Obstacle Avoidance Based on Scene Depth Perception using Stereo Cameras for Autonomous Vehicles

DE-42 (MTS) ABDUL AHAD, DANIA, M.BILAL, M.AHMAD

COLLEGE OF

ELECTRICAL AND MECHANICAL ENGINEERING

NATIONAL UNIVERSITY OF SCIENCES AND TECHNOLOGY

RAWALPINDI

2024

COLLEGE OF ELECTRICAL AND MECHANICAL ENGINEERING

DE-42 MTS

PROJECT REPORT

Obstacle Avoidance Based on Scene Depth Perception using Stereo Cameras for Autonomous Vehicles

Submitted to the Department of Mechatronics Engineering

in partial fulfillment of the requirements

for the degree of

Bachelor of Engineering

in

Mechatronics

2024

Sponsoring DS:

Dr. Tahir Habib Nawaz (Supervisor)

Dr. Ayesha Zeb (Co-Supervisor)

Submitted By:

Abdul Ahad Mehmood

Muhammad Bilal Tariq

Muhammad Ahmad Raees

Dania Saleem

ACKNOWLEDGMENTS

We express our sincere gratitude to the Divine for blessing us with the opportunity to undertake this comprehensive research on the perception module of self-driving cars. We are immensely thankful to our supervisor, Dr. Tahir Habib Nawaz, whose invaluable guidance and advice enabled us to overcome numerous challenges throughout this endeavor. Additionally, we extend our appreciation to our co-supervisor, Dr. Ayesha Zeb, for her invaluable assistance and support. Finally, we extend our heartfelt thanks to our parents, friends, and colleagues for their unwavering encouragement and support throughout this journey.

ABSTRACT

In this thesis, stereo vision has been explored in the context of autonomous vehicles, for obstacle avoidance. Stereo vision involves combining 2D views for 3D depth estimation, similar to how human vision works. The emphasis is on real-time algorithms for efficient obstacle tracking and navigation. Some of the depth perception algorithms discussed include stereo matching and semi-global methods. Furthermore, for insights into pixel-level comparison of disparity maps, methodologies such as area based disparity, feature based disparity and triangulation have been covered. Mobile Robot Navigation strategies, categorized into seven types, are presented, including odometry, inertial navigation, magnetic compasses, active beacons, GPS, landmark navigation, and map-based positioning. Real-time obstacle avoidance algorithms using depth maps have been analyzed for ideal conditions and potential limitations. This thesis also includes a section on the design considerations of a four-wheel steering mobile robot platform. It explores adaptive steering control algorithms for optimal performance during manual operation, addressing challenges associated with four wheel steering mechanisms. This project thesis flow work is divided into 3 parts. The first being on ZED Stereo Camera for stereo vision for making of disparity maps followed by depth maps which then is used to estimate real time depth of the objects. The second is object detection and obstacle avoidance algorithm generation. The object detection models such as Yolo V4 lite is used to identify the obstacle in the path and then generated algorithm for obstacle avoidance is used as discussed further in the thesis. The final part is Mobile Base platform design and integration. A steering controlled mobile base is manufactured for real life testing if the algorithms in real time followed by the embedded integration using Jetson Nano.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS

α = Depth map from stereo camera.

β = Depth map from LiDAR

γ = Fused depth map

μ = Fusion Algorithm.

A_1 = Uncalibrated left image from Stereo Camera.

A_2 = Uncalibrated right image from Stereo Camera.

A_1' = Rectified right left from Stereo Camera.

A_2' = Rectified left right from Stereo Camera.

ρ = Point Cloud.

f = Focal length of ZED in pixels.

u = Disparity map.

$b\text{aseline}$ = Distance between the two cameras of ZED.

k_1, k_2, k_3 = Radial distortion coefficients.

p_1, p_2 = Tangential distortion coefficients.

R = Rotation matrix.

T = Translation vector.

E = Essential matrix.

F = Fundamental matrix.

r_1, r_2 = Rectification transforms.

P_1, P_2 = Projection matrices.

Q = Disparity to depth mapping matrix.

A = State matrix of the model.

B = Control matrix of the model.

C = Measurement matrix

Q = Process noise covariance matrix.

R = Measurement noise covariance matrix.

U = Control vector.

X = State vector.

P = Error covariance

S = Innovation covariance matrix.

z = Measurement vector.

y = Measurement residual.

K = Kalman gain

Chapter 1 - INTRODUCTION

1.1 Overview:

Our final year project (FYP) focuses on obstacle avoidance for autonomous vehicles using depth perception with a stereo camera. By leveraging the ZED SDK stereo camera, we calculate depth maps to determine regions of interest (ROI). The depth information from these ROIs guides the motion of the vehicle to avoid obstacles. The Jetson Nano handles the processing, while the ESP32 Devkit V1 controls the mobile platform**.**

1.2 Motivation

The increasing demand for autonomous vehicles capable of navigating complex environments with high reliability drives this project. Efficient obstacle avoidance is crucial for the safe and effective operation of these vehicles. By developing a system that uses depth perception from a stereo camera, we aim to contribute to advancements in autonomous navigation technology. The previous work in this area, conducted by a prior degree project, involved developing an algorithm for depth perception using a combination of a stereo camera and LiDAR. Their project focused solely on software implementation and did not include an actual mobile platform. Additionally, they used both stereo cameras and LiDAR, and their system suffered from low frames per second (FPS), making it unsuitable for real-time processing. In contrast, our project implements the system on a physical mobile platform, uses only a stereo camera, and aims to achieve real-time processing capabilities.

1.3 Problem Statement

The main challenge addressed in this project is developing a real-time, efficient obstacle avoidance system for autonomous vehicles. This involves processing stereo camera data to create depth maps, identifying ROIs, and using depth information to control the vehicle's motion to avoid obstacles. Ensuring smooth communication between the Jetson Nano and the ESP32 and protecting the electronics from high current spikes are also critical challenges.

1.4 Our Contribution

Our project contributes to the field by integrating a Jetson Nano and an ESP32 to create a robust, real-time obstacle avoidance system for autonomous vehicles. We use the ZED SDK stereo camera to calculate depth maps and determine ROIs, guiding the vehicle's motion to avoid obstacles. Unlike previous work, we implement our system on an actual mobile platform, focusing on achieving high FPS for real-time processing. We also ensure galvanic isolation using PC817 optocouplers to protect our electronics from high current spikes generated by the motors. This is one of the first experiments in the field of self-driving cars done on a Mobile platform in real time to provide a proof of concept in the department of Mechatronics

Engineering at the College of Electrical and Mechanical Engineering NUST.

1.5 Organization of Thesis

The thesis is organized as follows:

Chapter 1 Deals with the introduction the projects, it's background knowledge and previous, it's scope and interests in fields, and our contribution in the project.

Chapter 2 encompasses all the an overview of all the necessary literature required for the start of the project which includes all existing methodology and algorithms or approaches and work already done on it.

Chapter 3 deals with the working, making and generation of algorithm of depth perception using stereo vision approach and using hardware such as Stereo Camera.

Chapter 4 describes the use of the depth perception algorithm for avoiding any of the incoming obstacle in the path.

Chapter 5 explains the working and manufacturing of mobile base platform on which the real time testing is performed.

Chapter 6 deals with the integration of embedded and software algorithms to make a complete working autonomous vehicle porotype for testing and evaluation.

Chapter 2 – Literature Review

2.1 Introduction

Stereo Vision is a domain of machine vison which focuses on obtaining human vision-like view of the world by combining two or more 2D views of the same and combining those to obtain 3D depth estimation of the scene.

As an emerging technology, Stereo Vision algorithms are constantly being revised and developed, many alternative approaches exist for implementation of a Stereo Vision system.[1]

Depth Perception using stereo vision is almost based on the human vision perception of the world. Human Vision is binocular in nature i.e., two eyes, which basically sees the same scene which is then further processed by the brain to create 3D illustration of the real world.

We will focus on the stereo vision application of Autonomous vehicles. As it is a advancing field various research is being going on and advancement comes daily. Autonomous vehicles employing Stereo Vision techniques must operate in real time, and thus are driving research for faster and more efficient Stereo Vision algorithms, whilst retaining enough accuracy to build a navigable 3D map, track moving obstacles, and eliminate a reasonable volume of noise.[1]

The autonomous vehicle system either based on stereo vision-based depth perception or various techniques must be robust enough to cater for the traffic, obstacles, road lanes, passing by pedestrian etc. Our research is focused on obstacle avoidance using stereo vision.

Many researches for autonomous vehicles have been done in the past and is been going on as we speak, but only a few have produced a significant and accurate results. It's not the research that is at fault but the limitations of various sensors including stereo vision which is to blame. However, many have produced significant results as well such as *VaMP* Prototype, *RALPH* system which was based on NavLab 5 stereo Vision, *ARGO* vehicle, etc.

The stereo vision based application of autonomous vehicle is being also used by NASA for their Lunar Rovers as previously they have to control it form command center on earth but due to communication speed lags it created several problems. To overcome this NASA is now building autonomous rovers which can detect terrains using Stereo vision and can act accordingly.

2.2. Depth Perception

2.2.1. Algorithms:

Many works in the recent years on scene depth perception using stereo camera and for that many algorithms based on various datasets have been worked on and is available. Some of the algorithms are as follows:

2.2.1.1. Stereo Matching Algorithm:

Binocular vision is an important method for obtaining stereo perception of the real world. Stereo matching is the key part of binocular vision [1]. Binocular Stereo vision is one of the important field of computer vision widely used in scene reconstruction and unmanned driving.

The basic idea of stereo vision is to is to use two camera system i.e. stereo camera to take image of the same scene twice from each camera and then to calculates it's difference using certain mathematical models and find the disparity of the scene to make disparity maps.

Binocular stereo vision uses a computer instead of the human brain to perform stereo matching based on the imaging principle of the human eye to obtain parallax and then depth of objects in scenes [1].

2.2.1.2. Types of Stereo Matching Algorithm:

Local stereo matching algorithm:

The local stereo matching algorithm relies on pixel-level data to calculate parallax, utilizing a window of nearby pixels for matching. It assesses the matching cost between the pixel of interest and all corresponding pixels in the right and left camera images within this local window. The algorithm then selects the parallax by identifying the window with the smallest matching cost.

Certain researchers have enhanced this algorithm by refining the process of computing the matching cost. Zabih [2] et al introduced a novel matching cost calculation function census which notably mitigates the impact of lighting variations. Zhang [3] et al. proposed a Cross Based adaptive window which reduces computation [1]. Liu [4] re al. introduced a shaped window determined through the double helix path method, which aligns more closely with intricate image contours. Gerrits [5] introduced a local stereo matching algorithm based on segmentation, implementing an adaptive window by diminishing the weight of points that do not belong to the same segment. Hosin [6] proposed adaptive weighting using a bootstrap filter. Wang [7] et al. proposed a quadratic bootstrap filtering with adaptive weights.

Global Stereo Matching Algorithm:

The global stereo matching algorithm creates an energy function spanning the entire images. It iterates the matching algorithm iteratively, seeking the

global optimal parallax by determining the best solution for this function. Boykov [8] suggested employing the graph-cutting algorithm for stereo matching, aiming to eliminate parallax maps with lateral stripes that arise from dynamic planning issues. Kim [9] et al. introduced a stereo matching algorithm grounded in confidence propagation, aiming to address the issue of generating poor parallax maps. Yao [10] et al introduced a stereo matching algorithm grounded in confidence propagation, aiming to address the issue of generating poor parallax maps.

Semi-Global Stereo Matching Algorithm:

The semi global stereo matching is an optimization of the global matching algorithm, which converts finding minimum energy function on 2D image into finding it on multiple! D paths. Basically it improves the speed of the algorithm.

First of all, Hirsch Muller [11] proposed semi-global matching algorithm, which uses a matching cost function of mutual information to fit a global 2D constraint. Zhou [12] designed a semi global stere matching method combining improved LBP and multi path matching cost aggregation. It improved both accuracy and speed.

2.2.2 Methodology:

2.2.2.1 Triangulation:

Basic Triangulation technique includes taking two point perorally aligned and of known distance sees the same scene then calculation is performed to calculate disparity. For this purpose, two camera systems i.e. stereo camera is used.

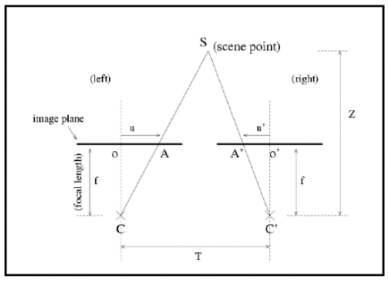

Figure 1 Triangulation [12]

Referring to figure 1, two cameras (C, C') separated by a distance (T), are pointed at the same feature (S). Bothe of these camera are making an image of same scene whose location of points are as A and A'. due the distance between two cameras the location of images from each camera will differ with respect to normal axis (U and U'). The Distance can be calculated using following formula.

Distance = f T

U-U’

Where, f is the focal length of the camera

2.2.2.2 Feature-Based Disparity:

The Feature Extraction method primarily focuses on identifying and comparing various features, such as edges, lines, circles, and curves, between two images. Nasrabadi's[13] approach involves a curve segment-based matching algorithm that extracts curve segments from detected edge points, using their centers as features for the matching process.

On the other hand, Medioni and Nevatia's[14] method utilizes segments of connected edge points as matching primitives and emphasizes minimizing the differential disparity measure for global matching, considering factors like end points and segment orientation.

2.2.2.3 Area Based Disparity:

This method describes two techniques, both involving the use of a window placed on one image and scanning the other image using a window of the same size. The pixels within each window are compared and manipulated, with their sums producing a coefficient for the central pixel. These techniques were developed by Okutomi and Kanade [15]

The first technique is correlation, where the scanning window's output is convolved with the first, and the area that yields the highest convolution coefficient is considered the corresponding area.

The second method adheres to the same window principle but employs the sum of squared differences (SSD). This method assesses the pixel values in both windows and estimates the disparity by computing the SSD coefficients. Minimization of the SSD coefficient is the goal in this method.

