CHALMERS
UNIVERSITY OF TECHNOLOGY

UNIVERSITY OF GOTHENBURG

A Solution for 3D Visualization on Soil Surface Using Stereo Camera

Master's thesis in Computer science and engineering

Zilong Wang
Qi Chang

Department of Computer Science and Engineering
CHALMERS UNIVERSITY OF TECHNOLOGY
UNIVERSITY OF GOTHENBURG
Gothenburg, Sweden 2023

Master's thesis 2023

A Solution for 3D Visualization on Soil Surface Using Stereo Camera

Zilong Wang Qi Chang

UNIVERSITY OF
GOTHENBURG

CHALMERS
UNIVERSITY OF TECHNOLOGY

Department of Computer Science and Engineering
Chalmers University of Technology
University of Gothenburg
Gothenburg, Sweden 2023

A Solution for 3D Visualization on Soil Surface

Zilong Wang
Qi Chang

Supervisor: Beatriz Cabrero-Daniel, University of Gothenburg
Advisor: Sara Afshar, Volvo Construction Equipment
Examiner: Hans-Martin Heyn, University of Gothenburg

Master's Thesis 2023
Department of Computer Science and Engineering
Chalmers University of Technology and University of Gothenburg
SE-412 96 Gothenburg
Telephone +46 31 772 1000

Typeset in LaTeX
Gothenburg, Sweden 2023

A Solution for 3D Visualization on Soil Surface

Qi Chang
Zilong Wang
Department of Computer Science and Engineering
Chalmers University of Technology and University of Gothenburg

Abstract

Ground surface monitoring using 3D visualization techniques has gained significant attention in recent years, particularly in the context of autonomous driving.

This research presents an implementation of a cost-efficient 3D visualization solution for soil surface analysis, aiming to explore an alternative approach to Light Detection and Ranging (LiDAR) technology. The research investigates the feasibility of utilizing stereo cameras as an affordable option for generating 3D visualization. A comparative study between various stereo vision 3D reconstruction methods is conducted to evaluate their performance.

Different stereo-matching methods are employed to extract depth information from the captured stereo images, including a machine learning method and a traditional semi-global block matching method. The resulting depth maps are then projected onto a 3D space, enabling the generation of point cloud data for visualization purposes. The visualized 3D representation provides an enhanced understanding of the soil surface conditions and facilitates detailed analysis.

To assess the effectiveness of the implemented visualization solution, several metrics are employed as a measure to compare the accuracy of the generated visualizations. Time measurement and Root Mean Square Error (RMSE) analysis serve as benchmarks for evaluating the performance and reliability of the proposed 3D visualization approach.

The findings of this research demonstrate the potential of stereo cameras as a cost-effective alternative to LiDAR sensors for soil surface analysis. The presented 3D visualization solution may contribute to autonomous construction vehicles by providing an efficient and affordable function for monitoring soil surface conditions.

Keywords: **3D reconstruction**, **Soil surface visualization**, **Stereo-matching application**, **Autonomous vehicles**

Acknowledgements

We would like to express our sincere gratitude to our supervisor, Beatriz Cabrero-Daniel, for her continuous guidance and support throughout this thesis.

We would also like to thank our advisor Sara Afshar, for her support, feedback, and provision of necessary resources.

Lastly, we are grateful to our family, friends, and colleagues for their encouragement, understanding, and support throughout this journey.

Zilong Wang, Qi Chang, Gothenburg, June 2023

Contents

List of Figures

List of Tables

1

Introduction

Ground surface condition is a critical factor to consider in engineering projects that operate on the ground. The condition of the ground can significantly affect the performance, safety, and efficiency of these projects. Traditional ground surface monitoring has primarily focused on flat surfaces, such as roads, with the goal of detecting abnormalities or objects. With the growing interest in automation technologies for construction vehicles, there is an increasing demand for more sophisticated and comprehensive ground surface monitoring systems that can support a wide range of applications.

In recent years, 3D visualization has become a popular topic in both industry and academia because of its wide applications in various fields, such as autonomous driving [5]. While these applications focus more on object detection or general observation, many 3D visualization approaches use LiDAR as the sensor [6]. LiDAR is widely used in capturing environments or surface conditions. It is a method for determining ranges by targeting an object or a surface with a laser and measuring the time for the reflected light to return. It can provide a point cloud in 3D space which includes relatively precise position information. However, the cost of high-precision LiDAR systems is relatively expensive, which limits their practical application in engineering projects.

In light of these considerations, this research aims to explore cost-effective and efficient solutions for soil surface monitoring and visualization using depth cameras, specifically stereo cameras. By selecting a more affordable sensor, we aim to investigate a different 3D reconstruction method using stereo vision camera in order to achieve an effective, high-precision, and low-latency soil surface monitoring system. The existing visualization system works with a sensor installed at the back of the vehicle. During the operation of the construction vehicle in the field, raw data will be gathered in the form of either point clouds or images, which will be transmitted to a server for further processing. The collected data will be processed in real-time to generate soil surface information specific to the vehicle's current working area. This processed data will then be displayed on a screen inside the vehicle, enabling operators to obtain a clear and intuitive visualization of the soil surface conditions. An overview of the system can be seen in Figure 1.1.

In this study, the primary objective is to conduct a comparative analysis of various stereo matching methods and identify the most suitable solution for engineering vehicles, considering multiple evaluation metrics. The software development aspect of this research primarily emphasizes the server-side of the pipeline. The findings

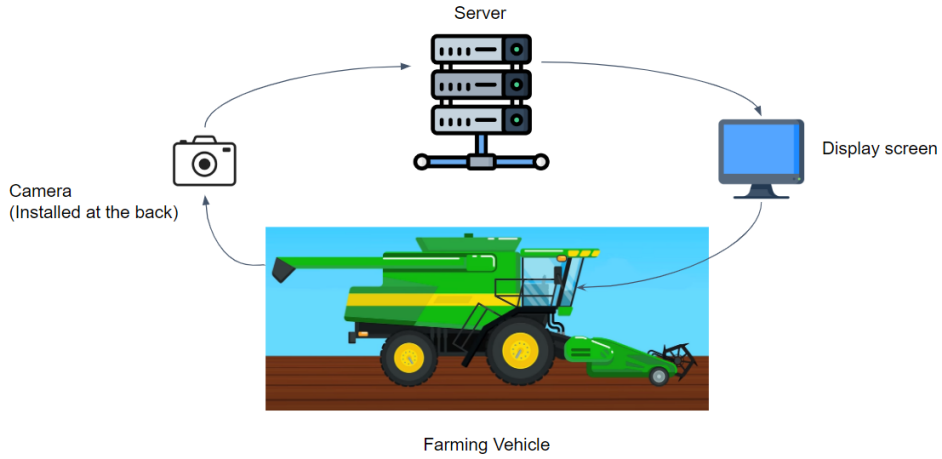

Figure 1.1: Overview of Visualization System

of this study hold significant potential for seamless integration into construction vehicles, thereby enhancing automation capabilities for soil surface monitoring and facilitating diverse applications in relevant domains.

1.1 Problem Statement

Current industrial real-time ground surface monitoring techniques emphasize on providing solutions for autonomous driving, which seek to achieve monitoring road conditions and surrounding objects [7]. However, there is a need for construction vehicles, such as mining vehicles or farming vehicles, to monitor the ground surface while operating. The ground surface information required for these construction vehicles differs significantly from those needed by regular driving vehicles. For instance, mining vehicles need to acquire information about the terrain they are operating on, including data on roughness, gradients, and potential obstacles. Detecting roughness enables the vehicle to adjust its speed, while obstacle information helps avoid collisions. Another distinguishing factor is that the visualization focus is primarily on the ground underneath rather than the general forward ground view.

There is not enough current research on real-time visualization solutions for soil surfaces. Most research either focuses on offline analysis [8] or uses relatively expensive and complicated sensors [9] which is not suitable for industrial products.

1.2 Research Questions

The research questions of this study include the following:

- **RQ1**: What techniques can be used for point cloud generation from stereo images?
- **RQ2**: What techniques can be used for curve extraction from a 3D scene?
- **RQ3**: What is the difference in performance between different techniques?

1.3 Goal

The motivation of this thesis is to provide a cost-friendly and efficient implementation for an embedded soil surface monitoring and visualization system. The proposed system aims to enable vehicle operators to conveniently assess the ground conditions located behind or beneath the vehicle. To achieve this objective, the research is guided by the following three specific goals:

- **Implement and compare various stereo matching methods**: The aim of this research is to implement an alternative visualization solution to using LiDAR, specifically using stereo cameras. One of the primary objectives of this study is to evaluate and compare various stereo matching methods to identify the most suitable solution that satisfies the imposed time constraints while concurrently delivering detailed 3D information. In this study, a LiDAR scanner on iPhone 13 will be utilized to provide a baseline point cloud for comparison as it is very precise. Stereo image pair will be collected by stereo cameras and visualized using corresponding stereo matching strategies.
- **Select interpolation method**: One challenge in achieving 3D reconstruction from stereo images is the lack of fine details due to the correlation between the number of points in the generated point cloud and the number of pixels in the images. To enhance the visual quality of the 3D reconstruction results, it is necessary to employ interpolation methods to increase the density of the points. Using interpolation not only enhances the visual effect of the 3D reconstruction results but also allows for precise control over the number of points in the final generated point cloud. This enables a more scientific and straightforward evaluation process.
- **Define evaluation metrics**: The goal for evaluation is to identify a method that is cost-effective, efficient, and can operate in a wide range of environments and soil conditions. When defining the evaluation metrics, it is important to consider not only the efficiency and effectiveness of visualization but also the practical requirements of the engineering and product domains. These practical requirements may include factors such as computational resources, real-time performance, and compatibility with existing systems or platforms.

1.4 Contribution

This paper makes several contributions to the field of 3D visualization solutions of soil surface.

Firstly, an experimental scene is built to represent a real-world scenario, which enables the collection of data using different sensors. This is important because it ensured that this research was conducted in a realistic environment, which increases the relevance and applicability of the findings.

Secondly, evaluation metrics are proposed to analyze the processed data collected from the sensors and compare the performance of different visualization solutions.

Thirdly, based on the evaluation metrics, a suitable solution is chosen. An end-to-end processing pipeline is built to enable vehicle operators to monitor the soil conditions. This pipeline is universal and it can be easily adopted in other similar scenarios.

Overall, our research provides a solution that can be a foundation for future research. The experimental scene, evaluation metric, and end-to-end processing pipeline are valuable tools that can be used to develop and test new solutions in similar scenarios. The research can also provide insights into the performance of different solutions, which can contribute to the development of more effective and efficient solutions for soil surface visualization in the future.

1.5 Structure of Paper

The following sections can provide an overview of this paper:

- **Introduction**: this section introduces the problem and background, and outlines the research questions.
- **Related work**: in this section reviews of previous related research are analyzed.
- **Theory**: this section presents the theories that are relevant to the research and explains why they are related to the study.
- **Methodology**: describes the experiment setup, data collection and evaluation metrics used in the research.
- **Result**: present the findings of experiments with visual graphs and tables.
- **Discussion**: Discuss the findings based on the result, address the limitations and some possible future improvements.

2

Related Work

In this chapter, an overview of existing research on soil surface visualization and 3D reconstruction methods will be presented.

2.1 Smart agriculture

Sustainable and smart agriculture has been a hot topic in recent years as the World Resources Institutes (WRI) reported in 2018 to meet food demand of 10 billion people by 2050 [10]. With the advancement of the Unmanned Aerial Vehicles (UAVs) and Internet of Things (IoT) technology, people are witnessing the emergence of greater possibilities in agriculture. These smart farms are designed to be run by interconnected devices and vehicles. Some enormous potentials can be achieved by the integration of different IoT technologies to achieve automated operations with minimum supervision [11].

Lots of IoT platforms appeared to provide low-cost systems for environmental parameter monitoring [12, 13], in order to facilitate better crop growth. These parameters been monitored include temperature, humidity, soil moisture, wind speed, wind direction, UV index, etc. The UAVs, on the other hand, are used to create useful 2D or 3D maps of an agricultural field. Certain information can be obtained from the maps, for instance area of the farmland, soil conditions and the statues of the crops. Gullen et al. [14] obtained high-resolution maps with spatial variation information using UAV images. The result maps are used for profitable precision agriculture tasks, such as the separation of fruit quality areas, identifying the area of deforestation and the agronomic control of homogeneous zones [11].

UVAs can also contribute to planting seeds and spraying pesticides or fertilizers. UAVs equipped with image recognition technology for an optimized planting technique is under research [15].

2.2 LiDAR-Based Visualization

Light detection and ranging (LiDAR) sensors can efficiently capture point cloud data that can be used for 3D reconstruction. It is widely used for mapping applications and autonomous driving. With the point cloud data collected by a LiDAR scanner, the actual scene can be effectively modeled with high accuracy. In 2003, Fabio et al. summarized a series of strategies that can apply to converting a measured point cloud

to a 3D model and then visualizing it on computers [16]. This research emphasizes solving problems of reconstructing computer graphic models from incomplete, noisy and sparse point cloud data.

