Traffic Scene Perception using Multiple Sensors for Vehicular Safety Purposes

Dissertation

Presented in Partial Fulfillment of the Requirements for the Degree Doctor of Philosophy in the Graduate School of The Ohio State University

By

Siavash Hosseinyalamdary, B.S., M.S.

Graduate Program in Civil Engineering

The Ohio State University

2016

Dissertation Committee:

Alper Yilmaz, Advisor
Charles Toth, Co-advisor
Ralph Von Frese
Abstract

Autonomous driving is an emerging technology, preventing accidents on the road in future. It, however, faces many challenges because of various environmental conditions and limitations of sensors. In this dissertation, we study multiple sensor integration to overcome their limitations and reliably perform missions enabling autonomous driving.

The laser scanner point cloud is a rich source of information, suffers from low resolution, especially for farther objects. We generalize 2D super-resolution approaches applied in image processing to the 3D point clouds. Two variants are developed for the 3D super-resolution: the dense point cloud is generated in such a way that it follows the geometry of the original point cloud; the brightness of the images are utilized to generate the dense point cloud. The results show our proposed approach successfully improves the density of the point cloud, preserves the edges and corners of objects, and provides more realistic dense point cloud of objects relative to the existing surface reconstruction approaches.

The static and moving objects must be detected on the road, the moving objects must be tracked, and trajectory of the platform must be designed to avoid accidents. The densified point clouds are integrated with other sources of information, including the GPS/IMU navigation solution and GIS maps, to detect the objects on the road.
and track the moving ones. The results show static and moving objects are detected, the moving objects are accurately tracked, and their pose is estimated.

In addition to obstacle avoidance, the autonomous vehicles must detect and obey the traffic lights and signs on the road. Due to the variations in the traffic lights, we propose Bayesian statistical approach to detect them. The spatio-temporal consistency constraint is applied to provide coherent traffic light detection in space and time. In addition, conic section geometry is utilized to estimate the position of the traffic lights with respect to camera mounted on the platform. The proposed traffic light detection approach is evaluated using Karlsruhe Institute of Technology (KITTI) and La Route Automatise (LARA) benchmarks. The results of the proposed traffic light detection approach are 98.7% precision rate and 94.7% recall rate in LARA benchmark, outperform the existing traffic light detection approaches tested in LARA benchmark.

In conclusion, we integrate multiple sensors to overcome their shortages, such as low resolution of point clouds, and propose obstacle avoidance and traffic light detection approaches based on the integrated sensors. Our results outperform the earlier studies in traffic light detection and provide more realistic surfaces in 3D super-resolution. Further studies may modify the proposed traffic light detection to detect the traffic signs.
To everybody who works on the open source projects
Acknowledgments

I would like to thank Professor Alper Yilmaz for his advice, encouragement, patience and kindness. I would like to thank Professor Charles Toth for his guidance, support and patience.

I would like to thank Professor Ralph von Frese for his time and comments, and Professor Lakshmi Dasi for serving as the graduate school representative during my dissertation defense.

I would like to thank Professor Dorota Brzezinska and my education was not possible without her support. I also thank Professor Christopher Jekeli for providing the opportunity to broaden my research interests, for his advice and support.

I would like to thank my fellow lab members, past and present, for all their help. Special thanks to my family for their care and support.
Vita

2003 .................................. B.S. Civil Engineering, University of Tabriz, Iran
2006 .................................. M.S. Civil Engineering, University of Tehran, Iran
2011-2013 ............................. M.S. Civil Engineering, The Ohio State University.
2013-present ........................... Graduate Research Associate, Civil Engineering, The Ohio State University.

Publications

Research Publications


S. Hosseinyalamdary “GIS aided traffic monitoring using multi-sensor integration”. *Book Chapter in Encyclopedia of GIS* 2017


**Fields of Study**

Major Field: Civil Engineering
### Table of Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Abstract</td>
<td>ii</td>
</tr>
<tr>
<td>Dedication</td>
<td>iv</td>
</tr>
<tr>
<td>Acknowledgments</td>
<td>v</td>
</tr>
<tr>
<td>Vita</td>
<td>vi</td>
</tr>
<tr>
<td>List of Tables</td>
<td>xi</td>
</tr>
<tr>
<td>List of Figures</td>
<td>xii</td>
</tr>
<tr>
<td>1. Introduction</td>
<td>1</td>
</tr>
<tr>
<td>1.1 Background</td>
<td>1</td>
</tr>
<tr>
<td>1.2 Problem Statement</td>
<td>3</td>
</tr>
<tr>
<td>1.3 Objective and Research Questions</td>
<td>4</td>
</tr>
<tr>
<td>1.4 Related work</td>
<td>8</td>
</tr>
<tr>
<td>1.5 Scope and limitation of the study</td>
<td>10</td>
</tr>
<tr>
<td>1.6 Key Contributions</td>
<td>10</td>
</tr>
<tr>
<td>1.7 Organization of this Dissertation</td>
<td>11</td>
</tr>
<tr>
<td>2. Multiple Sensor Integration</td>
<td>12</td>
</tr>
<tr>
<td>2.1 GPS</td>
<td>12</td>
</tr>
<tr>
<td>2.2 IMU</td>
<td>15</td>
</tr>
<tr>
<td>2.3 Integrated Navigation Solution</td>
<td>16</td>
</tr>
<tr>
<td>2.4 Camera</td>
<td>17</td>
</tr>
<tr>
<td>2.4.1 Projection $3D \rightarrow 2D$</td>
<td>17</td>
</tr>
<tr>
<td>2.4.2 Projection $2D \rightarrow 2D$</td>
<td>19</td>
</tr>
<tr>
<td>2.5 Laser Scanner</td>
<td>20</td>
</tr>
<tr>
<td>Section</td>
<td>Title</td>
</tr>
<tr>
<td>---------</td>
<td>-------</td>
</tr>
<tr>
<td>3.1</td>
<td>Introduction</td>
</tr>
<tr>
<td>3.2</td>
<td>Mathematical Framework</td>
</tr>
<tr>
<td>3.2.1</td>
<td>Geometrical 3D super-resolution</td>
</tr>
<tr>
<td>3.2.2</td>
<td>Brightness based 3D super-resolution</td>
</tr>
<tr>
<td>3.2.3</td>
<td>Global continuity</td>
</tr>
<tr>
<td>3.3</td>
<td>Implementation</td>
</tr>
<tr>
<td>3.4</td>
<td>Experiments</td>
</tr>
<tr>
<td>3.5</td>
<td>Results</td>
</tr>
<tr>
<td>3.5.1</td>
<td>Global smoothness</td>
</tr>
<tr>
<td>3.6</td>
<td>Discussion</td>
</tr>
<tr>
<td>3.7</td>
<td>Conclusion</td>
</tr>
<tr>
<td>4.1</td>
<td>Introduction</td>
</tr>
<tr>
<td>4.2</td>
<td>Multiple sensor integration</td>
</tr>
<tr>
<td>4.2.1</td>
<td>GPS/IMU navigation solution</td>
</tr>
<tr>
<td>4.2.2</td>
<td>GIS maps</td>
</tr>
<tr>
<td>4.2.3</td>
<td>Laser scanner’s point cloud</td>
</tr>
<tr>
<td>4.3</td>
<td>Pose Estimation</td>
</tr>
<tr>
<td>4.4</td>
<td>Point cloud enhancement</td>
</tr>
<tr>
<td>4.5</td>
<td>Tracking</td>
</tr>
<tr>
<td>4.6</td>
<td>Experiments</td>
</tr>
<tr>
<td>4.7</td>
<td>Results</td>
</tr>
<tr>
<td>4.8</td>
<td>Discussion</td>
</tr>
<tr>
<td>5.1</td>
<td>Introduction</td>
</tr>
<tr>
<td>5.1.1</td>
<td>Bayes’ theorem</td>
</tr>
<tr>
<td>5.2</td>
<td>Methodology</td>
</tr>
<tr>
<td>5.2.1</td>
<td>Binary labeling</td>
</tr>
<tr>
<td>5.2.2</td>
<td>Spatial coherency</td>
</tr>
<tr>
<td>5.2.3</td>
<td>Temporal coherency</td>
</tr>
<tr>
<td>5.2.4</td>
<td>Observation vector</td>
</tr>
<tr>
<td>5.2.5</td>
<td>Learning</td>
</tr>
<tr>
<td>5.3</td>
<td>Implementation</td>
</tr>
<tr>
<td>5.3.1</td>
<td>Reduction of the search space</td>
</tr>
</tbody>
</table>
List of Tables

<table>
<thead>
<tr>
<th>Table</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1</td>
<td>Some active safety features provided by automotive industry [17].</td>
<td>9</td>
</tr>
<tr>
<td>5.1</td>
<td>The initial parameters of the models in this study.</td>
<td>118</td>
</tr>
<tr>
<td>5.2</td>
<td>The confusion matrix; It shows that the traffic lights signals have been</td>
<td>121</td>
</tr>
<tr>
<td></td>
<td>correctly detected.</td>
<td></td>
</tr>
<tr>
<td>5.3</td>
<td>The traffic light detection algorithms are compared using LARA benchmark;</td>
<td>127</td>
</tr>
<tr>
<td></td>
<td>Our proposed approach outperforms the other approaches, it has the highest</td>
<td></td>
</tr>
<tr>
<td></td>
<td>precision rate.</td>
<td></td>
</tr>
</tbody>
</table>
## List of Figures

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1</td>
<td>Autonomous driving task is divided into its component. We propose an approach to increase the resolution of point clouds and focus on the obstacle avoidance and traffic light detection tasks using the enhanced point clouds in this dissertation.</td>
<td>6</td>
</tr>
<tr>
<td>2.1</td>
<td>The traffic scene perception is only achievable when multiple sensors are integrated. The navigation solution of the platform is provided using the integration of GPS and IMU. However, the images and point clouds are applied in addition to IMU sensor to navigate the platform in the absence of GPS. The road geometry is extracted from GIS maps and integrated with images to detect lanes of the road. Point clouds, GPS/IMU navigation solution and GIS maps are also integrated to detect and track moving objects. These sensors are complementary and are essential for automated driving.</td>
<td>13</td>
</tr>
<tr>
<td>2.2</td>
<td>Projection of a point into the image space [26].</td>
<td>18</td>
</tr>
<tr>
<td>2.3</td>
<td>The intrinsic geometry of the camera [26] is demonstrated in (a) the camera profile and (b) image view. The lens focal length, ( f ), and the principal point, ( p ), are the camera calibration parameters.</td>
<td>19</td>
</tr>
<tr>
<td>2.4</td>
<td>Projection of the planar points into the image space is the homography transformation [26]. The projection of the image of the planar points in two epochs is also the homography transformation. ( R ) and ( t ) are rotation matrix and translation vector between two camera exposure stations.</td>
<td>21</td>
</tr>
<tr>
<td>2.5</td>
<td>(a) The scene. (b) The laser scanner point cloud is projected into image space and the depth of points is shown by color.</td>
<td>22</td>
</tr>
</tbody>
</table>
2.6 The road links are extracted from vector OpenStreetMap and shown in pink. The road links are overlaid on the raster OpenStreetMap.

3.1 Linear interpolation results in different values with different weights of points. (a) Four points are located on two perpendicular planes. (b) If the points are incorrectly chosen or weighed, the interpolated point (red dot) will not follow the geometry of the surface (two planes). (c) The correct weights of the points leads to the interpolated points that follows the geometry of the surface.

3.2 The surface of an object is implicitly represented by an indicator function. This function is a signed distance function of the surface for each point and it has positive values for points inside the object and negative values for the points outside. The surface of this object is defined by the zero crossing of the indicator function.

3.3 The profile view of the neighborhood of point $X$. The point $X'$ is represented by the vector $[u_{X'}, v_{X'} W(u_{X'}, v_{X'})]^\top$. The surface is reconstructed if $W(u_{X'}, v_{X'})$ is determined.

3.4 The sampled points of the minivan’s visible surface; The red rectangular area shows the neighborhood of a point at the center of this rectangle.

3.5 A planar surface is estimated applying least squares to the points in a local neighborhood. This surface is applied to initialize the iterative gradient descent minimizations. The color of the surface is back-projected from the image.

3.6 The surface is estimated in the local neighborhood using the smoothness constraint and the observed points. The points of point cloud are assume accurate and are not smoothed. Therefore, the surface is estimated in the way that it passes through the observed points and observed points remain intact. The color of the surface is back-projected from the image.

3.7 The observed points are noisy and they are smoothed. Therefore, the surface passes between the observed points and generates a smooth surface. The color of the surface is back-projected from the image.
3.8 Surface of an object in the local neighborhood is constructed based on the brightness changes of points. Therefore, change in brightness between points leads to change in the surface. The smoothness rate of the observed points is 0 and these points remained intact. The color of the surface is back-projected from the image.

3.9 The smoothness rate of the observed points is larger than 0 and therefore, the observed points are smoothed and the resulted surface passes between these points. The color of the surface is back-projected from the image.

3.10 A 100 \times 100 grid generated in the local neighborhood of the red rectangle in Figure 3.4. The sampled points are transformed to the local coordinate system and displayed with red dots. The color of grid points show the distance of these points from tangent plane.

3.11 The sensors configuration; GPS/IMU navigation solution, cameras and laser scanner are mounted on the platform [20].

3.12 The segmented point cloud of the minivan in Figure 3.4; The blue line shows boundaries of this minivan extracted from images, and the background color is changed for better visualization. The color of the points is back-projected from the image.

3.13 The segmented point cloud of the minivan from top view; The color of the points is back-projected from the image.

3.14 The reconstructed surface of the minivan using Moving Least Squares (MLS) approach; The boundaries of the object are not preserved and the its shape is significantly distorted. The color of surface is back-projected from the image.

3.15 The reconstructed surface of the minivan using Moving Least Squares (MLS) approach from top view; The point cloud is severely distorted and many artifacts are added to surface of the minivan. The color of surface is back-projected from the image.

3.16 The reconstructed surface of the minivan using Algebraic Point Sets Surface (APSS); The boundaries of the minivan are not preserved and APSS overextend the point cloud. The color of surface is back-projected from the image.
3.17 The reconstructed surface of the minivan using Algebraic Point Sets Surface (APSS) from top view; There are many artifacts in the reconstructed surface and this approach does not preserve the edges and corners of the minivan. The color of surface is back-projected from the image. .................................................. 53

3.18 The reconstructed surface of the minivan using Robust Implicit Moving Least Squares (RIMLS); Similar to APSS, this approach cannot preserve the boundaries of the minivan. The color of surface is back-projected from the image. .................................................. 54

3.19 The reconstructed surface of the minivan using Robust Implicit Moving Least Squares (RIMLS) from top view; Similar to APSS, the edges and corners of the minivan are not preserved. The color of surface is back-projected from the image. .................................................. 55

3.20 The reconstructed surface of the minivan using the geometrical variant of the proposed approach; The boundaries of the minivan are preserved and distortion in the point cloud is not significant. The color of surface is back-projected from the image. .................................................. 56

3.21 The reconstructed surface of the minivan using the geometrical variant of the proposed approach from top view; The edges and corners of the minivan are preserved and the shape of the minivan is not distorted. The color of surface is back-projected from the image. .................................................. 57

3.22 The reconstructed surface of the minivan using the brightness based variant of the proposed approach; The brightness based reconstructed surface has less artifacts with respect to the geometrical variant of our approach. Therefore, it has the best performance over the studied approaches. The color of surface is back-projected from the image. .................................................. 58

3.23 The reconstructed surface of the minivan using the brightness based variant of the proposed approach from top view; Similar to geometrical variant, this approach preserves the edges and corners of the minivan. The color of surface is back-projected from the image. .................................................. 59
3.24 The proposed surface recovery approaches in sensor’s perspective (top row) and top view (bottom row) evaluated for a cyclist; (a) Original point cloud, (b) Geometry based surface reconstruction, (d) Brightness based surface reconstruction. Brightness based surface reconstruction has superior performance over geometrical surface reconstruction and provides smoother and more realistic surface. The color of surface is back-projected from the image.

3.25 The brightness based 3D super resolution (a) without and (b) with global constraint. Surface of the minivan is continuous using the global constraint. The color of surface is back-projected from the image.

4.1 The trajectory of the platform, represented by dots, is shown in this figure. The road boundaries are shown with green lines.

4.2 The point cloud collected from 360° horizontally rotating Velodyne laser scanner is shown in this figure. The red dots are the collected points.

4.3 The points that are not located in the field of view of the camera are removed. Therefore, every point in the remaining point cloud are assigned with a gray scale or color value.

4.4 The silver points are the ground points that are selected by fitting a plane to low elevation points. These points belong to the ground plane and should be removed.

4.5 The silver points are the ground points that are selected by fitting a plane to low elevation points. These points are uninformative and should be removed.

4.6 The objects are segmented using Euclidean distance segmentation. Every object is shown with a unique color.

4.7 The static objects are discriminated and shown with the light blue. The moving objects are uniquely colored and they keep their color if they are correctly tracked. If a moving object is detected, a unique color is assigned to the new object.
4.8  The point clouds in two consecutive epochs are shown in red and blue. The green lines show the points correspondence between two epochs. These corresponding points are used to estimate the pose of the object.

4.9  (a) The minivan observed by laser scanner has a sparse point cloud. (b) The 3D super-resolution approach is applied to the sparse point cloud to enhance its resolution and the 3D object tracking of the minivan is performed using the dense point cloud.

4.10 The sensors configuration; GPS/IMU navigation solution, cameras and laser scanner are mounted on the platform [20].

4.11 The moving object tracking from the beginning. All objects are considered static (light blue) in the first epoch.

4.12 The moving object tracking after 0.4 second. The moving objects (brown is minivan and green is cyclist) are detected as early as 0.4 second.

4.13 The moving object tracking after 3 seconds. The cyclist and van are correctly tracked since their color does not change over time.

4.14 The moving object tracking after 6 seconds. The cyclist and minivan are correctly detect, but a parked car is incorrectly labeled as moving object.

4.15 The moving object tracking after 9 seconds. The minivan is far from the platform and the proposed approach stops to track it further.

4.16 The moving object tracking after 12 seconds.

4.17 The moving object tracking after 14 seconds. The cyclist is correctly tracked and the parked cars are correctly labeled as static objects.

4.18 The moving object tracking after 15 seconds. A parked car is incorrectly labeled as moving object.

4.19 (a) The cyclist and (b) minivan has been tracked and their point clouds are shown. The results show that the proposed approach can correctly detect and track the moving objects.
4.20 The tracking error; the Kalman filter has been converged and the tracking error is significantly dropped after few epochs. The tracking error grows in time since the tracking error is the accumulation of inaccuracy in the pose estimation.

4.21 The moving object detection and tracking results for 9 seconds: (a) beginning; (b) after 3 seconds; (c) after 6 seconds; (d) after 9 seconds. There is no moving object during 9 seconds, but some ground points are not removed and incorrectly labeled as moving object in (d).

4.22 The moving object tracking after 12 seconds. Two objects are labeled as moving objects, but one of them is correctly labeled.

4.23 The moving object tracking after 13 seconds. Two objects are correctly labeled as moving objects.

5.1 The spatial coherency between pixels of a traffic light; If the pixels of a traffic light are white circles, \( \omega_t(x_{i-1,i,i+1}) \), and the pixels of the background are gray circles, \( \omega_t(x_{j-1,j,j+1}) \), the labels of white circles should be similar. In other words, the spatial coherency term encourages that pixels with similar color have similar labels.

5.2 If the pixel \( x_i \) is labeled as traffic light in the previous time, it is most likely to be labeled as traffic light in the next time, \( P(\omega_t(x_i) = 1|\omega_{t-1}(x_i) = 1) \gg 0 \). Therefore, the label of pixel \( x_i \) in the current time, \( \omega_t(x_i) \), depends on the label of this pixel in the previous time, \( \omega_{t-1}(x_i) \), and the observation in the current time, \( Z_t(x_i) \).

5.3 The lens activation pattern, \( z_c \), depends on the height cue, \( z_h \), the height cue depends on the shape cue, \( z_s \), the shape cue depends on the color cue, \( z_c \), the GIS cue, \( z_g \), does not depend on other cues, and all observations depend on the label of the pixel, \( \omega_t(x_i) \).

5.4 The color of the traffic light is represented in HSV color space and the color of each lens is modeled as Gaussian. Since HSV is circular a color space in hue component, it has the range of \([0 360)\).

5.5 The ellipse is fitted to the connected pixels and shown by blue line. The probability of the pixels belonging to a traffic light is represented by color which is large inside and small outside of the ellipse.
5.6 The height of the traffic light is modeled as mixture of Gaussians (solid line). The traffic light can be installed either on a pole or suspended over the road and each of these cases is modeled as normal distribution (dashed lines).

5.7 The lens activation pattern indicates that if the red lens of the traffic light is active, the green lens must be inactive. Therefore, it must be dark and its gray value should follow half-normal distribution.

5.8 (a) The traffic light position is retrieved from OpenStreetMap and is projected in the image space. (b) The probability of the pixels belonging to the traffic light is modeled as bivariate normal distribution and shown by color.

5.9 (a) The red color mask applied to the image. There are many red objects in the image. (b) The median filter is applied to remove color noise of camera. (c) An ellipse is fitted to every object and non-ellipse-shaped objects are removed. The search space has been significantly reduced for the potential traffic lights.

