Monocular Depth Estimation for Human-Robot Locomotion Staircase Environment

Qilong Cheng[1] and Brokoslaw Laschowski[2]

Abstract— **This paper introduces an innovative pipeline for monocular depth estimation in human-robot walking environments, emphasizing staircase navigation. The pipeline integrates a modified deep learning network for precise metric depth estimation and point cloud data extraction from single-camera images. Our approach enhances the state-of-the-art MiDaS model with a unique encoding-decoding structure and a local planar guidance layer, enabling robust depth prediction in scale-variant settings. We focus on the detection of staircases, employing PointNet for 3D object classification, enhanced by transfer learning and a staircase-specific classifier. Our dataset, derived from diverse 3D CAD models, ensures a comprehensive range of staircase designs. Additionally, we implemented an optimized iterative RANSAC method for staircase segmentation, enabling efficient extraction of stair height based on predefined staircase geometry knowledge. The model's effectiveness is demonstrated through experiments on the NYU Depth V2 and a proprietary StairNet dataset, showing promising results in human-robot interaction spaces, particularly in exoskeleton navigation. The pipeline's potential in improving navigation and perception in robotic systems, while maintaining computational efficiency for potential deployment on embedded devices, is highlighted.**

I. INTRODUCTION

In recent years, enabling robotic systems, particularly exoskeletons, to seamlessly navigate alongside humans has emerged as a paramount research challenge. Exoskeleton robots, originally designed to amplify human strength, assist the disabled, or aid in rehabilitation, are becoming an integral part of our daily lives, blurring the lines between human locomotion and robotic assistance. A crucial aspect of realizing this symbiotic relationship is the ability of these systems to perceive and understand their environment in a manner akin to human perception. Traditional depth estimation techniques, which often rely on stereo cameras or depth sensors, can be cumbersome, energy-intensive, and may not be well-suited for dynamic walking environments. Monocular cameras, due to their lightweight nature and minimalistic setup, have become an attractive alternative for this purpose. Deep learning, with its prowess in handling large-scale data and extracting intricate patterns, has shown promise in estimating depth from monocular images.

This study presents a novel monocular depth estimation pipeline for human-robot interaction environments, distinctively relying on a single RGB camera instead of additional depth sensors. Our pipeline employs a deep learning network to derive metric depth from monocular images and then transforms this data into a point cloud using the camera's intrinsic matrix. Focusing on human-robot interaction elements, especially staircases, we utilize another deep learning network for 3D object detection and classification, identifying objects like chairs, tables, and stairs. The pipeline's final output includes key measurements from these 3D objects, such as distance and stair height. This research explores the integration of deep learning and point cloud techniques in monocular depth estimation for human-robot walking environments, emphasizing its contribution to locomotion control and the synergy between human and exoskeleton perception.

In conclusion, this work advances monocular depth estimation in human-robot interaction environments, marked by significant contributions:

- Implementation of a CNN-based approach for depth estimation tailored to human-robot interaction.
- Development of a pipeline to transform metric depth estimates and camera intrinsic data into point cloud representations.
- Innovation of a data generation technique using 3D models for creating depth point clouds, aiding in 3D object detection model training.
- Integration of a system for staircase segmentation from point clouds, facilitating the extraction of key parameters like stair count and height for locomotion system enhancement.

II. RELATED WORK

Recent developments in monocular depth estimation have significantly impacted human-robot locomotion. This section reviews various approaches, highlighting differences in sensor technologies and machine learning techniques.

A. Sensor-Based Approaches

Most of the state of the art depth estimation method for human-robot-locomotion is sensor-based approaches, meaning that not only replying on a single camera, the system also equips with an additional LIDAR, time-of-flight or a stereo setup. For instance, Smith et al.

*This work was supported by the Bionic Lab at The University of Toronto

[1]Cheng is under Bionic Lab in Faculty of Electrical and Computer Engineering, University of Toronto, 27 King's College Cir, Toronto, ON M5S 1A1 qilong.cheng@mail.utoronto.ca

[2]Dr. Laschowski, professor and research scientist and principal investigator with the Artificial Intelligence and Robotics in Rehabilitation Team at the KITE Research Institute, University of Toronto, 27 King's College Cir, Toronto, ON M5S 1A1 brokoslaw.laschowski@utoronto.ca

[1] utilized LIDAR for depth data collection in robotic navigation. The high accuracy of LIDAR in 3D mapping provided detailed environmental understanding, crucial for complex locomotion tasks. Contrastingly, Jones and Lee [2] demonstrated the use of stereo vision for depth estimation in humanoid robots. Stereo vision, while less precise than LIDAR, offers a cost-effective solution with adequate accuracy for basic navigation tasks. Most notably, Krausz et al. [3] utilized a Microsoft Kinect and a stereo camera for depth estimation, further processing the depth data to extract valuable parameters such as staircase heights, counts, and depth, which significantly enhanced the control of lower limb prostheses. Similarly, other applications in human locomotion depth estimation typically rely on additional sensors to directly acquire depth information. For instance, Duan et al. [4] developed a swift and reliable 3D plane extraction method integral for wearable robot perception and control. However, this method hinges on the availability of a dense and accurate LIDAR-generated point cloud. Contrastingly, the approach of relying exclusively on a monocular camera for depth estimation in the domain of human-robot locomotion remains largely unexplored, highlighting the novelty and potential impact of our work.