LiDAR is widely used in retrieving available sight distance (ASD), for example, Ma et al. proposed a real-time visualization method for estimating 3D highway sight distance using LiDAR data [17]. The method aimed at providing a real-time monitor system for roadside views along the highway.

In recent years as deep learning becomes a popular topic, researchers started to apply deep learning to LiDAR data for solving depth completion. In 2019, Qiu et al. came up with DeepLiDAR, which is a deep learning architecture that produces accurate dense depth for outdoor scenes from a color image and sparse data from LiDAR [18]. The solution network consists of two pathways implemented with a stack of deep completion units. The first pathway is the color pathway takes an RGB image and the sparse depth as input and outputs the complete depth. The second pathway, which is the surface pathway, predicts a surface normal image from the input image, then combines this surface normal image with a confidence mask and the sparse depth to output the complete depth. Depth generated by the two pathways is integrated by a weighted sum to produce the final result. The model has achieved state-of-the-art performance on the KITTI dataset.

Apart from machine learning strategies, there are also some LiDAR depth completions that use non-learning completion methods. In 2021 Zhao et al. proposed a method for LiDAR depth completion that is based on a surface geometry model which is enhanced by an outlier removal algorithm [19]. The theory behind the method is each pixel with an unknown depth has a nearby LiDAR point, so it is assumed that those pixels in on the same surface with the nearby LiDAR point. The depth of the unknown point can be estimated by the nearby point depth plus a residual error which is calculated by physical parameters. With an outlier removal algorithm to remove points from occluded regions, the method is enhanced and achieves state-of-the-art performance on the KITTI dataset.

2.3 Stereo Vision 3D Visualization

Apart from LiDAR, stereo cameras are also an effective method for 3D reconstruction and visualization. Many researchers have completed monitoring or visualization systems with a stereo camera.

In 2009, Jokela et al. [20] presented a system and evaluation results to monitor road conditions, including ice, water, snow, and dry asphalt. However this project did not use image pairs from stereo cameras to construct a 3D scene. Instead, it analyzes graininess and light polarization changes to detect the category of the surface.

Due to the fact that most monitoring or visualization system do not need very detailed information of the surface, there are more researches that focus on using stereo cameras to do object detection tasks. Lagisetty et al. [21] developed real-time obstacle detection for autonomous navigation robots using a stereo camera. The system utilizes left and right images to do correlation-based stereo matching by searching

both images to maximize the similarity criterion. Then a 3D reconstruction is done through the singular value decomposition (SVD) technique. With the pose estimation result the system can confirm the position of the obstacles in order to avoid them during navigation.

In addition to providing monitoring systems with a basic visualization, it is also very important to generate high-quality depth maps or 3D models from stereo cameras. Some of the methods increased the quality of the original left and right image pairs. In 2003 D.Scharstein et al. used structured light to acquire high-quality stereo image pairs, which does not require light source calibration [22]. Nedevschi et al. presented a high-accuracy stereo vision system for far-distance obstacle detection [23]. The system uses stereo matching to do 3D reconstruction, then group the obtained points in space to detect the potential obstacles. Other methods for high-quality result uses a fusion of different techniques. Zhu et al. showed combining time-of-flight sensors and stereo cameras can result in high-accuracy depth maps in 2008 [24].

In 2011 Geiger et al. [25]proposed a novel stereo matching sparse feature matcher in conjunction with an efficient and robust visual odometry algorithm, which can runs at 25 frames per second to generate the 3D point cloud (depth maps at 3-4 fps). Another 3D reconstruction method based on a joint 3D reconstruction model for depth fusion, including a data fitting term and a robust low-rank matrix factorization constraint, is proposed by Tian et al. [26] in 2022. The algorithm has shown its efficiency and wide range of application by comparing to the state-of-the-art methods.

2.4 Limitation of Previous Research

Former researches mostly focus on the following two aspects:

- Apply 3D reconstruction methods to object detection at a far distance
- Increase the quality of generated depth maps or 3D models

We can conclude that there is insufficient research on close-up soil surface visualization using stereo cameras. To address this gap, our project proposes an end-to-end soil surface visualization strategy for construction vehicles.

3

Theory

3.1 3D Visualization

3.1.1 LiDAR

LiDAR, short for Light Detection and Ranging, is a remote sensing technology that uses laser pulses to measure distances. Remote sensing indicates we use sensors instead of actually physically measuring things with human hands. The sensors can provide information about the surroundings and record it.

A LiDAR works by emitting light from a rapidly firing laser. Therefore the main components of a typical LiDAR include a transmitter and a receiver [27]. Light pulses will be sent out, reflected off objects and received for interpretation. The distance information can be calculated by the time of light traversal or triangulation. The calculation methods can be seen in Figure 3.1.

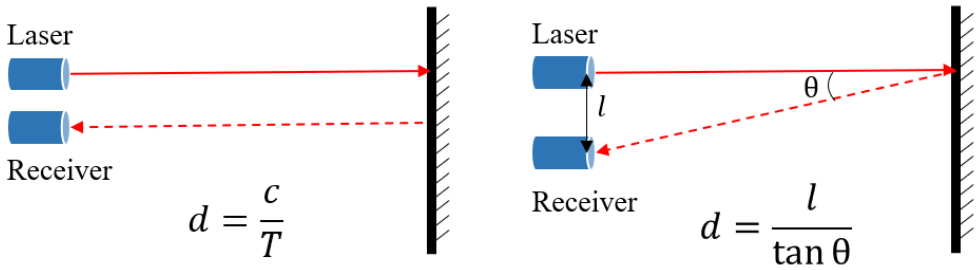

Figure 3.1: Time-of-Flight and Triangulation Distance Measure in LiDAR [1]

There are certain limitations of using LiDAR to achieve 3D visualization, as in outdoor environments bad weather can result in poor performance. For example, rain, fog or snow can make light reflect and make objects hard to define.

3.1.2 Stereo Vision

Stereo vision is a 3D information extraction method inspired by the way our eyes work to perceive depth and 3D information. Humans and other animals with binoc-

ular vision have two eyes that are slightly offset from each other, resulting in slightly different images being projected onto each retina. This is known as binocular disparity, which allows human brain to calculate the depth and 3D structure of objects in the scene. In the context of computer vision, stereo vision refers to the use of two or

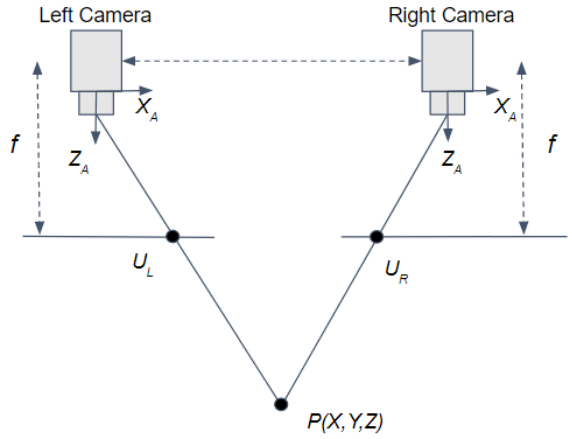

Figure 3.2: A Simplified Stereo Vision System [2]

more cameras to capture multiple images of a scene from different viewpoints, then use computer algorithms to match corresponding pixels in the images and calculate the disparities or differences between them. The resulting depth map can be used to reconstruct a 3D model of the scene. A simplified stereo vision system with two cameras can be seen in Figure 3.2.

The process of stereo vision involves several key steps, including camera calibration, image rectification, stereo matching, and 3D reconstruction. Camera calibration is the process of estimating the intrinsic and extrinsic parameters of the cameras to correct for lens distortion and align the images. Image rectification involves transforming the images so that corresponding pixels are aligned along the same scan line. Stereo matching involves finding correspondences between pixels in the left and right images based on features such as color, texture, or edge information. The disparity or difference between the corresponding pixels is then used to calculate the depth of the scene using triangulation. Finally, the depth map or point cloud is used to generate a 3D model of the scene.

The accuracy and quality of the 3D reconstruction from stereo vision depend on several factors, including the resolution and quality of the input images, the effectiveness of the stereo-matching algorithm, and the post-processing techniques used to filter and refine the point cloud. Many researchers have explored various techniques to improve the accuracy and efficiency of stereo vision, such as using machine learning algorithms for stereo matching or developing new post-processing techniques for filtering and smoothing the point cloud.

3.1.2.1 Passive Stereo

There are two fundamental approaches for stereo vision - passive and active stereo. They differ in the use of sensors and illumination sources.

Passive stereo uses passive sensors, typically a pair of cameras, to capture images of a scene. The camera itself does not provide any illumination so the scene is illuminated with ambient light, and the cameras capture the scene by reflected light. An advantage of passive stereo is that it is less expensive and requires less power than active stereo. The disadvantage is the quality of the depth information obtained from the passive stereo is dependent on the quality of the input images and can be affected by environmental factors such as lighting conditions and low contrast.

3.1.2.2 Active Stereo

Active stereo, on the other hand, uses active sensors such as structured light combined with multiple cameras to illuminate the scene with a known pattern of light. The sensors can then capture the reflected or emitted light and calculate the distance or depth of the objects. The advantage of active stereo is its ability to capture depth information in a wider range of environments and lighting conditions. However, it is more complex, expensive, and may require more power than passive stereo.

Although the working principle for active stereo cameras sounds very much like LiDAR sensors, they actually have a lot of differences. For example, LiDAR sensors use laser emission to calculate the distances while active stereo cameras use structured light patterns to capture images of the environment from different angles. In general, LiDAR provides more accurate and precise depth measurements than active stereo, especially over longer distances. LiDAR is also more complex and more expensive in price. Compared to LiDAR and passive stereo cameras, active stereo cameras are more suitable for scenarios requiring high data resolution and real-time processing. A comparison of active stereo and passive stereo can be seen in Figure 3.3.

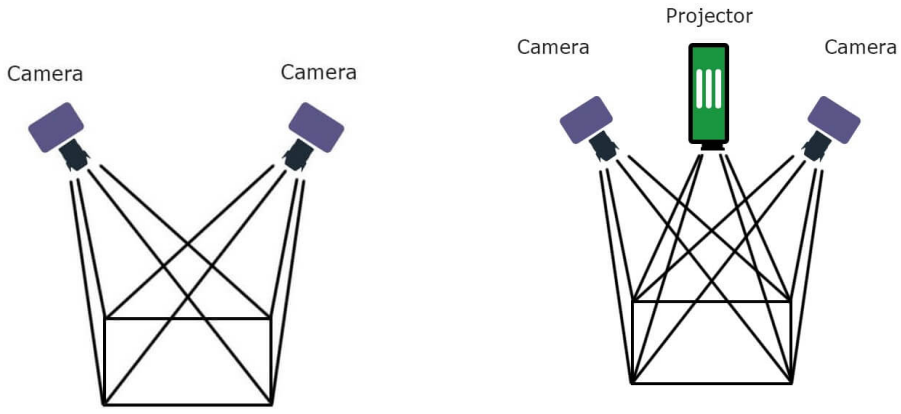

Figure 3.3: Passive and Active Stereo Camera [3]

3.1.3 Stereo Matching

Stereo matching is the crucial step in 3D reconstruction from stereo images. It refers to the process of finding correspondences between pixels in the left and the right images of a stereo image pair. The result of stereo matching is called a disparity map. Stereo matching can infer the depth of each point in the image by knowing the disparity, enabling 3D reconstruction of the scene.

There are several algorithms and methods for stereo matching. Scharstein et al. [28] developed a taxonomy scheme and evaluated several stereo matching algorithms in 2002. According to them, stereo algorithms generally perform the following four steps:

- 1. Matching cost computation
- 2. Cost aggregation
- 3. Disparity computation
- 4. Disparity refinement

The actual sequence and taken steps depend on the specific algorithm. For example, local (window-based) algorithms do not use disparity refinement, while global algorithms usually do not perform the aggregation step. Apart from algorithm methods, deep learning is also contributing to stereo matching tasks. The following are some popular stereo-matching methods.

- Sum of absolute differences (SAD): SAD is a simple and widely used method that calculates the absolute difference between the pixel values in the left and right images at each disparity level and selects the disparity with the minimum SAD.
- Normalized cross-correlation (NCC): NCC is similar to SAD but uses the normalized cross-correlation instead of the absolute difference to account for changes in brightness and contrast.
- Census algorithm: This method is proposed by Zabih and Woodfill [29] to calculate the correspondence in two steps. First, census transform is defined to describe the relationship between a pixel and its neighbors. The next step defines the displacement by taking the minimum of the Hamming distance sums for each horizontal line.
- Semi-global matching (SGM): SGM is a global optimization technique for stereo matching proposed by H. Hirschmuller [30]. It performs pixel-wise matching based on mutual information and approximation of global smoothness.
- Deep learning-based methods: Deep learning-based methods have shown promising results in stereo matching. convolutional neural networks (CNNs) are trained to perform a mapping function between the left and right images and output the result depth map.

3.1.3.1 The Modified H. Hirschmuller Algorithm (SGBM Algorithm)

In 2002, Scharstein et al. [28] implemented a stand-alone improved algorithm for stereo matching. In this algorithm, left and right images are divided into smaller blocks, a matching cost is calculated by comparing the corresponding pixel value. The matching cost is aggregated along different directions to form a cost volume. By using a bilateral filter, the cost volume can be filtered to preserve edges and reduce noise. Finally the disparity map is generated by finding the disparity value that has the minimum cost along a path in the cost volume.

