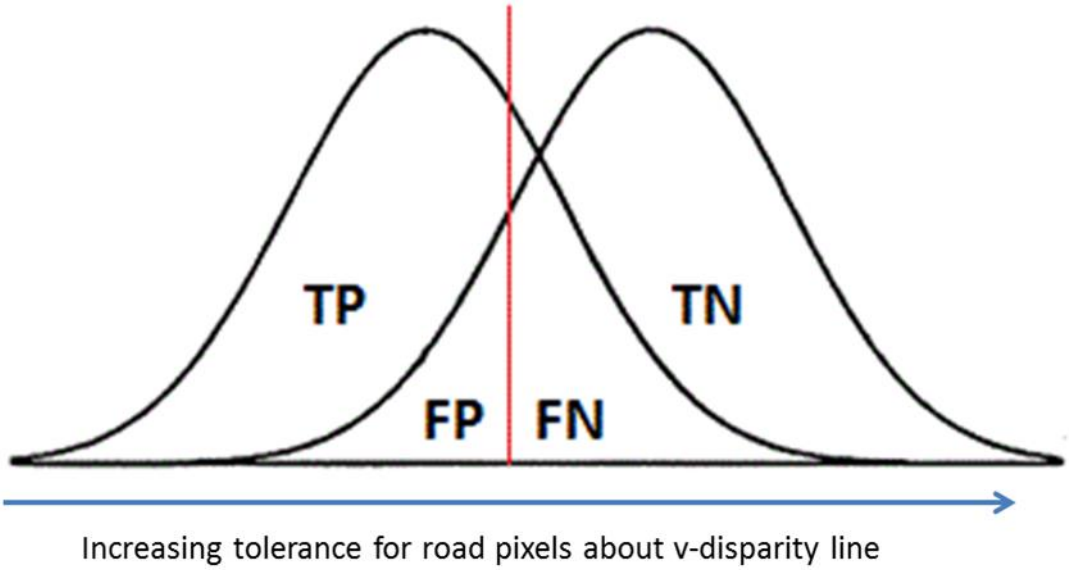

Figure 41. *Distribution of the 4 elements of confusion matrix as dictated by the threshold (red line)*

ison will be based on the quality of the detection and in some algorithms, the processing time will also be recorded to check if the algorithm meets the real time constraints. To quantify the detection quality we use the True Positive Rate (TPR) and False Positive Rate (FPR) as explained in Table 2 and illustrated in Figure 43. The quality of any detection algorithm varies with the key parameters that control their respective detections. This implies that to compare two algorithms we need to first find the thresholds that ensures the best performance for the individual algorithms and then compare their respective detection quality when they are exhibiting their individual best performance. This two-step process is accelerated by making use of the Receiver Operating Characteristics (ROC) curve. In the following section, we explain in detail how the ROC curves ensure such evaluation.

To generate ROC curve for any evaluation we need to vary a key parameter that segments the instance to be classified into the two categories; True (road) or False (background). The parameter for our classification is the tolerance for road pixels about the selected v-disparity line. Classifications lead to a 'confusion matrix' with 4 parameters True Positive, False Positive, True Negative & False Negative. Table 2 presents the meaning of these parameters in our context.

The Figure 41 presents the distribution of positives (road) and negative (background) pixels that are overlapping on the scale of the road tolerance about v-disparity. The red line indicates the threshold for road and background segmentation by the estimation algorithm. Points before the threshold line are classified as road; Points after the threshold are classified as background.

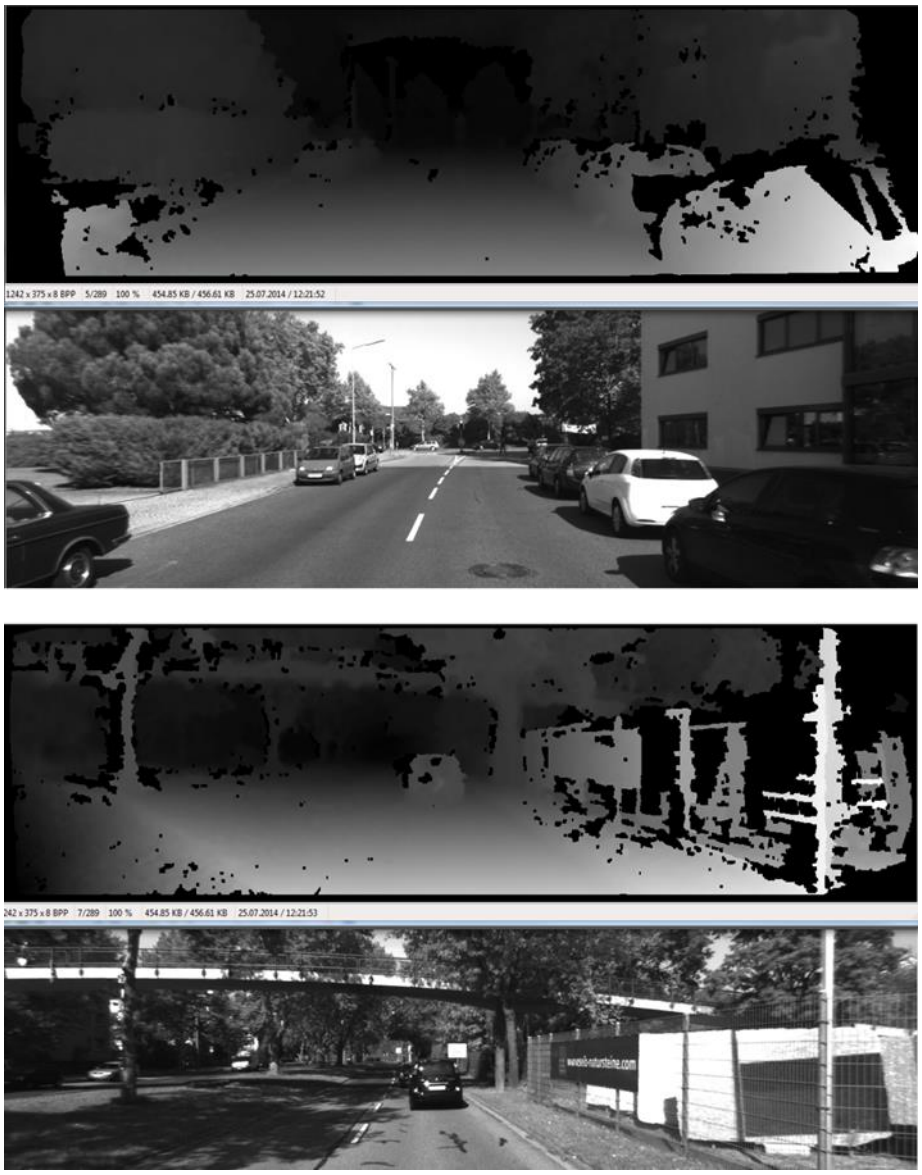

Figure 42. *disparity images for KIT dataset rendered with 'libelas' library*

Table 2. *Evaluation parameter definition*

Class	Description
True positive (TP)	Detected road pixels that in fact belong to road
False positive (FP)	Detected road pixels that do not belong to road
True negative (TN)	Detected background pixels that in fact belong to the background
False negative (FN)	Detected background pixels that do not belong to background
True positive rate (TPR)	$\frac{TP}{TP + FN}$
False positive rate (FPR)	$\frac{FP}{TN + FP}$

For zero tolerance we see that no pixels are classified, at low tolerances only road pixels are identified and all of them are true positive, increasing the tolerance still increases true positive and false positive rates. And by further increasing the tolerance, the percentage of false positives increases. Figure 43 presents the classification with an illustration. The figure shows a road and a cloud overlaid on top of the road. Points within the cloud are the estimated road pixels, while points outside the cloud are the estimated background. This estimation can be evaluated as TP, FP, TN & FN with the 4 colours shown in the Figure 43. .