5.10 Precision/Recall curve for the KITTI dataset. The best precision/recall rate balance occurs at $P(\omega_t) > 88.4\%$. Number of the traffic lights is 440.

5.11 The traffic light detection for a KITTI dataset; The lower part of the red signal has more illumination and it looks yellow. Therefore, the color of the signal should be evaluated for every pixel and the choice of the color should be estimated in a voting scheme.

5.12 The traffic light detection for a KITTI dataset; The traffic light is partially occluded, but the algorithm can detect the traffic light since it applies the temporal consistency.

5.13 The traffic light detection for a KITTI dataset. The red signal of the traffic light is correctly recognized.

5.14 The traffic light detection for a KITTI dataset. The red and amber lenses are simultaneously active and the proposed approach can correctly detect both signals. The amber traffic light is shown by blue for better visualization.
5.15 The traffic light detection for a KITTI dataset. The traffic light signal
is green and it is correctly recognized.

5.16 The traffic light detection for a KITTI dataset; There is a false positive
that has low probability (61%) and will be removed using the threshold
in (5.3). The amber traffic light is shown by blue for better visualization.

5.17 Multiple traffic light detection in KITTI dataset. The estimated lo-
cation of the traffic lights can be applied to find the traffic light that
corresponds to the each lane.

5.18 Precision/Recall curve for LARA dataset. The best precision and recall
rate balance happens at \( P(\omega_t) > 93.9\% \). Number of the traffic lights is
2486.

5.19 The LARA traffic light detection results from beginning.

5.20 The LARA traffic light detection results after 1 second.

5.21 The LARA traffic light detection results after 3 seconds.

5.22 The LARA traffic light detection results after 6.5 seconds.

5.23 The LARA traffic light detection results after 11 seconds.

5.24 The LARA traffic light detection results after 13.5 seconds.
Chapter 1: Introduction

1.1 Background

More than two million car crashes occurred in the U.S. between 2005 and 2007 [1]. In most of these situations (94%), the accidents happened in the consequence of the drivers' fault. There are various factors that prevented drivers from avoiding the accidents. Among many factors, recognition error (41%), decision error (33%), and performance error (11%) are the leading reasons [1].

We study the traffic perception for an automated vehicle in this dissertation. This system can provide necessary warnings to help the driver to avoid the accidents or it can autonomously take action and avoid the collisions. In brevity, the traffic understanding is studied in the dissertation resulting in the recognition error mitigation. There are scientific endeavors to minimize the intervention of the driver and develop a reliable system to safely drive the vehicle on the road. These systems are autonomous vehicles or driver assistant systems based on the level of automation. In the lowest level of autonomy, the system detects possible risk of accident and provides the driver proper warnings. In the second level of autonomy, the system can control the vehicle for a limited time in addition to the warnings it obtains. In the highest level of autonomy, the system is replaced with the driver and no driver is required to control the
vehicle. Disadvantages of the low level of autonomy is the fact that the warnings may distract or distress the driver and may cause accident. In the high level of autonomy, there is controversy about about the liability of the system in crashes and whether the use of fully automated vehicles is ethical and how it affects driving related jobs.

Among several tasks should be accomplished to detect possible risk of an accident, traffic scene understanding and monitoring is one of the most challenging tasks. Traffic scene monitoring is the procedure to detect the vehicles, cyclists and pedestrians on the road, track them, estimate any sudden break or lane changing, evaluate their aggressiveness and recognize any possible risk of collision. This is not possible unless multiple sensors collect various sources of information and the observations are integrated in a united coordinate system. Therefore, the multiple sensors integration is studied to accomplish the traffic scene understanding and monitoring on a moving platform.

DARPA Urban Challenge (DUC) has proved the reliability of the laser scanner for traffic scene understanding and collision avoidance [45]. We use the laser scanner to collect information of the environment such as road, buildings, and other adjacent vehicles, cyclists, and pedestrians. The navigation solution, which is provided by the integration of GPS and IMU, is applied to navigate the autonomous vehicle (also known as platform) and prior information such as Geospatial Information System (GIS) maps are used to improve the accuracy of the algorithms as well as it speeds up the point cloud processing, which is measured by laser scanner. It should be noted that we aim to provide a reliable solution when we try to minimize the cost. In other words, we refuse to use expensive sensors that are not cost-effective and affordable to be installed on an economy car.
1.2 Problem Statement

The autonomous driving is still a challenge. Data collected from sensors has limited quality, and different environments may affect the sensors’ performance and the assumptions in the algorithms. The radar and imaging sensors are prone to noise and may be cluttered [19]. In addition, the low-cost sensors may have low resolution and further processing is required to improve their resolution. The collected information from the radar and imaging sensors are view dependent and the objects are always self-occluded [19]. Since the imaging sensors are in motion with the platform, the images and point clouds may be prone to motion blurring and point cloud dragging [3]. Therefore, the motion of imaging sensors should be compensated in order to reveal the motion of neighboring objects. In addition to these sensor errors, the environmental factors can adversely affect the quality of the collected information. Snow, fog, and rain may impair the vision, and deteriorates the laser measurements. The lighting situation can range from dark shadows, ambient lighting, or specularity in metallic surfaces. The image algorithms are ambiguous in the textureless regions and consequently the image processing algorithms such as optical flow and stereo matching may deteriorate in these regions. Similarly, the surfaces may not have enough curvature changes and the pose estimation of objects may not be possible. Repetitive patterns such as the leaves of trees may cause incorrect matching in images. Porous objects such as fences can cause similar ambiguity in the point clouds.

In addition to the imaging sensors, navigation sensors are prone to many error sources and different factors can adversely affect their quality. GPS receivers can collect the satellites’ signal and provide the position of the platform. The accuracy of the position can range from a few ten meters to a few centimeters based on the
technologies applied to mitigate the error sources [44]. The GPS signals are blocked by high-rise buildings in urban canyon and the position of the platform may be lost. Furthermore, multipath effect can introduce significant systematic error and deteriorate the GPS position accuracy. IMU sensors are used to obtain navigation solution in the absence of GPS signals. High quality IMU sensors are expensive and may not be cost-effective for the economy cars and navigation solution quickly drifts away in low quality IMUs.

Prior information such as GIS maps can also be erroneous and some refinements may be required. The registration error is common in maps. There may exist some features, which are incorrectly labeled or located. Some details such as the width or number of lanes of the road may not be available in the geospatial databases.

1.3 Objective and Research Questions

The autonomous driving consists of two subtasks and safe driving requires these subtasks to be fully accomplished. The platform must be able to accurately perceive the environment. The other vehicles on the road must be detected and their trajectory must be tracked. The road geometry must be determined and lanes of the road must be detected. The traffic lights and signs must also be recognized. Pedestrians and cyclists must be detected and their position must be accurately estimated.

After environment around the platform is perceived, appropriate decisions must be made and proper actions must be taken. The decisions must be updated over time since the environment constantly changes. Researchers applied the Mrakov Decision Process Model (MDPM) to design the trajectory of the platform to avoid accidents.
The decision making process is not studied in these dissertation and it is an active research area by its own.

The environment perception and traffic scene understanding is studied in the research. The traffic scene understanding is also divided into subtasks, including lane keeping, obstacle avoidance, and traffic light and sign detection. Among these tasks, the obstacle avoidance and traffic light detection algorithms are developed in this dissertation. The autonomous driving and its subtasks are shown in Figure 1.1 and the research focus of this dissertation is highlighted.

The obstacle avoidance is factorized into the following objectives:

- To detect the obstacles and objects on the road.
- To discriminate the moving and static objects.
- To track the moving objects.
- To estimate the trajectory and motion dynamics of the tracked objects.
- To analyze the tracking error of the moving objects.

These research questions are studied in chapter 4, where we apply multiple sensors to detect the objects on the road, find the moving ones, track them, and estimate their dynamics.

The traffic light detection objective is also itemized as follows:

- To develop a Bayesian statistical framework to handle variations in traffic lights.
- To make traffic light detection approach coherent in space and time.
- To evaluate the results of the proposed traffic light detection approach.
• To recognize the traffic lights’ signal.

• To estimate the position of the traffic lights.

These research questions are studied in chapter 5, where we apply a Bayesian statistical framework to detect the traffic lights using color camera, the GPS/IMU navigation solution, and GIS maps. We also apply spatio-temporal consistency constraint to provide coherent traffic light detection approach in space and time. In addition, we apply conic section geometry to estimate the position of the traffic lights with respect to the camera.

Figure 1.1: Autonomous driving task is divided into its component. We propose an approach to increase the resolution of point clouds and focus on the obstacle avoidance and traffic light detection tasks using the enhanced point clouds in this dissertation.

The sensors applied in chapter 4 and chapter 5 have several limitations described in section 1.2. Therefore, these sensors must be integrated to overcome the limitations of
each sensor. For instance, the image based navigation and GIS based geo-localization approaches are utilized in the absence of GPS signals in urban canyon.

One of the prominent shortages of these sensors is the low resolution point clouds generated by low-cost laser scanners. The cost-effective laser scanners have low resolution and are noisy, and therefore, their use in object recognition or tracking is limited. This limitation is severe for the objects far from the platform and the object recognition and tracking approaches may fail to detect these objects or track them. Therefore, we propose 3D super-resolution approach to densify the point clouds and reduce their noise in chapter 3. We extend the existing 2D super-resolution algorithms developed for image processing to 3D point clouds and increase the resolution of the sparse point clouds. The proposed approach has two variants: geometrical and brightness based 3D super-resolution. The objective of this approach is itemized as follows:

- To develop an approach to reconstruct surface of an object.
- To modify the proposed approach to preserve the edges and corners of the object.
- To improve the reconstructed surface using image content.
- To compare the proposed approach to the earlier studies.

The high resolution point clouds are more robust for the object recognition and object tracking. Therefore, we apply 3D super-resolution approach to the point clouds in chapter 3 and utilize the densified point clouds in obstacle detection and moving object tracking to improve their results in chapter 4.
The traffic light detection approach in chapter 5 can be modified to detect the traffic signs. The traffic signs have also specific colors, the follow specific shapes, and there are installed based on the traffic sign installation standards. The lane detection is not studied in this dissertation, but there are many successful approaches that reliably recognize the road markers and detect the lanes of the road [10, 63, 68, 75].

1.4 Related work

This section focuses on the commercial resources and industrial improvements in automatic driver assistance system and autonomous vehicles. Technical literature review is left to the following chapters when the algorithms are explained in detail. As mentioned before, the DARPA Urban Challenge has demonstrated that the autonomous vehicles can drive on the road in future [45]. Google has become one the leading corporations and it has introduced its autonomous vehicle prototype [43]. It is claimed that the prototype has traveled more than 500,000 miles and no accident has happened. However, the prototype is designed to drive no more than 30 miles per hour [43].

Aside from autonomous driving, many automotive companies have introduced different features of the automatic driver assistance system. We describe some of these advancements and the reader is referred to [1] for recent features. Volvo has introduced “City Safety” feature that uses an infrared sensor to detect vehicles driving in the same direction within 18 feet radius [1]. This feature has some limitations in high relative speed and it cannot avoid the accident between two vehicles if the relative speed is more than 19 mph [1]. Mazda has applied radar sensors to detect adjacent vehicles in the blind spot and warn the driver if the turning signal is activated [1].
Buick has introduced ultrasonic sensors that can help the driver to safely park the
car. It alerts the driver if an object is detected closer than 1 foot [1]. Mercedes-Benz
has introduced an adaptive cruise control system that uses radar sensor to avoid
front collision. This system slows down if the platform comes too close to the front
vehicle [1]. Mercedes-benz has also introduced lane keeping assistance which uses
cameras to detect them and keeps the vehicle within the lanes for a few seconds [1].
Acura improves the visibility of the driver using an adaptive headlight. The headlight
direction is changed if the driver turns the steering wheel [1]. Tabled 1.1 states some
of the driver assistant system features that have already developed in automotive
industry. Readers are encouraged to visit manufacturer’s website for the updated list
of features.

<table>
<thead>
<tr>
<th>System</th>
<th>Maker</th>
</tr>
</thead>
<tbody>
<tr>
<td>Forward collision warning</td>
<td>Nissan</td>
</tr>
<tr>
<td>Adaptive cruise control</td>
<td>Mitsubishi</td>
</tr>
<tr>
<td>Lane keeping support</td>
<td>Nissan</td>
</tr>
<tr>
<td>Collision mitigation brake</td>
<td>Honda</td>
</tr>
<tr>
<td>Night vision</td>
<td>Honda</td>
</tr>
<tr>
<td>Lane passing alarm</td>
<td>Benz</td>
</tr>
<tr>
<td>Brake assist with navigation link</td>
<td>Toyota</td>
</tr>
<tr>
<td>Blind spot detection</td>
<td>BMW</td>
</tr>
</tbody>
</table>

Table 1.1: Some active safety features provided by automotive industry [17].
1.5 Scope and limitation of the study

- Navigation in the GPS denied environment is still a challenging topic and many alternatives are suggested in the absence of GPS signals. This paper does not cover any GPS denied environments.

- This paper does not register or modify inaccurate GIS maps and the extracted GIS information is conservatively utilized to avoid any wrong assumptions based on incorrect GIS information.

- There are many more sensors that can be used to improve the traffic scene understanding. However, we focus on the imaging sensors, camera and laser scanner, navigation sensors, GPS and IMU, and GIS maps. Radar and ultrasound, which are extensively used in the automatic driver assistance system, have not been studied in this dissertation.

1.6 Key Contributions

The theoretical contributions of this dissertation include:

- Moving objects detection framework is developed and a Kalman filter with non-holonomic constraints is applied to track the moving objects in the vicinity of the platform.

- 2D super-resolution approaches, which have been developed for image processing, are generalized for 3D point clouds and the diffusion equations (oriented Gaussian filter) are developed to densify the point cloud and filter its range noise.
• The brightness of the surface is connected to the curvature and consequently, images are utilized to improve the resolution of the point clouds.

• A Bayesian probabilistic traffic light detection approach is developed to robustly detect the traffic lights. The results outperform the previous work on a public benchmark.

• Conic section geometry is applied to estimate the position of the traffic lights with respect to the camera.

1.7 Organization of this Dissertation

The rest of this dissertation is organized as follows: Chapter 2 introduces the utilized sensors and explains how various information sources can be integrated together; Chapter 3 introduces a new method to reconstruct surface of the objects. It is a crucial step to densify and denoise the sparse point clouds and provide better tracking results; Chapter 4 describes the details of detecting the static and moving objects. It briefly states the use of different sensor information to detect and track the moving objects at the vicinity of the platform and avoid the potential accidents; Chapter 5 introduces a Bayesian probabilistic framework to detect the traffic lights in the color images and recognize their signal. It utilizes conic section geometry to localize the traffic lights with respect to the camera; Chapter 6 summarizes and concludes multiple sensor integration for autonomous driving and provides the future direction of this study. Appendix A provides the proof to some statements in chapter 3 and 5.
Chapter 2: Multiple Sensor Integration

The automatic driving requires efficient integration of different information sources of multiple sensors. The imaging sensors, including laser scanners and cameras, provide rich source of information about the environment. However, these sensors collect the information in their local coordinate system and changes in the sensor’s location or attitude leads to changes in the collected information. In contrast to imagery sensors, the GPS and IMU sensors measure the dynamics of the platform and it does not provide any information about the environment. Therefore, these sensors should be integrated overcome the drawbacks of imaging and navigation sensors.

Figure 2.1 shows the multiple sensor integration to perform the required tasks in the autonomous driving. The use of multiple sensors improves the reliability of system. For instance, the GPS/IMU navigation solution drifts away in the absence of GPS signals in urban canyon and image based navigation solution can correct the drift and provide a reliable and accurate navigation solution.

2.1 GPS

Global Positioning System (GPS) is a satellite based radio-navigation system that localizes a platform equipped with GPS receivers all over the world. This system measures distance of the satellites to the receiver based on the time of flight of the
Figure 2.1: The traffic scene perception is only achievable when multiple sensors are integrated. The navigation solution of the platform is provided using the integration of GPS and IMU. However, the images and point clouds are applied in addition to IMU sensor to navigate the platform in the absence of GPS. The road geometry is extracted from GIS maps and integrated with images to detect lanes of the road. point clouds, GPS/IMU navigation solution and GIS maps are also integrated to detect and track moving objects. These sensors are complementary and are essential for automated driving.
satellites' signals. The position of the satellites are precisely modulated on the signals and the position of the platform is estimated using the measured distances and the position of GPS satellites. The GPS positioning suffers from various error sources such as the satellites’ orbital and timing determination errors, the receiver’s timing error and hardware limitations, atmospheric delays, and multipath error. Therefore, accuracy of this system varies from a few millimeters to a few ten meters depending on the utilized technology and the error mitigation approaches.

If the low-cost standalone GPS receiver is used, the receiver can measure the code signals transmitted by the satellites and therefore, the positioning accuracy is about 8 meters. If the GPS corrections are received from base stations, most of systematic errors can be eliminated or mitigated and consequently, the positioning error drastically decreases. Wide Area Augmentation System (WAAS) is GPS correction service provider that calculates the GPS errors in the North America and sends the correction messages to the users from its geostationary satellites. Every GPS receiver which is WAAS-enabled significantly reduces its errors and improves its position accuracy to the range of 1 to 3 meters [30]. This system is free and publicly available to the users in the North America.

There are other GPS correction service providerz that cover the position error messages in a region or worldwide. Among these service providers, the OmniStar corrections service provider is a commercial service with worldwide cover. OmniStar uses around 100 base stations worldwide and can provide the position corrections and improve the position accuracy of the rover (the receiver mounted on the platform) to better than 10 centimeters [30]. Since the specific receivers and equipment should
be used and the initial cost of using these commercial corrections service providers is higher. The users may pay additional fees, too.

The most accurate GPS positioning technique is Real-Time Kinematic (RTK) approach. In this case, a carrier phase of GPS signals are used in addition to code in the base and rover stations and it can provide better than 10 centimeters position accuracy. A radio link is required to transfer the information collected in base station to the rover station and the rover station can achieve high accuracy positioning after it resolves the carrier phase ambiguities [44].

Finally, the satellites’ constellation has significant impact on the position accuracy of the rover. If there are few visible satellites or their geometry is poor, the accuracy is deteriorated. Dilution Of Precision (DOP) indicates the number of the satellites and their geometry and consequently, it represents the achievable accuracy of position.

2.2 IMU

Inertial Measurement Unit (IMU) is a set of three accelerometers and three gyroscopes that observes the platform’s acceleration and angular rate around its axes. The observed accelerations and angular rates are converted into rotation and translation of the platform and therefore, the pose of the platform can merely be estimated by IMU. However, the IMU measurements are prone to different error sources such as noise, bias and scale factor error.

Accuracy of IMUs depends on the technology applied in these sensors. Generally speaking, the mechanical and Laser Ring Gyro (LRG) are the most accurate
technologies, Fiber Optics Gyro (FOG) is medium accuracy IMU, and Micro-Electro-Mechanical System (MEMS) is the least accurate IMU sensor. MEMS sensors are inexpensive and they are frequently used in automotive industry.

2.3 Integrated Navigation Solution

As previously mentioned, the GPS positioning requires clear sky view, and therefore no GPS solution is available if the GPS signals are blocked by nearby objects such as the tree canopy and buildings. Therefore, an alternative sensor is required to provide the navigation solution in the absence of GPS positioning. IMU sensors obtain the navigation solution when the GPS signals are blocked for a limited period of time. Therefore, integrated GPS/IMU navigation solution provides a seamless pose estimation of the platform and it has become a standard navigation solution in recent years.

If the low quality IMUs are used, the navigation solution drifts away due to noise, bias and scale factor errors, in the absence of GPS positioning. Therefore, low quality IMUs are not able to bridge the GPS gaps for a long period of time. Higher quality IMUs are expensive and may not be cost-effective for many applications such as autonomous driving. There is an active research to integrate the IMU and vision based navigation system and calibrate (estimate bias and scale factor error) the IMU using the vision based navigation solution [39, 72].

The GPS and IMU integration is performed using estimators such as the Kalman Filter (KF). The Kalman filter is a linear estimator that combines the system model and the observation model and provides the best estimation of unknowns [35]. The
IMU system model predicts the position, velocity and orientation of the platform and the GPS measurements update the predicted values and improves their accuracy.

Since the Kalman filter is a linear estimator, the non-linear functions should be linearized. The Extended Kalman Filter (EKF) is designed to linearize the non-linear systems in an iterative scheme. If the system is highly non-linear, the extended Kalman filter may diverge and the Unscented Kalman Filter (UKF), another variant of the Kalman filter, can be used. The unscented Kalman filter resamples the input data, applies the system to the resampled data and estimates the mean and covariance of the resampled data. Since no linearization is applied, the linearization error is avoided [69]. The Particle Filter (PF) is another extension to the Kalman filter that can be applied to the non-Gaussian data. For more details, readers are referred to [69].

2.4 Camera

The camera sensors are cheap and provide a rich source of information. Image is actually a projection of the reflected light into the image plane. Therefore, images follow the perspective geometry and it is explained for two cases in the following sections.

2.4.1 Projection $3D \rightarrow 2D$

The image taken by camera is a projection of the 3D surfaces of the objects into 2D image plane that obeys the projective geometry. Figure 2.2 shows that 3D point, $X$, is projected into image plane, $x$. 