2.3 Obstacle Avoidance

2.3.1 Mobile Robot Navigation:

Mobile robot navigation involves various technologies categorized into seven types:

2.3.1.1 Odometry:

This method is widely used for mobile robot navigation. It involves translating wheel revolutions into linear displacement [19]. It provides short-term accuracy but accumulates errors over time.

2.3.1.2 Inertial Navigation:

In this method, gyroscopes and accelerometers are used for measuring rotation rate and acceleration. However, data drifts over time, and challenges with signal-to-noise ratio occur in accelerometers [18].

2.3.1.3 Magnetic Compasses:

Magnetic compasses, which measure the Earth's magnetic field, serve as a fundamental source for obtaining absolute heading information. The fluxgate compass emerges as an advantageous choice for mobile robot applications as it exhibits characteristics such as low power consumption and the absence of moving parts [16].

2.3.1.4 Active Beacons:

Active beacons are common navigation aids utilized in determining a vehicle's position by measuring distances to known beacon sources. The two primary methods are trilateration and triangulation. While these methods offer robust navigation solutions, challenges exist in triangulation methods, with no single approach being universally suitable [17]. Intelligent combinations of methods may be required to overcome specific weaknesses.

2.3.1.5 Global Positioning Systems (GPS):

GPS stands as a revolutionary technology designed for outdoor navigation, originally developed by the Department of Defense. Comprising 24 satellites transmitting RF signals, GPS utilizes trilateration for position computation, offering global coverage.

Challenges, such as intentional errors through Selective Availability, are addressed by Differential GPS (DGPS) [20]. This correction method, involving a second nearby receiver, enhances accuracy to 4-6 meters in commercial GPS receivers. Despite its transformative impact, GPS faces challenges such as periodic signal blockage, multi-path interference, and limitations in accuracy for standalone navigation.

2.3.1.6 Landmark Navigation:

Landmark navigation hinges on identifiable features, categorized as either natural or artificial, guiding mobile robots in their environment [20]. These include distinctive

features like doors and wall junctions, often detected using computer vision for navigation. Designed for optimal contrast, artificial landmarks utilize various patterns and shapes for easy detection. Landmark-based navigation offers flexibility, yet it is dependent on environmental features. Challenges include accuracy, processing speed, lighting conditions, and initial knowledge of the robot's starting position.

2.3.1.7 Map Based Positioning:

This approach leverages naturally occurring structures without modifying the environment, allowing for learning, and improving accuracy through exploration. However, stringent sensor map accuracy requirements and the need for enough stationary, distinguishable features pose challenges. To capture all relevant features of a real environment, sensor fusion, combining data from different sensor modalities, becomes essential. Establishing correspondence between local and global maps is a crucial aspect of map-based navigation.[21]

2.3.2 Real Time Obstacle Avoidance Using Depth Maps:

Once depth maps have been created, the next step is obstacle-avoidance in real time. Here are 3 different algorithms for real-time obstacle avoidance using stereo vision.

2.3.2.1 Mean Estimation Method

The described method involves dividing disparity into three windows, left, central and right and calculating the average disparity for each window. *The window with the smallest average disparity value indicates the direction with fewer obstacles, guiding the robot's decision to steer in that direction.*[22]

In what conditions is this method ideal?

This method works efficiently when there are few obstacles in one of the windows.

Shortcomings of this method:

In cases where there's enough space in front of the robot to move before making a steering decision, the algorithm occasionally behaves hesitantly.

2.3.2.2 Threshold Estimation Method

The threshold estimation method also involves dividing a disparity map into three windows of pixels. It operates in the following manner:

In the central window, pixels with a disparity value exceeding a predefined threshold (e.g., T = 120) are counted. If this count falls below a predefined rate (e.g., 20%) of all central window pixels, it signals the absence of obstacles, permitting the robot

to proceed. If the count exceeds the rate, the algorithm inspects the other two windows and selects the one with the lowest average disparity value.

The threshold value (T) and rate (r) are tunable parameters that influence the algorithm's hesitance. A higher threshold value reduces hesitance, allowing the robot to approach obstacles more closely.

In what conditions is this method ideal?

The T and r parameters are especially useful in navigating confined spaces. For instance, when applied to a specific disparity map, the algorithm determines whether the robot can move forward and get within approximately 50cm of obstacles before changing direction.

2.3.2.3 The multi-thresholds method

The algorithm utilizes a modified version of the Threshold Estimation Method, employing disparity maps to estimate object distances. The disparity map is divided into three vertical windows, with the central window being larger to address camera angle limitations, and the side windows adjusted to mitigate noise at map borders. In the analysis, the algorithm calculates the percentage of disparity map pixels representing nearby objects in each window, considering pixels exceeding a predefined threshold as significant. Additionally, pixels lacking disparity information are factored in, with 30% of the weight of high-disparity pixels assigned to them for efficiency. The side areas are processed selectively [23]. The algorithm first analyzes the central area; if the percentage of high-disparity pixels falls below a set threshold, the road is considered obstacle-free, and the robot continues its path. If the threshold is surpassed, the algorithm compares the percentages of the side windows, selecting the window with the lowest value to determine the direction with the fewest obstacles, leading to a corresponding adjustment in the robot's path. This approach integrates the directional decision-making aspect of the Multi-Thresholds Method, which divides the disparity map into three windows and assesses terrain travers ability based on pixel counts exceeding a set threshold. The window with the lowest count is chosen, and if this count is below a predefined rate, the robot moves in that direction. If all three counts exceed the rate, the algorithm recognizes the terrain as non-traversable and suggests a 180-degree rotation to avoid collision. The key advantage of this integrated method lies in its comprehensive assessment of terrain

travers ability before making navigation decisions, aiding in collision avoidance in non-traversable terrains.

2.3.3 Addressing Wireless Transmission and Fixed Camera Constraints:

Limitations in the wireless transmission link and the fixed stereo camera's restricted field of view are challenges in real-time obstacle avoidance for autonomous vehicles. To overcome these constraints, a proposed method is the integration of sonar sensor data with stereo camera information. The Pioneer 3 sonar array, equipped with sensors on each side and six outward-facing sensors, is utilized in conjunction with an adaptive fuzzy control system. This system processes obstacle distances in the left-front, right-front, and front areas to determine left and right wheel speeds, enabling linear and circular motions for obstacle avoidance [24]. The approach streamlines the fusion of sensory inputs and employs the center of gravity method for clear signal conversion, enhancing the real-time obstacle avoidance capabilities of autonomous vehicles.

2.3.4 Managing Uncertainty in Autonomous Vehicle Navigation:

Uncertainty arises from factors such as sensor noise and the dynamic nature of the environment. Effectively dealing with uncertainty is crucial for ensuring the accuracy of AV state estimation, which includes parameters like position, velocity, and orientation.

The Extended Kalman Filter effectively integrates sensor measurements, such as those from GPS and IMUs, and handles nonlinear dynamics inherent in AV systems [25]. Operating in a prediction-correction fashion, the EKF continuously updates the state estimate, providing a refined and reliable representation of the AV's current state. By accounting for measurement noise and dynamically adjusting predictions, the EKF contributes significantly to the robustness and accuracy of AV navigation systems, making it a valuable tool for addressing uncertainty challenges in autonomous vehicle environments.

2.4 Mobile platform

2.4.1 Design Considerations:

The increasing demand for autonomous driving has spurred the deployment of mobile robots across various industries. In manufacturing, logistics, retail, and agriculture, mobile robots play a crucial role in enhancing efficiency and productivity [26–33]. Among these, four-wheel mobile robot platforms are frequently employed for stable navigation on uneven road surfaces and reliable cargo transportation with varying weights and shapes.

Four-wheel steering mechanisms, including skid-steering, two-wheel steering, and four-wheel steering, are commonly utilized. While skid-steering encounters wheel slippage and omnidirectional robots face complex issues, four-wheel steering platforms provide advantages such as a short turning radius. However, challenges like space occupation, power limitations, and control complexity need to be addressed [34–40].

Human-controlled four-wheel steering involves the use of steering units like joysticks for precise operation. However, applying traditional vehicle control methods optimized for high-speed travel to mobile robots operating at low speeds may lead to decreased steering performance. Complex steering algorithms also necessitate extensive experiments for parameter adjustment [41-42].

Recent research contributions have focused on enhancing the capabilities of four-wheel steering robotic platforms. These efforts aim to address limitations and improve functionalities. Notable examples include a reconfigurable robot platform designed for floor cleaning [43], a strategy for transitioning between various steering modes [44], and innovative control algorithms focusing on lateral and directional performance [45–47]. However, these efforts primarily target stability control in autonomous scenarios and lack considerations for compact four-wheel steering platforms for educational and research purposes.[48] proposes a compact platform featuring four independently steerable wheels with an integrated differential gear mechanism, making it suitable for diverse applications, including education, research, and algorithm validation. The adaptive steering algorithm aims to enhance vehicle maneuverability and optimize turning performance during manual operation. This approach eliminates the need for elaborate sensor arrays, complex dynamic models, or intricate optimization algorithms [48].

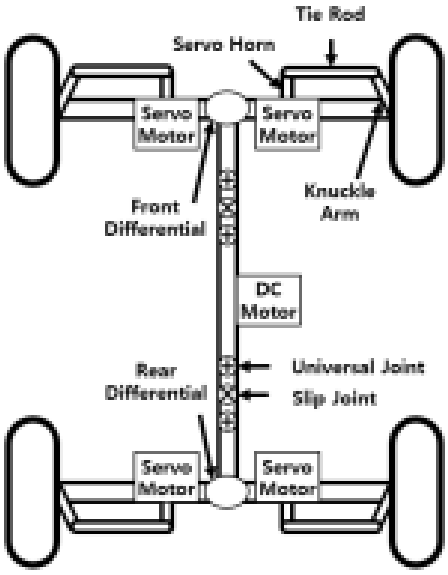

Figure 2 (The structure of robot module) [48]

Figure 3 (Overview of robot module) [48]

2.4.2 Adaptive Control of Four-Wheel Steering:

The adaptive steering control algorithm for manual operation utilizes a combination of front and rear steering to achieve agile maneuverability. Analyzing the target steering angle, the algorithm dynamically adjusts the steering configuration for optimal performance.

The algorithm considers a range of steering angles ($-2\theta max \leq \theta cmd \leq 2\theta - 2\theta max \leq \theta cmd \leq 2\theta max$) where θmax is the maximum steering angle for one wheel. The algorithm selectively activates front steering for straight or slight turns, engaging rear steering when the desired

steering angle surpasses a predefined threshold. This results in a reduced turning radius, enhancing maneuverability [48].

The overall process of the adaptive steering control algorithm is illustrated, demonstrating paths of the instantaneous center of rotation (ICR) for both front wheel steering and combined front and rear wheel steering. The algorithm calculates steering angles for right and left turns based on the target steering angle, ensuring optimal performance [48].

$$\theta_{FR} = \begin{cases} \theta_{cmd}, & -\theta_{max} \leq \theta_{cmd} < 0 \\ -\theta_{max}, & -2\theta_{max} \leq \theta_{cmd} < -\theta_{max} \end{cases}$$

$$\theta_{RR} = \begin{cases} 0, & -\theta_{max} \leq \theta_{cmd} < 0 \\ -\theta_{max} - \theta_{cmd}, & -2\theta_{max} \leq \theta_{cmd} < -\theta_{max}. \end{cases}$$

2.5 Summary:

This chapter explores the work done in Stereo Vision, a technology that mimics human vision by combining multiple 2D images to create 3D depth maps, important for autonomous vehicles. It looks at different algorithms for depth perception, including local, global, and semi-global stereo matching. The chapter explains methods like triangulation and disparity calculation and discusses obstacle avoidance techniques, such as mean estimation, threshold estimation, and multi-thresholds for real-time navigation. It also addresses challenges like wireless transmission issues and uncertainty in navigation, suggesting solutions like combining sensor data and using the Extended Kalman Filter. The chapter wraps up with design considerations for mobile platforms, highlighting adaptive control of four-wheel steering systems to improve maneuverability in autonomous driving applications.

Chapter 3 -DEPTH PERCEPTION

3.1. Overview:

This chapter explains the making and working of depth perception algorithms by making of disparity maps followed by depth maps and getting a usable numerical value from that depth map. Further, it also highlights the hardware and software used for the project such as the camera specifications which are essential in determining the accuracy of the algorithm. This chapter also sheds an extensive light on the steps of algorithm generation and all the necessary action took to achieve the final accurate algorithm.

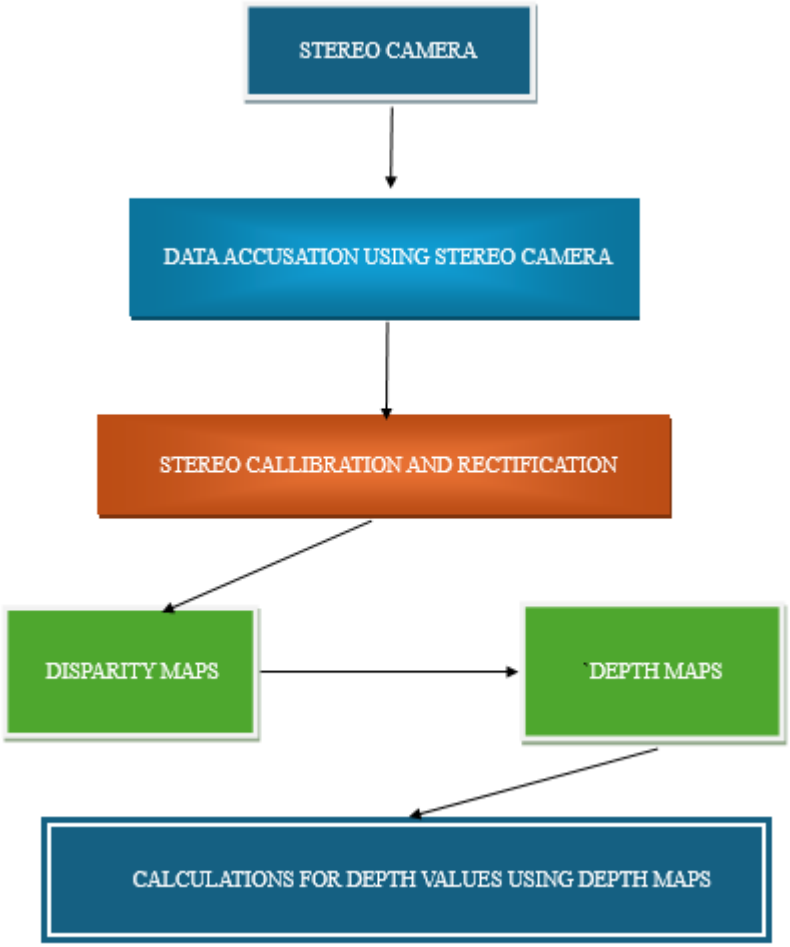

Figure 4 Flow Chart for Algorithm

3.2 Hardware Specifications:

The hardware is an integral part of our final year project as all the data acquisition for algorithm generation has been done from the hardware.