Since we are estimating the road surface only for points below the horizon (empirical constant), we must exclude the points outside this horizon while counting the road and background pixels in ground truth images. Also note that the disparity images generated by 'libelas' library usually has a smaller envelope than the stereo images. Since the disparity images with invalid pixels do not assist our estimation of ground pixels, we must exclude corresponding pixels in ground truth pixels while counting road and background pixels. Receiver operating characteristic curve or ROC curve is the trace of TPR vs FPR as the threshold parameter is varied from its minimum to maximum value. This ensures that both TPR and FPR swing from 0 to 1 as the threshold varies. The significance of ROC curve lies in the fact that it can highlight the best performance of an algorithm. The highest performance is presented by algorithm that has a TPR of 100% and FPR of 0% which corresponds to the top left corner of the ROC graph. Since real algorithms seldom have such capability we consider the point closest to top left corner on the curve traced by the real algorithm as the best performance that can be extracted from the algorithm.

3.2.2 ROC curves for key algorithm parameters

The ROC curves for different parameters used for line scan in v-disparity are presented in Figure 45. The parameter Accumulator_w_x represents the size of the window within which the score of the candidate line (the candidate line fit to v-disparity) is calculated.

The ROC plot indicates that the accumulator window of width one performs best against other window sizes. Another parameter that influences the estimation outcome is the horizon row. Figure 46 presents the ROC curve for the different horizon row values. Note that for low horizon rows the estimation is restricted to an envelope that is close to the vehicle. In this zone the disparity has lower error and hence estimation is better. As the horizon row increases, the size as well as the cumulative error increases and the estimation is more prone to error as evident from the plot.

3.2.3 ROC for v-disp algorithms

Our road surface estimations have been generated by 2 main classes of detection algorithms. The first is the v-disparity approach that looks for smooth and flat surfaces and

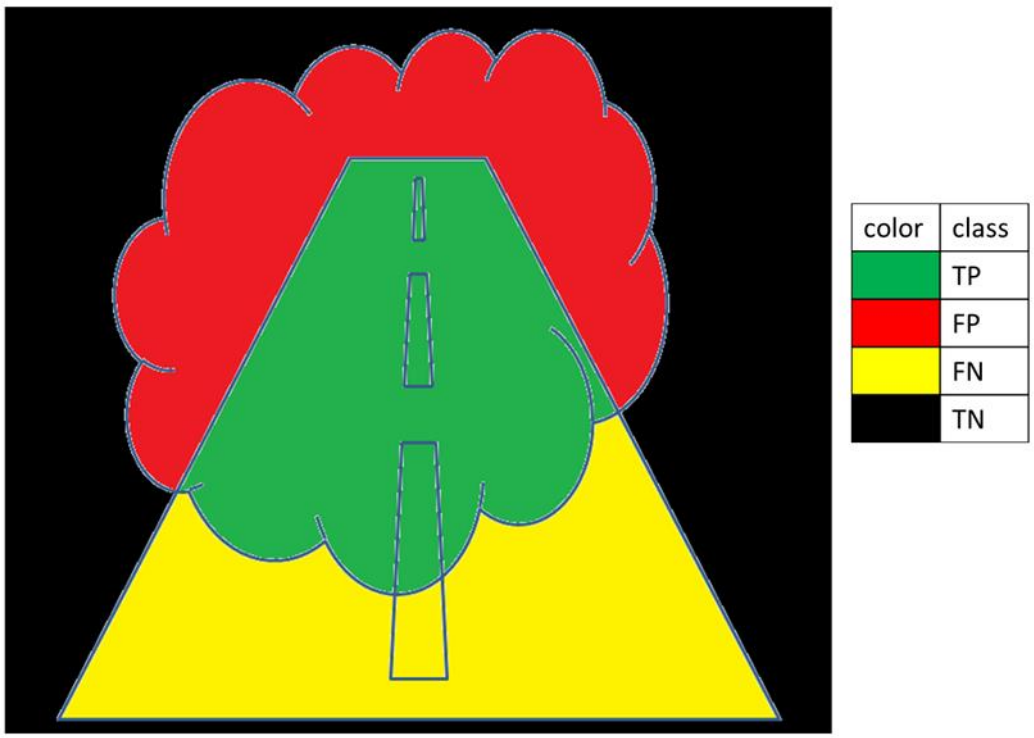

Figure 43. *The 4 elements of the confusion matrix and their interpretation for estimation (represented here by the cloud)*

the second one is the OGM approach that looks for obstacles and claims free space from the ego-vehicle up to these obstacles.

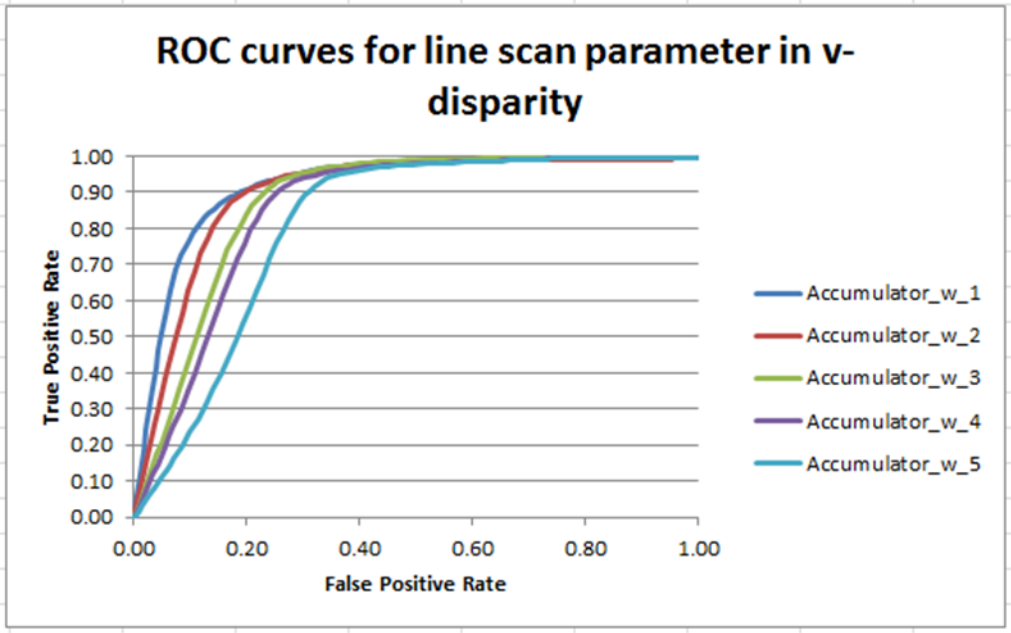

Figure 44. *The 4 elements of the confusion matrix and their interpretation for estimation (represented here by the cloud)*

Figure 45.