17
The projection matrix, \( P \), projects point \( X \) of an object into point \( x \) in image plane, such that:

\[
x = PX.
\]  
(2.1)

The projection matrix is decomposed into three components, such that:

\[
P = K[R|t],
\]  
(2.2)

where \( R \) is the rotation matrix between the world (object) coordinate system and image coordinate system, \( t \) is the translation vector between these two coordinate systems and \( K \) is the calibration matrix of the camera. Here, \( | \) is the matrix partitioning sign. The calibration matrix is a \( 3 \times 3 \) matrix, such that

\[
K = \begin{bmatrix}
f & 0 & p_x \\
0 & f & p_y \\
0 & 0 & 1
\end{bmatrix},
\]  
(2.3)

where \( f \) is the lens focal length and \( p_x \) and \( p_y \) are the coordinates of the principal point, \( p \), shown in Figure 2.2. Let’s assume that the number of pixels per unit is \( m_x \).
and \( m_y \) in \( x \) and \( y \) directions of the image coordinate system, then the calibration matrix in (2.3) is changed to:

\[
K = \begin{bmatrix}
\alpha_x & 0 & x_0 \\
0 & \alpha_y & y_0 \\
0 & 0 & 1 \\
\end{bmatrix},
\]  

(2.4)

where \( \alpha_x = f m_x \) and \( \alpha_y = f m_y \) are the lens focal length in pixel units. Also, \( x_0 = m_x p_x \) and \( y_0 = m_y p_y \) are the coordinates of principal point in pixel units. Figure 2.3 displays the principal point under the camera geometry. The camera geometry is shown in the image view in Figure 2.3a and the camera profile view in Figure 2.3b.

![Camera geometry](image)

Figure 2.3: The intrinsic geometry of the camera [26]. is demonstrated in (a) the camera profile and (b) image view. The lens focal length, \( f \), and the principal point, \( p \), are the camera calibration parameters.

### 2.4.2 Projection 2D → 2D

The 3D points may be located on a plane and the projection matrix is converted into the homography transformation. If the 3D points are located in a plane, the \( z \) coordinate of the points in the plane coordinate system is zero. Therefore, the
homography transformation is applied to transfer the points on the plane, \( X \), into the image space, \( x \), by \( x = K[R|t]X \). The rotation matrix consists of three column vectors, \( R = [r_1 \ r_2 \ r_3] \). A point on the plane is \( X = [X \ Y \ 0 \ 1]^T \) and 2D points on the image is \( x = [x \ y \ \omega]^T \) in homogenous coordinates. Therefore, the projection matrix is converted to the homography transformation, such that:

\[
\begin{bmatrix}
  x \\
  y \\
  \omega \\
\end{bmatrix} = K[r_1 \ r_2 \ t] \begin{bmatrix}
  X \\
  Y \\
  1 \\
\end{bmatrix} = KRCX,
\]  

(2.5)

where \( C = \begin{bmatrix}
  1 & 0 & c_1 \\
  0 & 1 & c_2 \\
  0 & 0 & c_3 \\
\end{bmatrix} \), and \( c = [c_1 \ c_2 \ c_3]^T \) is the translation vector of the camera center at the plane coordinate system, \( t = -Rc \). The homography transformation, \( H \) is estimated by

\[
H = KRC,
\]  

(2.6)

The homography matrix is a \( 3 \times 3 \) matrix with a rank of 3. Therefore, this matrix is invertible and it provides a one-to-one projection between the points on the plane and the points in the image space. Figure 2.4 shows the homography transformation between planar points and the image space. The transformation between two images of a planar object is homography transformation, too.

### 2.5 Laser Scanner

The laser scanner generates 3D coordinates of the points on surface of nearby objects. The coordinates of these points are estimated by measuring the travel time of the emitted laser beams and their reflection. In addition to 3D coordinates of the points, the laser scanners may integrate the point cloud with the camera and produce the color point cloud. However, this integration may increase the cost of the laser scanner and many laser scanners do not have color or brightness. If the lever arm
Figure 2.4: Projection of the planar points into the image space is the homography transformation [26]. The projection of the image of the planar points in two epochs is also the homography transformation. $\mathbf{R}$ and $\mathbf{t}$ are rotation matrix and translation vector between two camera exposure stations.

and boresight of these two sensors are known, the color is assigned to the 3D points using the projection matrix. Some of the laser scanners measure the intensity of the reflected laser beams. The intensity of the reflected laser beams can be used to weigh the observations. In other words, the stronger signal of the reflected laser beams indicate to the more accurate measurements.

The density of the point cloud depends on the angular resolution of the laser scanner. Laser scanners with high angular resolution are expensive. On the other hand, low-cost laser scanners are cost-effective for autonomous driving, but their angular resolution may not be sufficient for high level applications such as object tracking. Therefore, it is crucial to populate the low resolution point clouds and improve their resolution. The higher resolution point cloud are more robust for 3D tracking and object recognition. We propose an approach to improve the resolution
of the point clouds in chapter 3 and utilize the improved point clouds for the 3D tracking of the moving objects in chapter 4.

Figure 2.5 shows the point cloud projected into the image space and the depth of each point is shown by color. If the platform is in motion, the laser scanner point cloud will be dragged. In other words, the points at the beginning of the rotation cycle of the laser scanner are collected in a different coordinate system rather than the points in the end of the rotation cycle. This error, which is also called motion drag, can be mitigated at the hardware level [3]. Unfortunately, there is no effective motion drag removal at the software level if the laser beams are not time-tagged.

![Figure 2.5: (a) The scene. (b) The laser scanner point cloud is projected into image space and the depth of points is shown by color.](image)
2.6 GIS maps

Satellite imagery provides rich information sources such as photo images and thermal maps. These images are extensively used for many applications such as in agriculture and urban planning. However, this kind of information is not effective for traffic understanding. First, there are many objects that should be removed from these images to provide a useful map. For instance, the moving cars on the road should be excluded from the map. Second, it is not trivial to extract important information. For example, the roads are not easy to be extracted from these images. Third, there are some objects such as traffic lights and traffic signs that may not be visible from satellite images. For these reasons, satellite images are not efficient and these images should be converted into vector maps.

Information in vector maps are stored in the topological data structure [51]. The objects such as traffic lights and traffic signs are point features, the objects like roads are linear features represented by their vertices, the objects such as buildings are polygon features represented by the vertices. The vector maps are specialized for spatial queries. For example, if the navigation solution is inaccurate and the location of the platform does not lie on the road due to the inaccuracy of the navigation solution, the closest road link is retrieved from vector maps and the position of platform is projected into the road links. Unfortunately, generating the vector maps is costly. Therefore, most of vector maps are prepared for specific purposes and these maps cover a local region depending on the extent of a specific project.

OpenStreetMap (OSM) is the only worldwide vector map which is provided by users on the world. It is an open source database which users collect, maintain and update its information. Therefore, the accuracy of this database depends on the
accuracy of the information users provide. Since many users utilize smartphones and handheld GPS receivers to position the features, data may not be accurately collected. In addition, the database may not be complete and some features may be missed. The road links are extracted from the vector OSM map and overlaid on the raster OSM map (OSM has also raster GIS map) in Figure 2.6.

![Figure 2.6: The road links are extracted from vector OpenStreetMap and shown in pink. The road links are overlaid on the raster OpenStreetMap.](image)

In addition to OSM, Google Maps also provide an accurate vector GIS map. Google provides an API to retrieve the raster version of google maps. However, the
vector version of Google Maps is proprietary and it is not publicly available. The monopoly of vector version of Google maps makes Google company more advantageous over other commercial and academic projects especially for the detection of static objects such as roads and traffic lights.

There are certain criteria that should be satisfied for the information to be useful. First, the features should be close to the platform, otherwise these features are not immediate threat for the safety of the platform. Second, layers of information such as buildings can be used for navigating the platform in the GPS denied environments, but these layers should be removed for the moving object detection. Third, some levels of abstraction may be required for too detailed features. For instance, representing a curve with the thousands of points may be converted to a few points.
Chapter 3: 3D Super-resolution

3.1 Introduction

Low-cost laser scanners provide low resolution point cloud with range accuracy that may not be sufficient for many applications. Hence, the laser scanner’s angular resolution is fixed with fewer observed points on the farther surfaces. For instance, the observed points spacing for an object at 1 meter is 1.7 millimeters with the laser scanner angular resolution of 0.1°. In contrast, this spacing for an object at 50 meters is 8.7 centimeters. Therefore, a human may be represented by few points at 50 meters. The point cloud for the farther objects is too sparse and many computer vision tasks such as object tracking and object classification may not be feasible. In this study, we address an approach that improves the resolution of the point cloud and filters noise of the points.

The improved point clouds are utilized in object recognition and 3D tracking.

In order to find the optimum surface, the surface should be modeled using these observed points. However, the contribution of these points may not be equal. In Figure 3.1, it is shown that if different points are chosen, it results in different surfaces. For instance, if two planes are perpendicular to each other, and four points are located
at these two planes, the estimated surface using two points with small normal changes (small curvature) outperforms the choice of two points with large normal changes.

2D super-resolution in an active research and Partial Differential Equations (PDEs) have been utilized to increase the resolution of images [4, 71, 60]. In this section, we generalize the 2D super-resolution approaches applied to images into 3D point cloud and reconstruct surface of the objects. The proposed approach weighs the points in such a way that the reconstructed surface follows the geometry of the observed points.

![Figure 3.1: Linear interpolation results in different values with different weights of points. (a) Four points are located on two perpendicular planes. (b) If the points are incorrectly chosen or weighed, the interpolated point (red dot) will not follow the geometry of the surface (two planes). (c) The correct weights of the points leads to the interpolated points that follows the geometry of the surface.](image)

In order to formulate the surface reconstruction, different characteristics of the surfaces are briefly overviewed [7]. The surfaces are either open or closed. Surfaces with boundaries are open surfaces. Otherwise, they are close surfaces. The number of holes in the surface and its openness/closeness represent its topology. Since the laser scanner samples the points of surfaces from its own perspective, surfaces of the
objects are observed partially self-occluded. In other words, surface of an object is an open surface if it is self-occluded.

Surface of an object is represented in several ways. First, it can be represented by linear combination of linear or non-linear orthogonal or non-orthogonal functions. This representation is also called explicit representation of a surface of an object [47]. The spherical harmonics is one of these representations that uses linear combination of sine and cosine functions to model surface of objects. The spline, which is a piecewise polynomial function, and its variant have also frequently used. The surface represented by these functions are suitable for closed surfaces.

In addition, surface of object can be implicitly represented. Surface of an object may be sampled by points and therefore, the sampled points represent this surface [47]. The higher resolution and more accurate sampled points create a better presentation of surface. Moreover, surfaces may be represented by the geometric shapes such as triangle, which are called mesh [47]. For more details of surfaces’ different representations, readers are referred to [66].

Another approach to implicitly represent surface of the objects is the level sets [54]. In level sets, a function is defined in the way that it is positive inside the object and negative outside the object and it is schematically shown in Figure 3.2. Therefore, surface of the object is represented by zero crossing of this function. Level sets are originally defined for the closed surfaces and they should be modified to operate on the open surfaces. Therefore, level sets are three dimensional functions that their zero value is located at surface on the objects.
Figure 3.2: The surface of an object is implicitly represented by an indicator function. This function is a signed distance function of the surface for each point and it has positive values for points inside the object and negative values for the points outside. The surface of this object is defined by the zero crossing of the indicator function.

3.2 Mathematical Framework

Surface of an object is a 2D manifold embedded into three dimensional Euclidean space. If this manifold is smooth and differentiable, it is a Riemannian manifold and its tangent plane can be defined for every point on this surface. Let’s assume point $X_0$ is a point of the point cloud collected from an object’s surface. The tangent space $T(u, v)$ is defined where $u$ and $v$ are the orthonormal bases of this space and the normal vector, $n$, is defined at the point $X_0$. The vector $n$ is perpendicular to the vectors $u$ and $v$ and these three vectors construct three orthonormal vectors sufficient to represent every point in the neighborhood of point $X_0$. The point $X_0$ is the origin of this coordinate system and the normal vector $n$ is its third axis. The choice of $u$ and $v$ is not important as long as they are orthogonal. This coordinate system is
defined in the tangent space of point $X_0$ and is valid for the neighborhood of this point, $\Omega(X_0)$.

Let's assume that the point $X' = [u_{X'} \quad v_{X'} \quad w_{X'}]^\top$ is located in the neighborhood of point $X_0$, where $u_{X'}$ and $v_{X'}$ represent its coordinates in tangent space and $w_{X'}$ is its distance from the tangent plane. The distance of the point from tangent plane can be modeled using a function $W(u_{X'}, v_{X'})$. Therefore, this point can be represented as $X' = [u_{X'} \quad v_{X'} \quad W(u_{X'}, v_{X'})]^\top$.

Figure 3.3 shows profile of the local coordinate system. The point $X'$ located on the surface is unknown and this will be determined if $W(u_{X'}, v_{X'})$ is estimated.

Figure 3.3: The profile view of the neighborhood of point $X$. The point $X'$ is represented by the vector $[u_{X'} \quad v_{X'} \quad W(u_{X'}, v_{X'})]^\top$. The surface is reconstructed if $W(u_{X'}, v_{X'})$ is determined.
3.2.1 Geometrical 3D super-resolution

There are infinite number of surfaces pass through the finite number of points. Therefore, surface reconstruction based on the point cloud of an object is an ill-posed problem and it cannot be solved without additional observations or constraints. In order to overcome this problem, we assume the surface is smooth and it is differentiable everywhere. Therefore, the local surface in the neighborhood of point $X$ is smooth. The gradient of $W(u_X, v_X)$ indicates to the smoothness of the surface. The gradient operator is represented by $\nabla$ and $\nabla W(u_X, v_X) = [\frac{\partial W(u_X, v_X)}{\partial u}, \frac{\partial W(u_X, v_X)}{\partial v}]^\top$.

In the rest of this dissertation, we shorten $W(u_X, v_X)$ to $W$ and also we use $W_u$ and $W_v$ instead of $\frac{\partial W}{\partial u}$ and $\frac{\partial W}{\partial v}$.

The smoothness is guaranteed by minimization of the global energy function. The global energy function is formulated, such that:

$$E = \int\int_{\Omega} \Phi(||\nabla W||)dudv, \quad (3.1)$$

where $||.||$ is $L_2$ norm and $\Phi(.)$ is an arbitrary differentiable convex function. There are several choices of $\Phi$ in the literature [71], but we use the quadratic function for $\Phi$. Therefore, (3.1) changes to:

$$E = \int\int_{\Omega} ||\nabla W||^2dudv. \quad (3.2)$$

The Euler-Lagrange equations can be applied to minimize (3.2) and it results in (see the appendix for proof):

$$\frac{\partial W}{\partial t} = div(\nabla W) = \Delta W, \quad (3.3)$$

where $\Delta$ is the Laplacian operator, $div$ is the divergence operator, and $\frac{\partial W}{\partial t}$ shows the changes of $W$ over time $t$. 

31
Equation (3.3) indicates to a famous problem in partial differential equations known as heat flow. This equation can be solved using gradient decent minimization in a few iterations. In every step, the surface is estimated using the estimated surface in the previous iteration:

\[ W_k = W_{k-1} - \frac{\partial W}{\partial t} \delta, \quad (3.4) \]

where \( \delta \) is the step size of each iteration. \( W_{k-1} \) is the estimated value of \( W \) in the previous iteration and \( W_k \) is the estimated value of \( W \) in the current iteration. In the gradient descent approach, the initial value of \( W \) should be known in the first iteration.

Minimization of (3.1) guarantees the surface smoothness. However, this equation uniformly smoothes the surface and therefore, the edges and corners are smoothed, too. This is also called isotropic diffusion filtering. Since the edges and corners should be preserved in the point cloud, a diffusion tensor \( \mathcal{D} \) is used in (3.3) to suppress the smoothness in edges and corners, such that:

\[ \frac{\partial W}{\partial t} = div(\mathcal{D} \nabla W). \quad (3.5) \]

Equation (3.4) is also modified:

\[ W_{k+1} = W_k - \frac{\partial \mathcal{D} W}{\partial t} \delta. \quad (3.6) \]

Equation (3.5) is also called anisotropic diffusion filtering.

It is mentioned that surface estimation based on the point cloud is an ill-posed problem. Equation (3.5) is also an ill-posed problem without introducing other constraints. However, integrating the observed point cloud of an object and the smoothness constraint in (3.5) provides a unique solution.
If points of the point cloud are accurately observed from the surface, these points should not change and the surface should be estimated in the way that the estimated surface passes through these points. Therefore, these points should not change over time and $\frac{\partial W}{\partial t} = 0$. For every point $X'$ on the surface, it is estimated by:

$$\frac{\partial W}{\partial t} = \begin{cases} \text{div}(D\nabla W) & X' \notin X^i \\ 0 & X' \in X^i. \end{cases}$$ (3.7)

If points of the point cloud are noisy, these points can be smoothed and noise is filtered out. If the smoothness rate of these points is $\epsilon$, these points change over time with the rate of $\epsilon$ and therefore, $\frac{\partial W}{\partial t} = \epsilon$. In this case, for every point $X'$ on the surface, it is estimated by:

$$\frac{\partial W}{\partial t} = \begin{cases} \text{div}(D\nabla W) & X' \notin X^i \\ \epsilon & X' \in X^i. \end{cases}$$ (3.8)

In this case, the surface is reconstructed in the way that it passes between the points of the point cloud and provides smooth surface. Figure 3.4 shows the point cloud of a minivan and a local neighborhood of a point on this minivan is represented by a red rectangle. The point cloud of this minivan is segmented from the original point cloud. The segmentation procedure will be described in the next section and we do not address it in this section.

Figure 3.5 shows a planar surface of an object estimated from a few points of the minivan in this neighborhood in Figure 3.4. The planar surface estimated by least squares approach does not pass through some of the points in the surface. This planar surface is used to initialize the iterative gradient descent in (3.7) and (3.8). The surface of the minivan in Figure 3.6 is estimated assuming the points of point
cloud are accurate and do not need to be smoothed. Equation (3.8) is applied to estimate the surface of the minivan in Figure 3.7.

**Diffusion tensor**

In Figure 3.1, it is shown that the interpolated point depends on the weight of the observed points. Two planes of surface create an edge and this edge will be smoothed if the weights of the observed points are not properly chosen, shown in Figure 3.1b. If the weight of observed points is chosen higher along the edges, the interpolated points will follow the geometry of surface, as shown in Figure 3.1c.

The diffusion tensor should be selected in the way that it preserves the edges and corners and does not smooth them. Therefore, it should be chosen in the way that the
weight of the observed points increases along the edges and corners and the weight of observed points decreases across the edges and corners. The structure tensor, $G$, represents the geometry of surface, computed by:

$$G = \nabla W^T \nabla W.$$  \hspace{1cm} (3.9)

This tensor is a $3 \times 3$ symmetric matrix. If this matrix is decomposed to its eigenvalues and eigenvectors, the eigenvector corresponding to its smallest eigenvalue is the normal to the surface. Let’s call the eigenvalues of the structure tensor $\lambda_1 \geq \lambda_2 \geq \lambda_3 \geq 0$ and $\theta_1, \theta_2, \theta_3$ are eigenvectors corresponding to these eigenvalues. Therefore, $\theta_3$ represents the normal direction of surface and $\theta_1, \theta_2$ are located on surface. The diffusion
Figure 3.6: The surface is estimated in the local neighborhood using the smoothness constraint and the observed points. The points of point cloud are assume accurate and are not smoothed. Therefore, the surface is estimated in the way that it passes through the observed points and observed points remain intact. The color of the surface is back-projected from the image.

matrix is chosen as a function of two eigenvalues, $\lambda_1, \lambda_2$, such that:

$$D = f_1(\lambda_1, \lambda_2)\theta_1\theta_1^T + f_2(\lambda_1, \lambda_2)\theta_2\theta_2^T,$$

(3.10)

where $f_1(.)$ and $f_2(.)$ are monotonically decreasing functions.

The design of diffusion matrix has been extensively studied. In their seminal work, Perona and Malik have applied the inverse of exponential function of the squared gradient to penalize the smoothness in the direction of gradient [57]. We apply one of the popular choices of the diffusion matrix [4, 71, 60], such that:
Figure 3.7: The observed points are noisy and they are smoothed. Therefore, the surface passes between the observed points and generates a smooth surface. The color of the surface is back-projected from the image.

\[
 f_1(\lambda_1, \lambda_2) = \frac{1}{\sqrt{1 + \lambda_1 + \lambda_2}} \\
 f_2(\lambda_1, \lambda_2) = \frac{1}{1 + \lambda_1 + \lambda_2}. \tag{3.11}
\]

The diffusion matrix is calculated by (3.10) and (3.11) and is replaced in (3.8) to estimate the interpolated points of surface.
3.2.2 Brightness based 3D super-resolution

Shade from shaping is the surface reconstruction based on the brightness changes of surfaces. Horn and his colleagues applied partial differential equations to reconstruct surface based on the brightness in their seminal work [31]. This problem is an ill-posed problem and surface of an object can be reconstructed by adding some assumptions and constraints. Since illumination is complicated in the real world, this approach is only applied in laboratory environments with controlled illumination. If different controlled lightings are available, the 3D model of surface can be estimated using photometric stereo. This approach is limited and it does not work for the realistic scenarios with complicated illumination.