The SGBM algorithm uses a semi-global optimization that considers matching cost along many directions rather than one direction. As a result, it achieved better performance in root-mean-squared (RMS) error compared to traditional block-matching algorithms.

In 2008, H. Hirschmuller came up with the Semiglobal Matching (SGM) method [30]. The modified H. Hirschmuller algorithm is an implementation of the SGM algorithm by H. Hirschmuller [30] in 2008, combined with a pixel dissimilarity measure proposed by S. Birchfield et al. [31]. The block size can shrink to 1 to represent a block with only one pixel while matching. It is based on pixel-wise matching of mutual information and approximating a global, 2D smoothness constraint by combining many 1D constraints. The matching cost is calculated by mutual information (MI), by the following equation.

$$MI_{I_1,I_2} = H_{I1} + H_{I2} - H_{I_1,I_2} \tag{3.1}$$

H here refers to the separate and joint entropies which are calculated from the probability distributions P of intensities of the associated images. Apart from a basic computation of matching cost, the method also applied a cost aggregation to add support for the smoothness and avoid noise. The first term is the sum of all pixel matching costs of disparity D. P_1 is a constant penalty for all pixels q in the neighborhood N_p of p. P_2 in the last term is a larger constant penalty for larger disparity changes.

$$\begin{aligned} E(D) = \sum_p (C(p, D_p) &+ \sum_{q \in N_p} P_1 T|D_p - D_q = 1| \\ &+ \sum_{q \in N_p} P_2 T|D_p - D_q = 1|) \end{aligned} \tag{3.2}$$

The modified H. Hirschmuller resulted in a runtime of just 1-2 seconds on typical test images, it also showed a tolerance against a wide range of transformations and worked well on reconstructions from the huge aerial frame images.

3.1.3.2 HITNet

HITNet is a framework developed by Google for neural network based depth estimation [4]. Compared to other CNNs that focus on increasing the efficiency of disparity estimation for active stereo while maintaining a high level of accuracy,

HITNet aimed to achieve efficient and accurate results opening up the possibility of using passive stereo.

The idea of HITNet is from traditional patch match methods. In HITNet, a tile hypothesis is defined as a plantar patch with a learnable feature. It can be described as a vector that encodes a slanted 3D plane. HITNet uses a module called U-Net to extract features on a multi-scale. These feature extractors provide a set of multi-scale feature maps that are used for initial matching and warping in the propagation stage.

HITNet first initialized a disparity and feature vector, then extract features by running a $4 * 4$ convolution on each extracted feature map. After the new set of feature maps is generated, it is able to compute the initial disparity. The propagation step takes tile hypotheses as input and outputs the refined tile hypotheses. The disparity is upsampled during updating using the tile plane equation and the reference tile. An overview of the HITNet framework can be seen in Figure 3.4.

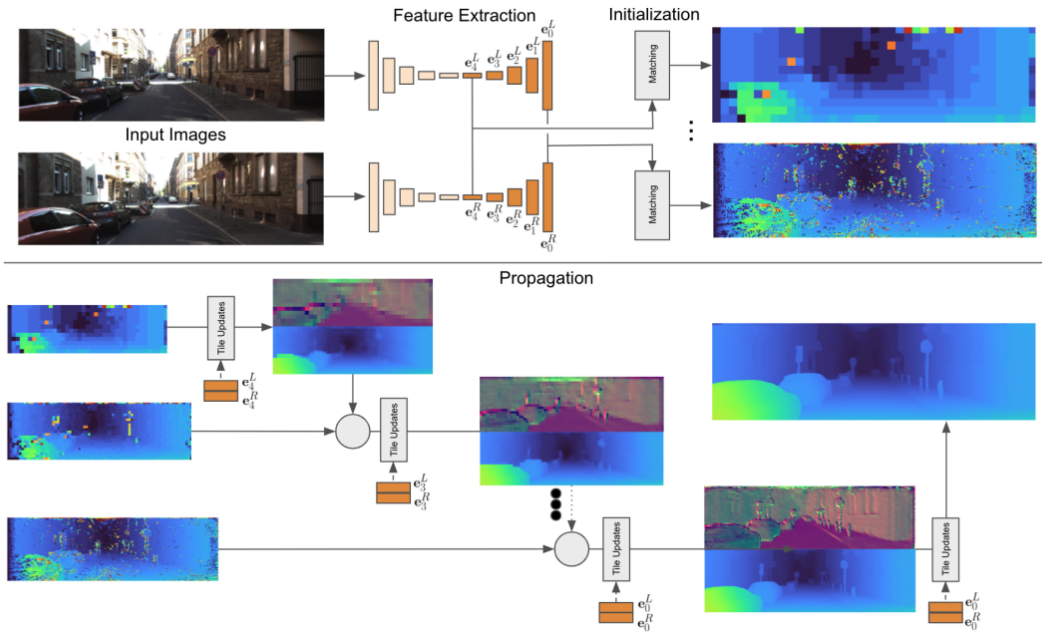

Figure 3.4: An overview of HITNet [4]

To conclude, HITNet uses a combination of multi-scale feature extraction, global context integration, and channel attention mechanisms to achieve state-of-the-art performance on several benchmark datasets. Compared to previous deep learning models for stereo matching, HITNet is particularly well-suited for low-resolution and small-object recognition tasks, while traditional CNNs may not perform well because of the limited amount of information available in the input images.

3.1.4 Interpolation

Interpolation is used for estimating values that lie between known data [32]. For spatial data, there are two main types of interpolation - point and areal data interpolation. According to Lam et al. 's taxonomy [33], point interpolation methods can

further be classified into exact and approximate methods. Exact methods refer to interpolation techniques that provide an exact estimate of the value at a target location based on the known data points, which include most distance-weighting methods, Kriging, spline interpolation, interpolating polynomials, and finite-difference methods. Approximate methods refer to techniques that provide an approximate value, which include power-series trend models, Fourier models, distance-weighted least-squares, and least-squares fitting with splines.

In this research, interpolation is applied to point cloud data generated from 3D visualization methods. It is used due to two reasons. The first reason is the point cloud after generation may not be dense enough to accurately represent the surface, another reason is that exact numbers of points are needed to measure performance during the evaluation step.

The interpolation methods that are used in this experiment to test the performance are listed as follows:

- Nearest Neighbor interpolation: Assume any unknown data point's value is the same as its nearest neighbor.
- Linear interpolation: Use linear polynomials to do curve fitting. To be specific, draw a straight line between two known nearest data points, then use the linear function to estimate the value of a point.
- Cubic interpolation: Cubic interpolation is another subcategory of polynomial interpolation. Instead of linear polynomials, it estimates values between known data points using cubic polynomials.

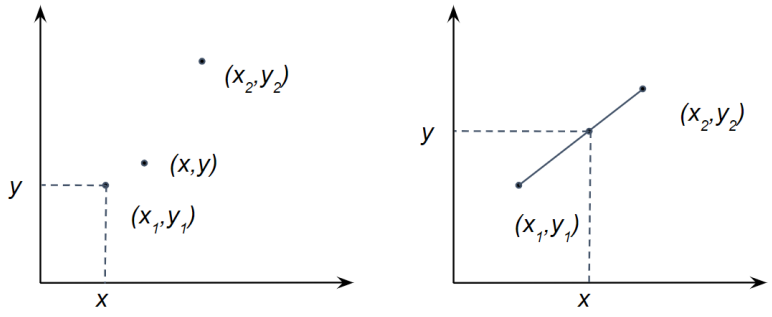

Figure 3.5: Illustration of Nearest Point Interpolation and Linear Interpolation

Figure 3.5 gives an example of nearest neighbor interpolation and linear interpolation in 2D space. In 3D space the procedure is similar. For instance, if a point $P(x, y, z)$ has a nearest neighbor $P_1(x_1, y_1, z_1)$, the value of P is the same as the value of P_1.

4

Methodology

This chapter presents a comprehensive study on soil surface visualization, encompassing data collection, evaluation, and identification of potential threats to research validity. A systematic experimental workflow has been designed around several key aspects, including the intricacies of data collection, the selection of appropriate metrics, and the evaluation of different methods. The primary objective is to address the research questions and determine the most suitable stereo matching and interpolation methods for soil surface visualization.

4.1 Data Collection

This section outlines the systematic approach used to collect the required data for the experiment. Detailed descriptions are provided regarding the construction of the scene, the development environment, and how different equipment was used.

The data collection consists of various steps, including the experimental setup, collection of the baseline point cloud, and generation of point clouds using different strategies. The overview process is shown in Figure 4.1, each step of which will be described in detail in the following sections. By following this process, high-quality and diverse data are made available for subsequent analysis and evaluation.

4.1.1 Experimental Setup

The experimental setup includes the utilization of specific equipment and a defined development environment to conduct the data collection process. The following equipment and tools were used:

- **iPhone 13 Pro**: The LiDAR sensor embedded within the iPhone was utilized to capture the baseline point cloud. It can help to measure distances and accurately capture 3D geometric information of the scene.
- **Stereo Camera ZED**: The ZED camera was utilized to generate point clouds using its built-in functions, as well as collect stereo image pairs for subsequent point cloud generation. The camera consists of two synchronized cameras that capture stereo image pairs, enabling depth perception and 3D reconstruction.
- **Camera Stand**: A camera stand was utilized to securely position and stabilize the ZED camera during data collection. This helped ensure consistent

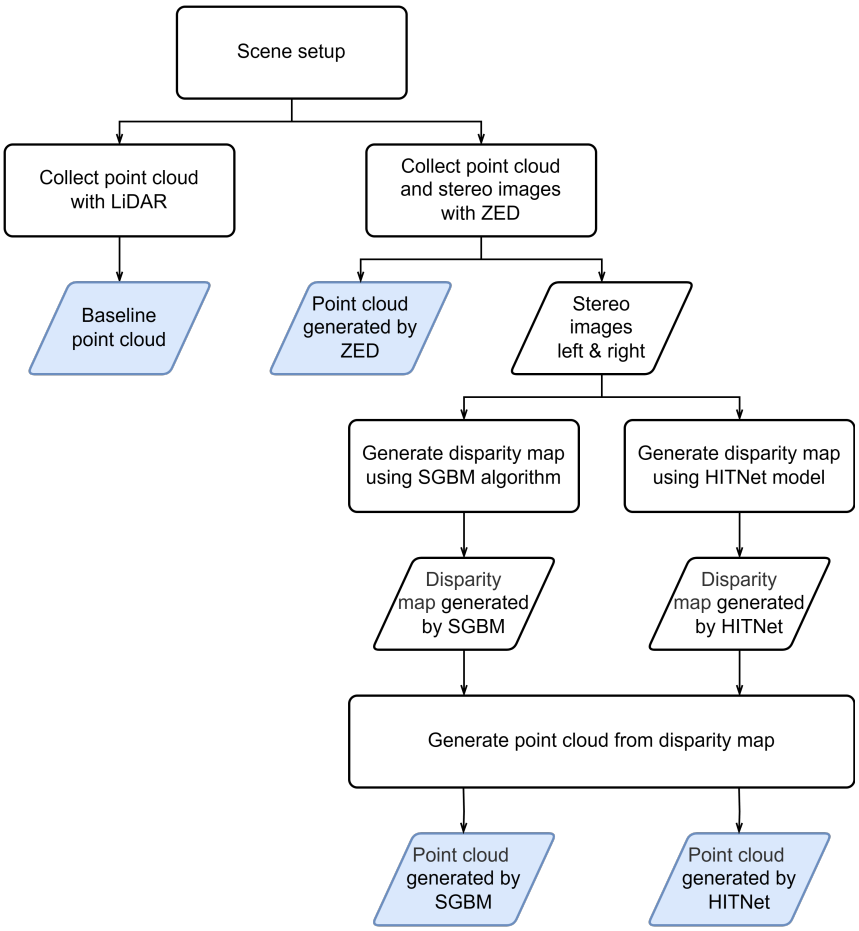

Figure 4.1: Overview of data collection process

camera positioning and minimized any undesired camera movement or vibrations.

The experiment was conducted in the following development environment:

- **CUDA 11.8**: CUDA is a parallel computing platform and application programming interface model developed by NVIDIA to harness the computational capabilities of the graphics processing unit.

- **ZED SDK 4.0**: ZED SDK is an official software framework provided by Stereolabs Inc. for capturing stereo image pairs and generating point clouds.

The indoor scene was constructed using 20 liters of actual soil, which was arranged to form a simulated surface with furrows and ridges, measuring 80cm in length and width, and 6cm in height, as shown in Figure 4.2. The four corners of the scene were marked to facilitate accurate positioning during data collection and following evaluation. The selection of actual soil aimed to capture realistic surface properties and enhance the authenticity of the indoor scene.