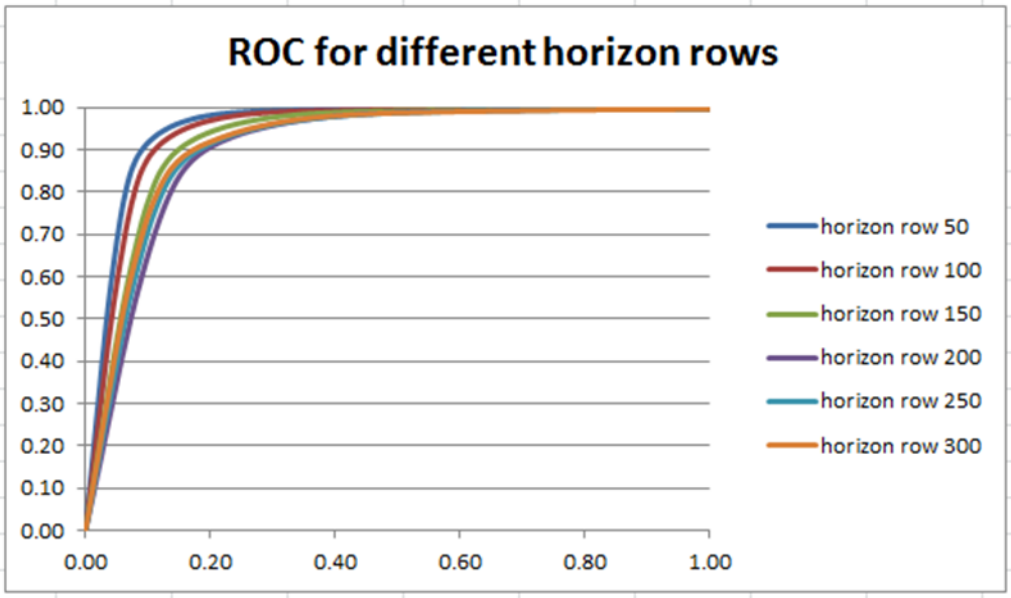

Figure 46. *The ROC curves for different horizon rows to which the estimation and hence evaluation is limited.*

The ROC curves for the 3 different algorithms implemented in Figure 23 and the later implemented algorithm as suggested in [17] (named "singapore" in plot) are presented in Figure 47. The evaluation of these algorithms have been carried out on a desktop PC with Core 2 Quad processor and a memory of 8 GB RAM The 2 best performances among these algorithms are as follows-

- Direct line fit algorithm gave a TPR of 84% and FPR of 13% at 2 frames/sec.
- "Singapore" algorithm gave a TPR of 84% and a FPR of 17% at 38 frames/sec.

It is to be noted that among the training images of the KIT dataset used for this evaluation, a vast majority of roads surfaces in scenes have a smooth horizontal surface this works in favor of the direct line fit algorithm which assumes that the road surfaces are flat and horizontal, the authors of [17] also point out this observation.

3.2.4 ROC for OGM algorithms

We also evaluated the OGM algorithms to study their performance. Figure 48 presents the results of this evaluation against the same dataset as used in previous evaluations. The 2 best preforming algorithms are as follows-

- udisp_OGM gave a TPR of 84% and a FPR of 22% @ 4.7 frames/sec
- polar_OGM gave a TPR of 83% and a FPR of 24% @ 1.4 frames/sec

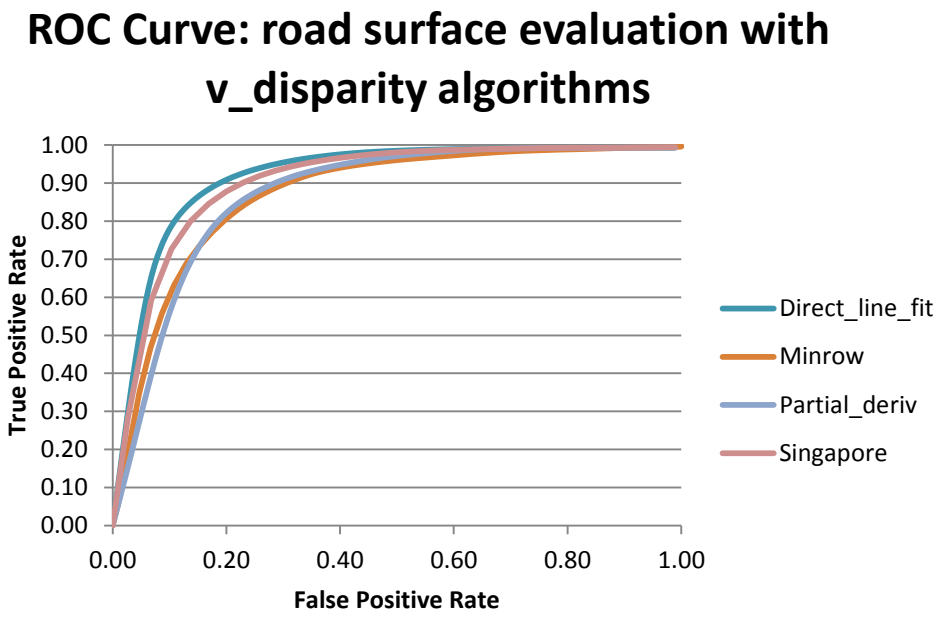

Figure 47. *The ROC curve for v-disparity ground surface detection algorithms.*

3.2.5 Comparison v-disp vs OGM algorithms

A comparison of the best algorithm from v-disparity approach and the best algorithm from OGM approach was also made. The plot of this comparison ROC curve is presented in Figure 49. It can be clearly inferred that “singapore” approach offer better estimates that udisp_OGM. Furthermore the “Singapore” approach provides results at 38 frames per second which is good enough for real time applications.