An image of surface is actually the reflected light from surface sensed by image sensors. The reflectance depends on the light source energy, its direction, the camera perspective, the material of surface, and the normal vector of surface. The function that relates the reflectance to its components is called Bidirectional Reflectance Distribution Function (BRDF).

Let’s assume the tangent space is defined for point \( \mathbf{X} \) and a point \( \mathbf{X}' \) is in the local neighborhood of this point. The gray value of these points are \( I(\mathbf{x}) \) and \( I(\mathbf{x}') \), respectively. The gray value difference of these points indicates change in the normal direction of surface at these points [33], such that:

\[
\arccos(n_{\mathbf{x}}^\top n_{\mathbf{x}'}) \simeq |I(\mathbf{x}) - I(\mathbf{x}')|, \tag{3.12}
\]

where \( n_{\mathbf{x}} \) and \( n_{\mathbf{x}'} \) are normal vectors of surface at points \( \mathbf{X} \) and \( \mathbf{X}' \). The normal vector at point \( \mathbf{X} \) is \( n_{\mathbf{X}} = [0, 0, 1]^\top \). The normal vector at point \( \mathbf{X}' \) is \( n_{\mathbf{X}'} = \frac{1}{\mu}[W_u, W_v, 1]^\top \), where \( \mu = \sqrt{1 + W_u^2 + W_v^2} \) enforces this vector to be a unit vector.
These two normal vectors are replaced into (3.12) and they lead to:

$$\|\nabla W\| \simeq \tan |I(x) - I(x')|.$$  \hfill (3.13)

Equation 3.13 is applied to estimate the interpolated points using the image content. In other words, it provides a smooth surface if the brightness of point \(X, I(x)\), and the brightness of point \(X', I(x')\), are similar. We utilize this equation as a regularizer to estimated the interpolated points on surface. The energy cost function is constructed based on (3.13), such that:

$$E = \int\int_\Omega \Phi(\|\nabla W\| - \tan |I(x) - I(x')|)dudv.$$  \hfill (3.14)

where \(\Phi(.)\) is an arbitrary differentiable convex function and we choose a quadratic function as \(\Phi(.)\):

$$E = \int\int_\Omega (\|\nabla W\| - \tan |I(x) - I(x')|)^2dudv.$$  \hfill (3.15)

Minimization of cost function (3.15) leads to a surface satisfying (3.13). The Euler-Lagrange equations are applied to minimize the energy cost function in (3.15) (the proof is given in the appendix), such that:

$$\frac{\partial W}{\partial t} = \text{div}(\|\nabla W\| - \tan |I(x) - I(x')|) \frac{\nabla W}{\|\nabla W\|},$$  \hfill (3.16)

where \(\text{div}\) is the divergence operator and \(\frac{\partial W}{\partial t}\) indicates to the changes of \(W\) over time. If the brightness does not significantly change between these two points, \(|I(x) - I(x')|\) is almost zero and (3.16) is converted to (3.3). If the brightness changes are significant, it smoothes the surface in the direction that it has the minimum brightness changes. Equation (3.16) is an ill-posed problem and observed points in the point cloud should be used to provide a unique solution to this equation. Points on surface of an object
are estimated, such that:

\[
\frac{\partial W}{\partial t} = \begin{cases} 
\text{div}(\sqrt{\|\nabla W\|} - \tan|I(x) - I(x')| \frac{\nabla W}{\|\nabla W\|}) & X' \notin X^i \\
\epsilon & X' \in X^i 
\end{cases}
\] (3.17)

where \(X^i\) are observed points in the local neighborhood and \(X'\) is an arbitrary point on surface. If smoothing rate of the observed point is chosen to be zero, the observed points will not change and the surface passes through these points. Figure 3.8 shows surface of an object in the local neighborhood based on (3.17), where \(\epsilon = 0\).

![Figure 3.8: Surface of an object in the local neighborhood is constructed based on the brightness changes of points. Therefore, change in brightness between points leads to change in the surface. The smoothness rate of the observed points is 0 and these points remained intact. The color of the surface is back-projected from the image.](image)

40
When the smoothness rate of the observed points, $\epsilon$, increased, these points are smoothed and the resulted surface passes between these points. Figure 3.9 shows smoothed surface passing between the observed points.

Figure 3.9: The smoothness rate of the observed points is larger than 0 and therefore, the observed points are smoothed and the resulted surface passes between these points. The color of the surface is back-projected from the image.

### 3.2.3 Global continuity

When local surfaces are estimated, they should be transformed into the object coordinate system. Since the normal estimation of the observed points does not provide uniform and seamless normal vectors in practice, the locally estimated surfaces will
not be perfect. Therefore, we develop a global constraint to smooth out the discontinuities and obtain a continuous surface. The smoothness filter can be applied, such that:

\[
\tilde{W}_{X'} = \int w_{W_{X'}} W_{X'} \, dW_{X'}
\]

\[
w_{W_{X'}} = \int \exp\left(\frac{(X' - X'^N) \mathbf{n}}{\|X' - X'^N\|}\right) \, dX'^N,
\]

where \(\mathbf{n}\) is the normal vector at the point \(X'\) and \(W_{X'}\) is the third component of the point \(X'\). The neighboring points of the point \(X'\) are represented by \(X'^N\) and \(w_{W_{X'}}\) is the weight of the points in this neighborhood. \(\tilde{W}_{X'}\) is the third component of the point \(X'\) after global constraint.

### 3.3 Implementation

Equation (3.6) is an iterative approach to minimize the proposed energy cost functions. Therefore, an initial surface is required and it converges to surface of object in (3.6). The initial surface is generated by a \(n \times n\) planar grid on the tangent plane in the local neighborhood of the point \(X\). Figure 3.10 shows a 100 \(\times\) 100 grid in this neighborhood and the color of points shows the depth of the grid points in the tangent plane. The observed points in this neighborhood, are converted to the local coordinate system and shown by red dots.

Since point \(X\) is located at the boundary of object, the observed points are located in one side of this point. The size of grid is chosen based on the density of the observed points. In some approaches such as Moving Least Squares (MLS), a circular neighborhood is chosen for the grid points within this neighborhood [2, 56]. However, the circular neighborhood overextend the neighborhood at the boundaries of object.
In this paper, an ellipse based neighborhood is utilized to preserve the boundaries. Therefore, the neighborhood is represented by the grid points inside the ellipse.

The ellipse that represents the neighborhood of point $X$ is constructed by $c = U\Lambda U^T$, where $\Lambda = \text{diag}(\lambda_1, \lambda_2)$ and $U$ is a matrix constructed from eigenvectors, $\theta_1$ and $\theta_2$. $c$ is a $2 \times 2$ symmetric matrix that represents an ellipse in the tangent space and the points inside this ellipse are $x^Tc x \leq 0$. Therefore, every grid point that obeys $x^Tc x \leq 0$ is considered as the point in the neighborhood of point $X$.

The observed points, $X^i$, are converted to the local coordinate system if the center of object coordinate system is transferred to the point $X$ and the third component of the object coordinate system, $z$, is aligned to the third component of the local coordinate system, $n$. This transformation is accomplished by

$$X^i_l = R_2(\beta)R_1(\alpha)(X^i - X).$$

(3.19)

where $\alpha = \arctan\left(\frac{n_2}{n_3}\right)$ is the rotation angle around the first axis of local coordinate system, $\beta = -\arccos\sqrt{n_2^2 + n_3^2}$ is the rotation angle around the second axis of local coordinate system and $n = [n_1, n_2, n_3]^T$ is the normal vector of point $X$ in the global coordinate system.

Spacing between the grid points is too small with respect to spacing between the observed points. Figure 3.10 shows the grid points are closely spaced in contrast to the observed points which are scattered. Therefore, the numerical conditioning [26, 27] may be required to numerically improve the geometry of the points and provide a more robust solution.

Let’s assume that the average of the observed points is $\bar{X}$, origin of the local coordinate system may be transferred to the average of the observed points, $\bar{X}$. In addition, the local coordinate system should be scaled in the way that the maximum
Figure 3.10: A $100 \times 100$ grid generated in the local neighborhood of the red rectangle in Figure 3.4. The sampled points are transformed to the local coordinate system and displayed with red dots. The color of grid points show the distance of these points from tangent plane.

distance to the observed points becomes $\sqrt{3}$. In other words, if distance of the farthest observed point to origin, $\overline{X}$, is $d_{\text{max}}$, all of the points should be scaled by $\frac{\sqrt{3}}{d_{\text{max}}}$ [26, 27].

3.4 Experiments

In order to evaluate the proposed approach, we use the Karlsruhe Institute of Technology (KITTI) benchmark. This benchmark consists of various datasets collected
from urban, rural, residential and highway environments. Each dataset includes the collected data from several sensors such as integrated GPS/IMU output, laser scanner, and stereo monochromic and color cameras [20, 21]. The boresights and lever arms of the sensors are provided and the sensors are synchronized with an acceptable accuracy (a few milliseconds). Here, we focus on the laser scanner’s point clouds and images to evaluate our propose 3D super-resolution approach.

The point cloud is collected from Velodyne HDL-64E laser scanner that generates a point cloud with 2 centimeters range accuracy and the maximum range of this laser scanner is 120 meters. Its horizontal and vertical fields of view are $360^\circ$ and $26.8^\circ$, respectively. It has $0.09^\circ$ and $0.4^\circ$ angular resolution horizontally and vertically.

The image that has been registered with this point cloud is taken from a PointGray Flea2 monochromic camera. It has $90^\circ$ opening angle and global shutter. The lens focal length of the camera is 4 millimeters with the pixel size of $4.65\,\mu m$. These sensors are installed on a car shown in Figure 3.11. Since the lower and upper parts of the image are cropped and the resolution of the cropped images is $1276 \times 333$ [20, 21].

Unfortunately, KITTI benchmark does not provide ground truth for the surface reconstruction. Therefore, we segmented two objects to reconstruct their surfaces. These objects are a minivan and a cyclist. The minivan is a challenging object to reconstruct its surface. The windows of the minivan permit the laser beams to pass through so that the windows of the minivan are observed as holes on this object. In addition, depth of the points reflected from the side of the minivan significantly changes and therefore, the density of the observed points changes across side of the minivan. The cyclist is also a challenging object. There are many small parts of the object that do not contain any observed points. The wheels of the bicycle is one of
Figure 3.11: The sensors configuration; GPS/IMU navigation solution, cameras and laser scanner are mounted on the platform [20].

these objects. The hands and feet of the cyclist are also sampled by relatively few points.

Berger et al. have prepared a benchmark that compares several surface reconstruction approaches [6]. They have compared explicit and implicit surface reconstruction approaches including Poisson, Wavelet, Radial Basis Function (RBF), Fourier, Multi-level Partition of Unity (MPU), Moving Least Squares (MLS), Algebraic Point Sets Surfaces (APSS), and Robust Implicit Moving Least Squares (RIMLS) [6]. Their inputs to these algorithms are the non-uniformly sampled point cloud of an objects. Since these point clouds represent closed surfaces and our proposed approach is designed for open surfaces, this benchmark is not applicable for our proposed approach.
In addition, the image content, required for our approach, is not available in this benchmark.

Berger et al. compared these approaches and concluded that MLS and its variants such as APSS and RIMLS, provide better results [6]. Fortunately, MLS and its variants are applicable for the open surfaces and we compare our results with these approaches. Since we do not have a ground truth, the quantitative comparison is not feasible. However, the qualitative comparison is provided to evaluate the results of proposed approach with respect to earlier studies.

3.5 Results

Figure 3.12 shows segmented point cloud of a minivan. The boundary of the minivan is extracted from the image and it is shown with a blue line in this figure. The background color is changed to provide better visualization. In addition, top view of this minivan is shown in Figure 3.13. This view provides more information about the estimated depth of these points.

The Moving Least Squares (MLS) approach has been used to reconstruct surface of the minivan. MLS approach uses least squares plane fitting to the points in the local neighborhood [2, 56]. MLS reconstructed surface of minivan and the results are shown in Figure 3.14 and Figure 3.15.

Since it uses the circular neighborhood, it overextends the boundaries and significantly distorts the geometry of surface. Therefore, it does not preserve the edges and corners of object and adds many artifacts to the point cloud.
Algebraic Point Sets Surface is another surface reconstruction approach. This approach also reconstructs surface of an object in the local neighborhood [23]. Figure 3.16 and Figure 3.17 show the reconstructed surface of the minivan from laser scanner’s perspective and top view.

However the results of APSS approach looks better than MLS and it introduces less artifacts, there are still a few points of the reconstructed surface which do not follow the geometry of surface. In addition, this approach does not preserve the edges and corners and estimated depth of the points has many outliers.

Robust Implicit Moving Least Squares (RIMLS) is a modification of MLS to overcome the shortages of MLS and preserve the edges and corners of objects [55]. The results of RIMLS surface reconstruction approach are shown in Figure 3.18 and 3.19.
Figure 3.13: The segmented point cloud of the minivan from top view; The color of the points is back-projected from the image.
Figure 3.14: The reconstructed surface of the minivan using Moving Least Squares (MLS) approach; The boundaries of the object are not preserved and the its shape is significantly distorted. The color of surface is back-projected from the image.
Figure 3.15: The reconstructed surface of the minivan using Moving Least Squares (MLS) approach from top view; The point cloud is severely distorted and many artifacts are added to surface of the minivan. The color of surface is back-projected from the image.
Figure 3.16: The reconstructed surface of the minivan using Algebraic Point Sets Surface (APSS); The boundaries of the minivan are not preserved and APSS overextend the point cloud. The color of surface is back-projected from the image.
Figure 3.17: The reconstructed surface of the minivan using Algebraic Point Sets Surface (APSS) from top view; There are many artifacts in the reconstructed surface and this approach does not preserve the edges and corners of the minivan. The color of surface is back-projected from the image.
The results of this approach are similar to the results of APSS and do not preserve the edges and corners of the minivan.

![Minivan reconstruction](image)

Figure 3.18: The reconstructed surface of the minivan using Robust Implicit Moving Least Squares (RIMLS); Similar to APSS, this approach cannot preserve the boundaries of the minivan. The color of surface is back-projected from the image.

The proposed 3D super-resolution approach in this study has two variants: geometrical and brightness based. The geometrical 3D super-resolution approach is applied to the point cloud of the minivan and the results are shown from laser scanner’s perspective in Figure 3.20 and top view in Figure 3.21. The geometrical 3D super-resolution approach follows the geometry of surface and it preserves the edges and corners. In addition, it does not overextend the boundaries of the minivan. Moreover, the windows of this minivan are not observed and they are appeared as holes in
Figure 3.19: The reconstructed surface of the minivan using Robust Implicit Moving Least Squares (RIMLS) from top view; Similar to APSS, the edges and corners of the minivan are not preserved. The color of surface is back-projected from the image.

the point cloud. This approach remains these holes in contrast to earlier studies that fill in them.

In addition, the image content is applied to reconstruct surface of the minivan. The results of brightness based surface reconstruction are shown from laser scanner’s perspective in Figure 3.22 and top view in Figure 3.23. This approach preserves the edges and corners and does not overextend the boundaries of the minivan. The results of this approach are similar to geometrical surface reconstruction.

Since the variants of the proposed 3D super-resolution approach results in similar (not identical) point clouds of minivan, we evaluate the results for a cyclist which has
a complicated surface. The segmented point cloud of the cyclist is shown in Figure 3.24a.

The geometrical and brightness based 3D super-resolution approaches are also shown in Figure 3.24b and Figure 3.24c. The results show that brightness based 3D super-resolution performs better than geometrical 3D super-resolution since it provides smoother surface of the cyclist.

### 3.5.1 Global smoothness

The locally estimated surfaces are accumulated and constitute surface of the object. However, these locally estimated surfaces may not coincide and therefore, surface
Figure 3.21: The reconstructed surface of the minivan using the geometrical variant of the proposed approach from top view; The edges and corners of the minivan are preserved and the shape of the minivan is not distorted. The color of surface is back-projected from the image.
Figure 3.22: The reconstructed surface of the minivan using the brightness based variant of the proposed approach: The brightness based reconstructed surface has less artifacts with respect to the geometrical variant of our approach. Therefore, it has the best performance over the studied approaches. The color of surface is back-projected from the image.
Figure 3.23: The reconstructed surface of the minivan using the brightness based variant of the proposed approach from top view; Similar to geometrical variant, this approach preserves the edges and corners of the minivan. The color of surface is back-projected from the image.
Figure 3.24: The proposed surface recovery approaches in sensor’s perspective (top row) and top view (bottom row) evaluated for a cyclist; (a) Original point cloud, (b) Geometry based surface reconstruction, (d) Brightness based surface reconstruction. Brightness based surface reconstruction has superior performance over geometrical surface reconstruction and provides smoother and more realistic surface. The color of surface is back-projected from the image.
may become discontinuous. The global constraint in (3.18) can be used to smooth discontinuities and generate smooth surface of the object. In Figure 3.25, the brightness based 3D super-resolution approach is applied with and without global constraint. Surface of the object is much smoother and more realistic with the global constraint. However, the global constraint is computationally expensive and may not be applied for real-time applications.

3.6 Discussion

The proposed approach for surface reconstruction provides better estimated surface of the objects. However, there are some problems using surface reconstruction approaches in practice:

The surface reconstruction approaches are computationally expensive and they require high power processors. Since the surface reconstruction for each point does not depend on other reconstructed points, it can be programmed on Graphics Processing Units (GPUs).

In addition, if an object is represented by few points, it cannot be tracked and recognized. If the object is represented by too many points, the object tracking or recognition will be slow. Therefore, an object closer to the laser scanner may not need any improvement in its resolution since it is represented with sufficient points and surface reconstruction approach may be applied for farther objects. The optimum number of points on an object is not straightforward.

However, the proposed approaches smooth the observed points of the point cloud and filter its noise, they are not able to remove the systematic errors and improve the accuracy of these points.
Figure 3.25: The brightness based 3D super resolution (a) without and (b) with global constraint. Surface of the minivan is continuous using the global constraint. The color of surface is back-projected from the image.

3.7 Conclusion

The geometrical and brightness based 3D super-resolution approaches have been introduced and qualitatively compared with previous work. The results show the proposed approaches outperform the previous approaches and they can preserve the
edges and corners of the objects. The reconstructed surface of the objects results in more robust the 3D object tracking or recognition. In the next chapter, we apply the 3D super-resolution to increase the resolution of the point cloud and improve the 3D moving object tracking approach.
Chapter 4: Moving object detection and tracking

4.1 Introduction

Autonomous driving is a complicated process that the human brain performs unconsciously. This process includes many subtasks that should be robustly accomplished to enable safe maneuvering on the roads. One of these subtasks is finding static and moving objects on the roads, also known as obstacles, and taking proper actions to prevent the risk of accident. In other words, everything which is located above the ground with a certain height can be defined as an object or an obstacle. The trajectory of the platform should be planned in such a way to avoid the accidents with static objects such as parked cars on the road. However, avoiding accidents with the moving objects on the road is much more complicated since their trajectory must be accurately predicted and the route of the platform designed to avoid any possible collision with moving objects in near future. Obviously, this prediction is only valid for few seconds and constant observation of moving objects is necessary for developing a safe maneuvering plan.

There are a number of studies that address this autonomous driving subtask. Some of these studies focus on the moving object detection in stereo images. If stereo images are taken in a period of time, the stereo matching and optical flow algorithms can be
combined to estimate the 3D motion of each pixel in time. This is an active research area [24, 49] that detects the moving objects based on the images and it is called scene flow. However, scene flow does not provide robust and accurate results that can be applied for the autonomous driving. Tracking of moving objects is also possible using the laser scanner. The point cloud can be tracked and aggregated in time [29, 28]. This approach improves the quality of the point cloud as it tracks the point cloud. The aggregated point cloud is more accurate, but also more computationally expensive. In the developed self-driving car at Stanford, it is suggested that a simple Kalman filter which is updated by the centroid of the object’s points has worked much better than more complicated algorithms [46].

4.2 Multiple sensor integration

In this section, we propose multiple sensor integration to detect and track the moving objects. The GPS/IMU navigation solution is used to estimate the pose of the platform. It is also necessary to transfer the sensors’ output collected in the local coordinate system into a global coordinate system. Moreover, if GIS maps are available, the position of the platform should be used to find the nearby road links and traffic features. Furthermore, point cloud collected from laser scanner should be refined and uninformative points such as points on trees and buildings should be removed. The nearby objects in the point cloud should be segmented and their pose between two epochs should be estimated. The estimated pose of the objects should be transferred to the global coordinate system and tracked over time. Some estimators such as the Kalman filter enables predicting the pose of the objects in the near future.
4.2.1 GPS/IMU navigation solution

Global Positioning System (GPS) receiver estimates position of the platform as long as the clear sky view is available. Inertial Measurement Unit (IMU) estimates the orientation of the platform and the translation between two IMU measurements. Therefore, combined GPS and IMU, also known as the GPS/IMU navigation solution, estimates pose of the platform. In other words, it obtains the rotation matrix and translation vector of the platform in the global coordinate system. There are a few definitions of the global coordinate system [35].