Figure 4.2: Indoor scene constructed with actual soil

4.1.2 Collect Point Cloud with LiDAR

Depth sensing on iPhone is based on a sensor fusion strategy, which integrates data from the camera and LiDAR sensor with other sensors such as the accelerometer, gyroscope, and magnetometer, to obtain more accurate and comprehensive information [34]. This strategy maximizes the strengths of each sensor and provides enhanced device orientation, motion tracking, and spatial awareness. According to the research conducted by Payton Chase et. al., the average error of the point cloud generated by the iPhone LiDAR is within 3cm on the X- and Y-axis and 7mm on the Z-axis [35]. These results highlight the accuracy and precision of the iPhone LiDAR sensor, demonstrating its reliability as a suitable baseline for further evaluation.

The scanning process utilized the Polycam, a mobile application specifically designed for capturing 3D scenes and generating point cloud data leveraging the LiDAR sensor on the phone. A 360-degree scan was conducted by systematically moving the camera around the scene, ensuring coverage of all details. The generated point cloud is shown in Figure 4.3, serving as the baseline for subsequent evaluation and comparison with point clouds generated using other methods. In this experiment, all point clouds were stored in files with the PCD format.

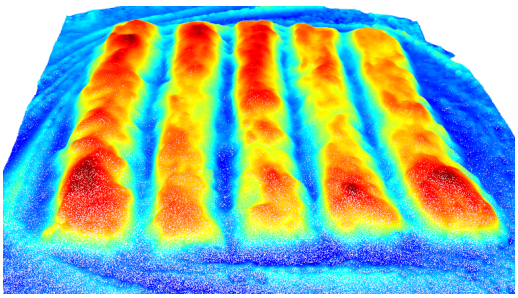

Figure 4.3: The point cloud collected with the iPhone LiDAR

4.1.3 Collect Point Cloud and Stereo Images with ZED

This section outlines the process of collecting point cloud and stereo images using the ZED camera. The initial camera configuration is listed in Table 4.1. The resolution was set to HD1080, ensuring high-quality stereo images and point cloud capture. The depth mode was selected as ULTRA for its ability to provide the

highest depth range and better Z-accuracy among all depth sensing modes supported by the ZED camera. The coordinate unit was set as meters, and the coordinate system followed a right-handed convention, with the Z-axis pointing upwards. These settings standardize the reference frame for point cloud generation, facilitating a more accurate evaluation.

Camera configuration	
Camera resolution	HD1080
Depth mode	ULTRA
Coordinate unit	Meter
Coordinate system	Right handed Z up

Table 4.1: Initial configuration for the ZED camera

The camera was installed at different heights and offset angles to capture data. As listed in Table 4.2, a total of eight frames of data were captured from two different heights with four different offset angles. Figure 4.4 illustrates how the ZED camera was applied for collecting required data. This approach ensures diverse and comprehensive data by capturing the scene from different perspectives.

Camera installation				
Height (cm)	50	65		
Offset angle (degree)	15	20	25	30

Table 4.2: Installation parameters for the ZED camera

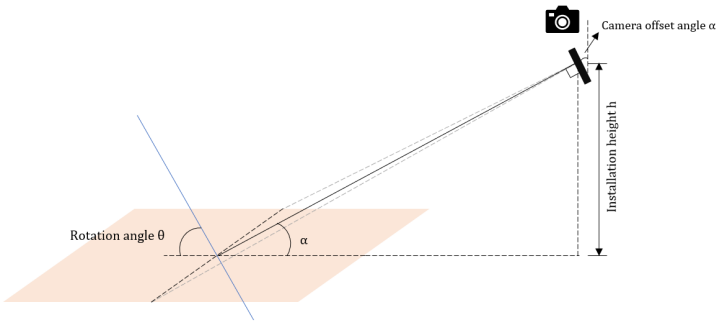

Figure 4.4: Illustration of data collection with the ZED camera

In this step, eight frames of data were collected and the average processing time was recorded. For each capture, calibration parameters of the camera obtained by the ZED SDK were also recorded, which is necessary for the subsequent steps that convert disparity maps to point clouds. A total of eight point clouds were collected and saved as PCD files, while the eight pairs of stereo images were saved as PNG files without compression. Algorithm 1 shows a code example for this step.

Algorithm 1 Collect data with ZED camera

```
import pyzed.sl as sl                                   ▷ import ZED SDK python package
zed = sl.Camera()
initParams = sl.InitParameters(selected parameter)
zed.open(initParams)                                    ▷ open the camera
zed.grab()                                              ▷ capture a frame
leftImage, rightImage, pcd = sl.Mat()
zed.retrieve_image(leftImage, sl.VIEW.LEFT)             ▷ retrieve the left image
zed.retrieve_image(rightImage, sl.VIEW.RIGHT)           ▷ retrieve the right image
zed.retrieve_measure(pcd, sl.MEASURE.XYZ)               ▷ retrieve the point cloud
Save leftImage and rightImage to PNG files
Save pcd to PCD file
zed.get_camera_information().camera_configuration.calibration_parameters
Save calibration parameters to TXT file
Record the processing time
```

4.1.4 Generate Disparity Map Using SGBM Algorithm

The SGBM algorithm is widely used for computing disparity maps from stereo image pairs. It works by comparing corresponding blocks of pixels in the left and right images to estimate the disparity, providing depth information in the form of a disparity map. The SGBM algorithm can be implemented through various libraries, and this experiment adopted the *StereoSGBM_create* function provided by the OpenCV Python library. The meanings of the different parameters used in the function and their selected values are summarized in Table 4.3.

Name	Description	Selected value
minDisparity	Minimum disparity value	16
numDisparities	Number of disparity levels	176
blockSize	Size of the matching block	3
P1	Penalty for disparity smoothness	600
P2	Penalty for disparity smoothness	2400
disp12MaxDiff	Maximum allowed difference in disparity	200
uniquenessRatio	Uniqueness ratio threshold	1
speckleWindowSize	Window size for disparity speckle filtering	3
speckleRange	Disparity difference threshold for speckle filtering	12

Table 4.3: Parameters for the StereoSGBM_create function

Values for different parameters were selected based on empirical evaluation and considering their impact on the quality of the generated disparity map, aiming to achieve a balance between accuracy and smoothness. Please note that these parameter values are specific to this experiment and scene conditions. Adjustments may be necessary for different datasets or application scenarios.

In this step, the raw data of eight generated disparity maps were saved in TXT files, and the average processing time was recorded. Algorithm 2 shows a code example for this step.

Algorithm 2 Generate disparity map using SGBM algorithm

Require: leftImagePath - Path to the left image
rightImagePath - Path to the right image

```
import cv2
leftImage = cv2.imread(leftImagePath)                    ▷ load left image
rightImage = cv2.imread(rightImagePath)                  ▷ load right image
stereo = cv2.StereoSGBM_create(selected parameters)
disparityMap = stereo.compute(leftImage, rightImage)
Save disparityMap to TXT file
Record the processing time
```

4.1.5 Generate Disparity Map Using HITNet Model

HITNet is a deep learning model with the key advantages of high accuracy and robustness in generating disparity maps. There are numerous pretrained open-source models available, enabling direct usage without the need for extensive training. In this experiment, an open-source HITNet model trained on the Middlebury dataset was used. Serving as a widely recognized benchmark dataset for stereo matching and disparity estimation, the Middlebury dataset offers diverse and challenging stereo image pairs with ground truth disparity maps, enhancing the reliability and generalizability of the model, and ensuring its good performance in various scenarios.

This experiment leveraged an open-source pretrained HITNet model as well as codes for using this model to predict disparity maps [36]. After loading the pretrained model and providing the stereo image pair as input, the model then predicts the disparity values for each pixel, producing the output disparity map. In this step, the raw data of eight generated disparity maps were saved in TXT files, and the average processing time was recorded.

4.1.6 Generate Point Cloud from Disparity Map

Converting disparity maps to point clouds is based on the triangulation principle. Disparity values represent the difference in pixel positions between left and right images. By leveraging these differences and knowing the camera calibration parameters, the depth of each pixel can be estimated, then projected into a 3D space to construct the point cloud. Figure 4.5 illustrates the process of converting a disparity map to a point cloud in this experiment.

Firstly, a depth map was generated from the disparity map using equation 4.1, in which *baseline* is the distance between the left and right cameras, and f_x refers to the focal length value of the camera lens in the horizontal direction. In the experimental scenario, the ZED camera has a baseline length of 12cm, and the value of f_x could be retrieved from the saved calibration parameters from the previous steps. The resulting depth map represents the distance of each pixel in the left image.

$$Depth = (baseline \cdot f_x)/Disparity \tag{4.1}$$

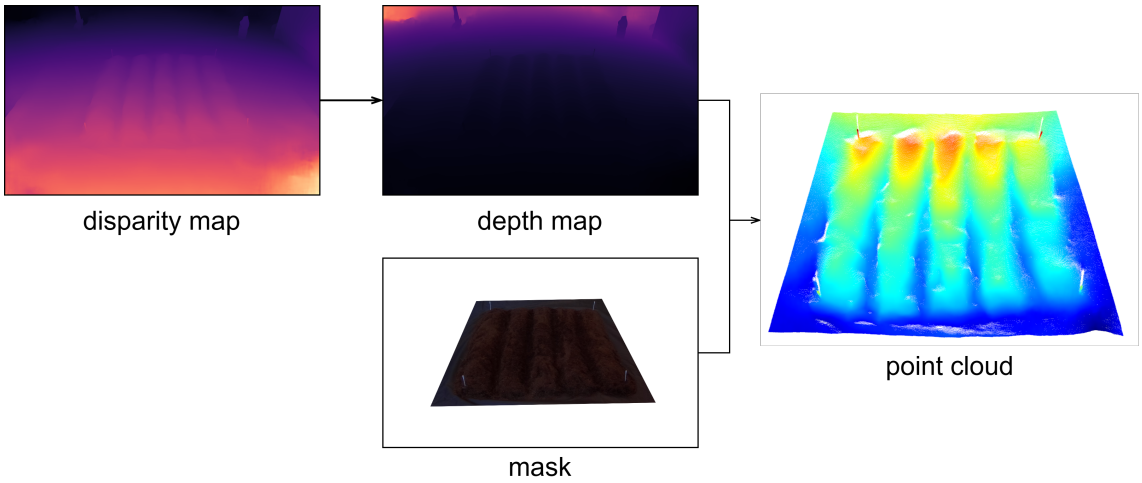

Figure 4.5: Process of generating point cloud from disparity map

Due to the limited area of soil, the camera also captured some irrelevant areas. In order to focus on a specific region of interest and reduce the processing time, a polygonal area that only contains the soil surface was manually marked out from the left image and saved as a mask. Only pixels within the defined mask region will participate in subsequent calculations.

Finally, the depth map was projected into a 3D space to generate the point cloud, which necessitates the determination of the x, y, and z values using the depth map and calibration parameters. As shown in Equation 4.2, the 3D z values were directly read from the depth values, while 3D x the and 3D y values needed some calculations based on the 2D x and y values, 3D z values, optical center $[C_x, C_y]$, and focal length (f_x, f_y).

$$\begin{aligned} z_{3D} &= Depth \\ x_{3D} &= \frac{(x_{2D} - C_x) \cdot z_{3D}}{f_x} \\ y_{3D} &= \frac{(y_{2D} - C_y) \cdot z_{3D}}{f_y} \end{aligned} \tag{4.2}$$

In this step, the two sets of eight disparity maps, respectively generated by the SGBM algorithm and HITNet model, were converted to two sets of point clouds and saved as PCD files. Algorithm 3 shows a code example for this step.

4.2 Evaluation

This section focuses on the evaluation of different interpolation and stereo matching methods. It outlines the various steps conducted to assess the performance of each method, as well as provides guidelines for data preprocessing and metric selection. Figure 4.6 illustrates the overall workflow, each step of which will be described in detail in subsequent sections.