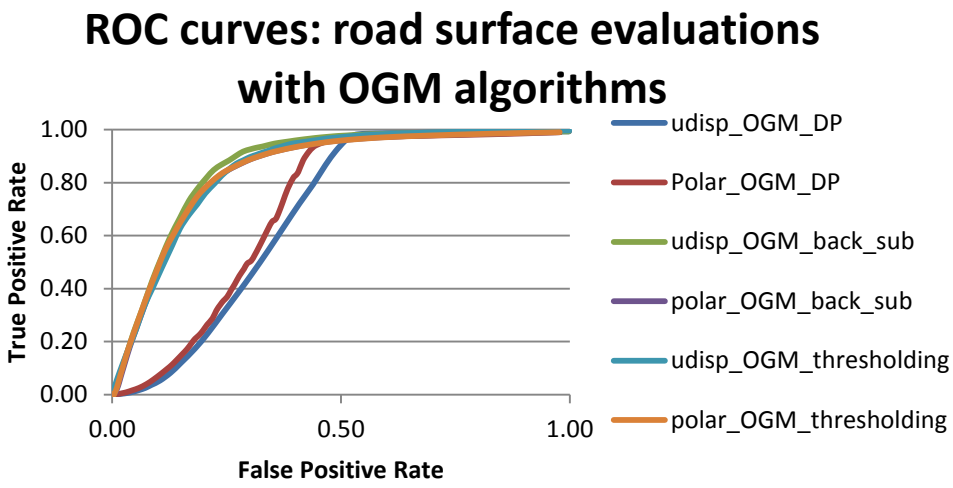

Figure 48. *The ROC curve for OGM ground surface detection algorithms*

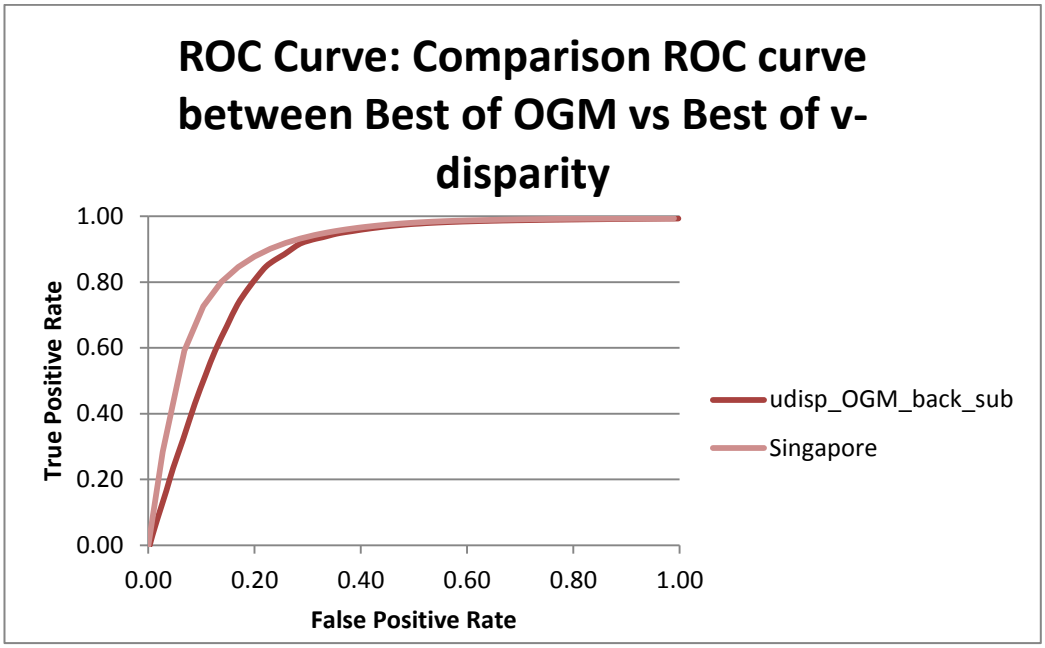

Figure 49. *Comparison ROC between the best of v-disparity and best of OGM ground surface detection algorithms*

4. CONTRIBUTIONS TO GROUND PLANE DETECTION ALGORITHMS.

4.1 Confidence metrics for road surface detection

One of the primary requirements of any estimate is the confidence of estimation. Naturally we had to prepare some confidence metrics to grade the ground plane estimates. Ground plane estimates are based on the assumption that along any row of the image (after obstacle elimination) the max number of pixels belong to the road and share roughly the same disparity. The confidence metrics are based on how closely the presented real images adhere to this assumption. We estimate the road surface based on the assumption that the roads present themselves in a particular mathematical representation, our confidence metrics are based on how well the situation presented "fits" this particular mathematical model.

The 3 confidence metrics used are –

1. Inverse of standard deviation of the disparity of pixels around the max intensity along each row of v-disparity
2. Percentage of the pixels in the vicinity (a fixed window) of the max intensity along each row of v-disparity
3. Density of pixels in the vicinity (a fixed window) of the max intensity along each row of v-disparity

Mathematically –

$$Confidence\_1_i = \frac{1}{\sigma_i} \tag{37}$$

$$mean\ \mu_i = \frac{1}{W_i} \sum_{j=j_{max}\ -\ ws}^{j=\ j_{max}\ +\ ws} j.w_{i,j} \quad where\ W_i = \sum_{j=j_{max}\ -\ ws}^{j=\ j_{max}\ +\ ws} w_{i,j} \tag{38}$$

The $w_{i,j}$ above represents the intensity at the row i and column jof the v-disparity image. The column corresponding to max intensity in a v-disparity row is indicated as j_{max} and the vicinity window size as ws.

$$\sigma_i = \frac{1}{W_i} \sum_{j\,=\,j_{max}\,-\,ws}^{j\,=\,j_{max}\,+\,ws} w_i.\ (j\ -\ \mu_i)^2 \tag{39}$$