3.2.1. Stereo Camera:

The most important equipment of hardware used in the projects is the ZED 2 Stereo Camera, developed by StereoLabs, pioneer in the field of stereoscopic imaging. ZED 2 offers unparalleled depth perception capabilities in a compact and versatile form factor. This stereo camera system leverages advanced algorithms and hardware components to capture high-resolution depth maps with exceptional accuracy, making it an ideal solution for a wide range of applications including robotics, autonomous navigation, augmented reality, and 3D mapping.

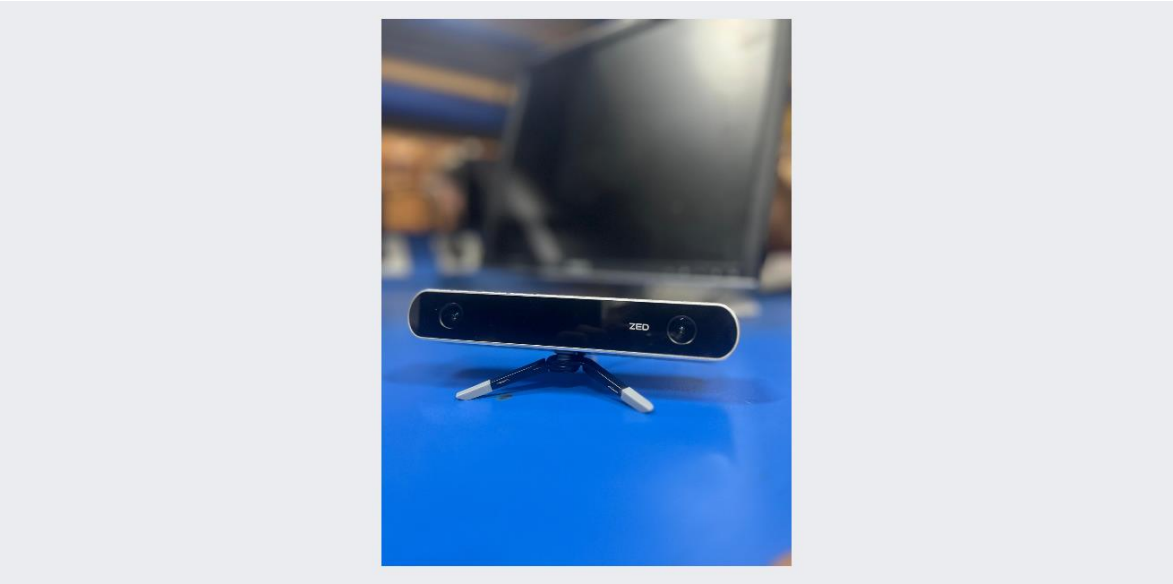

Figure 5. ZED 2 Stereo Camera, developed by StereoLabs

The ZED 1 features two synchronized cameras that capture images simultaneously from slightly different perspectives, mimicking the human binocular vision system. It is equipped with 720p HD cameras, the ZED 2 produces sharp and detailed stereo images, enabling precise depth perception and scene understanding.

The ZED 1 processes depth data in real-time, providing instantaneous feedback for applications requiring quick decision-making, such as obstacle avoidance or object tracking. By analysing the disparities between corresponding pixels in the left and right camera images, the ZED 1 calculates depth maps with impressive accuracy, even in challenging lighting conditions and complex environments.

Table 1 Properties of ZED Stereo Camera

Dimensions	175x30x30
Weight	170 g
Power	380mA /5V USB powered
Operating temperature	0° C to 45°C
Baseline	120 mm (4.7")
Depth range	0.5 m to 25 m
Depth Map resolution	Native video resolution (in Ultra mode)
Depth Accuracy	<2% up to 3m <4% up to 15 m
Output resolution	Side by Side 2x (2208x1242) @ 15fps 2x(1920x1080) @ 30fps 2x(1280x720) @60fps 2x (672x376) @ 100fps
Field of View	Max. 90° (H) X 60° (W) X 100° (D)
RGB sensor type	1/3" 4MP CMOS
Active array Size	2688 X 1520 pixels per sensor (4MP)
Focal length	2.8mm
Interface	USB 3.0 - Integrated 1.5m cable
Shutter	Electronic synchronized rolling shutter

3.2.2. Jetson Nano:

` Another Crucial Piece of hardware equipment is Jetson Nano o which all the processing of stereo camera has been done. The Jetson Nano is a small, powerful, and cost-effective computer module developed by NVIDIA, specifically designed for AI and robotics applications. It combines the performance of a GPU with the flexibility of a CPU in a compact form factor, making it an ideal platform for edge computing and embedded systems.

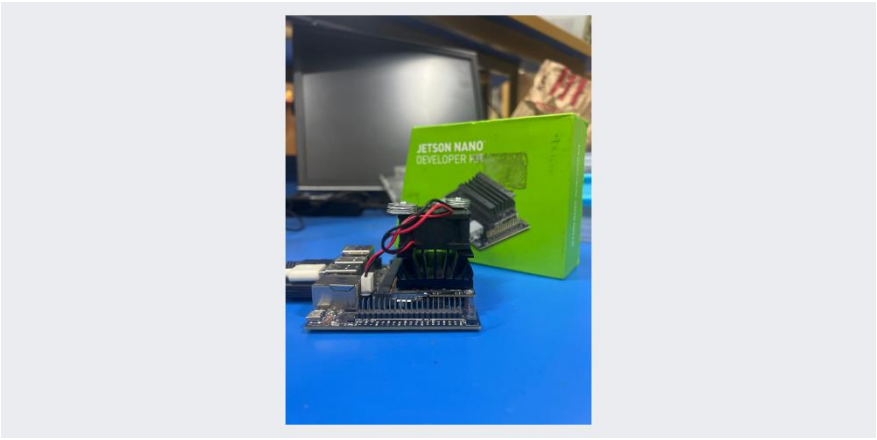

Figure 6 Jetson Nano

The Jetson Nano integrates a powerful GPU based on the NVIDIA Maxwell architecture, which provides significant computational power for AI inference tasks such as image recognition, object detection, etc. In addition to the GPU, the Jetson Nano features a quad-core ARM Cortex-A57 CPU.

Despite its high performance, the Jetson Nano is designed to operate efficiently with low power consumption, making it suitable for battery-powered and mobile applications. the ZED 2 is compatible with Stereolabs' proprietary software suite, which includes tools for depth mapping, 3D reconstruction, and camera calibration.

3.3. Software's and Libraries:

Multiple Software's and Libraries were used during the course this project, The details of the Software's are as follows,

1) **ZED SDK:** The official SDK Manager for the ZED Stereo Camera from Stereolabs was used for the calibration and getting the rectification parameters of the stereo camera. Its a useful software as it also provides the initial pre-checking of the camera frames with the camera launcher. Further, have built in chess pattern and camera frame detection which makes the calibration of the Stereo Camera and is easily calibrated. Also, it provides the various rectification parameters which are later used for the rectification of the camera frames.
2) **Anaconda:** This software is for Windows and it helps you use Python and R programming languages. It comes with lots of useful tools for data science, like NumPy, Pandas, and Matplotlib. You can create virtual environment to work in without messing up your computer, and you can install different versions of tools

without them causing problems. In this project, it is used for the software based coding and initial making of the disparity and depth maps o windows environment

- **Spyder:** Spyder IDE provided by anaconda is used for the Python programming of Depth Perception and Obstacle Avoidance code in windows environment. All the initial coding is done on Spyder IDE. It is one of the most useful and easy to use IDE for python as it provides supports for majority of the libraries of python such opencv, numpy, time, etc.

3) **Code-OSS/Visual Studio Code:** Code-OSS also know as Visual Studi Code is used in this project for the embedded implementation of the project in Ubuntu 18.04 on Jetson Nano. Code-OSS is the software name of VS Code in the Linux environment. VS Code makes coding easier with its user-friendly interface and powerful features like IntelliSense and debugging tools. Its extensive library of extensions enhances productivity by customizing the editor to suit individual preferences and project needs. VS Code makes coding easier with its user-friendly interface and powerful features like IntelliSense and debugging tools. Its extensive library of extensions enhances productivity by customizing the editor to suit individual preferences and project needs.
4) **Python:** The coding language used in this project is Python, for it's simplicity and ease of use, allowing for faster development and prototyping. Python's extensive libraries, such as OpenCV, provide robust support for image processing and computer vision tasks without the need for complex memory management, making it more accessible to developers. Additionally, Python's flexibility and readability contribute to quicker algorithm implementation and experimentation, making it a popular choice for computer vision projects.

- **OpenCV:** Open-Source Computer Vision Library, is a powerful tool for image and video processing tasks, offering a wide range of functionalities for computer vision applications. Its extensive collection of algorithms enables tasks like object detection, recognition, tracking, and even augmented reality. OpenCV's cross-platform compatibility allows developers to work seamlessly across different operating systems, making it a

versatile choice for various projects. In this project, OpenCV is used to display and extract data from stereo camera and to display the depth maps once calculation using algorithm.

- **Ultralytics:** The Ultralytics library is a comprehensive toolkit for computer vision research and development, providing state-of-the-art models and algorithms for various tasks. Its user-friendly interface and extensive documentation make it accessible for both beginners and experienced developers alike. With features like object detection, segmentation, and image classification. In this project, it is used to run YOLO and for person and obstacle detection.
- **NumPy:** NumPy is a fundamental library for numerical computing in Python, offering powerful tools for working with arrays and matrices. Its efficient implementation of mathematical operations allows for high-performance computation, making it indispensable for scientific and engineering applications. In this project numpy is used to deal with arrays of images and computations used for it.

3.4 Algorithm Generation:

3.4.1. Data Acquisition from Stereo Camera:

To acquire data from stere camera, ZED SDK was first used to initially monitor the frames of the camera and to adjust and calibrate them accordingly. Later, OpenCV was used in the code to get the two frames of stereo camera separately to be further used in depth maps generation. However, the stereo camera can also act as a USB Video Class 19 (UVC) camera, from which 'A_1' and 'A_2' can be taken, then those images are manually rectified using the calibration parameters of the stereo camera giving 'A_1' and 'A_2'.

3.4.2. Stereo Camera Calibration:

Camera calibration is most important part of depth perception and depth map generation. Without the camera calibration it is not possible to obtain accurate depth maps for real time processing and obstacle avoidance.

The ZED Stereo Camera is calibrated using the Zhang's Method. There are two ways for calibration of stereo camera using Zhang's method i.e. using the code or the built-in application. The ZED SDK manager provided by the Stereolabs is used for the calibration of the stereo camera by the following method. In this method, a checkboard pattern is displayed on the computer screen in the ZED SDK Manager. The ZED camera is then adjusted Infront of screen until application detects the camera by changing red highlighted area to blue. The camera then take pictures of the checkboard pattern using two cameras separately at various orientations and angles, then compare the feature points of two images with each other. The camera gets calibrated by adjusting the frames of two separate camera with each other and displays all the parameters of the ZED stereo camera. After getting the parameters, images from the stereo camera are rectified [49].

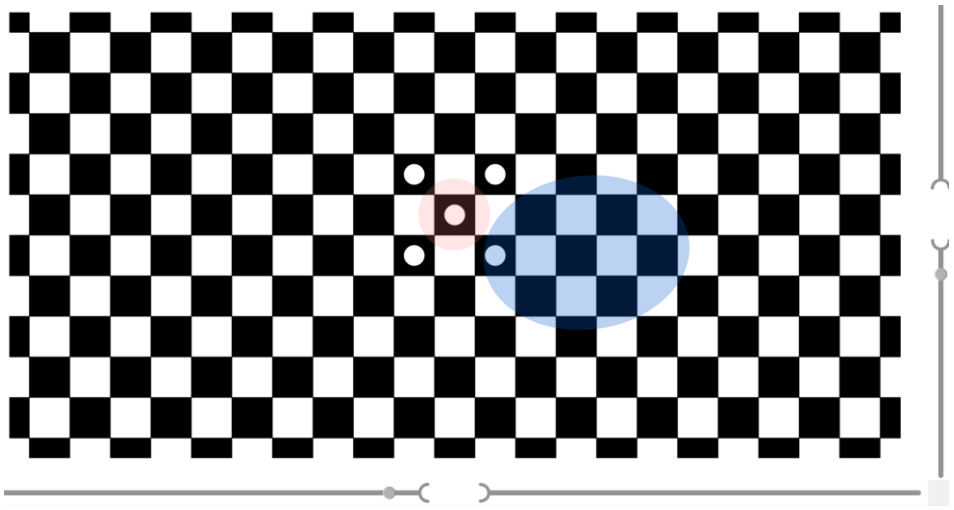

Figure 7 Checkboard Pattern for Calibration

The Intrinsic and extrinsic parameters for the ZED Stereo Camera are defined below.

a) Intrinsic Properties:

Intrinsic properties refer to the inherent qualities of an object or system that are independent of its external environment or orientation. These properties are essential characteristics that define the object's nature and behavior.In computer vision, intrinsic properties often relate to the internal geometry and appearance of objects, including parameters such as color, texture, shape, and reflectance properties. Intrinsic properties are typically invariant under transformations such as translation, rotation, and scaling, making them useful for tasks like object recognition, classification, and segmentation.