Algorithm 3 Generate point cloud from disparity map

Require: caliPath - Path to the calibration parameters (.txt format)
dispMapPath - Path to the disparity map raw data (.txt format)

```
import numpy as np
baseline = 0.12
[fx, fy, cx, cy] = np.loadtxt(caliPath)          ▷ load calibration parameters
dispMap = np.loadtxt(dispMapPath)                ▷ load disparity map
depthMap = baseline * fx / dispMap               ▷ calculate depth map
pcd = [ ]                                        ▷ calculate point cloud
for i in range(0, depthMap.height):
    for j in range(0, depthMap.width):
        if (i, j) in masked region:
            z = depthMap[i, j]
            x = (i - cx) * z / fx
            y = (j - cy) * z / fy
            pcd.append((x, y, z))
Save pcd to PCD file
```

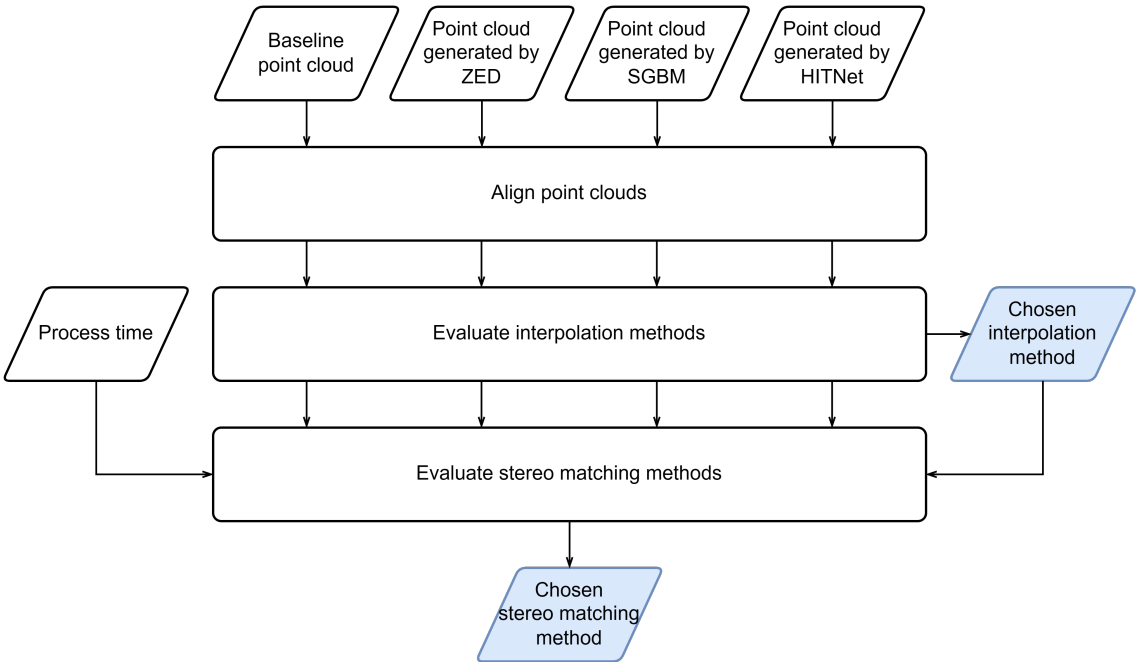

Figure 4.6: Overview of evaluation process

4.2.1 Align Point Clouds

The alignment of collected point clouds is a critical step in the evaluation process to ensure accurate comparisons and reliable analysis. This section presents the three key operations involved in aligning the point clouds: rotation, cropping, and affine transformation. Each operation is accompanied by its underlying mathematical principles and code examples.

As introduced in Section 4.1.3, all the point clouds in this experiment followed a standardized coordinate system, with the origin located at the left camera lens. As shown in Figure 4.7, the X-axis is defined as the line connecting the left and right camera lenses, pointing from the left to the right, the Y-axis is aligned with the line of sight, pointing towards the direction of observation, and the Z-axis is oriented upward. Knowing this lays a foundation for applying the following transformations on the collected point clouds.

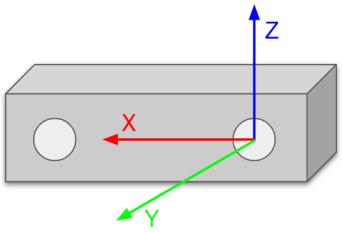

Figure 4.7: Coordinate system of the collected point clouds

To compensate for varying offset angles when capturing the data, the first operation is rotating the point clouds to align them parallel to the XY-plane. This was performed by applying a transformation matrix that rotates the point cloud clockwise around the X-axis, as illustrated in Figure 4.8. Equation 4.3 shows the matrix, in which θ is the camera offset angle.

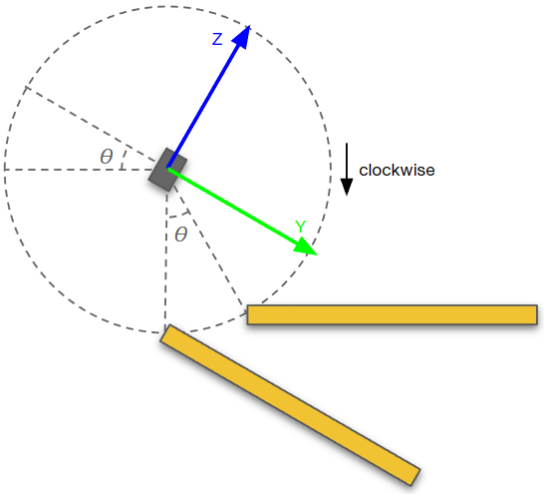

Figure 4.8: Illustration of the rotation transformation

$$R = \begin{bmatrix} 1 & 0 & 0 \\ 0 & cos\theta & sin\theta \\ 0 & -sin\theta & cos\theta \end{bmatrix} \tag{4.3}$$

The second operation is cropping the point clouds to ensure all of them correspond to the same region of the scene. A quadrilateral region on the XY-plane was defined using the positioning markers at the four corners, as mentioned in Section 4.1.1.

Only points within this region would be kept while the others be removed. To determine whether a point lies within the region, vector cross products can be applied. Consider a convex quadrilateral with vertices A, B, C, and D, and a random point P. Calculate the cross products of the vectors formed by point P with each consecutive pair of vertices, such as $\boldsymbol{AP} \times \boldsymbol{AB}$, $\boldsymbol{BP} \times \boldsymbol{BC}$, $\boldsymbol{CP} \times \boldsymbol{CD}$, and $\boldsymbol{DP} \times \boldsymbol{DA}$, if all these cross products have the same sign, it indicates that point P is located within the quadrilateral. Equation 4.4 shows the mathematical expression of the condition above.

$$\begin{aligned} (\boldsymbol{AP} \times \boldsymbol{AB}) \cdot (\boldsymbol{BP} \times \boldsymbol{BC}) \geq 0 \\ (\boldsymbol{BP} \times \boldsymbol{BC}) \cdot (\boldsymbol{CP} \times \boldsymbol{CD}) \geq 0 \\ (\boldsymbol{CP} \times \boldsymbol{CD}) \cdot (\boldsymbol{DP} \times \boldsymbol{DA}) \geq 0 \\ (\boldsymbol{DP} \times \boldsymbol{DA}) \cdot (\boldsymbol{AP} \times \boldsymbol{AB}) \geq 0 \end{aligned} \tag{4.4}$$

The final operation is to affine transform the cropped quadrilateral region into a square region on the XY-plane, which can align the point cloud and enables direct comparison and evaluation in subsequent steps. The affine transformation is a linear transformation that preserves parallelism and ratios of distances, which can be represented by the matrix shown in Equation 4.5, where $[a, b]$ and $[d, e]$ are scaling factors, $[c, f]$ are translation factors.

$$T = \begin{bmatrix} a & b & c \\ d & e & f \\ 0 & 0 & 1 \end{bmatrix} \tag{4.5}$$

In this experiment, all point clouds were affine transformed to a square area on the XY-plane with a side length of 1m. As illustrated in Figure 4.9, consider the original quadrilateral with vertices $A(x_1, y_1)$, $B(x_2, y_2)$, $C(x_3, y_3)$, and $D(x_4, y_4)$, and the corresponding vertices of the square area after transforming are $P_1(0, 0)$, $P_2(1, 0)$, $P_3(1, 1)$, and $P_4(0, 1)$, based on the transformation principle, the parameters in Equation 4.5 can be calculated by Equation 4.6.

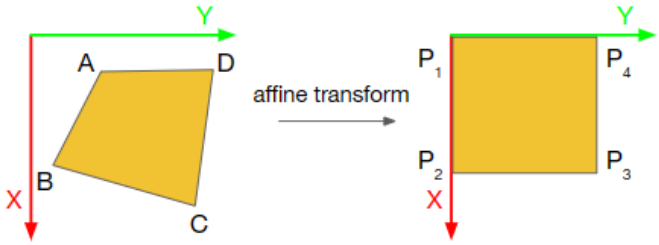

Figure 4.9: Illustration of the affine transformation

$$\begin{bmatrix} x_1 & y_1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & x_1 & y_1 & 1 \\ x_2 & y_2 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & x_2 & y_2 & 1 \\ x_3 & y_3 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & x_3 & y_3 & 1 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \\ d \\ e \\ f \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 1 \\ 1 \end{bmatrix} \tag{4.6}$$

Figure 4.10 illustrates the complete process of the aligning step, using the frame captured from a height of 65cm and an offset angle of 25 degrees. By implementing these alignment operations based on mathematical principles, all the collected point clouds were brought into a consistent and comparable state, minimizing possible errors, increasing the reliability of subsequent evaluations, as well as enabling accurate comparisons between different stereo matching methods.

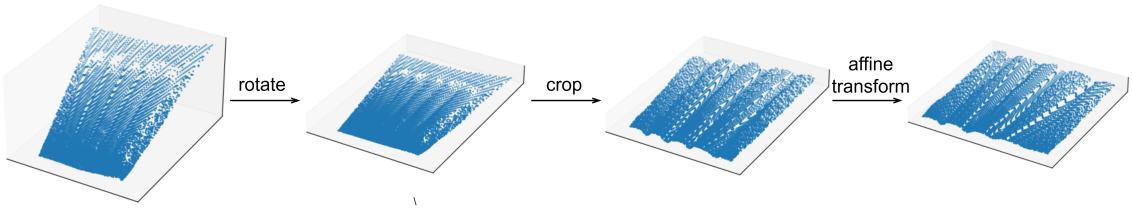

Figure 4.10: Process of aligning a point cloud

4.2.2 Evaluate Interpolation Methods

In this experiment, the three most common interpolation methods were chosen for comparison: nearest neighbor, linear, and cubic. Each of them has its own advantages and is suitable for specific scenarios. The nearest neighbor method is suitable for preserving sharp edges and discontinuities in the data, the linear method provides a smooth transition between neighboring points and is commonly used for uniformly distributed data, while the cubic method offers a more accurate representation of the underlying data, especially when the data has complex variations.

For point clouds collected by different strategies, it is important to evaluate the interpolation methods separately. This is because different collection strategies can cause variations in point cloud density and distribution, as shown in Figure 4.11, which may impact the performance of interpolation methods. By evaluating them individually, the best interpolation method can be assessed for each specific point cloud collection approach.

The metric adopted for evaluating interpolation methods is:

- **Mean absolute error (MAE)**: MAE measures the average difference between the predicted values and the actual values. It can be calculated by Equation 4.7, where n is the total number of points, y_i is the actual value for the i-th data point, and $\hat{y}_i$ is the predicted value for the i-th data point.

  Reason of selection: By using MAE, the overall accuracy of the interpolated results can be quantified, making it feasible to assess the performance of different methods and find the most reliable one.

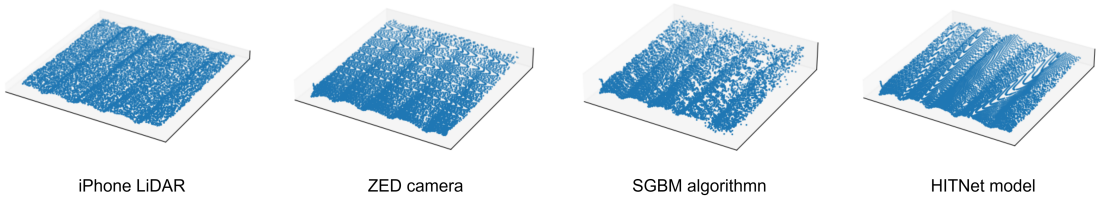

Figure 4.11: Examples of point clouds collected by different strategies

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \tag{4.7}$$

The evaluation was conducted following the steps below:

- For point clouds collected by each strategy, including the iPhone LiDAR sensor and the three stereo matching methods, all points were divided into training and testing sets using a 7:3 ratio.
- Based on training data, the three interpolation methods were employed respectively to predict the points in the testing sets.
- For each collection strategy, three MAE values representing three interpolation methods were calculated based on the predicted values and the ground truth values from testing sets.
- The interpolation method with the lowest MAE was selected as the best one for each specific collection strategy.

This evaluation process ensures a comprehensive comparison of different interpolation methods and enables the determination of the most accurate method for point clouds collected by each strategy. Algorithm 4 shows a code example for this step.

Algorithm 4 Calculate MAE for a specific interpolation method

Require: pcds - A list of point clouds
method - The chosen method for interpolation

```
import numpy as np
from scipy.interpolate import griddata
maeSum = 0
for pcd in pcds:
    trainXY, trainZ = Randomly pick 70 percent of the points in pcd
    testXY, testZ = The rest 30 percent of the points
    interpRst = griddata(trainXY, trainZ, testXY, method)
    maeSum += np.mean(np.abs(testZ - interpRst))
mae = maeSum/len(pcds)
```

4.2.3 Evaluate Stereo Matching Methods

In this experiment, three methods with different characteristics were used for stereo matching: the built-in functions provided by the ZED camera featuring convenience

and real-time performance due to the hardware integration, the SGBM algorithm with a good balance between accuracy and computational efficiency, and the HITNet deep learning model with a potential to achieve high accuracy by leveraging learned features.

As illustrated in Figure 4.11, different methods can produce point clouds with varying densities and distributions, making direct comparisons of the original values impossible. Then interpolation can be utilized to obtain the Z-values of each point on a standardized XY-grid, ensuring a consistent and fair basis for comparison. In this experiment, interpolated Z-values for all point clouds were obtained using the selected method determined in Section 4.2.2, with a standardized grid size of 200*200.

The metrics adopted for evaluating stereo matching methods are:

- **Time**: The total time spent from capturing a frame to generating the corresponding point cloud.