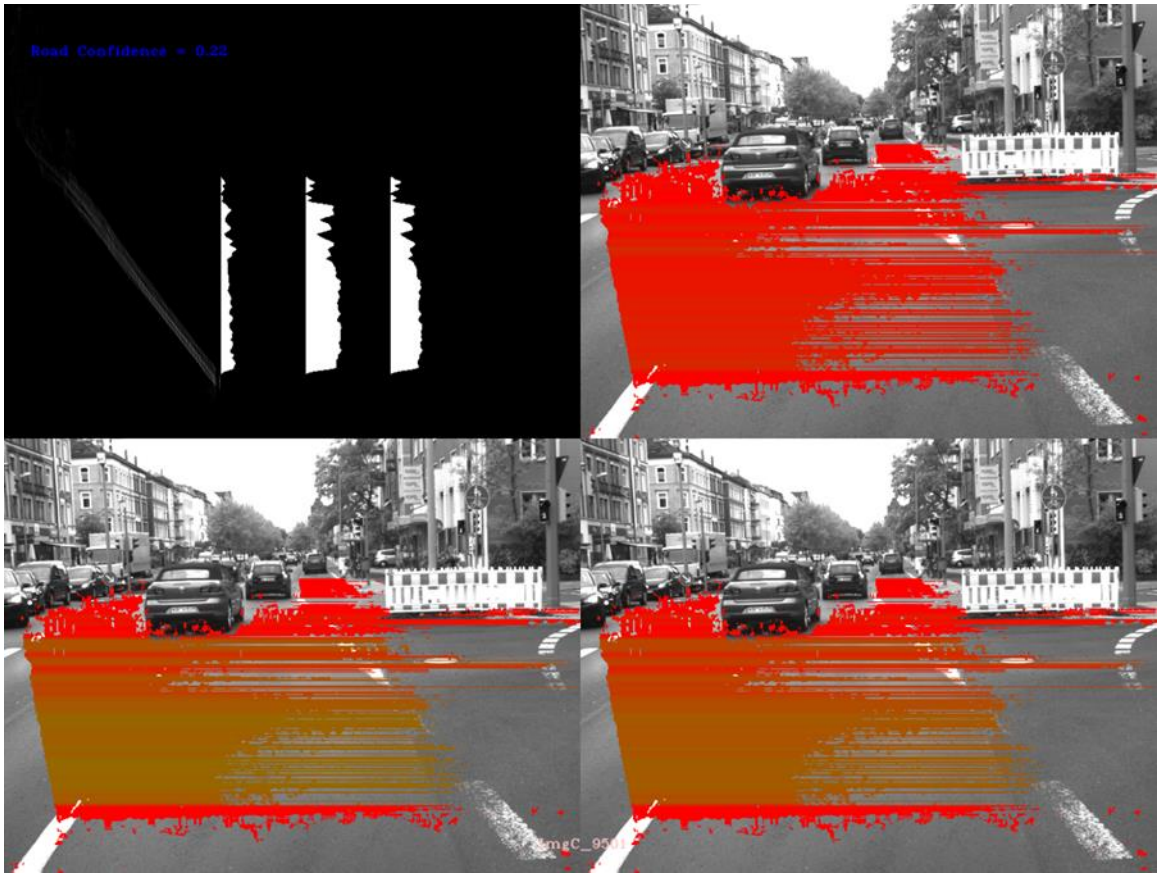

Figure 50. *Top left frame includes the v-disparity and 3 confidences (from left confidence_1, confidence_2 & confidence_3) plot sequentially. Top right is the ground plane estimate with color signifying the confidence_1. Similarly the bottom left corresponds to confidence_2 and bottom right to confidence_3. The road surface estimates in these frames are colored between green and red depending on the confidence.*

$$Confidence\_2_i = \frac{1}{avg\_population\_factor} \sum_{j\,=\,j_{max}\,-\,ws}^{j\,=\,j_{max}\,+\,ws} w_{i,j} \tag{40}$$

The $avg\_population\_factor$ presented in equation above is the average of the max $Confidence\_2$ observed in a dataset of 5000 images.

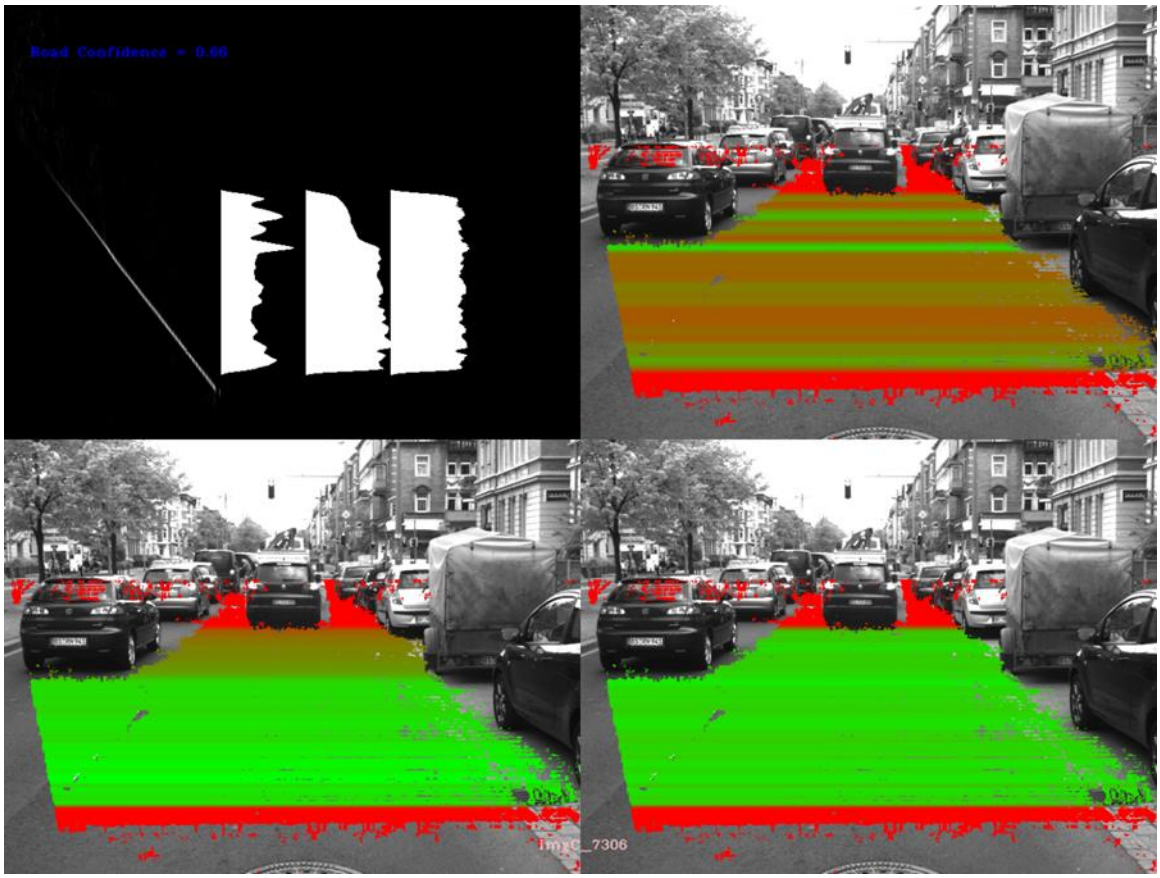

Figure 51. *Top left frame includes the v-disparity and 3 confidences (from left confidence_1, confidence_2 & confidence_3) plot sequentially. Top right is the ground plane estimate with color signifying the confidence_1. Similarly the bottom left corresponds to confidence_2 and bottom right to confidence_3.*

$$Confidence\_3_i = \frac{1}{population\_factor_i} \sum_{j=j_{max}-ws}^{j=j_{max}+ws} w_{i,j} \tag{41}$$

where

$$population\_factor_i = \frac{1}{distribution\_factor} \sum_{j=0}^{j=Image\_width} w_{i,j} \tag{42}$$

Notice that all of the above confidences are calculated around the max intensity along each row of v-disparity because the ground plane is theorized to exist around these pixels. This means that each row of the v-disparity and subsequently the ground plane estimate will have a particular confidence value. Assuming that a very optimistic road presentation will be one where all road pixels have disparities within a tolerance window of ±2, $distribution\_factor$ is set to 5. All the above confidences are calculated for rows where the maximum intensity in v-disparity is above a fixed threshold (which translates to - confidences being calculated for rows that have a sizable population of road pixels). Finally these confidences are filtered with Gaussian convolutions along the column.