Since the imaging sensors, including cameras and laser scanners, collect information in their local coordinate system, the GPS/IMU navigation solution is necessary to convert the sensors’ output in their local coordinate system to the global coordinate system. Due to the platform’s motion, static objects are observed as moving objects in local coordinate systems. In addition, every sensor has its own coordinate system and multiple sensor integration is only possible in a unified coordinate system.

Every local coordinate system is defined by an origin and three axes. Typically, the camera coordinate system is defined so that its origin is located at the camera center with the z axis aligned with its principal axis towards the scene. The x and y axes are also defined in width and height directions of the images. Typically, the y direction is chosen in such a way that it defines a right-handed coordinate system. The laser scanner coordinate system is defined at the center of its mirrors and its z axis is aligned with the rotation axis of laser scanner. If the laser scanner rotates horizontally, the y axis is defined in vertical direction and the x axis is chosen in the way that it generates a right-handed coordinate system. The GPS/IMU navigation solution has a coordinate system located at the phase center of the GPS receiver and
its axes are aligned with the IMU axes. The alignment of IMU is in the direction of its accelerometers and gyroscopes. The GIS information are provided in the global coordinate system where its origin and axes are precisely defined by the international conventions [35].

GPS/IMU navigation solution provides the rotation matrix and translation vector between the global coordinate system and IMU sensor’s local coordinate system. Since this local coordinate system changes with motion of the platform, it is dependent on time of the IMU measurement. Let’s define the rotation matrix from the IMU local coordinate system at time $t$ to the global coordinate system, $C_{i,k}^{global}$, where subscript $i$ is the IMU local coordinate system and subscript $k$ is epoch of the IMU measurement. The translation vector from the IMU local coordinate system at time $t$ into the global coordinate system is $t_{i,k}^{global}$. Typically, the original coordinate system is written as subscript and the destination coordinate system is written as superscript.

Let’s assume that the first epoch is considered as the reference epoch. The camera rotation matrix between the camera coordinate system at epoch $k$ and the camera coordinate system at reference epoch are calculated, such that

$$C_{c,0}^{c,k} = C_{i,k}^{c,i} C_{i,0}^{global} C_{i,0}^{c,i}$$

$$= C_{i}^{c,i} C_{i,k}^{global} C_{i,0}^{c,i}$$

(4.1)

where subscript and superscript ”0” are the reference epoch. Since the platform is rigid and the sensor configuration does not change over time, $C_{c,k}^{i,k} = C_{c,0}^{i,0}$. The rotation matrix between the camera’s local coordinate system and the IMU’s local coordinate system is called the boresight, $C_{c}^{i}$. Boresight does not change with time and it can be accurately measured in the calibration process. The IMU measurements lead to
the rotation matrix between IMU sensor’s local coordinate system and the global coordinate system, $C_{i,k}^{c,global}$.

The translation vector between the camera local coordinate systems at epoch $k$ and the IMU local coordinate system at the reference epoch is calculated, such that:

$$t_{i,0}^{c,k} = C_{i,k}^{c,i} (t_{i,k}^{i,global} - t_{i,0}^{i,global}) + C_{c,0}^{c,i} t_{c,i}^c$$

(4.2)

The configuration between camera and IMU does not change and therefore, $t_{c,0}^{i,0} = t_{c}^i$. This configuration is called lever arm and it should be accurately measured in the calibration process. The GPS/IMU navigation solution at epoch $k$ provides $t_{i,k}^{i,global}$. However, the transformations are given between the IMU and camera sensors, the laser scanner’s local coordinate system can be converted into the camera’s and IMU’s local coordinate systems or map’s global coordinate systems in similar fashion.

In conclusion, the multiple sensor integration requires the GPS/IMU navigation solution to convert the sensors’ local coordinate system into the global coordinate system and vice versa. Traditionally, the IMU sensors are installed close to the center of mass of the platform and its coordinate system is assumed as the coordinate system of the platform. The GPS/IMU navigation solution is also used to find nearby traffic features. Figure 4.1 shows the trajectory of the platform estimated by the GPS/IMU navigation solution. The green lines show the road boundaries which are approximated from GIS maps. In this figure, the coordinate system is shown at the first epoch of data collection, or reference epoch.
4.2.2 GIS maps

The progress in the satellite and aerial imagery has led to accurate raster GIS maps. In these maps, information are shown by the color values of the pixels. Therefore, these maps are bulky and are not easy to process. In contrast, the vector GIS maps are compact and easy to process. Features are represented by topological data structure that enables real-time processing and queries can be more efficiently requested and answered [51]. However, the vector map generation is expensive and human intervention may be required. The OpenStreetMap (OSM) is a publicly available spatial database. Users of this database are also responsible for data collection and updating the information of this database. Since there is no quality control on
the users’ input, its accuracy and completeness are not guaranteed and the features can be a few ten meters dislocated, mislabeled, or not stored in the database.

When the position of platform is estimated by the GPS/IMU navigation solution, a circular buffer which is centered at the estimated position of the platform is applied and the features in a certain radius are requested from OSM. The road links and traffic features inside this buffer are retrieved from OSM. Since the roads are classified based on their flow and importance, we approximate their width and request a buffer centered at the road links and with the radius of the approximated road widths. These road boundaries are not accurate and should be used in a conservative way. In other words, they should be selected in the way that they do not exclude the road and the objects on the road. This process is not required to be performed every epoch and the road boundaries may be reconstructed when the platform is far from the position that it has been used for buffering.

4.2.3 Laser scanner’s point cloud

The point cloud is generated from a 360° horizontally rotating laser scanner. The laser scanner emits laser beams, it detects the reflected beams and it measures the travel time of the laser beams and converts it into the distance. In other words, the laser scanner provides a sample set from the surface of the nearby objects. These objects mostly include points of the ground, building facades, trees, traffic signs, vehicles, pedestrians, and cyclists. Among these objects, the moving objects such as vehicles, pedestrians and cyclists are important for the moving object detection and tracking purposes and the other objects should be removed. The original points of point cloud are shown by red dots in Figure 4.2.
The points in 360° laser scanner field of view may be used for the autonomous driving. However, the cameras are frontal view cameras and the points that are not located at the field of view of the camera are removed since we need the gray or color value of each point of the point cloud. The remaining points that are located in front of the platform are shown in Figure 4.3.

The points reflected from the ground should be removed from the point cloud. This points are selected if a plane is fitted to the low elevation points of the point cloud. Since these points are contaminated with outliers, Random Sample Consensus (RANSAC) is used to estimate the plane coefficients. Therefore, the points that are
Figure 4.3: The points that are not located in the field of view of the camera are removed. Therefore, every point in the remaining point cloud are assigned with a gray scale or color value.
Figure 4.4: The silver points are the ground points that are selected by fitting a plane to low elevation points. These points belong to the ground plane and should be removed.

Located at a certain distance from this plane are labeled as ground points and are removed from the point cloud. The silver points in Figure 4.4 are labeled as ground points by fitting a plane to low elevation points. These points are removed from the static and moving object points.

In addition to the ground points, some static objects such as building and trees are not located on the road and therefore, these objects are not potential threat to the platform’s safety. Therefore, these objects should be removed from the point
Figure 4.5: The silver points are the ground points that are selected by fitting a plane to low elevation points. These points are uninformative and should be removed.

cloud. However, these objects may be used for other purposes, such as navigating the platform in the GPS-denied environments. The road boundaries are used to select and remove the points of the objects that are not located on the road. Since OSM road boundaries cannot be accurately determined, some of trees and buildings may still exist in the point cloud after removing the off-road points. The off-road points that include buildings and trees are shown in Figure 4.5 in dark green.

After removing the ground and off-road points, the remaining points of the point cloud are not connected. The Euclidean distance segmentation that clusters the
Figure 4.6: The objects are segmented using Euclidean distance segmentation. Every object is shown with a unique color. Nearby points is used to segment the remaining points. Every cluster represents the points of an object. These objects are static or in motion and further processing is required to discriminate the static and moving objects. Every object is shown with a unique color in Figure 4.6. The last step to detect the moving object is performed after the remaining objects are tracked over time. After the tracking process, which is explained in the next section, the objects with insignificant estimated motion are labeled as static objects. Figure 4.7 shows the results of static and moving objects discrimination. The static objects are shown with light blue.
Figure 4.7: The static objects are discriminated and shown with the light blue. The moving objects are uniquely colored and they keep their color if they are correctly tracked. If a moving object is detected, a unique color is assigned to the new object.
4.3 Pose Estimation

In order to determine the moving objects and track them in time, their pose should be estimated in two consecutive epochs. In other words, the rotation matrix and translation vector between two point clouds of an object should be estimated. In Figure 4.8a and 4.8b, the red and blue points have been collected at time \( t - 1 \) and time \( t \), respectively. The green lines show the matching correspondence between these two point clouds. Since the two point clouds are similar, the points are mostly matched correctly. However, the point cloud can change drastically between two epochs and the matching may be erroneous between the point clouds of two epochs. The systematic errors such as motion drag can deteriorate the point clouds, too.

Figure 4.8: The point clouds in two consecutive epochs are shown in red and blue. The green lines show the points correspondence between two epochs. These corresponding points are used to estimate the pose of the object.
The most popular approach to estimate the rotation and translation between two point clouds is Iterative Closest Point (ICP). ICP assumes that the point $X_i$ at time $t$ and the point $X_j$ at time $t-1$ are correspondence if their distance is minimum. By assuming the correspondence between two epochs, rotation matrix and translation vector are estimated by minimizing the distance between these two point clouds [7], such that:

$$\min_{\mathbf{R}, \mathbf{t}} \sum w_i \| (\mathbf{R}X_i + \mathbf{t} - X_j)n_i \|$$ \hspace{1cm} (4.3)

where $X_i$ is 3D position of a point at time $t$ and $X_j$ is 3D position of its corresponding point at time $t - 1$. In addition, $\mathbf{R}$ and $\mathbf{t}$ are rotation matrix and translation vector of the point clouds between these two epochs. The normal vector at the point $X_i$ is $n_i$, $w_i$ is the weight for the point $X_i$ and $\| . \|^2$ represents the L2 norm. The assumption that the closest points are corresponding, is not correct when two point clouds are far from each other. Therefore, this is an iterative approach that in every step, it minimizes (4.3) and transforms the point cloud using rotation matrix and rotation vector. Since this approach is a local minimizer, it can converge to the local minimum and may lead to incorrect results. In addition, ICP assumes that the object is rigid and the changes in two point clouds are due to change in the object’s pose and noise. ICP cannot handle non-rigidity of the objects, since the changes are assumed to be only as the result of rotation and translation of the objects.

In addition, there are a few extensions of ICP that use other information sources of the point clouds. In generalized ICP, the Principal Component Analysis (PCA) has been used to match the points of the similar planes [61]. In other words, it is a plane to plane matching algorithm and is more robust to the mismatches. In another ICP variant, the color information of the points are also applied to improve the ICP
point matching. Therefore, the closest points that share similar color are labeled as matched points in this algorithm [42].

Furthermore, 3D salient feature points can also be used to estimate the pose of an object between two epochs. Unlike ICP, only salient points contribute in feature matching and pose estimation. However, the rigidity assumption is necessary for the feature matching, it handles the non-rigidity much better than ICP since every two salient points is compared in their local neighborhood. The disadvantage of this approach is the fact that the objects may not have many salient points and pose estimation may fail due to lack of sufficient point matches. Some of the 3D features and descriptors are listed in [50].

In addition to these approaches, the centroid of the point clouds in two consecutive epochs are estimated and applied in the Kalman filter to estimate the translation vector. The rotation matrix cannot be directly estimated in this approach and other constraints such as non-holonomic constraint is required to estimate the rotation matrix of the platform.

4.4 Point cloud enhancement

In chapter 3, we addressed the 3D super-resolution approach to increase the resolution of the sparse point clouds. This enhancement in point clouds results in more reliable tracking of the moving objects. In other words, if the point clouds are improved, the salient points will be increased, and these features can be matched in the consecutive epochs. The mismatches in feature matching adversely affect the results of the pose estimation. The robust least squares approaches such as Random Sample Consensus (RANSAC) are utilized to remove the outliers and provide robust pose
estimation. However, the success of RANSAC approach depends on the number of feature applied in feature matching. Therefore, the pose estimation results will be improved if the enhanced point clouds are used.

Figure 4.9 shows the original point cloud of the minivan is enhanced using the 3D super-resolution approach in chapter 3. This dense point cloud is more robust to mismatches of the features and results in better pose estimation.

![Figure 4.9](image)

Figure 4.9: (a) The minivan observed by laser scanner has a sparse point cloud. (b) The 3D super-resolution approach is applied to the sparse point cloud to enhance its resolution and the 3D object tracking of the minivan is performed using the dense point cloud.

### 4.5 Tracking

Among the previously mentioned pose estimation approaches, utilization of centroids has shown the best results [45]. The Kalman filter is designed to remove centroid noise of the point cloud and create smooth trajectory of the nearby objects. The moving objects have not significant vertical movement. Therefore, the objects’
dynamics are modeled in 2D, using the Kinematic equations, such that:

\[
E_k = E_{k-1} + V_{E,k-1} \Delta t + \frac{1}{2} a_{E,k-1} \Delta t^2 \\
N_k = N_{k-1} + V_{N,k-1} \Delta t + \frac{1}{2} a_{N,k-1} \Delta t^2
\]

(4.4)

\[
V_{E,k} = V_{E,k-1} \Delta t + a_{E,k-1} \Delta t \\
V_{N,k} = V_{N,k-1} \Delta t + a_{N,k-1} \Delta t
\]

(4.5)

where \( E_k \) and \( N_k \) are easting and northing at epoch \( k \), \( V_{E,k} \) and \( V_{N,k} \) are the components of velocity in east and north directions at epoch \( k \) and \( a_{E,k} \) and \( a_{N,k} \) are the acceleration components of the object in east and north directions at epoch \( k \). \( \Delta t \) is the time interval, \( \Delta t = t_k - t_{k-1} \). In the implemented simplistic Kalman filter, state vector, \( \mathbf{x} \), is designed, such that:

\[
\mathbf{x} = \begin{bmatrix} E & N & V_E & V_N \end{bmatrix}^T
\]

(4.6)

The dynamic model of the Kalman filter is calculated, such that:

\[
\mathbf{x}_k = \Phi_{k-1} \mathbf{x}_{k-1} + \mathbf{G}_k \mu_k
\]

(4.7)

where \( \Phi \) is the transition matrix, \( \mathbf{G} \) is the matrix that relates the dynamic model noise to the state vector and \( \mu \) is the dynamic model noise. The acceleration is modeled as white noise, \( w_a \), since the probabilities of acceleration and deceleration are equal. The dynamic model is calculated by replacing (4.4) and (4.5) into (4.7), such that:

\[
\begin{bmatrix} E \\ N \\ V_E \\ V_N \end{bmatrix}_k = \begin{bmatrix} 1 & 0 & \Delta t & 0 \\ 0 & 1 & 0 & \Delta t \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} E \\ N \\ V_E \\ V_N \end{bmatrix}_{k-1} + w_a \begin{bmatrix} \frac{\Delta t^2}{2} \\ \frac{\Delta t^2}{2} \\ \Delta t \\ \Delta t \end{bmatrix}_{k-1}
\]

(4.8)

Furthermore, the observation model of the Kalman filter is calculated by

\[
\mathbf{z}_k = \mathbf{H}_k \mathbf{x}_k + \mathbf{v}_k
\]

(4.9)
where $z$ is the observation vector, $H$ is the design matrix, and $v$ is the observation noise. The observation model of the Kalman filter is calculated by

$$
\begin{bmatrix}
    E \\
    N \\
    E \\
    N
\end{bmatrix}^T_k = \begin{bmatrix}
    1 & 0 & 0 & 0 \\
    0 & 1 & 0 & 0 \\
    E_k \\
    N_k
\end{bmatrix} + \begin{bmatrix}
    \epsilon_E \\
    \epsilon_N
\end{bmatrix}_k
$$

where $c_E$ and $c_N$ are easting and northing components of centroid of the point cloud and $\epsilon_E$ and $\epsilon_N$ are the observation noise in easting and northing. The rotation angles cannot be estimated using centroid of the point clouds. However, non-holonomic constraints can be applied to estimate azimuth of the moving object. The non-holonomic constraints dictate that the moving objects do not have 6 degrees of freedom (3 rotation angles and 3 translation components). For instance, the vertical movement and roll rotation of the vehicle are not significant. Using the non-holonomic constraints, azimuth, $Az$, of the moving objects is estimated by

$$
Az = \tan^{-1}\left( \frac{V_E}{V_N} \right)
$$

In other words, the motion direction of the platform coincides with its front direction and the azimuth of the motion vector the same as the object’s azimuth.

### 4.6 Experiments

In order to evaluate the proposed moving object tracking approach, the Karlsruhe Institute of Technology (KITTI) benchmark [20] has been used. The KITTI benchmark is publicly available and specialized for applications of computer vision algorithms in autonomous driving. It has been applied to evaluate tracking results of the moving objects. The datasets of this benchmark are the collections of several sensors including the GPS/IMU navigation solution, stereo monochromatic and color
cameras, and laser scanner. The GPS/IMU navigation solution has been provided by OXTS RT 3003 navigation system. Dual frequency carrier phase GPS was used to resolve the carrier phase ambiguity. Therefore, the accuracy of the positioning is 2 centimeters and the orientation accuracy provided by the IMU is $0.1^\circ$ [20].

The images were taken from two PointGray Flea2 grayscale and two PointGray Flea2 color cameras in 0.1 second rate. The lower and upper parts of the images were removed because the boot and sky had covered the lower and upper parts of images and the images have $1276 \times 333$ resolution after cropping the lower and upper parts. The lens focal length is 4 millimeters, with the field of view about $90^\circ$ and $4.65\mu m$ pixel size.

The point cloud was collected by the Velodyne HDL-64E laser scanner. This laser scanner has 64 beams with a point cloud sampling frequency of 10 Hertz. Its range accuracy is better than 2 centimeters and its maximum range is about 120 meters. Its horizontal and vertical fields of view are $360^\circ$ and $26.8^\circ$ with horizontal and vertical angular resolution of $0.08^\circ$ and $0.4^\circ$, respectively.

The sensors’ configuration was determined by a calibration procedure with known lever arms and boresights. The sensors were synchronized using a hardware synchronizer with a few milliseconds accuracy. Figure 4.10 demonstrates the sensors configuration on the platform used to collect KITTI datasets.

4.7 Results

The results of tracking are shown in Figures 4.11-4.18. The segmented objects are assumed to be static at the first epoch and the Kalman filter is initialized with the zero velocity. The first epoch of the proposed approach is shown in Figure 4.11
Figure 4.10: The sensors configuration; GPS/IMU navigation solution, cameras and laser scanner are mounted on the platform [20].

and static objects are shown in light blue. The moving objects, cyclist and car, as detected after 0.4 seconds are shown in Figure 4.12. The results of the proposed approach for the moving objects are shown in Figure 4.13 and Figure 4.14 after 3 and 6 seconds. Since the colors of these objects have not changed in Figures 4.12-4.14, these objects are correctly tracked over time.

A static object, a parked car, is incorrectly labeled as the moving object in Figure 4.14. Since the point clouds of the static object in two epochs have been significantly changed, some static objects were labeled as moving objects. The objects that are incorrectly labeled as the moving objects are false positives in the tracking. However, the static object has been correctly labeled after a few epochs. The vehicle which has been tracked in Figure 4.14, sped up and went farther than 50 meters. Since the objects farther than 50 meters are not objects of interest, the vehicle was not tracked further in Figure 4.15. However, the cyclist was tracked throughout the whole dataset.
Figure 4.11: The moving object tracking from the beginning. All objects are considered static (light blue) in the first epoch.

Figure 4.12: The moving object tracking after 0.4 second. The moving objects (brown is minivan and green is cyclist) are detected as early as 0.4 second.
Figure 4.13: The moving object tracking after 3 seconds. The cyclist and van are correctly tracked since their color does not change over time.

Figure 4.14: The moving object tracking after 6 seconds. The cyclist and minivan are correctly detect, but a parked car is incorrectly labeled as moving object.
The tracking results are shown in Figures 4.16, 4.17, and 4.18 after 12, 14 and 15 seconds, respectively.

![Diagram](image)

**Figure 4.15:** The moving object tracking after 9 seconds. The minivan is far from the platform and the proposed approach stops to track it further.

The tracked cyclist and vehicle trajectories are shown in Figure 4.19. They were correctly tracked over time with the color unchanged. However, the vehicle’s tracking was shorter in Figure 4.19a since it traveled more than the violation criterion of 50 meters and we stopped to track it further.

Tracking results of the cyclist are compared to the ground truth and the tracking error is calculated and demonstrated in Figure 4.20. The Kalman filter converged with the tracking error significantly reduced in few epochs. Since the tracking is estimated based on pose estimation and it is an incremental measurement, the error is accumulated over time.
Figure 4.16: The moving object tracking after 12 seconds.

Figure 4.17: The moving object tracking after 14 seconds. The cyclist is correctly tracked and the parked cars are correctly labeled as static objects.
In another KITTI dataset with no moving objects until the vehicle stops at a crowded intersection. This dataset is more challenging since the moving objects has been observed for a few seconds for the proposed approach to detect them quickly. The results over 9 seconds of the moving object tracking are shown in Figure 4.21. The objects were considered static at the beginning in Figures 4.21a and there were no moving objects for 9 seconds. The static objects that were correctly detected as static objects are true negatives. However, many of the ground points could not be removed using plane fitting in 4.21d and these points were incorrectly labeled as moving objects. There also was the median in the road that has introduced the systematic error into the ground plane.