  Reason of selection: Evaluation with time enables the assessment for efficiency and real-time capability of different stereo matching methods, which is particularly significant for the autonomous system discussed in this research, where the ability to generate point clouds quickly is essential for timely decision-making and response.

- **Root Mean Squared Error (RMSE)**: RMSE measures the average magnitude of errors between the predicted values and the actual values. It can be calculated by Equation 4.8, where n is the total number of points, y_i is the baseline value for the i-th data point, and $\hat{y}_i$ is the collected value of each stereo matching method for the i-th data point.

  Reason of selection: Similar to MAE, RMSE enables quantification of the stereo matching accuracy, but it takes into account both the magnitude and distribution of errors, allowing for a more comprehensive and accurate assessment, which makes it more suitable for evaluating the stereo matching accuracy that requires higher precision.

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \tag{4.8}$$

The evaluation was conducted following the steps below:

- The best interpolation method determined for each collection strategy, as stated in Section 4.2.2, was used to interpolate the baseline and generated point clouds onto a standardized XY-grid of 200*200.
- The RMSE values between the point clouds generated by different stereo matching methods and the baseline point cloud were then calculated, using the points on the standardized XY-grid.
- The optimal stereo matching method was determined considering both of the selected metrics: time and RMSE.

Figure 4.12 illustrates the calculation of RMSE value for a point cloud generated by ZED built-in functions, and Algorithm 5 shows a code example for it. For each stereo matching method, eight point clouds collected from different perspectives were compared to the baseline, resulting in a total of eight RMSE values. The average of each set of RMSE values was then calculated and used to represent the performance of the corresponding method.

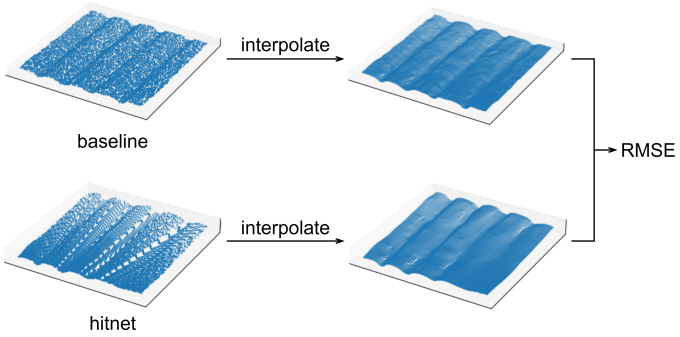

Figure 4.12: Illustration for the calculation of RMSE

Algorithm 5 Calculate RMSE for a specific point cloud

Require: baseline - The baseline point cloud
pcd - The point cloud to be compared
bslMethod - The selected interpolation method for baseline point cloud
pcdMethod - The selected interpolation method for generated point cloud

```
1: import numpy as np
2: from scipy.interpolate import griddata
3: gridXY = Grid of size 200*200 on 1m*1m region
4: bslXY, bslZ = All points in baseline
5: bslZInterp = griddata(bslXY, bslZ, gridXY, bslMethod)
6: pcdXY, pcdZ = All points in pcd
7: pcdZInterp = griddata(pcdXY, pcdZ, gridXY, pcdMethod)
8: rmse = np.mean(np.square(bslZInterp - pcdZInterp))
```

4.3 Threats to Validity

This section addresses potential threats to both internal and external research validity, as well as corresponding mitigation strategies. Validity threats refer to factors that can impact the accuracy, reliability, and generalizability of the study results. Understanding these threats is crucial for ensuring the reliability and credibility of the research.

Internal validity threats refer to factors that may compromise the ability to establish a causal relationship between variables. Here are several threats to internal validity identified in this experiment:

- **Variability in the experimental environment**: This includes factors such as variation in scene brightness, changes in camera settings, and effects of shadows and light reflections. These variations in the experimental conditions can potentially impact the accuracy and reliability of the results. Therefore, it is essential to control these environmental changes to ensure internal validity.
- **Biases in the data collection process**: Biases can arise from various sources, including inconsistent measurement techniques, human errors in data recording, or systematic differences in data collection between different observers. These biases can introduce inaccuracies and distortions in the collected data. To mitigate this threat, rigorous data collection protocols and standardized procedures should be ensured.

External validity threats refer to factors outside the control of the study that may impact the generalizability of the findings. Here are several threats to external validity identified in this experiment:

- **Selection of the soil sample**: Due to the variation in factors such as soil type, composition, and presence of contaminants like rocks, roots, or debris, the selection of the soil can significantly influence the outcomes and generalizability of the research. As a result, careful consideration and representative sampling of soil are crucial to minimizing this external threat.
- **Shaping of the soil surface**: Soil surfaces can exhibit diverse shapes depending on the environmental conditions and specific contexts. For instance, the soil surface in agricultural fields may have a relatively flat and uniform shape due to regular tilling and leveling practices. Therefore, it is important to consider the shapes of the soil surface in real-world scenarios, in order to select appropriate visualization methods that can accurately capture and represent the specific characteristics of the surface.

5

Results

This chapter presents the findings and outcomes of the experiment. It encompasses the exhibition of collected data, the evaluation of interpolation methods, the evaluation of stereo matching methods, and examples of visualization results. Evaluation metrics, including MAE, RMSE, and time, are used to assess the performance of different interpolation and stereo matching methods, providing insights into the strengths and limitations of each method. Furthermore, the results of visualization are presented, demonstrating the capability of the selected techniques in capturing the shape of the soil surface.

5.1 Collected Data

This section provides an overview of the collected experimental data, including the baseline point cloud collected through the iPhone LiDAR, the point clouds generated using different stereo matching methods, the processing time, and also the intermediate data generated and stored during the collection process.

Different types of data collected during the experiment are listed below, along with the respective descriptions and characteristics:

- **Stereo images**: The experiment involved capturing eight pairs of left and right images using the ZED camera. The camera installation parameters for each capture are listed in Table 5.1, and the image pairs collected by each capture are shown in Figure 5.1. Each image has a resolution of 1920*1080 and was saved in PNG format without compression. These stereo images provide the visual information required for subsequent depth estimation and point cloud generation.

	Capture							
	1	2	3	4	5	6	7	8
Height (cm)	50	50	50	50	65	65	65	65
Offset angle ($degree$)	15	20	25	30	15	20	25	30

Table 5.1: Installation parameters for each capture

- **Camera calibration parameters**: The calibration parameters of the ZED camera include the horizontal and vertical focal length of its lens, and the coordinates of its optical center. They were stored in a TXT file and used to

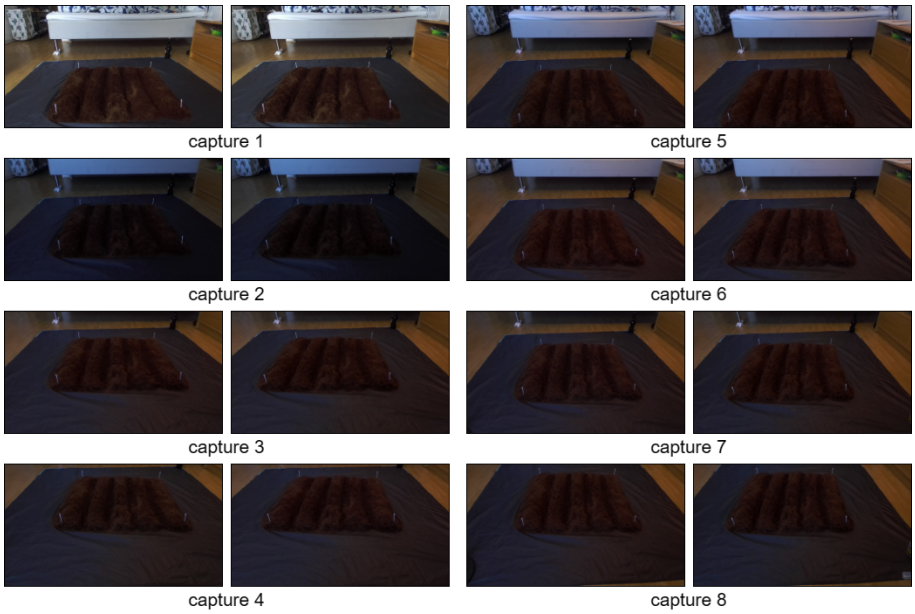

Figure 5.1: Stereo images collected by each capture

map the disparity values to a 3D space, constructing the point cloud. Table 5.2 displays these parameters.

Camera calibration parameters	
Horizontal focal length (*pixel*)	1.43e+03
Vertical focal length (*pixel*)	1.43e+03
X coordinate of the optical center (*pixel*)	9.79e+02
Y coordinate of the optical center (*pixel*)	5.96e+02

Table 5.2: Calibration parameters for the ZED camera

- **Disparity and depth maps**: For each pair of stereo images, a disparity map and a depth map were generated using the SGBM algorithm and HITNet model, both of which have the same resolution as the stereo image. In order to preserve the precision of the values, the raw data were saved in TXT files. Figure 5.2 shows the generated disparity maps using two different methods with the frame captured from a height of 65cm and an offset angle of 25 degrees. It is visible that the result of the HITNet model is smoother and more accurate compared to the one obtained using the SGBM algorithm, but further evaluation is still needed.
- **Point clouds**: One baseline point cloud was collected with the iPhone LiDAR, and for each stereo matching method, eight point clouds were generated from the corresponding stereo images. All the point clouds were aligned onto a standardized 1m*1m region for consistent evaluation and saved in PCD format, following the steps described in Section 4.2.1.
- **Processing time**: The processing time from starting the camera to generating the point cloud was manually recorded for each capture and stereo matching method. These data will be presented in the following section.

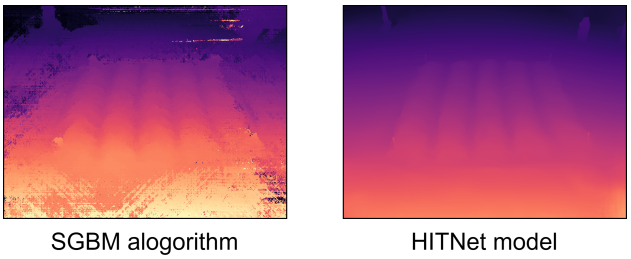

Figure 5.2: Generated disparity maps using different approaches

5.2 Evaluation of Interpolation Methods

The selected metric to evaluate different interpolation methods is MAE. Following the instructions in Section 4.2.2, the MAE values for point cloud collected by different strategies, including the baseline point cloud collected by iPhone LiDAR and the point clouds generated by the three stereo matching methods, were calculated for each interpolation method. The results are presented in Table 5.3, in which Nearest stands for the nearest neighbor method, and the MAE values of each capture with a 95% confidence interval, indicated by a line segment, are illustrated in Figure 5.3 to 5.5.

	MAE Average (m)			MAE Variance (m^2)		
	Nearest	Linear	Cubic	Nearest	Linear	Cubic
Baseline point cloud	3.25e-04	6.48e-05	4.94e-05	-	-	-
Built-in functions of ZED	6.76e-04	4.05e-04	4.98e-04	1.14e-08	8.35e-09	1.82e-08
SGBM algorithm	5.99e-03	5.90e-03	8.44e-03	7.81e-06	9.41e-06	1.96e-05
HITNet model	4.00e-04	8.33e-05	9.67e-05	1.59e-09	1.34e-10	3.14e-10

Table 5.3: Average and variance of MAE values for each collection strategy

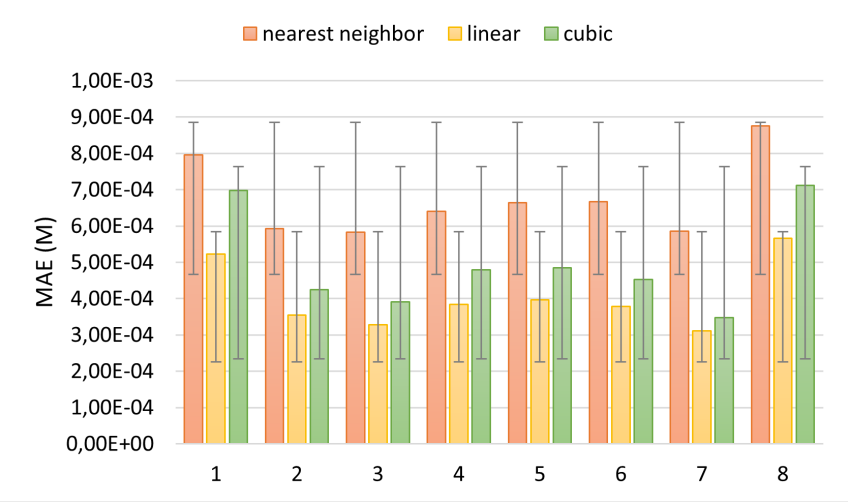

Figure 5.3: MAE values of each capture with a 95% confidence interval for the built-in functions of ZED

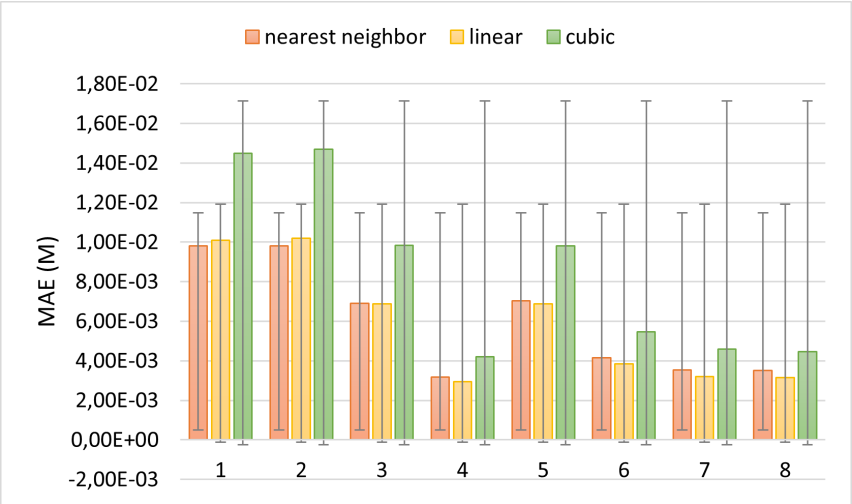

Figure 5.4: MAE values of each capture with a 95% confidence interval for the SGBM algorithm

According to Table 5.3, for the baseline point cloud which only contains one capture, the cubic method shows the highest accuracy because of its lowest average error, which can be due to its ability to capture more intricate variations in dense and evenly distributed points. For point clouds generated by built-in functions of ZED and HITNet model, the linear method has both the lowest average and variance, showing its capability of producing the most accurate and consistent results. For point clouds generated by SGBM algorithm, the nearest neighbor and the linear methods have the lowest average with a significantly small difference, but the former has a lower variance, indicating its predictions are more consistent and less scattered.