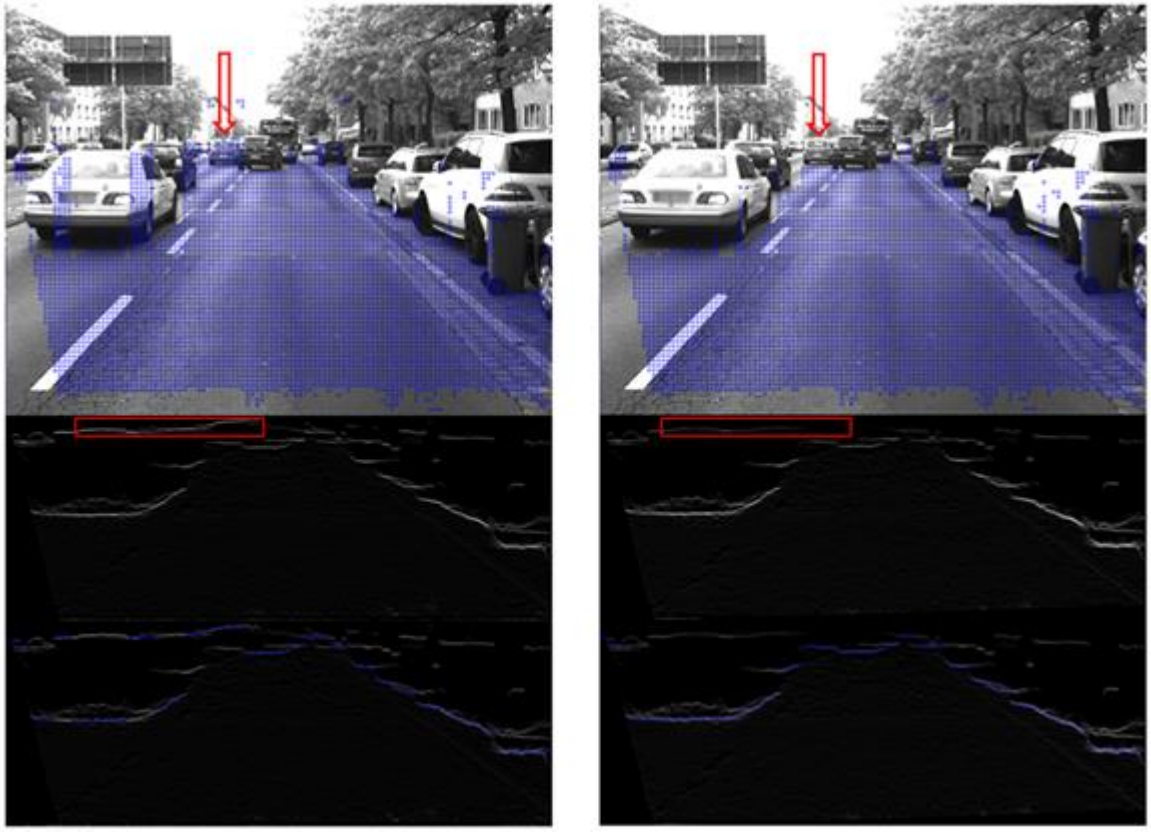

Figure 52. *Free space detection with extended u-disparity (left) and u-disparity (right). The red arrows highlight improved detection and the red windows in OGM highlight background suppression*

Figure 51 plots of the confidences thus defined for a particular frame. Note that the confidence_2 drops whenever a vehicle appears on the road (since the percentage of pixels belonging to road drops as well). But $confidence\_3$ does not exhibit such behavior, since we are normalizing it with the number of valid (non-zero) disparity along each row. An illustration of such behavior can be observed by comparing the the confidence rendering for lower left estimate ($confidence\_2$) and lower right estimate ($confidence\_3$) in Figure 51 below, due to the presence of a car in the adjacent lane, the confidence of road estimate drops adjacent to this car in lower left while such a drastic loss of confidence is not observed in lower right confidence rendering. Mathematically $confidence\_2$ is normalized with a constant and is hence sensitive to vehicle presence along the rows. The confidence of the entire image is calculated as the average of the $confidence\_3$ in the image (since $confidence\_3$ was found to be most meaningful from our observations). This can be seen written in blue in top left corner in Figure 51 and Figure 50.

Another illustration of the confidence rendering in presented in Figure 50. In the top left frame the v-disparity is sparse and not dense as usually observed. This means that the road is not flat, leading to poor estimates. Both the confidence rendering of road surface estimates and the average confidence for the entire estimate are hence poor.

Once the nominal disparity of ground plane is determined for each row (which in this case is the disparity corresponding to max intensity along any row of v-disparity), a suitable tolerance is given to estimate the ground plane pixels. Pixels along the image row which have disparities that lie within this tolerance are deemed to belong to road. Previously this tolerance was kept constant, about ±3. Now that we have a confidence measure for each row, we provide a more customized tolerance to each row. The tolerance for road surface segmentation in v-disparity (tolerance) is a linear function of the confidence as indicated by the equation below.

$$tolerance = 2 + \left[\frac{(100 - confidence\_3)}{100}.2\right] \tag{43}$$

This equation translates to conservative estimates (precision over sensitivity) for good confidence and liberal estimates (sensitivity over precision) for poor confidence.

4.2 Novel Extended u-disparity

In order to alleviate if not eliminate the problems discussed in section 0, an extended version of the u-disparity has been developed. As the name suggests, it is an extended version of the u-disparity. The construction of extended u-disparity is as follows –

- Each pixel in u-disparity is multiplied by a distance factor. The closer the pixel is to the vehicle, the larger the distance factor. This step ensures that the closer obstacles are better represented in OGM. This is also logical since closer vehicles hold more relevance from impending collision perspective. The other advantage is that this step dulls the background obstacles in OGM such as buildings (such as the one in Figure 35).
- For each pixel in u-disparity, a 2D Gaussian kernel is stacked on the corresponding pixel position in extended u-disparity. This is necessary since the obstacles are non-ideal and present themselves within a disparity window of a few pixels. This is especially true for sedans that do not have a flat rear surface and hence fairly inconsistent disparity. The Gaussian kernel is aimed to bring about a ‘spike’ around pixel cluster belonging to such vehicles in OGM

Figure 52 presents the improvement achieved with the extended u-disparity compared to estimates with normal u-disparity approach. Notice that the car highlighted by the red arrow is rendered as free space by the u-disparity approach (left) but is correctly identified as an obstacle by the extended u-disparity approach.

5. CONCLUSION & FUTURE WORK

5.1 Conclusion

The target of the thesis has been the implementation of a road surface detection algorithm suitable for ADAS systems. In this work we have implemented and subsequently compared several algorithms. The two best performing algorithms have been presenting along with their respective processing times.