The vehicle approached the crowded intersection and stopped behind the red light. The results of the crowded intersection are shown in Figure 4.22 and Figure 4.23. A
Figure 4.19: (a) The cyclist and (b) minivan has been tracked and their point clouds are shown. The results show that the proposed approach can correctly detect and track the moving objects.

few vehicles were correctly tracked. These vehicles were labeled after few epochs that they have been observed (about 0.4 seconds).

4.8 Discussion

The proposed algorithm is useful for driving in the highway when the nearby objects are moving with similar velocity and they can be observed for longer time. It has also been tested for an intersection, where the moving objects are visible for shorter period of time. However, this approach may not be able to detect abrupt object motion such as quick appearance of a pedestrian behind a car. Therefore, this algorithm may be backed up by other approaches and some image processing algorithms.
Figure 4.20: The tracking error; the Kalman filter has been converged and the tracking error is significantly dropped after few epochs. The tracking error grows in time since the tracking error is the accumulation of inaccuracy in the pose estimation.

In addition, most of moving objects can be detected in less than half of a second. However, the Kalman filter converges in a few seconds and therefore, an accurate motion estimation of the moving objects is provided after a few seconds. This latency is not acceptable and the Kalman filter convergence time must be shortened.

Since we remove the points on the ground, buildings, and trees, the remaining points of the point cloud is much less than the original point cloud. Therefore, the process of remaining points is fast and the proposed approach is applicable for real-time applications.
Figure 4.21: The moving object detection and tracking results for 9 seconds: (a) beginning; (b) after 3 seconds; (c) after 6 seconds; (d) after 9 seconds. There is no moving object during 9 seconds, but some ground points are not removed and incorrectly labeled as moving object in (d).
Figure 4.22: The moving object tracking after 12 seconds. Two objects are labeled as moving objects, but one of them is correctly labeled.

Figure 4.23: The moving object tracking after 13 seconds. Two objects are correctly labeled as moving objects.
Chapter 5: Traffic Light Detection and Mapping

5.1 Introduction

The traffic light and sign detection are avoidable tasks in the autonomous driving since these traffic features dictate the right of way. The traffic light detection is a critical task and failure in detecting the traffic lights may cause accident with other vehicles, cyclists, or pedestrians. There are scientific endeavors to provide a reliable solution for the traffic light detection. However, this task has still remained a challenge. This is due to the fact that there are many variations of traffic lights with different colors, shapes, and installation standards.

The traffic lights have red, amber, and green lenses and therefore, the color of these lenses are applied to detect the traffic lights. However, the traffic light lenses are not illuminated uniformly and the traffic light lenses have different colors in center and border of the lenses. Generally speaking, the center of the traffic light lens is brighter than its borders. In addition, there are many objects in background, such as trees, with similar color to the traffic light lenses. Therefore, the color information are not sufficient for the traffic light detection. The traffic lights have circular lenses, but the size of these lenses may change from one state to another. The height of the traffic lights follows the traffic light installation standards. However, traffic lights have
different height for the ones on the pole and the ones suspended on the road. Some of the standards may not be also obeyed and some traffic lights may be installed in a height which are not specified in the installation standards.

In addition, the images may suffer from error sources such as image blurring and adverse illumination. Since the platform is in motion, images are blurry and the image blurriness may severely distort the color and shape of the traffic lights. The image may also be overexposed or underexposed leading to failure in the traffic light detection.

The color has been extensively applied to detect the traffic lights [36]. The Red, Green, and Blue (RGB) color space is not suitable for the traffic light detection since these channels are not independent and this color space changes in different illuminations. Researchers have studied several color space, such as, normalized RGB [14, 15, 16, 52, 53], Hue, Saturation, and Value (HSV) [37, 67], YCbCr [8], YUV [62] and the CIELab color space [38, 65]. Multiple exposures may be applied to improve the color of the traffic lights in different illuminations [34].

The lenses of the traffic lights are circular. If a traffic light is located in front of the camera, the traffic light will be a circle in the image space. Some researchers exploited the circle based Hough transform to detect the traffic lights [9, 32]. Since the Hough transform is computationally expensive and is not suitable for real-time applications, fast radial symmetry transform (FRST) has been used to detect the circular shape of the traffic lights [65].

Researchers have also used the fixture of the traffic lights to detect them. The fixture of the traffic lights does not have geometric shape and more complicated approaches are applied to detect the traffic light fixtures. Template matching has
been applied for the traffic light fixture detection [11, 12, 70, 73]. The classifiers such as Support Vector Machine (SVM) have been also utilized to detect traffic light fixtures [22, 38, 41].

Some of GIS maps have traffic features including traffic lights. If the position of the traffic lights is projected to the image space, it provides the location of the traffic lights in the image space. In order to accurately project the traffic lights to the image space, the traffic lights must be accurately positioned in the geospatial database. In addition, the camera calibration parameters must be accurately known and the navigation solution must be accurately observed. Some researchers have addressed the use of accurate GIS maps, such as Google Maps, for the traffic light detection [5, 18, 38, 46]. However, Google Maps are not publicly available.

In addition to traffic light detection, some studies investigated the traffic light mapping too [13, 18, 36]. The autonomous vehicle should estimate the position of traffic lights with respect to the camera to stop behind the traffic lights in an appropriate distance. Furthermore, the autonomous vehicle should select the traffic light of its lane, if multiple traffic lights are detected.

5.1.1 Bayes’ theorem

We develop a statistical framework based on the Bayes’ theorem. Let’s assume there are two events, $A$ and $B$, and the probability of these events are $P(A)$ and $P(B)$. Knowledge of the event $B$ can change $P(A)$, if two events $A$ and $B$ are not independent. The probability of the event $A$ based on knowledge of the event $B$ is computed by:

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

(5.1)
where \( P(A|B) \) is the conditional probability of the event \( A \) given the event \( B \).

The joint probability of two events \( A \) and \( B \), \( P(A, B) \) is the probability that these two events occur. If two events \( A \) and \( B \) are the independent events, \( P(A, B) = P(A)P(B) \). Therefore, if two events are independent, knowledge of the event \( B \) does not change the probability of the event \( A \), and therefore, \( P(A|B) = P(A) \).

5.2 Methodology

The traffic lights and their signal status (i.e. red, amber, or green) have to be observed by the color cameras. Other sensors such as the GPS/IMU navigation solution, laser scanner, and GIS maps may be used to improve the traffic light detection, but are not essential for this task. We explain the use of the GPS/IMU navigation solution and GIS maps to improve the traffic light detection.

5.2.1 Binary labeling

Let’s assume the image \( I_t \) is taken in the current time, \( t \), and the pixel \( x_i \) belongs to this image. We define a binary function that returns one, if \( x_i \) belongs to a traffic light and it returns zero, otherwise. This function, \( \omega \), is defined by:

\[
\omega_t(x_i) = \begin{cases} 
1 & \text{If } x_i \text{ belongs to traffic light} \\
0 & \text{Otherwise}, 
\end{cases}
\]

(5.2)

where \( \omega_t(x_i) \) is the label of the pixel \( x_i \) in the image \( I_t \).

In order to label the pixels of the image, we utilize a Bayesian statistical framework. A pixel is labeled as the pixel of a traffic light, if the probability of this pixel belonging to the traffic light is larger than a threshold, such that:

\[
\omega_t(x_i) = \begin{cases} 
1 & P(\omega(x_i)) > Th \\
0 & \text{Otherwise}. 
\end{cases}
\]

(5.3)
where $Th$ is the pre-selected threshold and the choice of this threshold is empirically determined using precision/recall curve.

Suppose number of the pixels in an image is $n$ and all pixels of this image are shown by $x_{1:n}$. Let’s assume there are some observations extracted from this image at time $t$, $Z_t(x_{1:n})$. Since this probabilistic framework is designed for real-time applications, the future observations, such as $Z_{t+1}(x_{1:n})$, are not available. Therefore, the best estimate of the traffic lights is calculated when all observations up to this time, $Z_{1:t}(x_{1:n})$, are used. The probability of the labels for the pixels in this image is estimated using Bayes’ theorem by:

$$P(\omega_t(x_{1:n})|Z_{1:t}(x_{1:n})) = \frac{P(Z_t(x_{1:n})|\omega_t(x_{1:n}))P(\omega_t(x_{1:n})|Z_{1:t-1}(x_{1:n}))}{\int P(Z_t(x_{1:n})|\omega_t(x_{1:n}))P(\omega_t(x_{1:n})|Z_{1:t-1}(x_{1:n}))d\omega_t(x_{1:n})}. \tag{5.4}$$

The observation vector at the current time $Z_t(x_{1:n})$ is independent to the previous observation vectors, $Z_{1:t-1}(x_{1:n})$ given the labels of the pixels $\omega_t(x_{1:n})$. Therefore, $P(Z_t(x_{1:n})|Z_{t-1}(x_{1:n}), \omega_t(x_{1:n})) = P(Z_t(x_{1:n})|\omega_t(x_{1:n})).$

$P(Z_t|\omega_t(x_{1:n}))$, is called likelihood term and it relates the observation vector and the label estimation at the current time. $P(\omega_t(x_{1:n})|Z_{1:t-1})$ is called prior term since it relates the label estimation to the previous observations. The denominator is also applied to normalize the probability and restrict its range to $[0 1]$.

The likelihood term of (5.4) is estimated, such that:

$$P(Z_t(x_{1:n})|\omega_t(x_{1:n})) = \prod_{i=1}^{n} P(Z_t(x_i)|\omega_t(x_i))P(\omega_t(x_i)|\omega_t(x_{1:n\neq i})). \tag{5.5}$$

Equation (5.5) indicates that the observation vector of the pixel $x_i$ is independent to the observation vector of the pixel $x_j$ given the label of the pixel $x_i$, and therefore, $P(Z_t(x_i)|Z_t(x_j), \omega_t(x_j)) = P(Z_t(x_i)|\omega_t(x_i))$. $P(\omega_t(x_i)|\omega_t(x_{1:n\neq i}))$ represents the
consistency between pixels within the image. In other words, it indicates that a probability of a pixel belonging to the traffic light depends on the probability of its neighbors belonging to the traffic light.

The prior term, \( P(\omega_t(x_{1:n})|Z_{1:t-1}) \), is calculated by marginalization of the term \( P(\omega_{1:t}(x_{1:n})|Z_{1:t-1}) \) over \( \omega_{1:t-1}(x_{1:n}) \). By Markovian assumption, the labels at the current time only depend on the previous time and therefore, \( \omega_t(x_{1:n}) \) and \( \omega_{1:t-2}(x_{1:n}) \) are independent given \( \omega_{t-1}(x_{1:n}) \). Moreover, \( P(\omega_t(x_{1:n})|Z_{1:t-1}) \) is estimated by:

\[
P(\omega_t(x_{1:n})|Z_{1:t-1}) = \int P(\omega_t(x_{1:n})|\omega_{t-1}(x_{1:n}))P(\omega_{t-1}(x_{1:n})|Z_{1:t-1})d\omega_{t-1}(x_{1:n}) =
\]

\[
P(\omega_t(x_{1:n})|\omega_{t-1}(x_{1:n}) = 1)P(\omega_{t-1}(x_{1:n}) = 1|Z_{1:t-1}) +
\]

\[
P(\omega_t(x_{1:n})|\omega_{t-1}(x_{1:n}) = 0)P(\omega_{t-1}(x_{1:n}) = 0|Z_{1:t-1}).
\]

Finally, the denominator of (5.4) is a normalization parameter. The label of each pixel is binary and the denominator of (5.4) is simplified by:

\[
\int P(Z_t|\omega_t(x_{1:n}))P(\omega_t(x_{1:n})|Z_{1:t-1})d\omega_t(x_{1:n}) =
\]

\[
P(Z_t|\omega_t(x_{1:n}) = 1)P(\omega_t(x_{1:n}) = 1|Z_{1:t-1}) +
\]

\[
P(Z_t|\omega_t(x_{1:n}) = 0)P(\omega_t(x_{1:n}) = 0|Z_{1:t-1}).
\]

### 5.2.2 Spatial coherency

Let’s assume that the pixel \( x_j \) belongs to the neighboring pixels of the pixel \( x_i \).

If the pixel \( x_j \) belongs to a traffic light, and therefore, \( \omega_t(x_j) = 1 \), is most likely that \( \omega_t(x_i) = 1 \). Therefore, the probability these pixels sharing the same label is:

\[
P(\omega_t(x_i),\omega_t(x_j)) = \frac{1}{\eta} \exp(-\psi(\omega_t(x_i),\omega_t(x_j))),
\]
where $\eta$ is the normalization constant enforcing the probability to be in the range of $[0 \ 1]$. The function $\psi$ relates the neighboring pixels to their joint probability and represents the cost function of connecting these pixels.

Considering all neighboring pixels of pixel $x_i$, $P(\omega_t(x_{1:n}))$ is calculated by:

$$P(\omega_t(x_{1:n})) = \frac{1}{\eta} \exp(-\sum_{j \in N(i)} \psi(\omega_t(x_i), \omega_t(x_j))), \quad (5.9)$$

where $N(i)$ is the neighboring pixel set of pixel $x_i$.

The choice of the function $\psi$ is arbitrary, but one of the frequently used cost functions is the Gaussian Markov Random Field (GMRF) [59]. This function relates the color variations of two pixels with their labels using a Gaussian function by:

$$P(\omega_t(x_{1:n})) = \lambda (1 - \delta(\omega_t(x_i) - \omega_t(x_j))) \exp(-\frac{||I(x_i) - I(x_j)||^2}{2\beta}), \quad (5.10)$$

where $\lambda (1 - \delta(\omega_t(x_i) - \omega_t(x_j)))$ is a constant value enforcing the probability to be in the range of $[0 \ 1]$. In addition, $\beta$ is the average of color variations and it is estimated by:

$$\beta = \frac{1}{n} \sum_{i=1}^{n} \sum_{j \in N(x_i)} ||I(x_i) - I(x_j)||^2. \quad (5.11)$$

Figure 5.1 shows pixels of the image in the neighborhood of pixel $x_i$. The pixels belonging to a traffic light are represented by white circles and the background pixels by gray circles. The cost function $\psi$ results in higher joint probability for the pixels with similar color values and lower joint probability for the pixels with different color values.

### 5.2.3 Temporal coherency

The prior term in (5.6), $P(\omega_t(x_{1:n})|Z_{1:t-1})$, predicts the labels in the current time using the posterior estimation of the labels in the previous time, $P(\omega_{t-1}(x_{1:n})|Z_{1:t-1})$. 
The transition term, $P(\omega_t(x_{1:n})|\omega_{t-1}(x_{1:n}))$, relates the current and previous labels. Assume the relationship between the labels in current time and previous time is linear and given by:

$$\omega_t(x_{1:n}) = \mu_p + \Psi \omega_{t-1}(x_{1:n}) + \epsilon_p,$$

where $\mu_p$ is the constant change of labels, $\Psi$ is the coefficient of the linear function between the current time and the previous time and $\epsilon_p$ is noise in the transition term.

Assuming noise in the transition term is normally distributed, the transition term is estimated by:

$$P(\omega_t(x_{1:n})|\omega_{t-1}(x_{1:n})) = \mathcal{N}(\mu_p + \Psi \omega_{t-1}(x_{1:n}), \sigma_p^2)$$

where $\mathcal{N}(a, b)$ is the normal distribution with the mean $a$ and the variance $b$. 

Figure 5.1: The spatial coherency between pixels of a traffic light; If the pixels of a traffic lights are white circles, $\omega_t(x_{i-1,i,i+1})$, and the pixels of the background are gray circles, $\omega_t(x_{j-1,j,j+1})$, the labels of white circles should be similar. In other words, the spatial coherency term encourages that pixels with similar color have similar labels.
If the platform is stationary, the images in the current time and previous time are identical and the transition is normal distribution parameters are \( \mu_p = 0, \Psi = 1 \). Therefore, the changes in the labels may occur due to noise of the transition.

When the platform is in motion, dynamics of the platform affect \( \mu \) and \( \psi \). Since the traffic light is planar, the homography transformation, \( H \), is applied to relate the traffic light of current time and previous time. In addition, conic section geometry is utilized since the traffic light is circular.

If a point of the traffic light, \( X_i \), is located in 3D object coordinate system and its projection at the current and previous times are \( x_i \) and \( x'_i \), the homography transformation is applied to project \( X_i \) to \( x'_i \) and \( x_i \) by:

\[
\begin{align*}
x'_i &= H_{t-1}X_i \\
x_i &= H_tX_i,
\end{align*}
\]

where \( H_{t-1} \) and \( H_t \) are the homography transformations in the previous and current times. The correspondence of the pixel \( x_i \) in the previous time is estimated by:

\[
x_i = H_tH_{t-1}^{-1}x'_i.
\]  

(5.15)

Since the traffic light is stationary, the label of two corresponding pixels in the consecutive times should be the same. Therefore, labels in two consecutive times are:

\[
P(\omega_t(x_i)|\omega_{t-1}(x'_i)) = \mathcal{N}_\omega(H_tH_{t-1}^{-1}\omega_{t-1}(x'_i), \sigma^2_p).
\]  

(5.16)

Figure 5.2 illustrates that the label of the pixel \( x_i \) at time \( t \) depends on the label of pixel \( x_i \) at time \( t-1 \) and the observation of pixel \( x_i \) at time \( t \).

In the beginning, prior label estimation, \( \omega_{t-1}(x'_i) \), does not exist. Therefore, the labels are modeled as binomial distribution, such that:

\[
P(\omega_1(x_i) = 0) = 1 - \kappa
\]

\[
P(\omega_1(x_i) = 1) = \kappa.
\]  

(5.17)
Figure 5.2: If the pixel \( x_i \) is labeled as traffic light in the previous time, it is most likely to be labeled as traffic light in the next time, \( P(\omega_t(x_i) = 1|\omega_{t-1}(x_i) = 1) \gg 0 \). Therefore, the label of pixel \( x_i \) in the current time, \( \omega_t(x_i) \), depends on the label of this pixel in the previous time, \( \omega_{t-1}(x_i) \), and the observation in the current time, \( Z_t(x_i) \).

**Homography Estimation**

Since the lenses of the traffic lights are circular and they lie in a plane, they are conic sections and follow conic section geometry. A conic section can be represented by a \( 3 \times 3 \) symmetric matrix. If the radius of the traffic light lens is \( r \), the circular shape of the traffic light is represented by conic section as:

\[
C = \begin{bmatrix}
1 & \frac{1}{r^2} & 0 \\
\frac{1}{r^2} & 0 & 0 \\
0 & 0 & -1
\end{bmatrix}.
\]  

The projection of the traffic light conic section, \( C \), to the image space is an ellipse which is still a conic section, \( C' \). Since the traffic light is planar, the relationship between the traffic light lens and its image is the homography transformation. Therefore, the homography transformation at time \( t \), \( H_t \), is used to map \( C \) into \( C' \):

\[
C'_t = kH_t^{-T}CH_t^{-1}.
\]  

103
Equation (5.19) estimates the homography transformation up to the scale, $k$. By inverting (5.19), the conic section in the image space is converted to the conic section in the traffic light coordinate system by:

$$ kC = H_t^T C'_t H_t. $$  \hfill (5.20)

The homography transformation consists of rotation matrix, $R$, and translation vector, $t$, between the traffic light coordinate system and the camera coordinate system and the camera calibration matrix, $K$. Therefore, the homography transformation at time $t$, $H_t$, is decomposed to:

$$ H_t = K[r_1 r_2 t_c] = KR_tT_t, $$  \hfill (5.21)

where $r_1$ and $r_2$ are first and second column vector of rotation matrix and $T = \begin{bmatrix} 1 & 0 & -t_1 \\ 0 & 1 & -t_2 \\ 0 & 0 & -t_3 \end{bmatrix}$. $t_o = [t_1 \ t_2 \ t_3]^T$ is the translation vector in the traffic light (object) coordinate system, $t_c$ is the translation vector in the camera coordinate system and these translation vectors are related using $t_c = -R t_o$.

Replacing (5.21) into (5.20), the homography transformation is estimated by:

$$ kC = T_t^T R_t^TK^T C'_t KR_t T_t. $$  \hfill (5.22)

Moving $T_t$ to the left side of (5.22) leads to:

$$ kT_t^{-T} CT_t^{-1} = R_t^T K^T C'_t K R_t. $$  \hfill (5.23)

If two sides of (5.23) are equal, their eigenvalues are equal too. If $\lambda_1, \lambda_2,$ and $\lambda_3$ are the eigenvalues of $K^T C'_t K$, it has been shown that [48]:

$$ \lambda_1 \lambda_2 \lambda_3 = -\frac{k^3}{t_3^2 r^4} $$

$$ \lambda_1 \lambda_2 + \lambda_1 \lambda_3 + \lambda_2 \lambda_3 = k^2 \frac{d^2 - 2r^2}{t_3^2 r^4} $$  \hfill (5.24)

$$ \lambda_1 + \lambda_2 + \lambda_3 = k \frac{d^2 + t_3^2 - r^2}{t_3^2 r^2}. $$
Equation (5.24) can be utilized to estimate the rotation matrix and translation vector between the traffic light coordinate system and the camera coordinate system [48], such that:

\[ k = \lambda_1 r^2 \]
\[ t_3^2 = -\frac{\lambda_1^2 r^2}{\lambda_2 \lambda_3} \]
\[ d^2 = 2r^2 - \frac{(\lambda_1 \lambda_2 + \lambda_1 \lambda_3 + \lambda_2 \lambda_3)r^2}{\lambda_2 \lambda_3} \]

where \( r \) is the radius of the traffic light lens and \( d^2 = t_1^2 + t_2^2 + t_3^2 \) is the distance between the origins of the camera and traffic light coordinate systems. When \( t_3 \) and \( d \) are calculated, the choice of \( t_1 \) and \( t_2 \) is not important since the traffic light lens is a circle. Every choice \( t_1 \) and \( t_2 \) is valid as long as it satisfies \( d^2 = t_1^2 + t_2^2 + t_3^2 \). Therefore, \( \mathbf{t}_o = [t_1 \ t_2 \ t_3]^\top \) is reconstructed from (5.25).