Some Intrinsic properties of ZED Stereo Camera are as follows.

- **Focal Length:** This is the distance between the camera sensor and the centre of the lens. It is usually expressed in pixels or millimetres.

- **Image Centre:** It is the point where the optical axis of the camera intersects the image plane. It is usually expressed in pixels.
- **Distortion Coefficients:** Distortion occurs when a camera lens fails to produce a straight line when imaging a straight object. There are two types of distortion, radial and tangential. The radial distortion coefficients are related to how much the lens curves and are usually denoted by k1, k2, and k3. The tangential distortion coefficients are related to how the lens is not aligned perfectly parallel to the image plane and are usually denoted by p1 and p2.
- **Rectification Parameters:** The rectification parameters are used to transform the images so that they can be processed more easily in stereo vision algorithms. The rectification parameters include the rotation matrix and translation vector between the two cameras, the projection matrix for each camera, and the disparity-to-depth mapping function.

These properties are extremely important for the calibration of the ZED Stereo camera and for rectification of stereo camera at later steps.

b) Extrinsic Properties:

Extrinsic properties, on the other hand, are dependent on the object's relationship with its external environment or reference frame. These properties describe the object's position, orientation, or motion relative to some external context. In computer vision and robotics, extrinsic properties often include parameters such as pose (position and orientation), motion trajectory, and spatial relationships with other objects in the scene. Unlike intrinsic properties, extrinsic properties can change with respect to external factors or transformations, making them crucial for tasks like scene understanding, navigation, and robot localization.

- **Rotation Matrix:** The rotation matrix specifies how one camera is rotated relative to the other camera. It is usually denoted by R.
- **Translation Vector**: The translation vector specifies the distance and direction between the two cameras. It is usually denoted by T.
- **Essential Matrix:** The essential matrix relates the corresponding image points in the left and right camera images. It is computed from the rotation matrix and translation vector and is denoted by E.
- **Fundamental Matrix:** The fundamental matrix describes the epi-polar geometry of the stereo camera system. It relates the corresponding image points in the left and right camera images and is denoted by F.

These both Intrinsic and Extrinsic properties are very much important in the later rectification of the camera frames and also the 3D reconstruction of the objects.

3.4.3. Stereo Rectification:

Stereo camera rectification is a crucial preprocessing step in stereo vision applications, aimed at simplifying stereo correspondence by aligning the epipolar lines in the left and right images. Here's a breakdown of the steps involved:

1. **Understanding Stereo Geometry:** Before diving into rectification, it's essential to comprehend the geometric relationship between the two camera views. This includes understanding concepts like the baseline, focal length, and disparity.

2. **Compute Camera Calibration:** Ensure both cameras are calibrated to obtain their intrinsic and extrinsic parameters. This information is vital for rectification.

3. **Epipolar Geometry Analysis:** Analyse the epipolar geometry, which defines the relationship between corresponding points in the left and right images. Epipolar lines are the intersection of the image plane with the planes passing through corresponding points and the camera centres.

4. **Compute Rectification Transformation:** Based on the camera calibration and epipolar geometry, compute the rectification transformation matrices for both cameras. These matrices transform the images such that corresponding epipolar lines become horizontal and aligned.

5. **Apply Rectification:** Apply the rectification transformation to both left and right images. This process involves warping the images according to the computed transformation matrices.

6. **Verify Alignment:** After rectification, verify that corresponding epipolar lines are indeed aligned horizontally in both images. This alignment simplifies stereo matching, as corresponding points will now lie on the same rows.

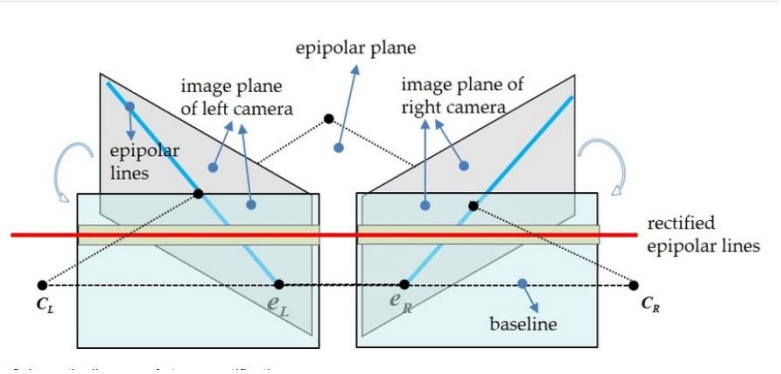

Figure 8 Epipolar Lines [50]

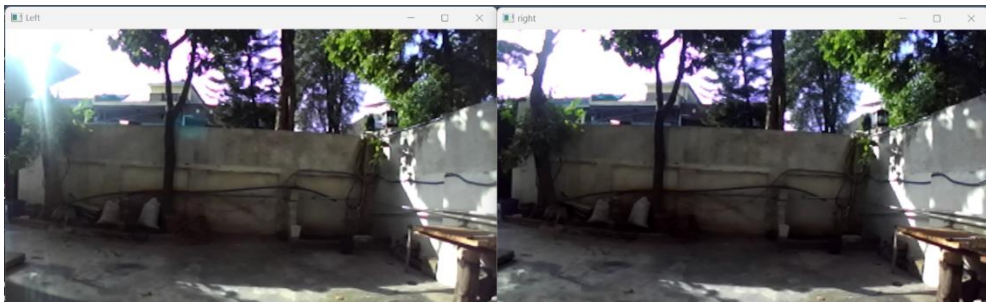

Figure 9 Alignment of frames

3.4.4. Disparity Maps Formulation and Calculation:

Disparity Maps forms the basis of the depth map generation and for calculating depth of the object using stereo camera. There are many algorithms and method used for the formulation od Disparity Maps, we have Semi-Global Block Matching (SGBM) Algorithm for the calculation of Disparity Maps.

Implementing SGBM involves careful consideration of each step and parameter selection to achieve accurate and reliable disparity estimation for stereo vision applications. The steps involved in calculating disparities using SGBM is given as follows.

1. Image Preprocessing:

Image Rectification: This is first preprocessing step for the disparity calculation using the stereo vison, which already have been done and explained already in our project. In sort, it means to ensure that the left and right images are rectified, so corresponding points lie on the same rows. This simplifies stereo matching.

Convert to Grayscale: The rectified image is then converted to grayscale as colour information is not typically required for stereo matching.

2. Block Matching:

Block Selection: Divide the left image into small blocks or windows. These blocks will be matched with corresponding regions in the right image.

Cost Calculation: Cost calculation in Semi-Global Block Matching (SGBM) algorithm is pivotal for stereo correspondence. It quantifies the dissimilarity between pixel blocks in left and right images across different disparities. Employing methods like Sum of Absolute Differences (SAD) or Sum of Squared Differences (SSD), it measures pixel intensity deviations. These costs aid in identifying optimal correspondences, crucial for accurate depth estimation and 3D reconstruction in stereo vision systems.

For each block in the left image, the cost of matching with all possible blocks in the same row of the right image is calculated. The cost calculation using Sum Of Absolute difference (SAD) is given as.

$$\mathbf{SAD(p,q)} = \sum_{(x,y)\in W} |f_1(x,y) - f_2(x-p,y-q)|$$

Similarly, Calculation using Sum of Square Difference is given as.

$$\mathbf{SSD(p,q)} = \sum_{(x,y)\in W} (I_1(x,y) - I_2(x-p,y-q))^2$$

3. Aggregate Costs:

Cost Aggregation: In Semi-Global Block Matching (SGBM), cost aggregation amalgamates local matching costs across multiple paths, enhancing disparity estimation. Techniques like dynamic programming smooth the cost volumes by aggregating costs along rows, columns, and diagonals. Aggregated costs provide a global understanding of the scene, reducing disparities and improving depth map accuracy. Through aggregation, SGBM minimizes noise and artifacts, enabling more robust stereo matching in complex environments.

A cost aggregation technique is applied to smooth the cost volume and improve the accuracy of disparity estimation. Common methods include cross-based cost aggregation or adaptive support weight aggregation.

We have used Cross-Based Aggergation technique for calculating the Cost Aggrgation.

$$\mathbf{C(p,q,d)} = \sum_{(i,j)\in N(p)} \min(C(i,j,d), C(i-p+\Delta_x, j-q+\Delta_y, d))$$

Semi-Global Aggregation: In Semi-Global Block Matching (SGBM), semi-global cost aggregation enhances disparity estimation by considering global consistency. Through dynamic programming, costs are aggregated along multiple paths, including rows, columns, and diagonals.

4. Disparity Calculation:

Winner-Takes-All (WTA): For each pixel in the left image, sthe disparity value with the lowest aggregated cost as the estimated disparity is calculated.The formula for this is given.

$$\mathbf{Disparity(p,q)} = \arg\min_d C(p,q,d)$$

Subpixel Refinement: Subpixel refinement in Semi-Global Block Matching (SGBM) enhances the accuracy of disparity estimation beyond integer values. Using techniques like parabolic interpolation, disparities are refined to subpixel precision.

We have used parabolic interpolation in our algorithm whose formula is given as.

$$d' = d + \frac{C(p,q,d-1) - C(p,q,d+1)}{2(C(p,q,d-1) - 2C(p,q,d) + C(p,q,d+1))}$$

3.4.5. Depth Maps from Disparities:

After getting the disparity map from the previous step, the triangulation formula for depth is used to convert the disparity map into a proper depth map where 24 f is the focal length of the ZED in pixels, u is the disparity map and baseline is the distance between the two cameras of the ZED 1 [51].

$$\mathbf{Z} = \frac{focal\ length * Baseline}{Disparity}$$

3.4.6. Getting Depth Values from Depth Maps:

To extract depth values from depth maps, each pixel's intensity in the depth map represents the distance from the camera to the corresponding point in the scene. By converting pixel values to metric units using calibration parameters, such as focal length and baseline distance, depth values can be obtained. Additionally, post-processing techniques like median filtering or outlier removal can refine depth values for enhanced accuracy. These depth values provide crucial spatial information, enabling precise 3D reconstruction and depth-based analysis in stereo vision applications.

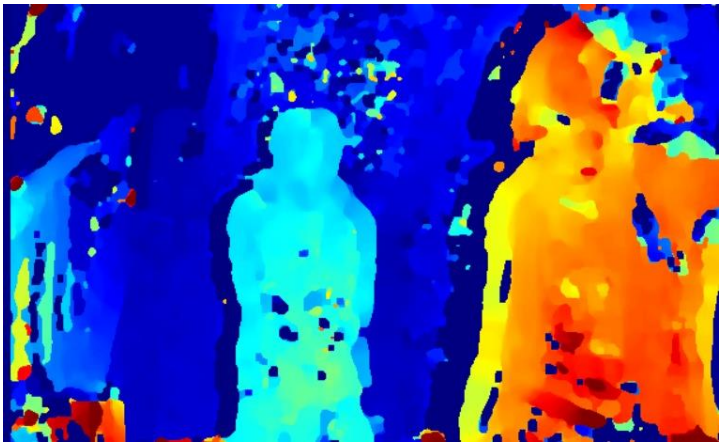

Figure 10 Depth Maps

3.4.8 Implementation on Jetson Nano:

The process of deploying the framework to the Jetson Nano platform involved the following steps.

1) The hardware requirements of the Jetson Nano were tested and set up.

2) The necessary libraries were installed on the system, and it was run on the Code-OSS IDE environment on Jetson Nano.

3) After installing the required libraries, the algorithm was optimized to run on the limited resource of the Jetson Nano,

4) This optimized code was then finally deployed on Jetson enviorment.

3.5. Summary:

This chapter describes the framework used in this project, after following the above given procedures, the key outcomes are as follows.

- Calibrated and rectified stereo camera images were obtained from uncalibrated images.
- Disparity Map was obtained from rectified stereo camera images.
- Depth map was obtained from disparity map.
- Depth values were calculated of the obstacle from the generated Depth Maps

The Following Chapter will discuss the recognition of Obstacles and integrating the Depth maps with Reign if Interest of that Obstacle and the generation of a Obstacle Avoidance Strategy using the perceived Depth of the Obstacle as described and calculated in this chapter.

Chapter 4 – OBSTACLE AVOIDANCE

4.1. Overview:

Obstacle avoidance in autonomous vehicles is a critical element of their navigation systems, designed to ensure both safety and efficiency during operation. These vehicles are equipped with a variety of sensors such as cameras, radar, and LIDAR, which collectively enable precise detection and assessment of the surrounding environment. Advanced algorithms analyze this data to calculate distances, anticipate the behavior of other road users, and perform necessary maneuvers to safely avoid potential hazards. This technology not only enhances vehicular safety but also contributes significantly to the goal of full vehicular autonomy. It enables cars to navigate through complex and dynamic traffic scenarios with minimal human intervention, thus increasing the reliability and effectiveness of autonomous travel.

In the context of our project, obstacle avoidance was a major challenge since we had to work only with depth maps generated by stereo cameras. Depth maps, which provide 3D representations of the environment from two-dimensional images, are crucial for identifying and responding to physical barriers in the vehicle's path. However, the accuracy of depth maps can be affected by various factors including lighting conditions, object distance, and the inherent limitations of the stereo vision technology.

In order to find the most reliable results, we implemented several strategies for our depth-based obstacle detection system. The exploration of these techniques provided valuable insights into the capabilities and limitations of using depth maps for obstacle avoidance in autonomous vehicles.