The theory of estimation as relevant to ground surface detection has been presented along with the use and advantages of stereo camera to this cause. Ground surface estimation with v-disparity approach as explained in [9] has been implemented by fitting straight lines to v-disparity images. Two additional algorithms have been developed; one assuming that the road pixels have the lowest disparity along each row (called 'min-row') and the other with separation of obstacles from v-disparity using partial derivatives. For portability of the detected road surface, line-fit and poly-line fit algorithms have been developed to represent the road pixels in v-disparity.The new approach of road surface estimation with crude obstacle elimination as presented in [17] has also been implemented; this approach is found to be better behaved around obstacles since we prepare v-disparity with pixels other than those belonging to obstacles. By adjusting the tolerance of road surface estimation we can distinctly detect road surface even in presence of elevated sidewalks.

Although simple, all the above approaches have poor perception of the horizon; estimations must be manually limited to a certain fixes row to observe sensible estimates at large depths. Such behavior arises due to poor disparity resolution at large depths. Furthermore since all these approaches have made some or the other assumption regarding the road surface topography, they are sensitive to unusual presentation of road surface, for instance when the road is twisted about the direction of heading. These algorithms work with data in image space [u, v, d]. 3 confidence metrics have been developed that describe how well the presented scene fits the mathematical model assumed. These metrics are also representative of how good the estimations are. The estimates are rendered with their respective confidences. Since these confidences are a function of the row number, we have a confidence for every row of the estimate. Average confidence along all the rows is assumed to be the confidence for the entire estimate. A simple 'Near Vehicle Warning' function has been developed to discard the detection when the front vehicle get close to the ego car. This function is based on the percentage of the vehicle pixels in front of the ego-vehicle.

Free space estimation with OGM looks at the same challenge from a different perspective with the core idea that obstacles limit road surface, and hence road surface extends upto the detected obstacles. Since the elevated sidewalks have height that is much smaller that the vehicles in traffic, we do not have the finesse to distinctly detect road surface in presence of elevated sidewalks. Although we do concede that this approach gives a better perception of the horizon and that the obstacle detection does not suffer on banked roads. Although the authors prefer Polar OGM in [4], we found that the polar OGM has poor data resolution at higher depths. Furthermore with polar OGM we must first transform data from image space to the real world coordinates, segment the OGM and then back project the points in image space to get the road surface estimates. With the col-disparity OGM we have uniform data resolution, we can work with data in image space itself and hence it is faster than the polar OGM. Our biggest concern with OGM approach is the processing time for dynamically segmenting the OGM. Although this issue can be offset by using the background separation technique as suggested in [20]. Additionally the OGM approach can estimate the free space upto surfaces that have larger vertical surface area than the obstacles closer to the ego-vehicle. We could get around this problem to some extent by limiting the pixels in image space that contribute to the OGM. Using a constant threshold for 'background subtraction' as suggested in [21] makes little sense, as the footprint of an obstacle in OGM depends on the size and distance of the obstacle from the ego-vehicle, both of which vary to a considerable extent. Polar OGM has a limited 3D scope (In our implementations the max depth of polar OGM was usually 76.8m) This means that if there are obstacles beyond the scope of OGM then they cannot be detected in OGM, the free space along those columns is wrongly estimated. On the other hand since u-disparity OGM has the complete disparity range, its scope is much larger than that of the polar OGM.

It has been proved in this report that the pixels that conform to the equation of plane 3D space, have corresponding points in image space also conforming to plane equation. Thus instead of fitting points in the 3D space, we directly fit the points in the image space to get the plane equation. The advantage of the plane fitting is that we can easily communicate this information to other functions instead of using sending the whole point cloud. We used the least sum of squared error approach to fit the plane to the points.

The final task of the thesis work has been the evaluation of the algorithms against the KIT dataset. The primary objectives of estimating road/ground plane and ego-lane have been accomplished. Three algorithms; direct line fit, minrow and partial derivative approaches for ground plane detection with v-disparity images were implemented in Visual Studio C++ with the help of OpenCV library. A derivation that concludes that flat horizontal roads correspond to straight lines in v-disparity has been presented. We proceed to detect such straight lines in v-disparity images and then overlay the road pixels onto real image through back-projection. The evaluation of these 3 algorithms revealed

that the direct line fit to the v-disparity images generated the best results. Although the authors of [17] argue that the KITTI datasets lack considerable variation in road surface topography. The ground plane estimates with the minrow and partial derivative approach provided comparable results. A recent publication that claims improved road surface estimation [17] has been implemented. Furthermore, confidence measures were developed as representative of the detection trueness. These measures are generated on how close the presented situation fits the ideal road model. A function to trigger warning message when the ego-vehicle is very close to another vehicle in front is developed. The free space detection with occupancy grid maps as published by authors at Daimler has been implemented as well. The Polar OGM claims to offer linear free space perception although it suffers from poor data resolution at higher depths. Dynamic programming is implemented to optimally segment the OGM (based on the spatial continuity & intensity of OGM pixels) and subsequently chart the free space. Use of dynamic programming is redundant to some degree after the 'background subtraction' as done in [21]. The Dynamic programming is a computationally intensive segmentation algorithm. A novel occupancy grid map by extending the u-disparity has been proposed. This approach does not require background subtraction and also does not suffer from poor data resolution at higher depths. Table 3 presents the comparison of the algorithms that we implemented.

Table 3. *Comparison of different road surface detection algorithms*

Approach	Best performance	Processing speed (frames/s)
v-disparity direct line fit	TPR: 84%, FPR: 13%	2
v-disparity "singapore"	TPR: 84%, FPR: 17%	38
udisp_OGM	TPR: 84%, FPR: 22%	4.7
polar_OGM	TPR: 83%, FPR: 24%	1.4

5.2 Future work:

So far the algorithms implemented make a new detection for every successive image frame. If we make use detection tracking, we can predict the road surface for the next frames using the ego-vehicle kinematic data. Furthermore, instead of making a new detection for every frame, we can make detections at lower frequency and fuse this data with the predicted detection using filters like Kalman filter.

Also the algorithms implemented in this work rely on the 3D topographical scene data for detecting road surfaces. We can also detect road surface using the physical appearance of the road using the intensity, color etc. A more accurate road surface detection can hence be perceived by fusion of detections from the 3D scene data and the detections using the road appearance.