Figure 5.3 and Figure 5.5 show that for point clouds generated by the built-in functions of ZED and the HITNet model, the linear method produces results with the highest accuracy under all circumstances, while the nearest neighbor method always produces the lowest. And the confidence intervals also show the linear method results in the narrowest range and lowest uncertainty. This can be related to the point cloud distribution. The point clouds generated by these two methods contain regions with sparse or unevenly distributed points, as shown in Figure 4.11, and since the nearest neighbor method simply assigns the value of the nearest point, it can result in larger errors in those regions. On the other hand, both the linear and cubic methods predict missing values based on the neighboring points, which can help reduce the error and result in a more accurate estimation, but the cubic method may introduce more noise than the linear one due to its higher complexity.

Figure 5.4 shows that for point clouds generated by the SGBM algorithm, the nearest neighbor and linear methods perform the best almost equally for all captures, while the cubic method always results in the worst prediction accuracy. And the confidence intervals indicate the nearest neighbor method results in the narrowest range and lowest uncertainty. As shown in Figure 4.11, the point cloud generated by the SGBM algorithm contains a significant amount of noise, reducing the overall accuracy. However, the nearest neighbor and linear methods are generally more

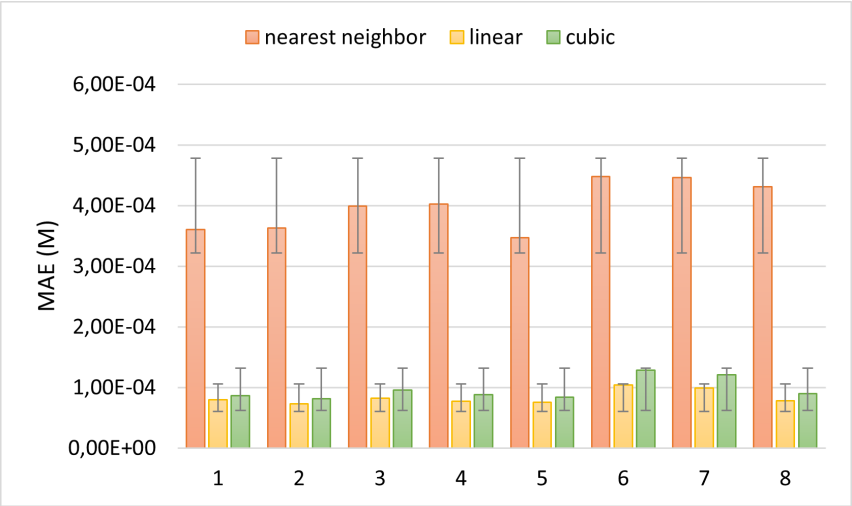

Figure 5.5: MAE values of each capture with a 95% confidence interval for the HITNet model

robust to noise compared to the cubic one. This is because the former two methods directly use nearby points for prediction, mitigating the impact of noise, while the nearest neighbor method performs better in cases where the points exhibit variations or non-linear relationships.

In conclusion, the cubic method will be selected as the most suitable one for the baseline point cloud, the linear method will be selected for the point clouds generated by the built-in functions of ZED and HITNet model, and the nearest neighbor method will be selected for those generated by th SGBM algorithm.

5.3 Evaluation of Stereo Matching Methods

The selected metrics to evaluate different stereo matching methods are time and RMSE. The RMSE values are computed based on the methodology described in Section 4.2.3. To facilitate the evaluation process, the point clouds are first interpolated to generate standard point clouds containing 200*200 points, utilizing the respectively selected interpolation methods. Subsequently, the RMSE is calculated by comparing the baseline point cloud with each specific point cloud generated by different methods. The average and variance of time and RMSE values are presented in Table 5.4, and the time and RMSE values across various captures with a 95% confidence interval, indicated by a line segment, are illustrated in Figure 5.6 and Figure 5.7.

	Time Average (s)	Time Variance (s^2)	RMSE Average (m)	RMSE Variance (m^2)
Built-in functions of ZED	2.25	0.22	1.30e-02	9.70e-06
SGBM algorithm	4.51	0.26	3.18e-02	1.72e-04
HITNet model	26.97	4.87	1.35e-02	1.59e-05

Table 5.4: Average and variance of time and RMSE value for each stereo matching method

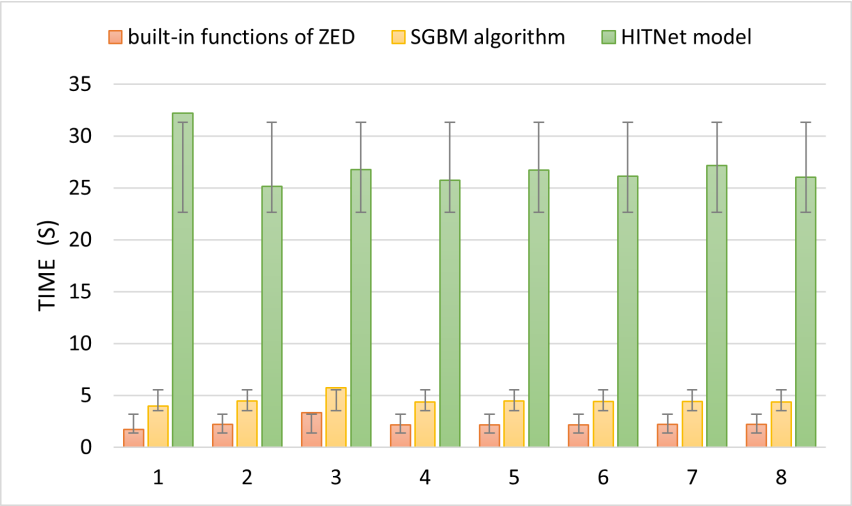

Figure 5.6: Processing time with 95% confidence interval of each capture

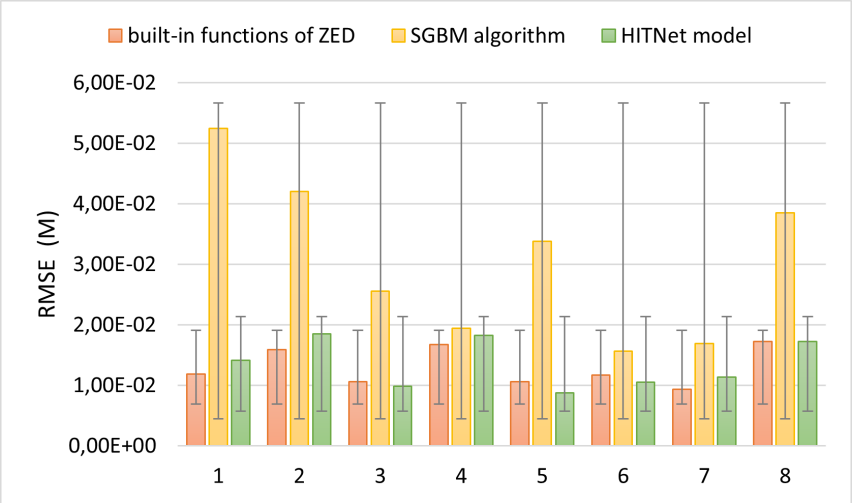

Figure 5.7: RMSE values with 95% confidence interval of each capture

According to Table 5.4, the average and variance of processing time for the HITNet model is significantly higher than the other two methods, with the built-in functions of ZED being the lowest in both way. The same result can be observed in all captures, as illustrated in Figure 5.6. This can be due to the complex neural network architectures and extensive computations of the HITNet model, which require more time to process a large amount of data and perform multiple layers of calculations. On the other hand, the ZED camera is designed with optimized hardware and software integration, allowing for efficient data processing and real-time performance.

From Table 5.4, it is also evident that the SGBM algorithm results in the highest RMSE average and variance, while the other two methods exhibit similar average while the built-in functions of ZED shows lower variance. This outcome indicates that the SGBM algorithm performs the worst in terms of accuracy and consistency, aligning with the visual analysis depicted in Figure 5.2, where the disparity map

generated by the SGBM algorithm contains a higher presence of noise compared to the one generated by the HITNet model. It is worth noting that the quality of the disparity map directly affects the subsequent computation of the point cloud. Therefore, a noisy disparity map leads to the generation of a correspondingly noisy point cloud, consequently contributing to lower accuracy. Figure 5.7 illustrates fluctuations in the RMSE values of the two methods with better performance across different captures, but the confidence interval of the buit-in functions of ZED is narrower, complying to its lower variance.

Based on the high accuracy and consistency achieved in generating point clouds and the significantly faster processing speed, the built-in functions of ZED will be selected and utilized for the final visualization of the soil surface.

5.4 Results of Soil Surface Visualization

The visualization of soil surface involves two critical steps: point cloud construction using stereo matching algorithms and curve extraction using interpolation methods. These steps can be achieved through various approaches, and the main objective of this experiment is to determine the optimal methods.

By evaluating the performance of different stereo matching methods, the built-in functions of ZED have been identified as the best choice for constructing point clouds from the soil surface. This method effectively captures the spatial information of the scene, generating accurate and detailed point clouds. Furthermore, for curve extraction, the linear interpolation method has been identified as the most suitable approach. This method complements the collected point clouds by filling in missing points along specified cross-sections of the soil, facilitating the extraction of smooth and continuous curves.

Figure 5.8 provides an illustrative example of soil surface visualization. It shows a point cloud obtained from the ZED camera using the built-in algorithms, together with three curves extracted using the linear interpolation method, representing three different cross-sections of the soil. Each of the curves comprises 100 interpolated points, clearly depicting the shape of the soil surface.

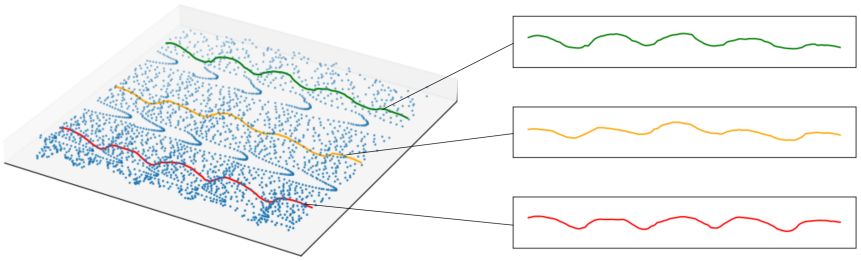

Figure 5.8: Illustration of soil surface visualization

Overall, the results demonstrate the effectiveness of the selected methods. The combination of the built-in functions of ZED for point cloud construction and the

linear interpolation method for curve extraction provides a reliable and efficient approach to visualizing and analyzing the complex shape of the soil surface.

6

Conclusion

In the conclusion chapter the analysis and interpretation of the experiment result are discussed. Additionally, the potential contribution to the related industry of this research, the imitations and some possible future research directions are also presented

6.1 Discussion

The primary objective of this study was to develop a low-cost, real-time visualization system for construction vehicles, specifically aimed at visualizing the soil surface. This was achieved by utilizing stereo cameras as a cost-effective alternative to expensive LiDAR systems. The study focused on the development of server-side processing methods for stereo images, enabling three-dimensional reconstruction. The resulting visualization system provides operators with visibility of the soil surface, allowing them to effectively monitor its conditions.

Based on the goal of the study, experiments were conducted to test three 3D reconstruction methods using stereo camera, including using ZED built-in functions, the SGBM algorithm and the HITNet model. To enhance the visual quality of the generated 3D point clouds and facilitate accuracy assessment, another evaluation was conducted on different interpolation methods, namely nearest neighbor, linear, and cubic interpolation.