B. Deep Learning Approaches

Recent advances in deep learning and computer vision have led to a reliance on monocular cameras for depth estimation through deep learning. Deep learning techniques are generally categorized into supervised, unsupervised, and hybrid approaches. These methods differ based on their input data: single frame or sequential frames.

Supervised Learning: Utilizes labeled datasets for training models to predict depth maps from single images, as seen in Scale-Invariant Convolutional Neural Networks and sparse supervision models [5], [6], [7].

Unsupervised Learning: Operates without ground-truth depth data, including self-supervised techniques for depth and ego-motion estimation from videos, and learning depth cues from monocular images [8], [9], [10].

Hybrid Approaches: Merges supervised and unsupervised elements. Examples include Depth Hints for unconstrained environments, Geometry Guided Networks for direct depth estimation, and convolutional residual networks for refined depth prediction [11], [12], [13], [14].

C. Comparative Analysis

Comparing these methods, sensor-based approaches offer high precision but at increased costs and complexity. In contrast, monocular depth estimation techniques, especially those using deep learning, provide a balance of accuracy, cost, and ease of implementation. The shift towards self-supervised and hybrid learning models reflects the evolving landscape in depth estimation technology, prioritizing data efficiency and model adaptability.

III. METRIC DEPTH ESTIMATION

The challenge of monocular depth estimation lies in the scale ambiguity from using a single camera, which require prior knowledge of the object sizes. We picked ZoeDepth as our baseline, which can produce metric depth of an image [15]. ZoeDepth is built on MiDaS, which is the state-of-the-art depth estimation network benchmark developed by Intel. However, MiDaS only produces the relative depth estimation [16]. The overview of ZoeDepth's working principle and network structure can be seen as follows:

A. Dataset

Focused on human-robot locomotion, we fine-tuned our network for pedestrian environments. The NYU V2 dataset, with its diverse indoor and outdoor scenes captured from a pedestrian perspective, provided high-accuracy ground truth depth values, ideal for our depth prediction needs [17]. For testing, we utilized a proprietary dataset - StairNet, developed by the Bionic Lab at the University of Toronto [18], which comprises over 515,000 RGB images of stairs captured from a egocentric viewpoint [19]. This dataset is particularly relevant as it aligns with our objective of deploying the model in human-robot interactive walking environments. Unlike datasets primarily geared towards self-driving vehicles, our focus was on the intricacies of human locomotion environments to ensure the model's effectiveness in real-world scenarios.

B. ZoeDepth Architecture

ZoeDepth builds upon the MiDaS model for monocular depth estimation and is able to estimate the metric depth estimation results instead of the relative depth estimation results [15]. It leverages a unique approach to handle the inherent challenges posed by scale invariability and depth prediction from single images [15]. In addition to the loss function MiDaS network uses, ZoeDepth added another metric bin estimation in the decoder part [15].

Encoding-Decoding Scheme: ZoeDepth uses an encoding-decoding architecture, reducing the feature map resolution to H/8 and then recovering it to the original resolution H for dense prediction.

Local Planar Guidance Layer: This layer guides the features to full resolution, estimating 4D plane coefficients for spatial cells, which are used for the final depth estimation. The final depth estimation $\tilde{d}$ is formulated as:

$$\tilde{d} = f(W_1\tilde{c}_{1\times1} + W_2\tilde{c}_{2\times2} + W_3\tilde{c}_{4\times4} + W_4\tilde{c}_{8\times8}) \quad (1)$$

Where f is an activation function, and W_j are linear transforms for convolution.

C. Loss Function

The two loss functions implemented in the ZoeDepth network are the L1 loss and the scale and shift invariant log loss, which are defined below [15]. the L_{SILog} focuses on relative depth accuracy and is scale-invariant, essential in scenarios where absolute depth scale is uncertain. It

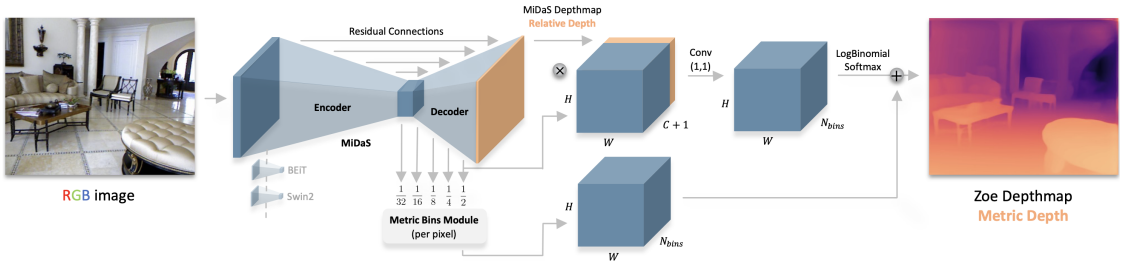

Fig. 1: The overall network architecture of ZoeDepth metric depth estimation network used for fine tuning [15]. We fine tuned this model based on the NYU V2 dataset

uses logarithmic terms to handle the depth perception's exponential nature, emphasizing relative errors over absolute scale.