Machine learning algorithms have been successfully used in medical applications; for instance, to predict the malignance of tumors. The machine learning algorithms are trained using prior instances of tumor image in patients along with their known malignance. The algorithms use the training tumor images to model the malignance of the tumor as a function of image pixels of the tumor. Support Vector Machine (SVM) is one of the most popular choices in machine learning from images. 'Weka' is an open source machine learning tool developed at the University of Waikato. A wide range of classifier functions are made available in this software. Modified versions of the SVM have been implemented in road surface estimation at the Karlsruhe Institute of Technology. A lot of effort has been made to benchmark algorithms involved in the ADAS [15]. The KIT team has implemented the machine learning algorithms on the graphics processor; a luxury as far as the current vehicle architecture is considered. But with the looming automated driving onset, graphics processors will see increasing usage. There also have been efforts by Daimler to record the road scene data that is relevant to driving vehicles (road lanes, intersections, road borders, etc) to a central online database [8]. Much like the google street view, this online data is the sequentially accessed depending on the live vehicle position. Scene captured from camera and the online scene repository are compared to give a complete scene interpretation. This reinforcement helps not only improve the accuracy of scene interpretation but also update the online scene repository whenever the road infrastructure changes – accidents, construction, deteriorates, etc. The online repository will be robust since we have vehicle constantly plying on roads providing with the latest road infrastructure data. Such system still needs to cope with the dynamic traffic constituents like pedestrians, cyclists and other vehicles. But it is one of the promising approaches to scene interpretation where all vehicles can support and benefit from the central repository. Another way to ensure robust performance is to include redundancy in the system. Use of multiple sensors to perceive the environment ensures reinforcement of scene perception. Using RADAR and LIDAR one can generate 3D cloud of points. Information from multiple sensors can be combined at either the low level (characterized by 3D point data) or the high level (characterized by road infrastructure entities).

6. REFERENCES

[1] Zehang Sun, George Bebis & Ronald Miller; "On-Road Vehicle Detection: A Review," in IEEE transactions on pattern analysis and machine learning, Vol 28, No. 5, May 2006.

[2] Florin Oniga and Sergiu Nedevschi, "Processing dense stereo data using elevation maps: road surfaces, Traffic isle and obstacle detection," in IEEE transactions on Vehicular Technology, vol. 59 No.3, March 2010.

[3] J.C. McCallVideo and M.M.Trivedi, "Video-based lane estimation and tracking for driver assistance: Survey, system and evaluation," IEEE transactions on intelligent transportation systems, Vol. 7, No. 1, March 2006.

[4] Hernan Badino, Uwe Frank and Rudolf Mester, "Free Space Computation Using Stochastic Occupancy Grids and Dynamic Programming," Workshop on dynamical vision, ICCV Rio De Janeiro, Brazil, 20 Oct 2007.

[5] Don Murray and Jim Little, "Using Real-Time Stereo Vision for Mobile Robot Navigation," Autonomous Robots, Vol 8, Issue 2, April 2000.

[6] Jun Zhao, Jayantha Katupitiya and James Ward, "Global Correlation Based Ground Plane Estimation Using V-disparity Image," IEEE Conference on Robotics and Automation, Roma, Italy. Pages 529-534, April 2004.

[7] Andreas Wedel, Hernan Badino, Clemens Rabe, Heidi Loose, Uwe Frank and Daniel Cremers, "B-Spline Modelling of Road Surfaces with an Application to Free Space Estimation," IEEE transactions on Intelligent transportation systems, Vol. 10, Issue 4, Dec 2009.

[8] Uwe Frank, David Pfeiffer, Clemens Rabe, Carsten Knoeppel, Markus Enzweiler, Fridtjof Stein and Ralf G. Herrtwich, "Making Bertha See," IEEE Conference on Computer Vision Workshops, Pages 214-221, Dec 2013

[9] Raphael Labayrade, Didier Aubert and Jean- Philippe Tarel, "Real Time Obstacle Detection in Stereovision on Non Flat Road Geometry Through V-disparity Representation," IEEE conference on Intelligent Vehicle symposium, pages 646-651, June 2002.

[10] [Zhencheng Hu and Keiichi Uchimura, "U-V-Disparity: An efficient algorithm for Stereovision Bases Scene Analysis," Intelligent Vehicles Symposium, Proceedings, IEEE, Pages 48-54, June 2005.

[11] A. Lopez, J. Serrat, C. Canero, F. Lumbreras and T. Graf, "Robust Lane Markings Detection and Road Geometry Computation," International Journal of Automotive Technology, Vol. 11 No. 3, Pages 395-407, May 2010.

[12] Adrian Kaehler and Gary Rost Bradski, "Learning OpenCV", 2008.

[13] Ryan Seghers, http://www.codeproject.com/Articles/560163/Csharp-Cubic-Spline-Interpolation, Jul 2014.

[14] Jannik Fritsch, Tobias Kühnl and Andreas Geiger, "A New Performance Measure and Evaluation Benchmark for Road Detection Algorithms," IEEE conference on Intelligent Transportation systems, Pages 1693-1700, Oct 2013.

[15] A. Geiger, P. Lenz, C. Stiller, R. Urtasun, "Vision meets Robotics: The KITTI dataset," International Journal of Robotics Research, Vol. 32, Issue 11, Sept 2013.

[16] Mohamed Aly, "Real time detection of lane markers in urban streets," Intelligent Vehicles Symposium, IEEE, Pages 7-12, June 2008.

[17] Wu, M. ; Lam, S.-K. ; Srikanthan, T. "Nonparametric technique based high-speed Road surface detection", Intelligent Transportation Systems, IEEE Transactions on (Volume:PP , Issue: 99)

[18] Sebastian Thrun; Wolfram Burgard; Dieter Fox, Probabilistic Robotics

[19] Perrollaz, M. ; Yoder, J.-D. ; Laugier, C., "Using obstacles and road pixels in the disparity-space computation of stereo-vision based occupancy grids", Intelligent Transportation Systems (ITSC), 2010. Pages 1147-1152.

[20] Badino, H. ; Mester, R. ; Vaudrey, T. ; Franke, U., "Stereo-based Free Space Computation in Complex Traffic Scenarios", Image Analysis and Interpretation, 2008. SSIAI 2008. Pages 189-192.

[21] Hernán Badino; Uwe Franke; David Pfeiffer, "The Stixel World- A Compact Medium Level Representation of the 3D-World", 31st DAGM Symposium on Pattern Recognition, 2009.

[22] Stephen P. Bradley; Arnoldo C. Hax; Thomas L. Magnanti, "Applied Mathematical Programming", Pages: 320 – 330.

[23] David Eberly, "Least Squares Fitting of Data".

[24] Greg Welch, Gary Bishop; “An Introduction to the Kalman Filter”, SIGGRAPH 2001.

[25] Robert N. Charette, “This car runs on code”, IEEE Spectrum, Feb 2009

[26] Chandrashekara N, “Basics of Automotive ECU”, ETAS webinar, Jan 2014

[27] Unselt, T.; Breuer, J.; Eckstein, L.; Frank, P. Avoidance of "loss of control crashes" through the benefit of ESP FISITA Conference paper no. F2004V295

[28] Paul Salmon, Michael Regan, Ian Johnston, “Human Error and Road Transport: Phase one – Literature review”; December 2005.

[29] National highway and traffic safety administration, “The Economic and Societal Impact of Motor Vehicle Crashes, 2010”