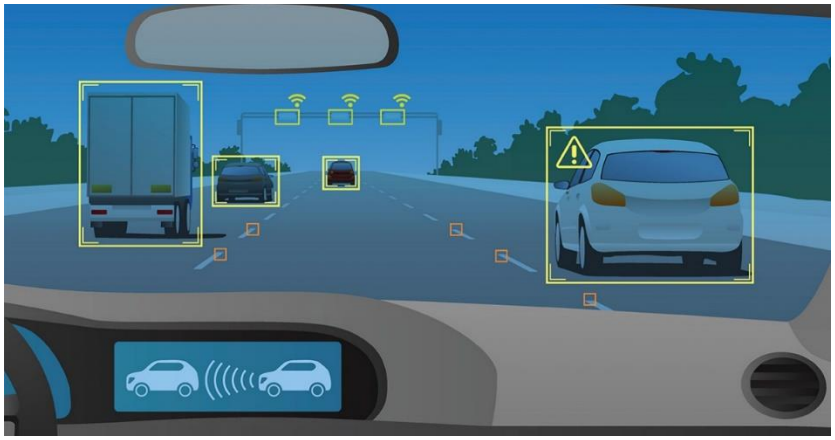

Figure 11 Autonomous Vehicle demo [61]

4.2. Detection of Obstacles: Methodologies and Challenges

4.2.1. Method I - Depth Map Division

The first method we explored was obstacle detection through the strategic segmentation of depth maps into five specific windows. For this division, the frame width was calculated and then divided into 5 parts. This technique was aimed at analyzing the depth within each section to detect potential obstacles effectively. By calculating the average depth for each segmented window and setting a comparison against a predetermined threshold, we were able to identify areas likely to contain obstacles. This threshold was not arbitrary but was established based on empirical data and expert assessments to ensure its relevance and accuracy in real-world conditions.

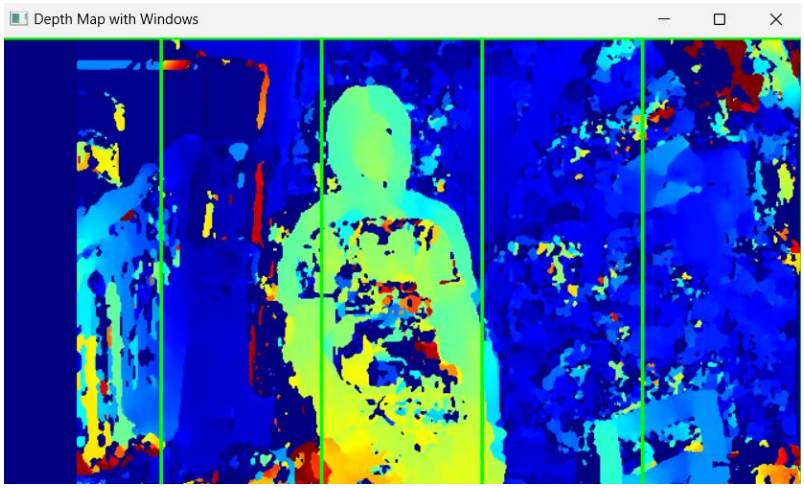

Figure 12 Depth Maps with Windows

```
width = frame.shape[1]
window_width = width // 5
windows = {
    'Left-most': (0, window_width),
    'Left': (window_width, 2 * window_width),
    'Center': (2 * window_width, 3 * window_width),
    'Right': (3 * window_width, 4 * window_width),
    'Right-most': (4 * window_width, 5 * window_width)
```

Figure 13 Code Screen Shot

4.2.1.1. Comparative Analysis and Directional Assessment

Upon identifying multiple windows that exceeded the obstacle threshold, we proceeded with a comparative analysis. This involved an in-depth evaluation to determine which window not only met the threshold but contained the closest obstacle based on its depth reading. Prioritizing obstacles based on their proximity is critical as it directly impacts the decision-making process regarding the vehicle's navigation and immediate actions required. Furthermore, the direction of the vehicle’s movement was under continuous surveillance. Integrating directional data with feedback from the depth

maps allowed for dynamic adjustments. This proactive approach ensured that the vehicle could adeptly navigate and maneuver around obstacles, enhancing both safety and efficiency in operation.

4.2.1.2. Drawbacks and Limitations

However, during testing, we encountered significant challenges with the accuracy of the depth computations. Background depth data frequently contaminated the readings within each window, complicating the distinction between true obstacles and mere depth anomalies in the vehicle's path. This limitation was critical as it directly affected the reliability of the obstacle detection process, potentially leading to unsafe navigation decisions. The difficulties highlighted the necessity for more advanced computational techniques to discriminate between relevant and irrelevant depth information more effectively.

4.2.2. Method II – Object Detection

To overcome the issues faced in method I, we shifted to object detection for detecting obstacles. By combining the left and right frames of the stereo camera and using ultralytics and YOLO [52], we managed to detect obstacles in real-time.

4.2.2.1. Reasons behind choosing YOLO (You Only Look Once)

YOLO represents a significant shift in the landscape of object detection algorithms by emphasizing unparalleled speed and accuracy in a singular evaluation framework. This innovative method stands out by processing images in one swift computational pass, directly translating image data into spatial bounding boxes and associated class probabilities.

YOLO's architecture allows for **rapid image processing**, a critical feature that supports real-time applications. By treating object detection as a single regression problem, where the system predicts multiple bounding boxes and class probabilities across the entire image simultaneously, YOLO achieves remarkable speeds [53]. In addition, its approach differs from traditional methods that use separate systems to predict regions and then classify each region independently, as YOLO integrates these steps into one continuous prediction mechanism, reducing processing time and improving the efficiency of the detection system.

With its capability to generalize across different scenes, YOLO demonstrates strong performance even in new or varied environments [54]. This robustness makes it

particularly useful for applications that require the algorithm to adapt to different operational contexts without needing extensive retraining. So, overall, the algorithm's efficiency ensures that vehicles respond in real-time to objects like pedestrians, other vehicles, or unexpected obstacles.

4.2.2.2. Finding Region of Interest

For our prototype, we decided to specifically detect humans and mark them as obstacles for avoidance. After detecting objects with YOLOv4-tiny, the system searches for people and then draws bounding boxes around each of them. This marked area, or region of interest, highlights where the vehicle needs to focus its attention. Once a person is detected and the ROI is set, our system uses this information to avoid collisions. The coordinates of the bounding box help the vehicle understand where the person is and adjust its path accordingly. If there's more than one obstacle, then each detected object is enclosed within a bounding box, and for each bounding box, a corresponding region of interest is defined. The work by Liu et al. (2020) describe similar methods, where bounding boxes are used to refine the focus on specific areas for depth estimation in cluttered environments. This technique allows for more precise measurements by isolating the object from its surrounding context, which is crucial especially for small or partially obscured objects [55].

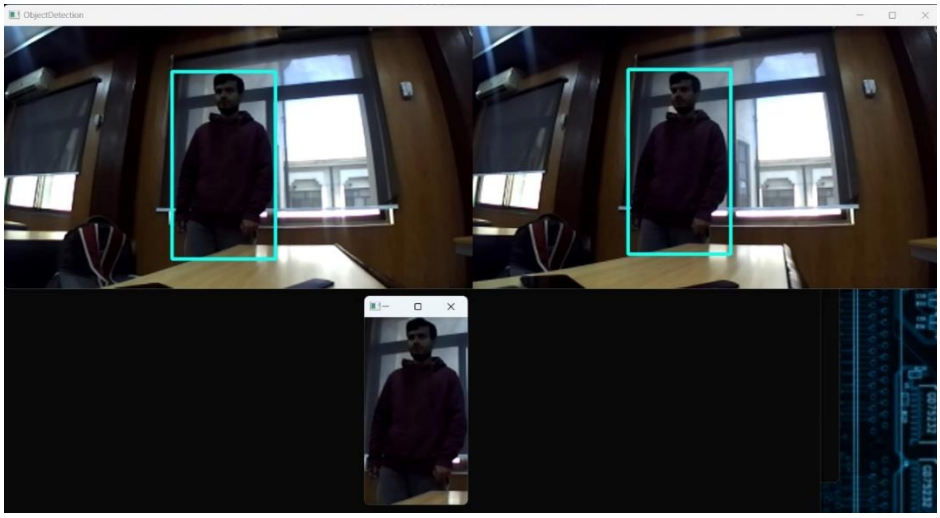

Figure 14 Obstacle Detection and Selection of ROI

4.3. Computing Depth of ROI

In the process of obstacle detection and avoidance in autonomous vehicle systems, accurately determining the depth of objects within the region of interest (ROI) is essential. This section details the methodology employed to compute the depth of each ROI, leveraging previously generated depth maps.

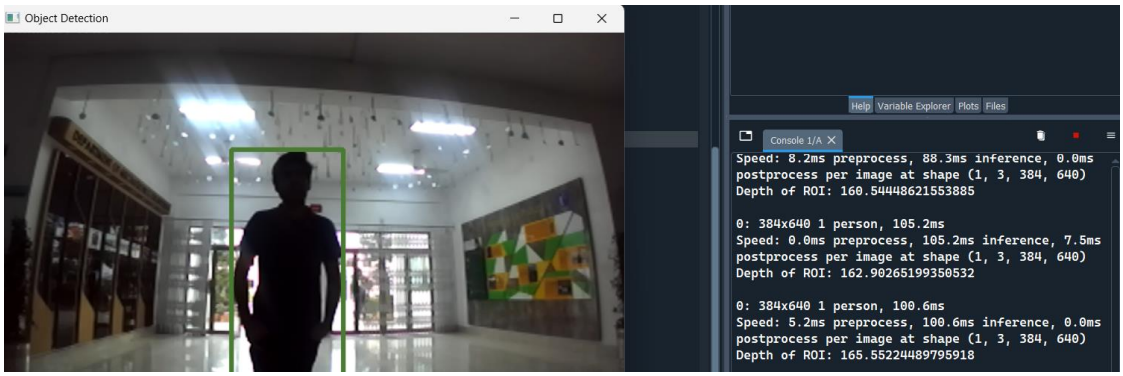

Figure 15 Computing Depth of ROI

4.3.1. Utilization of Depth Maps:

Depth maps that we had already computer are utilized here as foundational data. These maps offer detailed pixel – by – pixel estimates of disparity [56]. Disparity refers to the difference in the images of an object seen by two cameras from slightly different viewpoints. It is a crucial metric as it is inversely proportional to the actual distance from the camera to the object in the scene; greater disparity corresponds to closer objects.

Figure 16 Selection of Multiple ROI

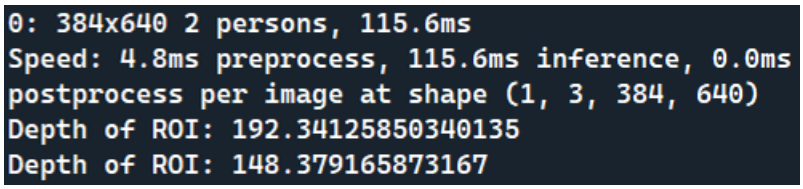

Figure 17 Depth of ROI

4.3.2. Average Disparity Calculation:

For each defined ROI, the average disparity is calculated. This involves summing the disparity values across all pixels within the ROI and then dividing by the total number of pixels to obtain a mean disparity value for that particular region.

4.3.3. Depth Calculation from Disparity:

Once the average disparity is obtained, the actual depth of the object or obstacle within the ROI is computed using the camera's intrinsic parameters. The formula to calculate depth D from disparity d involves the focal length f of the camera and the baseline distance b between the stereo cameras, given by the relationship:

$D=(f\times b)/d$

The focal length represents the distance over which the rays of light converge to form a sharp image, while the baseline distance is the horizontal distance between the stereo cameras.

This comprehensive approach ensures a reliable estimation of the depth of each obstacle. The accuracy of this depth calculation is paramount, as it directly influences the vehicle's ability to respond appropriately to its surroundings, ensuring safety and effective navigation.

4.4. Avoiding Obstacles based on Depth of ROI

After computing ROI and the obstacle's depth, the next step was locating the obstacles when they are in the path of the mobile base and based on that, decide the direction in which the base should move.

4.4.1. Window-Based Navigation Strategy

This approach involves segmenting the field of view into several discrete windows, typically five, focusing on key areas to enhance navigation decisions.

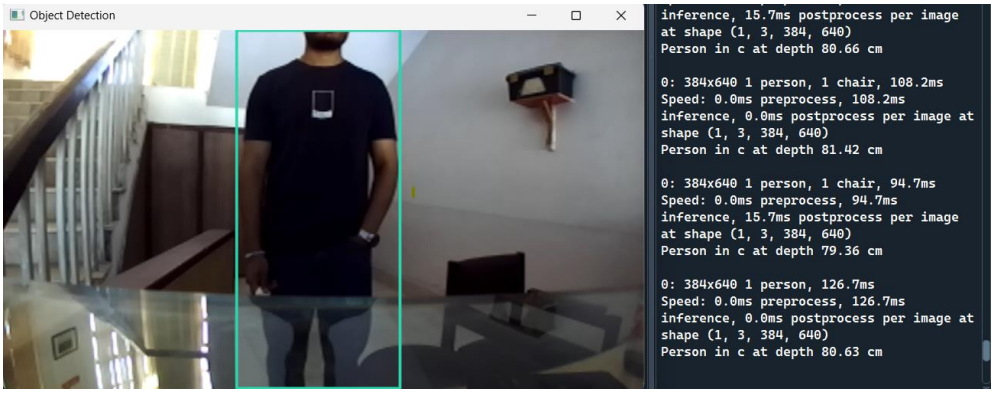

Figure 18 Window-Based Navigation Strategy

4.4.1.1. Division of the Field of View

The entire field of view is subdivided into five distinct windows. These windows are strategically positioned to cover essential viewing angles, namely the left, center, and right windows. This segmentation is vital for concentrating computational resources on

areas of highest navigational importance, particularly when the robot or vehicle is moving forward.

4.4.1.2. Finding Region of Interest

The rationale for focusing primarily on the left, center, and right windows stems from the need to manage computational load effectively. According to Chen et al. (2019), implementing a selective attention mechanism can significantly increase navigation efficiency. This is achieved by minimizing the real-time processing of peripheral visual data, which is often less relevant to immediate navigation tasks [57]. This targeted approach ensures that the system remains responsive and agile, avoiding delays that could compromise operational safety and efficiency.