The answers of the research questions are:

- **RQ1: What techniques can be used for point cloud generation from stereo images?**
  Generating point clouds from stereo images involves several steps including generating the disparity map, calculating the depth map, and projecting to 3D space. This study aimed to compare stereo matching solutions that generate the depth map. There are various solutions including algorithm-based approaches and deep learning approaches. In this experiment, the SGBM algorithm, HITNet model (trained on the Middlebury dataset), and the built-in functions of ZED camera is applied to generate point cloud from stereo images.
- **RQ2: What techniques can be used for curve extraction from a 3D scene?**
  2D curve extraction involves projecting the 3D point cloud onto a 2D plane

and extracting curves from cross-sections. To achieve smoother curves, interpolation methods can be employed to increase the number of points. In this experiment, three different interpolation methods - nearest neighbor, linear, and cubic - were used to achieve smooth 2D curve extraction.

- **RQ3: What is the difference in performance between different techniques?**
  Based on the experimental findings, it was observed that among the three interpolation methods utilized, linear interpolation exhibited superior performance because its lowest mean absolute error (MAE). The evaluation of stereo matching techniques was conducted using two metrics, namely the root mean square error (RMSE) and the processing time. The results of the experiment demonstrated that the ZED built-in functions exhibited the highest level of accuracy in generating point clouds, surpassing the performance of the SGBM stereo matching algorithm. Additionally, the processing time required by the ZED built-in functions was significantly shorter compared to the other methods.

This combination of superior accuracy and faster processing time makes the ZED built-in functions combined with linear interpolation the preferred choice for generating point clouds in this study.

It is important to point out that compared to the state-of-the-art research that evaluated stereo matching methods on existing datasets, our solution has a slower processing time. This observation should be interpreted with caution, as the measured time in this experiment encompasses various stages such as camera setup and image capturing, and may not necessarily reflect the real-time performance of our solution.

Although this study focused on a specific scenario, namely the artificially constructed soil surface in an indoor environment, the findings of the experiment can imply applicability in similar situations. The research contributes to addressing the gap in the field of smart agriculture by providing a soil surface monitoring system, thereby enhancing the capabilities of agricultural practices.

6.2 Limitations

Despite the achievements and findings of this study, there are several limitations that should be acknowledged:

- **Limited test scenarios**: The experiments in the research are conducted in a indoor environment with artificially constructed soil surfaces, provided with stable lighting sources. The performance of the selected solution in various scenarios are not tested.
- **Limited sample size**: In this study, experiments for comparing the three different stereo matching methods are only based on 8 pairs of stereo image sample. Although these 8 pair of samples are taken with different installation heights and offset angles to avoid sampling errors and bias, it would be better

to collect more samples to provide a larger number of data, increasing the credibility and confidence of the analyses.

- **External factors**: Based on existing research, it can be inferred that weather conditions such as rain or fog, as well as varying lighting conditions, may adversely affect the visual quality of the system. In addition, the presence of obstacles may also influence the visualization performance.
- **Evaluation metrics**: The study utilized various evaluation metrics in the selection of the stereo matching and interpolation methods. Nevertheless, it is important to acknowledge that additional metrics beyond time constraints, Mean Absolute Error (MAE), and Root Mean Square Error (RMSE) could be considered for a more comprehensive assessment of the visualization solution's performance. Furthermore, there is potential for the development of an evaluation equation that incorporates multiple metrics, to provide a more holistic and robust approach to evaluating the effectiveness and efficiency of the visualization solution.
- **System application**: The selected solution for this experiment requires GPU support for processing, and currently, all processing is performed on the PC using the CUDA development toolkit. However, a visualization system that can be deployed on construction vehicles should ideally be an embedded system. At present, the study have not include any implementation of porting the processing onto micro-controllers.
- **Connectivity**: Communication poses a significant challenge in the context of real-world applications, particularly in the realm of IoT-based smart farming. Seamless connectivity, as highlighted in a review by Islam et al. [11], is a critical aspect to consider. The integration and connectivity of the selected solution between the computing platform and the control platform remain open for discussion and further exploration.

6.3 Future Research Directions

To address the limitations mentioned previously, future research can be pursued to further explore the possibilities of utilizing stereo cameras for 3D reconstruction.

One important research direction is the practical deployment, which involves applying the visualization processing pipeline to embedded platforms, such as microcontrollers or other edge devices. This can enable the integration into construction vehicles. Such application requires more investigation on visualization effect in outdoor scenarios to eliminate the disturbance of external factors like illumination and weather condition. Adapting the system to solve these challenges is a valuable aspect to explore.

In order to enhance the visual quality and accuracy of the generated point clouds, further research efforts may be required to explore advanced interpolation methods. As discussed in Section 3.1.4, various interpolation algorithms, such as spline interpolation and finite-difference method, hold promise for improving the interpolation process. Therefore, it is recommended to investigate these interpolation techniques

in greater detail as part of future research endeavors. This exploration would contribute to the refinement and enhancement of the visualization solution.

Another avenue for research is conducting a comprehensive comparative study involving a broader range of sensors and 3D reconstruction methods. Although the experiments have demonstrated the excellent visualization performance of the ZED camera, there might exist alternative sensors that are more cost-effective and better suited for integration into embedded systems. Therefore, it is worthwhile to explore and investigate such sensors to enable a more comprehensive analysis and comparison of the proposed solution with alternative approaches. This analysis should encompass not only the visualization quality but also considerations of equipment cost and maintenance expenses to provide a holistic assessment.

The last crucial research direction that necessitates in-depth investigation is real-time performance. In this study, the selected solution's processing rate is relatively slow compared to the state-of-the-art stereo matching techniques, which may not be sufficient for a monitoring system requiring real-time capabilities. Therefore, it is essential to explore methods for improving real-time performance. Possible improvements include enhancing the hardware components of the processing machine or optimizing the steps and algorithms within the pipeline to achieve faster processing times.

Bibliography

[1] rnjena (25), partiko (66), and steemitboard (66). Low cost solid-state 2d lidar.

[2] VP3D. Viewpoint 3d.

[3] Vision Team. What is a stereo vision camera? - e-con systems, Jan 2023.

[4] Vladimir Tankovich, Christian Häne, Yinda Zhang, Adarsh Kowdle, Sean Fanello, and Sofien Bouaziz. Hitnet: Hierarchical iterative tile refinement network for real-time stereo matching, 2023.

[5] Renate Häuslschmid, Max von Bülow, Bastian Pfleging, and Andreas Butz. Supportingtrust in autonomous driving. In Proceedings of the 22nd International Conference on Intelligent User Interfaces, IUI '17, page 319–329, New York, NY, USA, 2017. Association for Computing Machinery.

[6] Xinzhu Ma, Zhihui Wang, Haojie Li, Pengbo Zhang, Wanli Ouyang, and Xin Fan. Accurate monocular 3d object detection via color-embedded 3d reconstruction for autonomous driving. In Proceedings of the IEEE/CVF International Conference on Computer Vision, pages 6851–6860, 2019.

[7] Ankit Dhall, Dengxin Dai, and Luc Van Gool. Real-time 3d traffic cone detection for autonomous driving. In 2019 IEEE Intelligent Vehicles Symposium (IV), pages 494–501, 2019.

[8] Jason Mah, Claire Samson, Stephen D. McKinnon, and Denis Thibodeau. 3d laser imaging for surface roughness analysis. International Journal of Rock Mechanics and Mining Sciences, 58:111–117, 2013.

[9] Philip Marzahn, Moritz Seidel, and Ralf Ludwig. Decomposing dual scale soil surface roughness for microwave remote sensing applications. Remote Sensing, 4(7):2016–2032, 2012.

[10] Muhammad Ayaz, Mohammad Ammad-Uddin, Zubair Sharif, Ali Mansour, and El-Hadi M Aggoune. Internet-of-things (iot)-based smart agriculture: Toward making the fields talk. IEEE access, 7:129551–129583, 2019.

[11] Nahina Islam, Md Mamunur Rashid, Faezeh Pasandideh, Biplob Ray, Steven Moore, and Rajan Kadel. A review of applications and communication technologies for internet of things (iot) and unmanned aerial vehicle (uav) based sustainable smart farming. Sustainability, 13(4):1821, 2021.

[12] Jash Doshi, Tirthkumar Patel, and Santosh kumar Bharti. Smart farming using iot, a solution for optimally monitoring farming conditions. Procedia Computer Science, 160:746–751, 2019.

[13] Faris A Almalki, Ben Othman Soufiene, Saeed H Alsamhi, and Hedi Sakli. A low-cost platform for environmental smart farming monitoring system based on iot and uavs. Sustainability, 13(11):5908, 2021.

[14] M Luz Guillén-Climent, Pablo J Zarco-Tejada, JAJ Berni, Peter RJ North, and Francisco J Villalobos. Mapping radiation interception in row-structured orchards using 3d simulation and high-resolution airborne imagery acquired from a uav. Precision Agriculture, 13:473–500, 2012.

[15] Jeongeun Kim, Seungwon Kim, Chanyoung Ju, and Hyoung Il Son. Unmanned aerial vehicles in agriculture: A review of perspective of platform, control, and applications. Ieee Access, 7:105100–105115, 2019.

[16] Fabio Remondino. From point cloud to surface: the modeling and visualization problem. International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, 34, 2003.

[17] Yang Ma, Yubing Zheng, Jianchuan Cheng, and Said Easa. Real-time visualization method for estimating 3d highway sight distance using lidar data. Journal of Transportation Engineering, Part A: Systems, 145(4):04019006, 2019.

[18] Jiaxiong Qiu, Zhaopeng Cui, Yinda Zhang, Xingdi Zhang, Shuaicheng Liu, Bing Zeng, and Marc Pollefeys. Deeplidar: Deep surface normal guided depth prediction for outdoor scene from sparse lidar data and single color image. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 3313–3322, 2019.

[19] Yiming Zhao, Lin Bai, Ziming Zhang, and Xinming Huang. A surface geometry model for lidar depth completion. IEEE Robotics and Automation Letters, 6(3):4457–4464, 2021.

[20] Maria Jokela, Matti Kutila, and Long Le. Road condition monitoring system based on a stereo camera. In 2009 IEEE 5th International Conference on Intelligent Computer Communication and Processing, pages 423–428, 2009.

[21] R. Lagisetty, N. K. Philip, R. Padhi, and M. S. Bhat. Object detection and obstacle avoidance for mobile robot using stereo camera. In 2013 IEEE International Conference on Control Applications (CCA), pages 605–610, 2013.

[22] D. Scharstein and R. Szeliski. High-accuracy stereo depth maps using structured light. In 2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2003. Proceedings., volume 1, pages I–I, 2003.

[23] S. Nedevschi, R. Danescu, D. Frentiu, T. Marita, F. Oniga, C. Pocol, R. Schmidt, and T. Graf. High accuracy stereo vision system for far distance obstacle detection. In IEEE Intelligent Vehicles Symposium, 2004, pages 292–297, 2004.

[24] Jiejie Zhu, Liang Wang, Ruigang Yang, and James Davis. Fusion of time-of-flight depth and stereo for high accuracy depth maps. In 2008 IEEE Conference on Computer Vision and Pattern Recognition, pages 1–8, 2008.

[25] Andreas Geiger, Julius Ziegler, and Christoph Stiller. Stereoscan: Dense 3d reconstruction in real-time. In 2011 IEEE intelligent vehicles symposium (IV), pages 963–968. Ieee, 2011.

[26] Xin Tian, Rui Liu, Zhongyuan Wang, and Jiayi Ma. High quality 3d reconstruction based on fusion of polarization imaging and binocular stereo vision. Information Fusion, 77:19–28, 2022.

[27] Behnam Behroozpour, Phillip A. M. Sandborn, Ming C. Wu, and Bernhard E. Boser. Lidar system architectures and circuits. IEEE Communications Magazine, 55(10):135–142, 2017.

[28] Daniel Scharstein and Richard Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. International journal of computer vision, 47:7–42, 2002.

[29] Ramin Zabih and John Woodfill. Non-parametric local transforms for computing visual correspondence. In Computer Vision—ECCV'94: Third European Conference on Computer Vision Stockholm, Sweden, May 2–6 1994 Proceedings, Volume II 3, pages 151–158. Springer, 1994.

[30] Heiko Hirschmuller. Stereo processing by semiglobal matching and mutual information. IEEE Transactions on Pattern Analysis and Machine Intelligence, 30(2):328–341, 2008.

[31] S. Birchfield and C. Tomasi. A pixel dissimilarity measure that is insensitive to image sampling. IEEE Transactions on Pattern Analysis and Machine Intelligence, 20(4):401–406, 1998.

[32] Philip J Davis. Interpolation and approximation. Courier Corporation, 1975.

[33] Nina Siu-Ngan Lam. Spatial interpolation methods: a review. The American Cartographer, 10(2):129–150, 1983.

[34] AV/Robot For 4D Sensing, AR and AIoT. Apple lidar demystified: Spad, vcsel, and fusion, 2021.

[35] P. Chase, K. Clarke, A. Hawkes, S. Jabari, and J. Jakus. Apple iphone 13 pro lidar accuracy assessment for engineering applications. 2022: The Digital Reality of Tomorrow, 2022.

[36] ibaiGorordo. Onnx-hitnet-stereo-depth-estimation.