In addition, decomposing both sides of (5.19) into their eigenmatrices and eigenvalues using singular value decomposition gives

\[ \mathbf{K}^\top \mathbf{C}_t \mathbf{K} = \mathbf{UDU}^\top \] and
\[ \mathbf{T}_t^{-\top} \mathbf{C}_t \mathbf{T}_t^{-1} = \mathbf{VDV}^\top, \]

where \( \mathbf{U} \) and \( \mathbf{V} \) are eigenmatrices and \( \mathbf{D} \) is a diagonal matrix containing the eigenvalues, \( \lambda_1, \lambda_2, \) and \( \lambda_3 \). Equation (5.19) is applied to estimate the rotation matrix, \( \mathbf{R}_t \), by:

\[ \mathbf{UDU}^\top = \mathbf{R}_t^\top \mathbf{VDV}^\top \mathbf{R}_t, \]  
(5.26)

and rotation matrix is calculated by [48]:

\[ \mathbf{R}_t = \mathbf{UWV}^{-1}, \]  
(5.27)

where \( \mathbf{W} = \pm \mathbf{I}_{3\times3} \) and \( \mathbf{I} \) is the identity matrix. The translation vector in the camera coordinate system is computed by \( \mathbf{t}_c = -\mathbf{R}_t \mathbf{t}_o \). After rotation matrix and translation vector, \( \mathbf{R}_t \) and \( \mathbf{t}_c \), are estimated, the homography transformation is calculated in (5.21).
5.2.4 Observation vector

The observation vector includes different cues extracted from the images. These cues indicate whether a pixel belongs to the traffic light. For instance, if the color value of pixel \( x_i \) is close to green, it may belong to a green traffic light lens. If the color value of this pixel is close to black or blue, it is most likely this pixel belongs to the background. Similarly, if the pixel does not belong to an ellipse-shaped object, it is most likely the pixel belongs to the background. If the pixel back-projected into real world and it does not follow the traffic light installation standards, it does not belong to a traffic light. In addition to the traffic light lens, the fixture of the traffic light should be detected. If the traffic light fixture is detected, the pixel may belong to the traffic light. Moreover, if the traffic lights are stored in the GIS maps and we project these traffic lights to the image space, the probability of pixel \( x_i \) belonging to the traffic lights is higher if its location is close to the projected traffic lights.

Let’s define the observation vector, \( Z_t(x_i) = \{z_c, z_s, z_h, z_l, z_g\} \), where the observation vector includes the color \((z_c)\), shape \((z_s)\), height \((z_h)\), lens activation pattern \((z_l)\) and GIS \((z_g)\) cues of the traffic lights, respectively. The observation vector is constructed from its components by:

\[
P(Z_t(x_i) | \omega_t(x_i)) =
\]

\[
P(z_l | z_h, \omega_t(x_i))P(z_h | z_s, \omega_t(x_i))P(z_s | z_c, \omega_t(x_i))P(z_c | \omega_t(x_i))P(z_g | \omega_t(x_i)).
\]  

(5.28)

Shape of the traffic lights is observed based on the connected pixels with similar color and therefore, the shape cue depends on the color cue. Height of the traffic lights are also calculated based on shape of the traffic lights and therefore, the height cue depends on the shape cue. The lens activation pattern also depends on the
height of the traffic lights. The color and GIS cues do not depend on other cues. All observations depend on the labels of the pixels. Figure 5.3 shows these dependencies among observations and label of pixel $x_i$.

![Diagram showing dependencies among observations and label of pixel $x_i$]

Figure 5.3: The lens activation pattern, $z_c$, depends on the height cue, $z_h$, the height cue depends on the shape cue, $z_s$, the shape cue depends on the color cue, $z_c$, the GIS cue, $z_g$, does not depend on other cues, and all observations depend on the label of the pixel, $\omega_t(x_i)$.

Color

The color can be applied to discriminate the pixels of the traffic lights from the background. However, the color may change because of the different illumination variations and camera responses. We use the Hue, Saturation, Value (HSV) color space since it has better performance in different illuminations. Therefore, the color is represented by the hue component of the HSV color camera.

Red, amber, and green signals of the traffic lights are modeled by normal distributions. Figure 5.4 shows the normal distribution for these red, amber, and green...
colors. Therefore, the color of the traffic lights is represented by adding these three distributions and it is modeled by a mixture of Gaussians. Let’s assume there is a hidden value, it has three values 1 for red, 2 for amber, and 3 for green. The normal distributions of red, amber and green have the mean $\mu_1$, $\mu_2$, $\mu_3$ and variance $\sigma_1^2$, $\sigma_2^2$, $\sigma_3^2$, respectively. Therefore, the mixture of Gaussian is modeled the color of the traffic light by:

$$ P(z_c|\omega_t(x_i) = 1) = \sum_{k=1}^{3} w_k \frac{1}{\sqrt{2\pi\sigma_k^2}} \exp\left(-\frac{(h-\mu_k)^2}{2\sigma_k^2}\right), \quad (5.29) $$

where $w_k$ is the weight of each color and $w_1 + w_2 + w_3 = 1$. The hue component of pixel $x_i$, is represented by $h$ in (5.29).

The color of the background should be modeled too. Since the pixel belonging to the background can have any color value, it may be modeled as a uniform distribution in the range of [0 359] by $P(z_c|\omega_t(x_i) = 0) = \frac{1}{360}$. However, the background is not uniformly distributed. Sky is a significant part of images, asphalt of the road has its own color characteristics, and therefore, the uniform distribution of the background color is not a realistic model. Therefore, we use a normal distribution with the mean $\mu_0$ and the variance $\sigma_0^2$ to model the color of the background. Therefore, the probability of the color based on different label values are:

$$ P(z_c|\omega_t(x_i) = 1) = \sum_{k=1}^{3} w_k \frac{1}{\sqrt{2\pi\sigma_k^2}} \exp\left(-\frac{(h-\mu_k)^2}{2\sigma_k^2}\right) $$

$$ P(z_c|\omega_t(x_i) = 0) = \frac{1}{\sqrt{2\pi\sigma_0^2}} \exp\left(-\frac{(h-\mu_0)^2}{2\sigma_0^2}\right). \quad (5.30) $$
Figure 5.4: The color of the traffic light is represented in HSV color space and the color of each lens is modeled as Gaussian. Since HSV is circular a color space in hue component, it has the range of [0 360).

Shape

The traffic lights are circular in real world and there will be ellipses when they are projected to the image space. Let’s assume that we have segmented similar pixels and fitted an ellipse to these pixels. This ellipse is a conic section and it can be represented by a $3 \times 3$ symmetric matrix, $C'$. If pixel $x_i$ belongs to the traffic light, it should be located inside the ellipse, $x_i^T C' x_i \leq 0$. where $x_i = [x_i ; 1]$ is homogeneous coordinates of the pixel $x_i$. The probability of this inequality can be represented by a sigmoid function, such that:

$$P(z_s | \omega_t(x_i) = 1) = \frac{1}{1 + \exp(\beta x_i^T C' x_i)}$$

(5.31)

where $\beta$ is the slope parameter of the sigmoid function.

The probability of every pixel inside or outside the ellipse is represented by color in Figure 5.5. The fitted ellipse is represented by a blue line, the probability of the pixel belonging to a traffic light is large inside and small outside the ellipse.
Figure 5.5: The ellipse is fitted to the connected pixels and shown by blue line. The probability of the pixels belonging to a traffic light is represented by color which is large inside and small outside of the ellipse.

**Height**

If an ellipse is fitted to the segmented pixels, this ellipse may be used in (5.25) to estimate the position of the traffic light with respect to the camera. If the camera height is known from the ground, the traffic light height is calculated. The estimated height is applied to detect the traffic lights.

However, the traffic light can be installed in different heights. They are installed on the pole, suspended over the road, or have other installation standards. Therefore, we assume there are $\kappa$ different traffic light installation standards, and we model every traffic light installation standard as a normal distribution.

In order to model the height of background clutters, we assume that the height of background clutter follows a normal distribution. Therefore, the height models for...
the traffic lights and background clutters are:

\[
P(z_h|z_s, \omega_t(x_i) = 1) = \sum_{k=1}^{\kappa} w_k \frac{1}{\sqrt{2\pi\sigma_k^2}} \exp\left(-\frac{(h - \mu_k)^2}{2\sigma_k^2}\right)
\]

\[
P(z_h|z_s, \omega_t(x_i) = 0) = \frac{1}{\sqrt{2\pi\sigma_0^2}} \exp\left(-\frac{(h - \mu_0)^2}{2\sigma_0^2}\right)
\]

Figure 5.10 shows the height model of the traffic lights. The traffic lights installed on a pole and suspended over the road are model as normal distribution (dashed lines) and the resultant mixture of Gaussians distribution is shown by solid line.

Figure 5.6: The height of the traffic light is modeled as mixture of Gaussians (solid line). The traffic light can be installed either on a pole or suspended over the road and each of these cases is modeled as normal distribution (dashed lines).

**Lens activation pattern**

If the position of the traffic light is estimated with respect to the camera, the position of its red, amber and green lenses are also known. Therefore, these lenses can be projected into the image space. If one lens is active, the other lenses may be inactive based on the traffic light standards. The inactive lenses are dark and their gray values follow the half-normal distribution with the variance \(\sigma^2\).
If the object is not a traffic light, the gray value can be uniformly distributed in the range of $[0, 255]$ by $P(z_l|z_h, \omega_t(x'_i) = 0) = \frac{1}{128}$. It can also be modeled as a normal distribution with the mean $\mu_0$ and the variance $\sigma_0^2$:

$$P(z_l|z_h, \omega_t(x'_i) = 1) = \frac{2\theta}{\pi} \exp\left(-\frac{I(x'_i)^2\theta^2}{\pi}\right)$$

$$P(z_l|z_h, \omega_t(x'_i) = 0) = \frac{1}{\sqrt{2\pi}\sigma_0^2} \exp\left(-\frac{(I(x'_i) - \mu_0)^2}{2\sigma_0^2}\right),$$

where $I(x'_i)$ is the gray value of the inactive lenses and $\theta = \frac{\sqrt{\pi}}{\sigma\sqrt{2}}$. (5.33)

Figure 5.7: The lens activation pattern indicates that if the red lens of the traffic light is active, the green lens must be inactive. Therefore, it must be dark and its gray value should follow half-normal distribution.

**GIS maps**

Prior knowledge is utilized to improve the traffic light detection approaches. For instance, if the position of a traffic light is stored in a geospatial database, this traffic light can be projected to the image space. The accuracy of the projected traffic lights depends on the accuracy of their position in the GIS maps, the camera calibration matrix, and the accuracy of the GPS/IMU navigation solution. Therefore,
the projected traffic lights may not be accurate. In addition, the GIS maps may have 2D position of the traffic lights and 3D position of the traffic lights require the information about their height. Moreover, most of GIS maps have general information such as signalized intersection rather than more detailed information such as the number of traffic lights in an intersection.

OpenStreetMap, which is publicly available, contains 2D location of the traffic lights. Since the users provide the information in OpenStreetMap, its accuracy and completeness depends on the user’s input and there is no guarantee of its accuracy and completeness. In addition, an inaccurate navigation solution leads to inaccurate position of the projected traffic lights in image space.

If the position of a projected traffic light is $\tilde{x}$, the probability of the pixel $x_i$ belongs to the traffic light follows a bivariate normal distribution with the mean $\tilde{x}$ and the variance $\Sigma_g$. The probability of the background pixels are also modeled as bivariate normal distribution with the mean $\tilde{x}_0$ and the variance $\Sigma_0$. Therefore, the traffic light detection based on prior knowledge is estimated by:

$$P(z_g|\omega_t(x_i) = 1) = \frac{1}{\sqrt{2\pi \Sigma_g^2}} \exp\left(-\frac{(x_i - \tilde{x}_g)^2}{2\Sigma_g^2}\right)$$

$$P(z_g|\omega_t(x_i) = 0) = \frac{1}{\sqrt{2\pi \Sigma_0^2}} \exp\left(-\frac{(x_i - \tilde{x}_0)^2}{2\Sigma_0^2}\right).$$  \hspace{1cm} (5.34)

Figure 5.8a shows the projected traffic light in the image space, represented by a red rectangle. The probability of the pixels belonging to the traffic light are demonstrated by color in Figure 5.8b.

5.2.5 Learning

In the previous sections, the statistical models mostly follow normal distribution. The observations are modeled by several distribution functions. These models have
some parameters such as mean and variance and these parameters may not be known prior the operation. Therefore, these parameters can be learned during the operation and the models are improved over time when more information are observed.

This learning approach is called unlabeled features learning and the Expectation Maximization (EM) is one of most frequently used approaches to learn from unlabeled data [40]. The Expectation Maximization consists of two steps: E-step, where the traffic lights are detected and M-step, where the model parameters are learned based on the detected traffic lights. It can be proven that this approach converges to the optimum solution as long as the problem is modeled as convex function [58]. It can be shown that in the E-step (the proof is given in the appendix):

$$q^{[j]}(\omega_t(x_i) = 0) = \frac{(1 - \kappa)\mathcal{N}(\mu_0^{[j-1]}, \Sigma_0^{[j-1]})}{\kappa \mathcal{N}(\mu_0^{[j-1]}, \Sigma_0^{[j-1]}) + (1 - \kappa)\mathcal{N}(\mu_0^{[j-1]}, \Sigma_0^{[j-1]})}$$

$$q^{[j]}(\omega_t(x_i) = 1) = \frac{\kappa \mathcal{N}(\mu_0^{[j-1]}, \Sigma_0^{[j-1]})}{\kappa \mathcal{N}(\mu_0^{[j-1]}, \Sigma_0^{[j-1]}) + (1 - \kappa)\mathcal{N}(\mu_0^{[j-1]}, \Sigma_0^{[j-1]})},$$

(5.35)
where $q^j(\omega_t(x_i)) = P(\omega_t(x_i)|Z_t(x_i), \mu_0^{[j-1]}, \Sigma_0^{[j-1]})$. The parameters can be estimated in M-step based on the estimated $q^j(\omega_t(x_i))$ by:

$$
\hat{\mu}_0^j, \hat{\Sigma}_0^j = \arg\max_{\mu_0, \Sigma_0} \sum_{i=1}^I q^j(\omega_t(x_i) = 0) \log((1 - \kappa)N(\mu_0^{[j-1]}, \Sigma_0^{[j-1]}))
$$

$$
\hat{\mu}_1^j, \hat{\Sigma}_1^j = \arg\max_{\mu_1, \Sigma_1} \sum_{i=1}^I q^j(\omega_t(x_i) = 1) \log(\kappa N(\mu_1^{[j-1]}, \Sigma_1^{[j-1]}))
$$

In the E-step and M-step, the superscript $[j]$ indicates to $j$-th iteration of the expectation maximization.

### 5.3 Implementation

The Bayesian statistical framework estimates the labels of the pixels in the images. However, this is computationally expensive if the proposed approach is applied to every pixel of the images. Many pixels are not similar to the traffic lights and can be deterministically detected. For instance, too dark or too bright objects, or the objects with non-ellipse shape are most likely form the background. Therefore, they can be labeled as background in the deterministic fashion. Therefore, the proposed Bayesian statistical approach is applied to the remaining objects. This speeds up the traffic light detection approach since it significantly reduces the search space.

In addition, the observations are modeled by various distribution functions. The choice of the model parameters, such as mean and variance, affects the results of the traffic light detection. These parameters can be learned as mentioned in the previous section. However, it requires initial model parameters in the Expectation Maximization approach. This section also discusses the initial values of the model parameters.
5.3.1 Reduction of the search space

The objects with the colors far from red, amber, and green belong to the background. Therefore, a threshold may be applied to the hue component of the HSV color space to filter these objects. In addition, too bright or too dark objects are not traffic lights and these objects are removed by utilizing thresholds in the saturation and value of the HSV color space. The results of the applied red color mask is shown in Figure 5.9a. However, the search space has been significantly reduced, the number of the red objects is still high.

The color camera is noisy and there are many red artifacts as a result of noise. Therefore, a median filter is applied to the image in Figure 5.9a and the color noise is removed, as shown in Figure 5.9b.

There are still many non-ellipse-shaped objects which belong to the background. Therefore, ellipse fitting is applied to the connected pixels and the non-ellipse-shaped objects are removed. Figure 5.9c shows the remaining objects after remove non-ellipse-shaped objects. In addition to the red objects in 5.9c, the amber and green objects are added and create a set of the potential traffic lights. The proposed Bayesian statistical approach should be applied to these object to verify whether these objects are the traffic lights. The deterministic background removal approach should be designed in a conservative way that it only filters the objects with very low probability of being a traffic light. Otherwise, it may filter some traffic lights.

5.3.2 Initial parameters

The labels of an image are modeled by Binomial distribution function in the beginning with the parameter, $\kappa$. This parameter is chosen as $10^{-4}$, representing a
Figure 5.9: (a) The red color mask applied to the image. There are many red objects in the image. (b) The median filter is applied to remove color noise of camera. (c) An ellipse is fitted to every object and non-ellipse-shaped objects are removed. The search space has been significantly reduced for the potential traffic lights.

traffic light with size of 5 × 5 in an image with size of 640 × 480. Each color is modeled as normal distribution with the mean, \( \mu_{1,2,3} \) and the variance \( \sigma^2_{1,2,3} \). These parameters are chosen in such a way that it obeys the hue histogram of the traffic lights provided by Levinson et al. [46]. However, the hue histogram provides better probability distribution, it requires a Look Up Table (LUT) and therefore, it is slower than our normal distribution model. The background is also modeled as the
normal distribution. Its large variance shows the large variation in the color of the background. The initial parameters of the observation models are given in Table 5.1.

Table 5.1: The initial parameters of the models in this study.

| Distribution | $P(\omega)$ | $P(Z_c|\omega)$ | $P(Z_g|\omega)$ | $P(Z_h|\omega)$ | $P(Z_l|\omega)$ |
|--------------|-------------|----------------|----------------|----------------|----------------|
| $P(\omega = 1)$ | $\kappa = 10^{-4}$ | $N_1(10, 8)$ | $N_1(\tilde{\mu}, 30)$ | $N_1(5.5, 0.5)$ | $N_1(1, 30)$ |
| $P(\omega = 0)$ | $1 - \kappa$ | $N_0(180, 90)$ | $N_0(\tilde{\mu}, 100)$ | $N_0(0, 20)$ | $N_0(125, 70)$ |

5.4 Experiments

The proposed Bayesian framework has been evaluated based on two distinct publicly available benchmarks: Karlsruhe Institute of Technology (KITTI) and La Route Automatisse (LARA) benchmarks. In the KITTI benchmark, which has been described in the previous sections, several datasets have been collected from rural and urban environments. These datasets include multiple sensors such as color cameras and the GPS/IMU navigation solution. The GPS/IMU navigation solution is also utilized to find and retrieve the traffic features in the vicinity of the platform.

Since the traffic light detection depends on the color camera, properties of this sensor is described. The color camera has $1027 \times 333$ resolution with $90^\circ$ opening angle. The lens focal length is 4 millimeters with $4.65\mu m$ pixel size [20]. The KITTI benchmark has been collected in Germany, and therefore, the traffic lights follow the German traffic light standards. Since this benchmark is not specifically collected for
the traffic light detection, there are only 440 traffic lights with 404 red lights, 10 amber lights, and 26 green lights. The ground truth of the traffic lights have not been provided in this benchmark and we manually annotated them to evaluate our results.

The LARA benchmark has a dataset collected in downtown of Paris, France. Therefore, the traffic lights obey the French traffic light standards. The dataset contains the images taken from a single uncalibrated color camera. Since height of the camera is not provided, we have approximated the camera height and applied this approximation to estimate height of the traffic lights. In addition, the camera calibration matrix is not available in this dataset and we approximated this matrix based on the nominal lens focal length of the camera. These approximations affect the results of the traffic light detection. However, we witnessed even the use of the approximated camera height and camera calibration matrix improves the traffic light detection results.

The color camera is the LARA benchmark has 640 × 480 resolution with 12 millimeters lens focal length. This benchmark contains 5280 red lights, 58 amber lights, 3381 green lights. Since this benchmark is lengthy and the platform was static in a long period of time, we chose a section of dataset with many traffic lights. This section has more than 1880 frames and 2486 traffic lights.