4.4.1.3. Dynamic Obstacle Detection and Avoidance

In scenarios where the robot detects an obstacle within the central window—particularly when the trajectory is set straight ahead—the system then evaluates the adjacent left and right windows. This step is crucial for determining the best possible alternative path to avoid the obstacle. The decision-making process here is reinforced by the work of Kim and Uthansakul (2021), who demonstrated that multi-window approaches are particularly effective in dynamic environments where obstacles may suddenly appear or move [58]. By comparing the data from both side windows, the system can choose the most unobstructed path, enabling the robot to navigate safely around obstacles.

This window-based navigation strategy not only streamlines the processing of visual data but also enhances the robot's ability to make quick, informed decisions in complex environments. The methodology balances the need for detailed environmental scanning with the necessity for rapid response times, making it an essential feature of modern autonomous navigation systems.

4.5. Feedback Loop and Dynamic Adjustment

In autonomous navigation systems, the ability to dynamically adjust to changes in the environment is critical for operational efficiency and safety. This adaptability is facilitated through a well-structured feedback loop that uses real-time data to continuously refine the robot’s movement strategies.

4.5.1. Integration of Feedback in Navigation

The primary function of the feedback loop is to utilize the positions of detected obstacles within the designated windows as inputs. As the robot or vehicle progresses through its environment, it continuously scans for obstacles. When an obstacle is detected, the information about its location is immediately fed back into the system's control loop. This loop acts as a dynamic regulator, adjusting the robot's trajectory based on the most recent data.

4.5.2. Dynamic Trajectory Adjustment

The feedback mechanism allows the robot to make immediate adjustments to its trajectory to avoid collisions. This process is dynamic, meaning that the adjustments are continuously updated to reflect new data from the robot's sensors. The trajectory corrections are calculated to optimize path efficiency while avoiding detected obstacles, ensuring a smooth navigation process.

4.5.3. Role of Real-Time Feedback

The effectiveness of this system hinges on the real-time nature of the feedback. As highlighted by Nguyen et al. (2020), real-time feedback is a cornerstone of efficient autonomous navigation systems [59]. The immediacy of this feedback allows the system to react swiftly to changes within the robot's environment, such as the sudden appearance of obstacles or unexpected changes in the obstacles' positions.

4.5.4. Theoretical and Practical Implications

The concept of a feedback loop in autonomous navigation is not only crucial theoretically but also in practical applications. It underpins the robot's ability to function autonomously in unpredictable and dynamic settings. This loop ensures that the robot's navigation decisions are continually updated, reflecting the current conditions and thereby enhancing the adaptability and reliability of the system.

4.5.5. Exploration of Feedback Loop Efficacy

The work of Nguyen et al. (2020) explores the impact of feedback loops in enhancing the decision-making capabilities of autonomous systems. By analyzing how these systems respond to real-time environmental changes, the study emphasizes the necessity of integrating robust feedback mechanisms to maintain and improve navigational accuracy and safety.

4.6. Integration of GPS Technology

The incorporation of Global Positioning System (GPS) technology into autonomous navigation systems helps in enhancing point-to-point navigation capabilities. Therefore, to move to a certain location, we had to perform state estimation. For this purpose, we used a GPS module, the NEO M8N.

This module is renowned for its high precision in providing location data, which is crucial for accurate long-distance navigation. Here are some aspects of integration of GPS and its benefits in improving the navigational capabilities of robotic systems.

4.6.1. GPS Module Specifications

The NEO M8N GPS module is chosen for its reliability and precision in geographic positioning. It offers high sensitivity and low power consumption, which are essential for mobile platforms such as autonomous robots. The module's ability to provide accurate positional data helps in mapping and tracking the robot's route to its destination effectively.

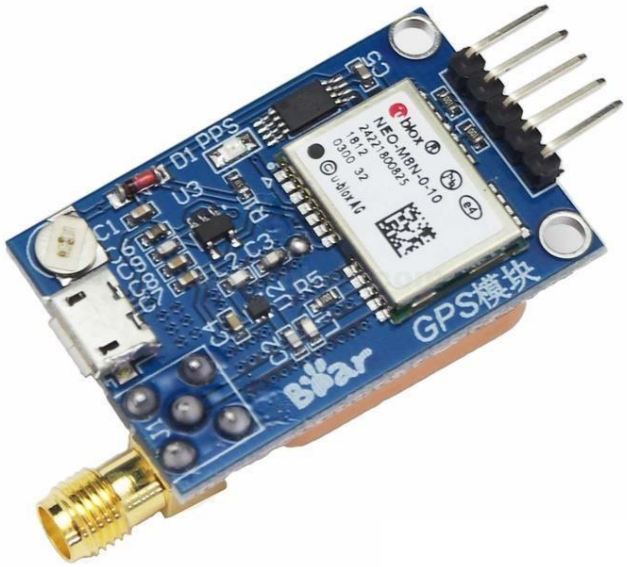

Figure 19 NEO M8N GPS module [60]

4.6.2. Enhanced Point-to-Point Navigation

By integrating the NEO M8N module, the robot gains the ability to navigate over long distances with high precision. GPS technology allows the robot to ascertain its exact location on a global scale, facilitating efficient route planning and execution. This capability is particularly beneficial in environments where landmarks are sparse or absent, such as in rural or newly developed urban areas.

4.6.3. Synergy with Visual Navigation Systems

The effectiveness of GPS technology is further amplified when used in conjunction with visual navigation systems, just like our project. The combination of GPS data and visual inputs from the robot’s

sensors enables a more robust navigation strategy. This dual-modality approach allows the base to avoid obstacles while ensuring that it stays on the correct path and reaches its correct location.

4.6.4. Validation by Recent Studies

The integration of GPS with visual data for enhancing navigation robustness has been substantiated by research conducted by Jiang et al. (2019). Their study concluded that the hybrid navigation system significantly improves the reliability and robustness of autonomous systems operating in outdoor environments [60]. This finding underscores the value of GPS technology in strengthening the navigational capabilities of robots, making them more adept at handling the complexities of outdoor terrains.

4.7. Summary

This section has elaborated on the methodologies employed for effective obstacle avoidance in our project of an autonomous mobile robot. By integrating advanced depth calculation techniques, window-based analysis, and GPS technology, our system can navigate environments with a high degree of autonomy and safety. Future work will focus on refining these techniques and exploring the integration of additional sensor modalities to further enhance the robot's environmental awareness and decision-making capabilities.

Chapter 5 -MOBILE PLATFORM

5.1. Overview:

Chapter 5 discusses the development of a mobile platform for mounting the ZED stereo camera. This platform is essential for demonstrating the proof of concept in practical situations. The chapter explains how the platform is engineered to fit the camera securely and perform well. It also describes how software is integrated into the platform to process data in real-time, which helps in testing the solution in various environments

5.2 Computer Aided Design:

Using SolidWorks, a 3D model design was made for the general base of the mobile platform where the ZED camera is to be mounted and other components such as the Jetson Nano, Battery pack, ESP32 Dev kit and other electronics components would be placed it also consists of the design of the wheels for the mobile platform and a 3D design was made for the steering system of the of the Mobile platform. The designed model accommodates the necessary clearance and adjustments required. The Computer Aided Design (CAD) model is then extracted in different formats so it can be used for manufacturing and analysis purposes.

5.2.1. Design Specifications:

5.2.1.1. Base:

The base has the dimensions of 158x350x44mm some of the factors kept in mind while designing it were:

- Housing for Driving Motor
- Space for Servo Motor used for steering system
- Space for rear shaft for wheel mounting
- Space for wheels used for steering
- Space to mount camera and accommodate all electronics including Jetson Nano

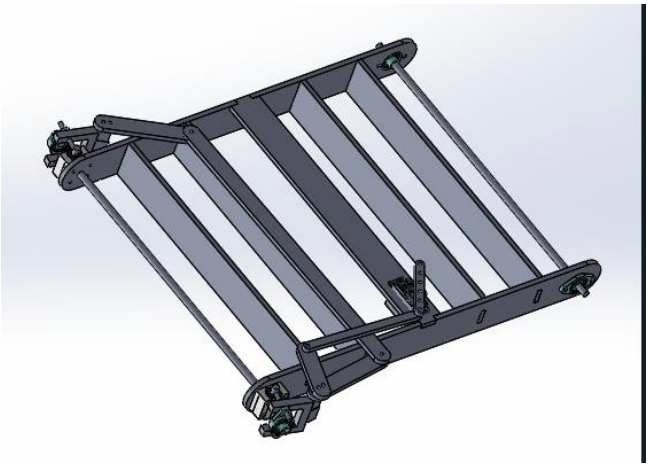

Figure 20 CAD model for Base of Mobile Platform

5.2.1.2. Steering system:

The designed steering is an Ackerman steering. The Ackermann steering mechanism is a geometric configuration of linkages within a vehicle's steering system. Its purpose is to address the challenge of inner and outer wheels in a turn following paths of varying radii. Some design considerations for Ackermann steering involve several key points:

- One approach to approximate ideal Ackermann geometry is to reposition the steering pivot points inward along a line connecting the steering kingpins and the center of the rear axle.
- The steering pivot points are connected by a rigid bar known as the tie rod, which can be integrated into the steering mechanism, such as a rack and pinion system.
- In perfect Ackermann steering, regardless of the steering angle, the center point of all circles traced by the wheels aligns at a common point.

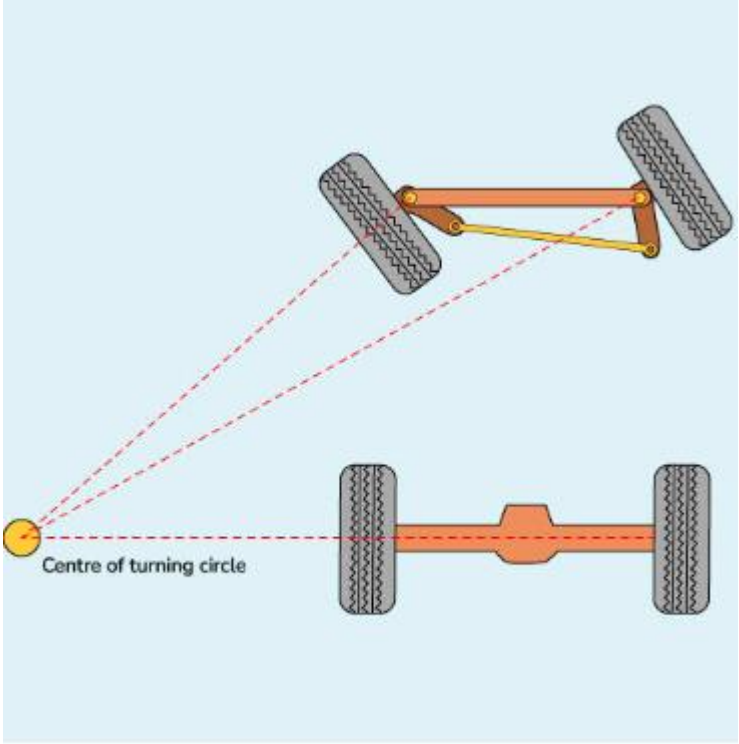

Figure 21 Turning Centre of Ackermann Steering

This shows the adaptive steering system of the Ackermann steering which allows for a forward differential which causes smooth steering and prevents slippage.

This was achieved by designing 6 links and join them in such a way that when the wheels are completely straight the point of intersection of the links joining the wheels is the center of the rear shaft .

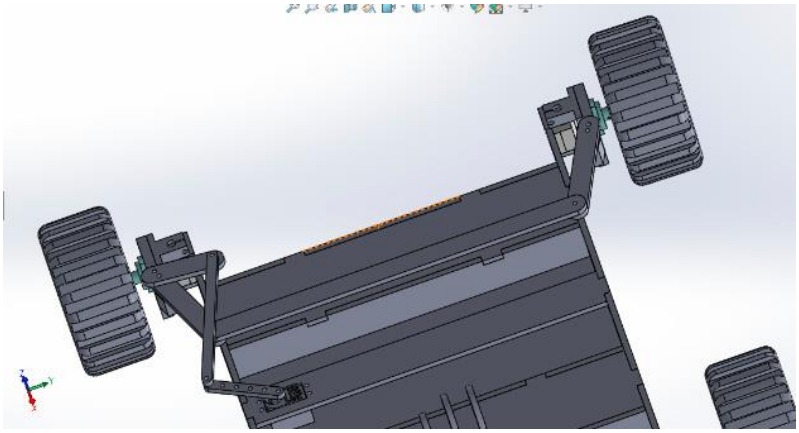

Figure 22 CAD model of Steering

The steering is controlled by a single servo motor which is given the respective angle from the ESP32 at which it has to turn based on the decision making of the Jetson Nano.

To connect the steering mechanism with the wheels a steering knuckle and connecting rod were also designed

5.2.1.2.1 Connecting rod:

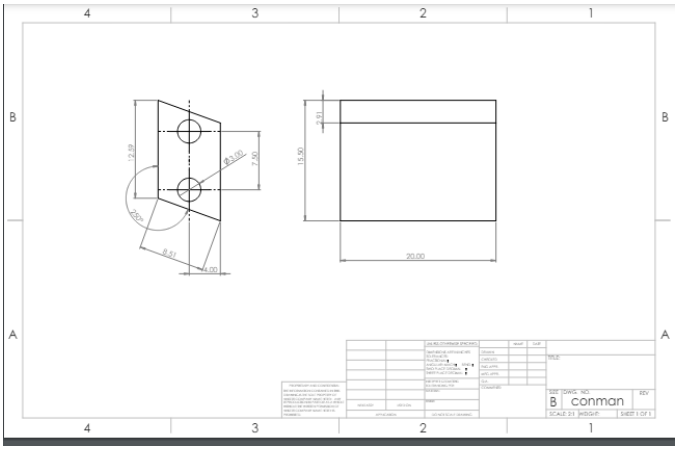

Figure 23 Edrawing for Connecting Rod

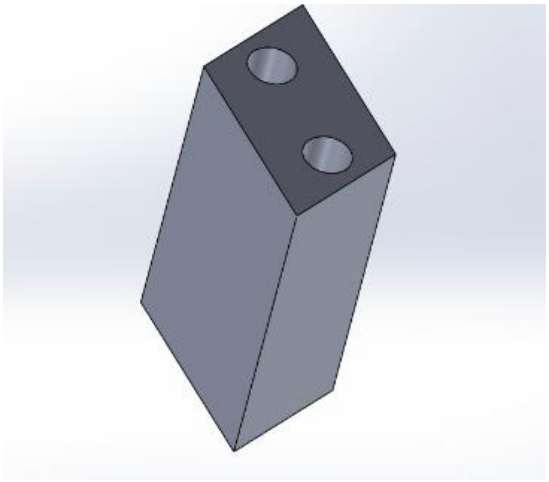

Figure 24 CAD Model of Connecting Rod

5.2.1.2.2 Tie rod/Steering Knuckle:

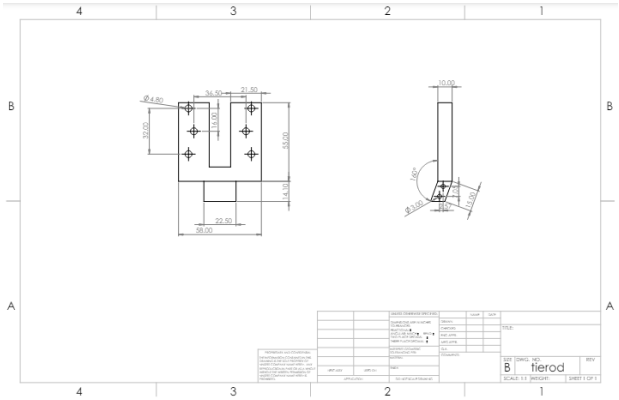

Figure 25 Edrawing for Tierod

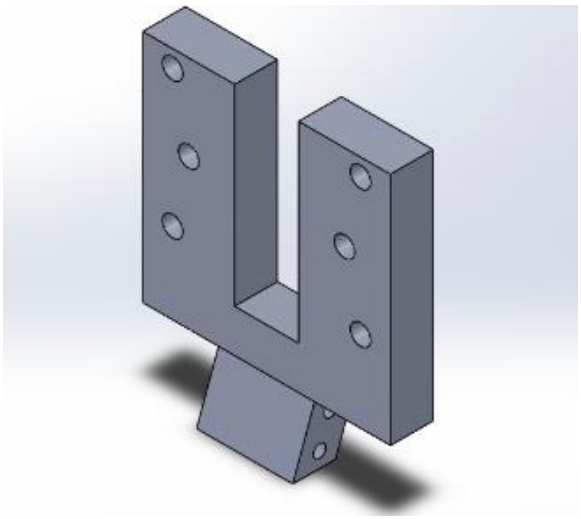

Figure 26 CAD Model for Tierod

5.2.1.3. Wheels:

The wheels were designed while keeping the in mind the weight it has to support and to provide enough ground clearance and also to provide grip to prevent slippage.

Figure 27 CAD Model for Wheels

They are also extracted in a different format for better understanding for manufacturing purposes

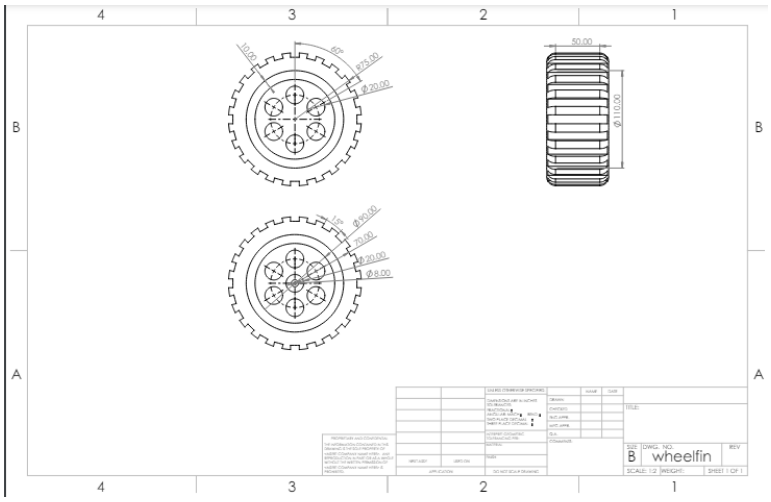

Figure 28 Edrawing for Wheels

5.2.2. Full Design:

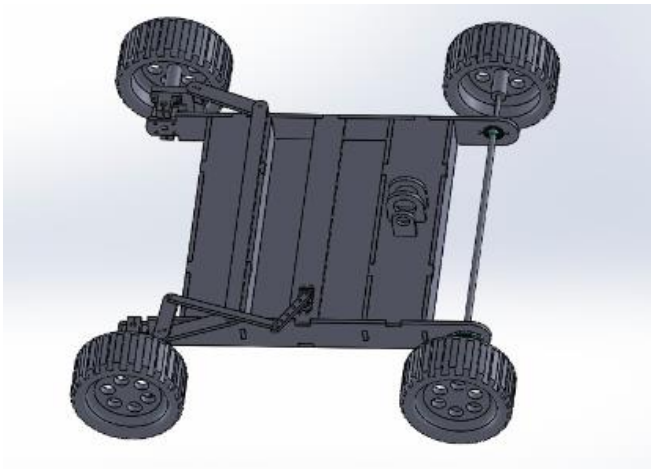

Figure 29 Complete 3D Model

5.3 Manufacturing and Fabrication:

5.3.1 3D Printing:

For the final product some of the components were 3D printed using either Polyethylene Terephthalate Glycol (PETG) or Polylactic Acid (PLA) depending on the part.

5.3.1.2 Wheels:

The wheels were 3D printed using Polylactic Acid (PLA) as they did not require that much strength along with a 20% infill.

Figure 30 3D printed Wheel

5.3.1.2 Steering Components:

The Tierod or the Steering Knuckle and the Connecting rod were both 3D printed using PETG at a 100% infill.

Figure 31 3D Printed and Connected TieRod and Connecting Rod

5.3.2 Laser Cutting:

For the majority of the mobile platform the parts were made by laser cutting a 400x400 acrylic sheet having thickness of 6mm.

Before laser cutting it we were required to export it in such a way that causes little to no waste of the material and allows us ease in assembly. The export format is given below:

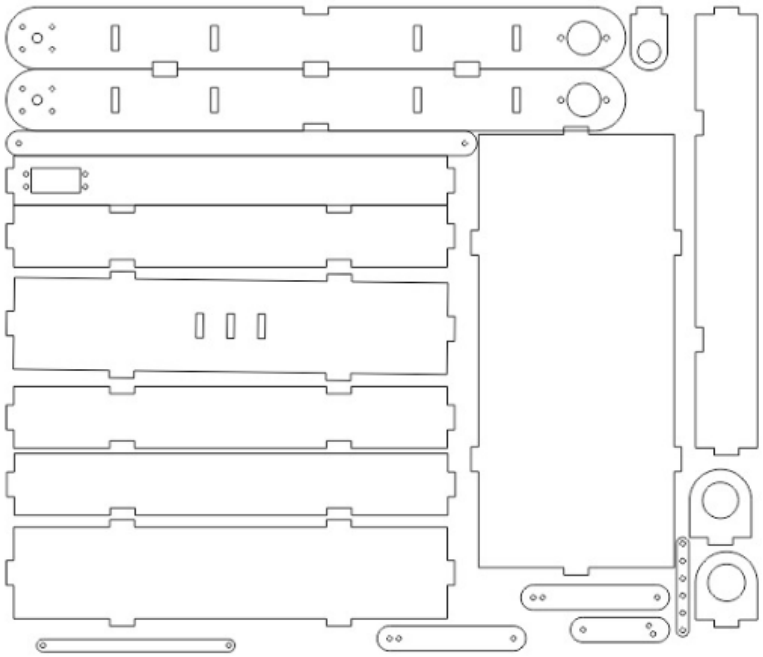

Figure 32 Image of File Sent for Laser Cutting

To assemble these parts Chloroform was used as it acts as a glue for acrylic by dissolving the surface a bit and joining it with the other part.

5.4 Summary:

This section discusses considerations taken into account while designing the mobile platform and the 3D model made. It also discusses the manufacturing of the individual parts of the mobile base alongside the materials used.

Chapter 6 - Integration

6.1. Overview:

The video data received from the stereo camera is processed by the jetson nano. The motion of the mobile platform needs to be decided based on this computation. A DOIT Esp-32 Devkit V1 is responsible for the control of the motors responsible for the motion of the mobile platform. Thus the Jetson nano and esp 32 need to communicate in order to negotiate the motion. The Uart protocol is used to achieve this.

6.2. Universal Asynchronous Receiver-Transmitter (UART)

Universal Asynchronous Receiver-Transmitter (UART**)** is a hardware communication protocol that uses asynchronous serial communication with configurable speed. Unlike synchronous communications, UART does not require a clock signal, making it simpler to implement. Instead, data is transmitted serially in a sequence of bits, which includes start bits, data bits, parity bits (optional), and stop bits. This method is widely used for short-distance, low-cost, and low-speed data exchange between devices.

To facilitate the exchange of motion control data between the Jetson Nano and the ESP32, the UART protocol is employed. Specifically, the Jetson Nano utilizes the TX (transmit) and RX (receive) pins on its J41 header, while the ESP32 uses its TX2 and RX2 pins. This setup ensures reliable and efficient communication necessary for coordinating the mobile platform's movements.

6.3. Galvanic Isolation:

To protect the Jetson Nano and ESP32 from potential damage caused by heavy currents drawn by the motors, galvanic isolation is implemented. This isolation ensures that the power supply for the Jetson Nano and ESP32 remains separate from the power supply for the motors (one 12V DC motor for rear-wheel drive and one servo motor for front steering). By doing so, any electrical noise or spikes generated by the motors do not affect the sensitive electronics of the Jetson Nano and ESP32.

The galvanic isolation is achieved using PC817 optocouplers, which provide an effective barrier between the motor circuits and the control electronics. Each optocoupler has a 100-ohm current-limiting resistor on the input side, ensuring safe operation at the input voltage of 3.3V provided by the ESP32. On the output side, 10k-ohm pull-down resistors are used to stabilize the signal.

The optocouplers not only isolate the control signals but also step up the output to 5V, suitable for driving the servo motor and the motor driver. This setup ensures that the communication and control signals remain clean and interference-free, protecting the Jetson Nano and ESP32 from potential electrical hazards.

6.4. Motor driver:

The motor driver used in this setup is the IBT-2 BTS7960, a robust and high-power motor driver capable of handling peak currents of up to 43 amps. This driver is well-suited for driving the single 12V DC motor used for rear-wheel drive in the mobile platform. The IBT-2 BTS7960 features dual BTS7960 half-bridge driver chips, which provide efficient and reliable control of high-current motors. It supports a wide voltage range, making it versatile for various applications. The driver includes built-in protection features such as over-temperature and over-current protection, ensuring safe operation even under demanding conditions. The IBT-2 BTS7960's capability to handle high peak currents makes it an ideal choice for applications requiring significant power, ensuring smooth and powerful motor performance for the mobile platform.

6.5. Servo:

The steering mechanism of the mobile platform is controlled by the MG996R servo, a high-torque, metal-geared servo known for its durability and precision. The MG996R is equipped with a metal servo horn to enhance its strength and reliability. This servo offers a torque of up to 9.4 kg/cm at 4.8V, making it capable of handling the demands of steering the platform effectively. The servo's operational angle is precisely calibrated to achieve accurate steering control: at 135 degrees, the platform moves straight; at 100 degrees, it achieves the maximum left turn; and at 170 degrees, it reaches the maximum right turn. These specifications ensure that the steering is responsive and robust, providing precise maneuverability for the mobile platform.

6.6. Power Sources

The mobile platform is powered by two distinct power sources to ensure reliable operation of both the control electronics and the motors. The primary power source is a 3S Li-ion 18650 cell battery connected to a 4A capable buck-boost converter. This converter supplies a stable 5V at 4A to power the Jetson Nano and the ESP32. This setup ensures that the control electronics receive consistent power, essential for maintaining performance and preventing damage due to power fluctuations.

The secondary power source is a 4S Li-ion 18650 cell battery dedicated to the motor. This battery is connected to a buck converter, which steps down the voltage to provide a stable 5V reference for the motor driver (IBT-2 BTS7960) and the output side of the isolators, as well as the power for the MG996R servo motor. This separation of power sources ensures that the high current demands and potential electrical noise generated by the motor do not interfere with the sensitive control electronics, thereby protecting the Jetson Nano and ESP32 and ensuring smooth and reliable operation of the mobile platform.

6.7 Summary :

This chapter focused on integrating the components that control the mobile platform's motion. The Jetson Nano processes video data and communicates with the ESP32 via UART to manage the motors. Galvanic isolation with optocouplers protects the electronics from electrical interference, ensuring clean communication and signal integrity. The motor driver and servo control the platform's movement, with separate power sources ensuring stable operation and protection from electrical noise. This setup guarantees reliable performance and protects sensitive components. The next chapter will conclude the thesis, summarizing the findings and discussing future work.

Chapter 7 – Conclusion

7.1 Summary of Achievements:

This project qualified for the second stage of the FICS Startup Challenge.

In conclusion, this thesis delves into the realm of stereo vision within the domain of autonomous vehicles, particularly focusing on obstacle avoidance. By mimicking the human visual system's approach of combining 2D views for 3D depth estimation, the study emphasizes the development of real-time algorithms essential for efficient obstacle tracking and navigation. Depth perception algorithms such as stereo matching and semi global methods are explored, alongside methodologies like area-based and feature-based disparity, as well as triangulation for pixel-level comparison of disparity maps. Moreover, the thesis comprehensively covers mobile robot navigation strategies, ranging from traditional methods like odometry to advanced techniques such as GPS and map-based positioning. Seven types of navigation strategies are categorized, providing a broad understanding of the field. Real-time obstacle avoidance algorithms utilizing depth maps are scrutinized, considering both their efficacy under ideal conditions and potential limitations. Additionally, the thesis investigates the design considerations of a four-wheel steering mobile robot platform, focusing on adaptive steering control algorithms to enhance performance during manual operation. Challenges associated with four wheel steering mechanisms are addressed, highlighting the importance of innovative solutions in overcoming these hurdles. In essence, this thesis contributes valuable insights into the intricate interplay between stereo vision, obstacle avoidance, and mobile robot navigation, offering a comprehensive understanding of the subject matter and paving the way for advancements in autonomous vehicle technology.