The number of red traffic lights is higher in both benchmarks since the platform stops behind the red light and observes the traffic light for longer period of time. The number of amber traffic lights is small in both benchmarks since the amber light shortly lasts.

When a traffic light is detected, we apply a bounding box representing the traffic light. The traffic lights are also represented by bounding boxes in the ground truth.
Based on the PASCAL criterion, we consider a traffic light to be correctly detected if the intersection of these two bounding boxes is larger than half of their union. These traffic lights are true positives, $TP$. Otherwise, we consider the traffic lights as incorrectly detected. These traffic lights are false positives, $FP$. Traffic lights not detected by the proposed approached are false negatives, $FN$.

In order to evaluate the proposed traffic light detection, the precision rate and recall rate are applied. The precision rate is the ratio of the true positives to the detected traffic lights (summation of correctly and incorrectly detected traffic lights). In other words, the precision rate is computed by:

$$\text{Precision} = \frac{TP}{TP + FP}.$$  \hspace{1cm} (5.37)

The recall rate is the ratio of the true positives to the number of traffic lights in the ground truth (summation of correctly detected traffic lights and the ones that are not detected). In other words, the recall rate is computed by:

$$\text{Recall} = \frac{TP}{TP + FN}.$$  \hspace{1cm} (5.38)

Since true negatives are not available, the accuracy criterion cannot be calculated.

### 5.5 Results

Results of the traffic light detection depends on the threshold in (5.3). In order to find the optimum value for this threshold, we plot the precision/recall curve and choose the value providing the best balance between the precision and recall rates. If the threshold is chosen too high, it only detect the traffic lights with very high probability. Therefore, the precision rate increases while the recall rate decreases. In
contrast, if the threshold is chosen too low, low probability objects are also labeled as the traffic light. Therefore, the precision rate decreases, but the recall rate increases. The ideal point is where both the precision rate and the recall rate are 100%. The best balance between the precision rate and the recall rate is the point which has the least distance to the ideal point.

### 5.5.1 KITTI dataset

The precision/recall curve of the proposed traffic light detection applied on KITTI benchmark is plotted in Figure 5.10. The maximum precision and recall rate happens where the threshold is 88.4%. In other words, a pixel is labeled as a pixel belonging to the traffic light if its probability is higher than 88.4%. The precision rate and recall rate are 95.8% and 93.0%, respectively. Moreover, it is important to verify whether the color of traffic lights are correctly recognized. Table 5.2, shows the confusion matrix for the classified colors of the traffic lights. This table shows there is no incorrect color classification of the traffic lights.

**Table 5.2: The confusion matrix; It shows that the traffic lights signals have been correctly detected.**

<table>
<thead>
<tr>
<th></th>
<th>System classification</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Red</td>
</tr>
<tr>
<td>Red</td>
<td>380</td>
</tr>
<tr>
<td>Amber</td>
<td>404</td>
</tr>
<tr>
<td>Green</td>
<td>10</td>
</tr>
<tr>
<td></td>
<td>26</td>
</tr>
<tr>
<td>Recall rate</td>
<td>94%</td>
</tr>
</tbody>
</table>
Figure 5.10: Precision/Recall curve for the KITTI dataset. The best precision/recall rate balance occurs at $P(\omega_t) > 88.4\%$. Number of the traffic lights is 440.

The traditional traffic lights use a light source and a color filter. In these traffic lights, the light energy is not uniformly distributed on the lens and the color of the traffic light changes from lens center to its borders. Figure 5.7 shows an example of this situation where the lower part of the traffic light is yellow although the red lens of the traffic light is active.

In Figure 5.7, the red lens of the traffic light is partially occluded, but the proposed approach is able to correctly detect the traffic light using the temporal coherency constraint.
Figure 5.11: The traffic light detection for a KITTI dataset; The lower part of the red signal has more illumination and it looks yellow. Therefore, the color of the signal should be evaluated for every pixel and the choice of the color should be estimated in a voting scheme.

Figure 5.12: The traffic light detection for a KITTI dataset; The traffic light is partially occluded, but the algorithm can detect the traffic light since it applies the temporal consistency.

Figure 5.13 to Figure 5.16 show a scenario when the platform stops behind the red light and waits until it turns to green. The red light, which is correctly detected, is shown in Figure 5.13. In many countries, two lenses cannot be active simultaneously,
but the red and amber lights may be active simultaneously in Germany. The proposed approach is able to correctly detect the red and amber lenses in Figure 5.14. The light turns to green and the proposed approach correctly detects the green traffic light in Figure 5.15. Figure 5.16 shows a false positive object incorrectly labeled as amber light. If the threshold is chosen as 88.4%, as previously mentioned, this object with the probability of 61% will be removed.

Figure 5.13: The traffic light detection for a KITTI dataset. The red signal of the traffic light is correctly recognized.

In Figure 5.17, there are multiple traffic lights on the road. The proposed approach is correctly detected the traffic lights and their position is estimated with respect to the camera. Therefore, the platform can select the traffic light of its lane.
Figure 5.14: The traffic light detection for a KITTI dataset. The red and amber lenses are simultaneously active and the proposed approach can correctly detect both signals. The amber traffic light is shown by blue for better visualization.

Figure 5.15: The traffic light detection for a KITTI dataset. The traffic light signal is green and it is correctly recognized.

5.5.2 LARA dataset

The proposed Bayesian statistical approach is also tested using the LARA benchmark. The Precision/Recall curve is plotted in Figure 5.18 and the best balance between the precision rate and recall rate occurs where the threshold is chosen as
Figure 5.16: The traffic light detection for a KITTI dataset; There is a false positive that has low probability (61%) and will be removed using the threshold in (5.3). The amber traffic light is shown by blue for better visualization.

Figure 5.17: Multiple traffic light detection in KITTI dataset. The estimated location of the traffic lights can be applied to find the traffic light that corresponds to the each lane.

93.8%. In other words, the objects with the probability higher than this threshold are labeled as the traffic lights. Using this threshold, the precision rate and recall rate are 98.7% and 94.7%.

The precision and recall rate were compared with the earlier studies on this dataset and the results are given in Table 5.3. Our proposed traffic light detection approach
Figure 5.18: Precision/Recall curve for LARA dataset. The best precision and recall rate balance happens at $P(\omega_t) > 93.9\%$. Number of the traffic lights is 2486.

has the highest precision rate and its recall rate is the second highest recall rate. Therefore, the proposed traffic light detection approach outperforms the previous traffic light detection approaches.

Table 5.3: The traffic light detection algorithms are compared using LARA benchmark; Our proposed approach outperforms the other approaches, it has the highest precision rate.

<table>
<thead>
<tr>
<th></th>
<th>[11]</th>
<th>[12]</th>
<th>[25]</th>
<th>[64]</th>
<th>[73]</th>
<th>Ours</th>
</tr>
</thead>
<tbody>
<tr>
<td>Precision Rate</td>
<td>95.38%</td>
<td>84.5%</td>
<td>72.83%</td>
<td>61.22%</td>
<td>96.95%</td>
<td>98.66%</td>
</tr>
<tr>
<td>Recall Rate</td>
<td><strong>98.41%</strong></td>
<td>53.5%</td>
<td>80.13%</td>
<td>93.75%</td>
<td>94.4%</td>
<td>94.65%</td>
</tr>
</tbody>
</table>
Figure 5.19 to Figure 5.24 demonstrate the results of the proposed traffic light detection approach applied to 15 seconds of LARA dataset. The traffic lights were correctly detected during this 15 seconds sequence, where the platform approaches a red light, stops behind the traffic light, and waits until it turns to green light.

5.6 Conclusion

In this paper, a Bayesian statistical framework is proposed to detect the traffic lights. We have applied spatio-temporal consistency constraints to improve the proposed traffic light detection approach. In addition to traffic light detection, we utilized the conic section geometry to estimate the position of the traffic lights with respect to the camera.
Figure 5.20: The LARA traffic light detection results after 1 second.

Figure 5.21: The LARA traffic light detection results after 3 seconds.
Figure 5.22: The LARA traffic light detection results after 6.5 seconds.

Figure 5.23: The LARA traffic light detection results after 11 seconds.
Figure 5.24: The LARA traffic light detection results after 13.5 seconds.
Chapter 6: Conclusion

In this dissertation, the reliability of the autonomous driving was studied and improved algorithms were developed:

• The proposed traffic light detection approach produced 98.7% and 94.7% precision and recall rates in LARA traffic light benchmark and outperformed the earlier studies. In this approach we utilized conic section geometry to estimate the position of the traffic lights with respect to the camera coordinate system.

• The 3D moving object detection and tracking was introduced based on multiple sensor integration. We applied the Kalman filter with non-holonomic constraints to track the moving objects.

• A 3D super-resolution approach was developed to improve the resolution of the point clouds and filter out their range noise. The results show the proposed approach has superior performance over the existing approaches and it preserves boundaries on an objects as well as its edges and corners.
  
  – We generalized 2D super-resolution applied in image processing to the point clouds and provided diffusion equations to reconstruct the surface of an object. The diffusion tensor was introduced to preserve the edges and corners of the point clouds.
We developed a regularizer to reconstruct the surface of an object based on the brightness changes in the surface.

6.1 Future work

The object classification improves the tracking results since it determines the dynamic model of the object. In other words, the dynamic model of vehicles, cyclists, and pedestrians are different and knowledge about type of the object improves the dynamic model in the 3D tracking. The tracking results are also important to correctly classify the object. For instance, the highest speed objects are most likely the vehicles. Therefore, a combination of the object classification and tracking will improve both tasks. In particular, multiple hypothesis tracking can be used for each class of object when the object classification is benefited from the accuracy of tracked objects.

In addition, the object tracking, object classification, traffic light and sign detection, lane detection and other subtasks benefit from online learning scheme. In this approach, the models applied in each task are improved while the observation is collected. This approach has been applied for the traffic light detection, but also can be extended to the other subtasks.
Appendix A: Proofs

Problem statement

Assuming

\[ E = \int_{\Omega} \Phi(||\nabla W||) dudv, \tag{A.1} \]

it should be proven that

\[ \frac{\partial W}{\partial t} = \Delta W, \tag{A.2} \]

Proof

Let’s assume that there is a function, \( \Phi \) and it has the energy of \( E \), such that

\[ E = \int_{\Omega} \Phi(u, v, W, W_u, W_v) dudv, \tag{A.3} \]

where \( u \) and \( v \) are two orthogonal vector that constitute the space bases and \( W \) is a function of these two vector bases. This energy function is minimized if

\[ \frac{\partial \Phi}{\partial W} - \frac{\partial}{\partial u} \frac{\partial \Phi}{\partial W_u} - \frac{\partial}{\partial v} \frac{\partial \Phi}{\partial W_v} = 0, \tag{A.4} \]

This is a sufficient condition that solution of (A.3) is minimum. If \( \Phi \) is the quadratic function, \( \Phi(||\nabla W||) = W_u^2 + W_v^2 \) and its derivatives are

\[ \begin{align*}
\frac{\partial \Phi}{\partial W} &= 0 \\
\frac{\partial \Phi}{\partial W_u} &= 2W_u \\
\frac{\partial \Phi}{\partial W_v} &= 2W_v
\end{align*} \tag{A.5} \]
Therefore, Equation (A.4) is simplified such that

\[ W_{uu} + W_{vv} = \Delta W = 0 \]  

(A.6)

If the parameter \( W \) is considered as a function of time, \( W_t \) is changed such that

\[ W_t = \Delta W \]  

(A.7)

**Problem statement**

Assuming

\[ E = \int\int_{\Omega} \Phi(||\nabla W|| - \tan |I(x) - (x')|)dudv, \]  

(A.8)

it should be proven that

\[ \frac{\partial W}{\partial t} = div(||\nabla W|| - \tan |I(x) - I(x')|) \frac{\nabla W}{||\nabla W||} \]  

(A.9)

**Proof**

Let’s assume that there is a function, \( \Phi \) and it has the energy of \( E \), such that

\[ E = \int\int_{\Omega} \Phi(u, v, W, W_u, W_v)dudv, \]  

(A.10)

where \( u \) and \( v \) are two orthogonal vector that constitute the space bases and \( W \) is a function of these two vector bases. This energy function is minimized if

\[ \frac{\partial \Phi}{\partial W} - \frac{\partial}{\partial u} \frac{\partial \Phi}{\partial W_u} - \frac{\partial}{\partial v} \frac{\partial \Phi}{\partial W_v} = 0, \]  

(A.11)

This is a sufficient condition that solution of (A.10) is minimum. If \( \Phi \) is the quadratic function, such that

\[ \Phi(||\nabla W|| - \tan |I(x) - (x')|) = W_u^2 + W_v^2 + \tan^2 |I(x) - (x')| - 2\sqrt{(W_u^2 + W_v^2) \tan |I(x) - (x')|}. \]  

(A.12)
The derivatives of function $\Phi$ are calculated such that

$$
\frac{\partial \Phi}{\partial W} = 0
$$

$$
\frac{\partial \Phi}{\partial W_u} = 2W_u - 2 \tan |I(x) - (x')| \frac{W_u}{||\nabla W||}
$$

$$
\frac{\partial \Phi}{\partial W_v} = 2W_v - 2 \tan |I(x) - (x')| \frac{W_v}{||\nabla W||}
$$

(A.13)

Therefore, Equation (A.11) is simplified such that

$$
div((||\nabla W|| - tan |I(x) - I(x')|) \frac{\nabla W}{||\nabla W||}) = 0
$$

(A.14)

**Problem statement**

Assuming

$$
Pr(\omega_1(x_i) = 1) = \kappa
$$

$$
Pr(\omega_1(x_i) = 0) = 1 - \kappa
$$

(A.15)

and

$$
Pr(Z_i|\omega_t(x_i) = 1) = \mathcal{N}_Z(\mu_1, \Sigma_1)
$$

$$
Pr(Z_i|\omega_t(x_i) = 0) = \mathcal{N}_Z(\mu_0, \Sigma_0)
$$

(A.16)

it should be proven that

$$
\hat{\kappa} =
$$

$$
\hat{\mu}_0^{[j]} , \hat{\Sigma}_0^{[j]} =
$$

$$
argmax_{\mu_0, \Sigma_0}(\sum_{i=1}^q [\omega_t(x_i) = 0) \log[(1 - \kappa)\mathcal{N}(\mu_0^{[j-1]}, \Sigma_0^{[j-1]})]}
$$

(A.17)

$$
\hat{\mu}_1^{[j]} , \hat{\Sigma}_1^{[j]} =
$$

$$
argmax_{\mu_0, \Sigma_0}(\sum_{i=1}^q [\omega_t(x_i) = 1) \log[\kappa\mathcal{N}(\mu_1^{[j-1]}, \Sigma_1^{[j-1]})])
$$

Proof

The parameters of these two normal distributions, $\mu_1$, $\Sigma_1$, $\mu_0$, and $\Sigma_0$ and also binomial parameter, $\kappa$ may not be accurately available. These parameters are estimated
using the maximization of the observations, such that

\[
\hat{\mu}_0, \hat{\Sigma}_0, \hat{\kappa} = \text{argmax}_{\mu_0, \Sigma_0} \left( \prod_{i=1}^{I} Pr(Z_i | \mu_0, \Sigma_0) \right) =
\text{argmax}_{\mu_0, \Sigma_0, \kappa} (\Sigma_i \prod_{i=1}^{I} \log Pr(Z_i | \mu_0, \Sigma_0))
\]  \tag{A.18}

\[
\hat{\mu}_1, \hat{\Sigma}_1, \hat{\kappa} = \text{argmax}_{\mu_1, \Sigma_1} \left( \prod_{i=1}^{I} Pr(Z_i | \mu_1, \Sigma_1) \right)
\text{argmax}_{\mu_1, \Sigma_1, \kappa} (\Sigma_i \prod_{i=1}^{I} \log Pr(Z_i | \mu_1, \Sigma_1))
\]

The labeling is estimated using the initial parameters of the model, such that

\[
\hat{q}_i(\omega_i) = \frac{Pr(\omega_i | Z, \kappa, \hat{\mu}_0, \hat{\Sigma}_0, \hat{\mu}_1, \hat{\Sigma}_1)}{Pr(Z)}
\]  \tag{A.19}

The parameters of the model is updated using the labeled pixels, such that

\[
\hat{\mu}_0, \hat{\Sigma}_0, \hat{\kappa} = \text{argmax}_{\mu_0, \Sigma_0} (\Sigma_i \prod_{i=1}^{I} \log [Pr(Z_i | \omega_0, \mu_0, \Sigma_0) Pr(\omega_0)]) =
\text{argmax}_{\mu_0, \Sigma_0, \kappa} (\Sigma_i \prod_{i=1}^{I} \log [(1 - \kappa) \mathcal{N}(\mu_0, \Sigma_0)])
\]  \tag{A.20}

\[
\hat{\mu}_1, \hat{\Sigma}_1, \hat{\kappa} = \text{argmax}_{\mu_1, \Sigma_1} (\Sigma_i \prod_{i=1}^{I} \log [Pr(Z_i | \omega_1, \mu_1, \Sigma_1) Pr(\omega_1)])
\text{argmax}_{\mu_1, \Sigma_1, \kappa} (\Sigma_i \prod_{i=1}^{I} \log [(\kappa \mathcal{N}(\mu_1, \Sigma_1)])
\]

Since the model parameters \( \mu_0 \) and \( \Sigma_0 \) are dependent to \( Pr(\omega_t = 0) \) and the model parameters \( \mu_1 \) and \( \Sigma_1 \) are dependent to \( Pr(\omega_t = 1) \), these parameters cannot be estimated directly. We apply the iterated expectation maximization approach to solve this problem. The expectation maximization has two steps in every iteration: E-step which estimates \( Pr(\omega_t) \) based on the initial model parameters and M-step which estimates the model parameters based on the estimated \( Pr(\omega_t) \). In the E-step

137
Pr(ωₜ) is estimated, such that

\[ q^{[j]}(ωₜ(xᵢ) = 0) = Pr(ωₜ(xᵢ) = 0 | Zᵢ, μ₀^{[j-1]}, Σ₀^{[j-1]}) = \]

\[ \frac{Pr(Zₜ | ωₜ(xᵢ) = 0, μ₀^{[j-1]}, Σ₀^{[j-1]}) Pr(ωₜ(xᵢ) = 0 | μ₀^{[j-1]}, Σ₀^{[j-1]})}{\sum_{k \in \{0,1\}} Pr(Zₜ | ωₜ(xᵢ) = k, μₖ^{[j-1]}, Σₖ^{[j-1]}) Pr(ωₜ(xᵢ) = k | μₖ^{[j-1]}, Σₖ^{[j-1]})} \]

\( (1 - \kappa)N(μ₀^{[j-1]}, Σ₀^{[j-1]}) \)

\[ \kappa N(μ₀^{[j-1]}, Σ₀^{[j-1]}) + (1 - \kappa)N(μ₀^{[j-1]}, Σ₀^{[j-1]}) = r_{i0} \]

\[ q^{[j]}(ωₜ(xᵢ) = 1) = Pr(ωₜ(xᵢ) = 1 | Zᵢ, μ₁^{[j-1]}, Σ₁^{[j-1]}) = \]

\[ \frac{Pr(Zₜ | ωₜ(xᵢ) = 1, μ₁^{[j-1]}, Σ₁^{[j-1]}) Pr(ωₜ(xᵢ) = 1 | μ₁^{[j-1]}, Σ₁^{[j-1]})}{\sum_{k \in \{0,1\}} Pr(Zₜ | ωₜ(xᵢ) = k, μₖ^{[j-1]}, Σₖ^{[j-1]}) Pr(ωₜ(xᵢ) = k | μₖ^{[j-1]}, Σₖ^{[j-1]})} \]

\( \kappa N(μ₁^{[j-1]}, Σ₁^{[j-1]}) \)

\[ \kappa N(μ₁^{[j-1]}, Σ₁^{[j-1]}) + (1 - \kappa)N(μ₁^{[j-1]}, Σ₁^{[j-1]}) = r_{i1} \]

The model parameters can be estimated in M-step based on the estimated \( q^{[j]}(ωₜ(xᵢ)) \), such that

\[ \hat{κ}^{[j]} = \frac{Σ_{i=1}^{I} r_{i0}}{Σ_{i=1}^{I} r_{i0} + Σ_{i=1}^{I} r_{i1}} \]

\[ \hat{μ}₀^{[j]} = \frac{Σ_{i=1}^{I} r_{i0} Zᵢ}{Σ_{i=1}^{I} r_{i0}} \]

\[ \hat{Σ}₀^{[j]} = \frac{Σ_{i=1}^{I} r_{i0} (Zᵢ - \hat{μ}₀^{[j]}) (Zᵢ - \hat{μ}₀^{[j]})ᵀ}{Σ_{i=1}^{I} r_{i0}} \]

\[ \hat{μ}₁^{[j]} = \frac{Σ_{i=1}^{I} r_{i1} Zᵢ}{Σ_{i=1}^{I} r_{i1}} \]

\[ \hat{Σ}₀^{[j]} = \frac{Σ_{i=1}^{I} r_{i1} (Zᵢ - \hat{μ}₁^{[j]}) (Zᵢ - \hat{μ}₁^{[j]})ᵀ}{Σ_{i=1}^{I} r_{i1}} \]
Bibliography