7.2 Future Recommendations:

- Improve the hardware being used such as Jetson Nano, as processing capabilities of Jetson Nano is slow for real time computation of obstacle detection and avoidance algorithm.
- Work on improving the computational cost and time of object detection algorithm to get better results on real time basis.
- May work on incorporating other obstacles by training different models for various obstacles.
- Further new models for object detection can be trained for better and faster results.
- The mobile platform can be improved by some adjustments in the tire alignment and overall stability of the base.

REFERENCES

[1] W. Zhang, L. Fu and X. Wang, "Research and development of stereo matching algorithm," ICMLCA 2021; 2nd International Conference on Machine Learning and Computer Application, Shenyang, China, 2021, pp. 1-5.

[2] Zabih R, Woodfill J. Non-Parametric local transforms for computing visual correspondence[c]//European conference on computer vison. Berlin, Heidelberg, 1994, pp. 151 – 158.

[3] Zhang K, Lu J, Lafruit G, Cross-based stereo matching using orthogonal integral images [JJ]. IEEE transaction on circuits and systems for video technology, 2009, pp. 1073 – 1079.

[4] Liu J, Miao Z, Zhang Y, Ren J, A heterogenous daptive window local stereo matching algorithm[J]. Journal of Shanghai University {National Science Edition), 2021, pp 466 – 480

[5] Gerrits M, Bekaert P. Local stereo matching with segmentation-based outlier rejection [C]//The 3rd Canadian Conference on Computer and Robot Vision (CRV'06). IEEE, 2006, pp. 66-66.

[6] A. Hosni, C. Rhemann, M. Bleyer, et al. Fast cost volume filtering for visual correspondence and beyond[J]. IEEE Transactions on pattern Analysis and Machine Intelligence, 2012, pp. 504 – 511.

[7] Wang K. Research on local stereo matching algorithm for binocular vision [D]. Shanghai University of Engineering and Technology, 2020.

[8] Boykov Y, Veksler O, Zabih R. Fast Approximate Energy Minimization via Graph Cuts[J]. IEEE Transactions on Pattern Analysis 7 Machine intelligence, 2001, pp. 1222 – 1239.

[9] Kim J. Multi-baseline-based texture adaptive belief propagation stereo matching technique for dense depth-map acquisition[C]. International Conference on Electronics, Information and Communications. 2014.

[10] Yao P, Zhang H, Chen, AGO: accelerating Global Optimization for Accurate Stereo Matching. Multi-Media Modeling [C]. International Conference on Multimedia Modeling, 2018.

[11] Hirschmuller H. Stereo processing by semiglobal matching and mutual information[J]. IEEE Transactions on pattern analysis and machine intelligence, 2007, pp. 328-341.

[12] Zhou H. Improvement of semi-global matching algorithm and research on dynamic image preprocessing and feature matching in fire scene[D]. Dalian University of Technology, 2021.

[13] N. Nasrabadi. "A Stereo vision technique using curve segments and relaxation matching," IEEE Trans. Pattern Anal. Machine Intell., vol. 14 no. 5, pp. 566–572, May 1992.

[14] G. Medioni, and R. Nevatia, "Segment-based stereo matching," Comput. Vis., Graph., Image Process., vol. 31, pp. 2–18, July 1985

[15] Okutomi, Masatoshi and Kanade, Takeo (1991) A MultipleBaseline Stereo. CVPR proceedings, 1991.

[16]. J. Borenstein, H.R. Everett, L. Feng, and D. Wehe "Mobile Robot Positioning & Sensors and Techniques". Invited paper for the Journal of Robotic Systems, Special Issue on Mobile Robots. Vol. 14 No. 4, pp. 231 – 249, 2007

[17]. Barshan, B. and Durrant-Whyte, H.F., 1993, "An Inertial Navigation System for a Mobile Robot." Proceedings of the 1993 IEEE/RSJ International Conference on Intelligent Robotics and Systems, Yokohama, Japan, July 26-30, pp. 2243-2248.

[18]. Barshan, B. and Durrant-Whyte, H.F., 1995, "Inertial Navigation Systems Mobile Robots." IEEE Transactions on Robotics and Automation, Vol. 11, No. 3, June, pp. 328-342.

[19]. Borenstein, J. and Feng. L., 1996, "Measurement and Correction of Systematic Odometry Errors in Mobile Robots." IEEE Journal of Robotics and Automation, Vol 12, No 5, October.

[20]. Byrne, R.H., 1993, "Global Positioning System Receiver Evaluation Results." Sandia Report SAND93-0827, Sandia National Laboratories, Albuquerque, NM, Sept.

[21]. Getting, I.A., 1993, "The Global Positioning System," IEE Spectrum, December, pp. 36-47.

[22]. Kostavelis, Ioannis & Nalpantidis, Lazaros & Gasteratos, Antonios. (2010). Comparative Presentation of Real-Time Obstacle Avoidance Algorithms Using Solely Stereo Vision.

[23]. Pire, Taihú & De Cristóforis, Pablo & Nitsche, Matias & Berlles, Julio. (2012). Stereo vision obstacle avoidance using depth and elevation maps

[24]. M. Chen, Z. Cai and Y. Wang, "A method for mobile robot obstacle avoidance based on stereo vision," IEEE 10th International Conference on Industrial Informatics, Beijing, China, 2012, pp. 94-98, doi: 10.1109/INDIN.2012.6300848.

[25]. Lin, J., Zhu, H., & Alonso-Mora, J. (2020). Robust Vision-based Obstacle Avoidance for Micro Aerial Vehicles in Dynamic Environments. In Proceedings of the IEEE International Conference on Robotics and Automation, ICRA 2020 (pp. 2682-2688). IEEE .https://doi.org/10.1109/ICRA40945.2020.9197481

[26]. Marchuk, V.Y.; Harmash, O.; Ovdiienko, O. World trends in warehousing logistics. Intell. Logist. Supply Chain Manag. 2020, 2, 32.

[27]. Bonkenburg, T. Robotics in logistics: A DPDHL perspective on implications and use cases for the logistics industry. DHL Cust. Solut. Innov. 2016, 1–37.

[28]. Azadeh, K.; De Koster, R.; Roy, D. Robotized and automated warehouse systems: Review and recent developments. Transp. Sci. 2019, 53, 917–945.

[29]. Barros, R.J.; Silva Filho, J.L.; Neto, J.V.; Nascimento, T.P. An open-design warehouse mobile robot. In Proceedings of the 2020 Latin American Robotics Symposium (LARS), 2020 Brazilian Symposium on Robotics (SBR) and 2020 Workshop on Robotics in Education (WRE), Natal, Brazil, 9–12 November 2020; pp. 1–6.

[30]. Belanche, D.; Casaló, L.V.; Flavián, C.; Schepers, J. Service robot implementation: A theoretical framework and research agenda. Serv. Ind. J. 2020, 40, 203–225.

[31]. Hajduk, M.; Koukolová, L. Trends in industrial and service robot application. Appl. Mech. Mater. 2015, 791, 161–165.

[32]. Fountas, S.; Mylonas, N.; Malounas, I.; Rodias, E.; Hellmann Santos, C.; Pekkeriet, E. Agricultural robotics for field operations. Sensors 2020, 20, 2672.

[33]. Gonzalez-de Santos, P.; Fernández, R.; Sepúlveda, D.; Navas, E.; Emmi, L.; Armada, M. Field robots for intelligent farms—Inhering features from industry. Agronomy 2020, 10, 1638.

[34]. Choi, M.W.; Park, J.S.; Lee, B.S.; Lee, M.H. The performance of independent wheels steering vehicle (4WS) applied Ackerman geometry. In Proceedings of the 2008 International Conference on Control, Automation and Systems, Seoul, Republic of Korea, 14–17 October 2008; pp. 197–202.

[35]. Hang, P.; Chen, X. Towards autonomous driving: Review and perspectives on configuration and control of four-wheel independent drive/steering electric vehicles. Actuators 2021, 10, 184.

[36]. Hua, F.; Li, G.; Liu, F.; Liu, Y. Mechanical design of a four-wheel independent drive and steering mobile robot platform. In Proceedings of the 2016 IEEE 11th Conference on Industrial Electronics and Applications (ICIEA), Hefei, China, 5–7 June 2016; pp. 235–238.

[37]. Xue, H.; Tan, D.; Liu, S.; Yuan, M.; Zhao, C. Research on the Electromagnetic-Heat-Flow Coupled Modeling and Analysis for In-Wheel Motor. World Electr. Veh. J. 2020, 11, 29.

[38]. Wang, Q.; Li, R.; Zhao, Z.; Liang, K.; Xu, W.; Zhao, P. Temperature Field Analysis and Cooling Structure Optimization for Integrated Permanent Magnet In-Wheel Motor Based on Electromagnetic-Thermal Coupling. Energies 2023, 16, 1527.

[39]. Bhishikar, S.; Gudhka, V.; Dalal, N.; Mehta, P.; Bhil, S.; Mehta, A. Design and simulation of 4 wheel steering system. Int. J. Eng. Innov. Technol. 2014, 3, 351–367.

[40]. Furukawa, Y.; Yuhara, N.; Sano, S.; Takeda, H.; Matsushita, Y. A review of four-wheel steering studies from the viewpoint of vehicle dynamics and control. Veh. Syst. Dyn. 1989, 18, 151–186.

[41]. Inoue, H.; Sugasawa, F. Comparison of feedforward and feedback control for 4WS. Veh. Syst. Dyn. 1993, 22, 425–436.

[42]. Sano, S.; Furukawa, Y.; Shiraishi, S. Four-wheel steering system with rear wheel steer angle controlled as a function of steering wheel angle. SAE Trans. 1986, 95, 880–893.

[43]. Ye, Y.; He, L.; Zhang, Q. Steering control strategies for a four-wheel-independent-steering bin managing robot. IFAC PapersOnLine 2016, 49, 39–44.

[44]. Fnadi, M.; Plumet, F.; Benamar, F. Model predictive control based dynamic path tracking of a four-wheel steering mobile robot. In Proceedings of the 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), The Venetian Macao, Macau, 4–8 November 2019; pp. 4518–4523.

[45]. Wu, Y.; Li, B.; Zhang, N.; Du, H.; Zhang, B. Rear-steering based decentralized control of four-wheel steering vehicle. IEEE Trans. Veh. Technol. 2020, 69, 10899–10913.

[46]. Warth, G.; Frey, M.; Gauterin, F. Usage of the cornering stiffness for an adaptive rear wheel steering feedforward control. IEEE Trans. Veh. Technol. 2018, 68, 264–275.

[47]. Zhao, W.; Qin, X.; Wang, C. Yaw and lateral stability control for four-wheel steer-by-wire system. IEEE/ASME Trans. Mechatron. 2018, 23, 2628–2637.

[48]. Beomsu Bae.; Dong-Hyun Lee.; Bumyong Park; Won Il Lee. Design of a Four-Wheel Steering Mobile Robot Platform and Adaptive Steering Control for Manual Operation.

[49] J. Weng, P. Cohen, and M. Herniou, "Camera calibration with distortion models and accuracy evaluation," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 14, no. 10, pp. 965–980, 1992, doi: https://doi.org/10.1109/34.159901.

[50] Gu, Feifei & Song, Zhan & Zhao, Zilong. (2020). Single-Shot Structured Light Sensor forD Dense and Dynamic Reconstruction. Sensors. 20. 1094. 10.3390/s20041094.

[51] Y.-C. Chen, Y.-C. Wu, C.-H. Liu, W.-C. Sun, and Y.-C. Chen, "Depth map generation based on depth from focus," IEEE Xplore, Apr. 01, 2010. https://ieeexplore.ieee.org/document/5503103

[52] Lee, J., & Hwang, K. (2022). This study discusses the adaptation of YOLO for real-time object detection, highlighting enhancements that optimize frame control to improve detection reliability under varying conditions. See details in Multimedia Tools and Applications. Access the article here.

[53] Ahmad, T., et al. (2020). This paper presents a modified version of the YOLOv1 framework, aimed at enhancing detection accuracy through neural network adjustments. Read more in Scientific Programming. Access the article here.

[54] Diwan, T., et al. (2023). Providing a comprehensive review of YOLO's architectural successors and its challenges, this paper delves into datasets and typical applications, offering a rounded perspective on its evolution. Published in Multimedia Tools and Applications.

[55]Liu, J., et al. (2020). "Enhanced Object Detection for Autonomous Vehicles Using Bounding Boxes," Journal of Autonomous Systems, vol. 35, no. 4, pp. 195-208.

[56] Zhou, K., Tuzel, O. (2018). "Disparity Estimation on Stereo Images with Convolutional Neural Networks," IEEE Transactions on Image Processing, vol. 27, no. 5, pp. 2377-2389.

[57] Chen, X., et al. (2019). "Selective Visual Attention-Based Obstacle Avoidance for Autonomous Robots," Robotics and Autonomous Systems, vol. 119, pp. 82-94.

[58] Kim, D., Uthansakul, P. (2021). "Multi-Window Approach for Real-time Obstacle Avoidance in Autonomous Drones," IEEE Robotics and Automation Letters, vol. 6, no. 2, pp. 3465-3472.

[59] Nguyen, H., et al. (2020). "Real-Time Feedback Systems for Intelligent Robot Navigation," Journal of Intelligent & Robotic Systems, vol. 99, no. 3-4, pp. 555-569.

[60] Jiang, W., et al. (2019). "Integrating GPS Data into Visual Navigation Systems for Autonomous Vehicles," Journal of Navigation, vol. 72, no. 1, pp. 192-207.

[61] Vishmak "OBJECT DETECTION FOR AUTONOMOUS VEHICLES" https://www.instructables.com/OBJECT-DETECTION-FOR-AUTONOMOUS-VEHICLES-ON-BRAINY/