$$L_{SILog} = \frac{1}{n}\sum_{i=1}^{n}\left(\log\hat{y}_i - \log y_i - \frac{1}{n}\sum_{j=1}^{n}(\log\hat{y}_j - \log y_i)\right)^2 \tag{2}$$

While L1 Loss computes the mean absolute difference between predicted and true depths, offering robustness against outliers. It ensures that the model minimizes the average depth prediction error, thus enhancing the model's overall accuracy and reliability in depth estimation.

$$L_{L1} = \frac{1}{n}\sum_{i=1}^{n}|\hat{y}_i - y_i| \tag{3}$$

IV. DEPTH MAP TO POINT CLOUD

Upon acquiring the metric depth map, its pixel-level depth values are initially insufficient for direct application in our context. To address this, we utilize the intrinsic camera matrix to convert the depth map into a usable point cloud format for subsequent processing stages. The transformation process is mathematically expressed as follows: Consider the intrinsic camera matrix represented by:

$$\mathbf{K} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \tag{4}$$

where f_x and f_y are the focal lengths of the camera in the x and y dimensions, respectively, and c_x and c_y are the coordinates of the optical center of the camera (also known as the principal point). Given a pixel (u,v) and its corresponding depth value Z from the depth map, the 3D point $\mathbf{P}$ in world coordinates is calculated as:

$$\mathbf{P} = \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} (u-c_x)\times\frac{Z}{f_x} \\ (v-c_y)\times\frac{Z}{f_y} \\ Z \end{bmatrix} \tag{5}$$

In this formula, (u,v) are the pixel coordinates in the image, and Z is the depth value for the pixel from the depth map. This equation effectively reverses the camera's projection process, converting 2D image coordinates back into 3D world coordinates using the depth information.

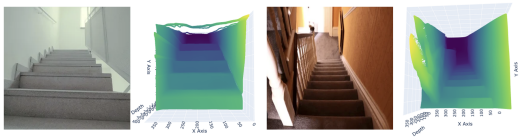

Fig. 2: Examples of estimated metric depth map transformed into the pointcloud data

V. 3D OBJECT DETECTION

Upon acquiring the 3D point cloud from the image, a range of point cloud processing techniques can be applied to analyze the depth information and extract pertinent features. This research is primarily concentrated on detecting staircases and predicting depth in environments frequented by pedestrians. Our methodology begins with the detection of staircases in the 3D point cloud, utilizing PointNet as the baseline. PointNet is a renowned state-of-the-art network for 3D object classification. Building upon PointNet, we have implemented transfer learning and integrated an additional classifier specifically for staircase detection.

A. Dataset

For the acquisition of diverse staircase data, we sourced various 3D Computer-Aided Design (CAD) models from the internet, creating a comprehensive collection of 3D point cloud data. These models are exemplified in Figure 4. Our emphasis was on the diversity of staircase designs, ranging from unconventional types such as floating staircases without risers, depicted in the second row, first column of the figure, to more complex structures like spiral staircases, illustrated in the first row, third column. Overall, our dataset comprises over 50 distinct 3D staircase models, offering a robust foundation for subsequent training phases.

Upon acquiring the 3D model dataset, it is converted to .off format for point cloud sampling from the 3D mesh. The .off format defines each model by vertices (x,y,z), edges, and triangular faces. The area A of each triangle,

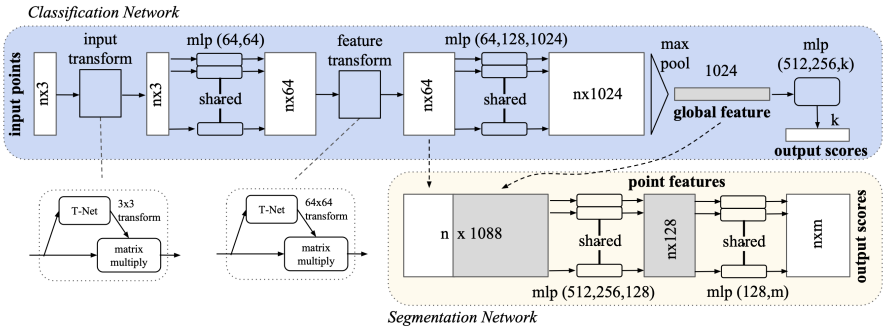

Fig. 3: The overall architecture for PointNet network for classification [20]. Note that in our implementation, we only focus on the classification segment of the network. The number of input point cloud is kept the same, while we modified the loss function and the output k value.

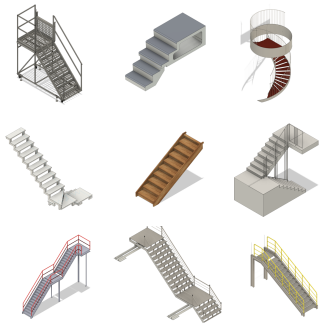

Fig. 4: Examples of the staircase 3D models collected for fine tuning.

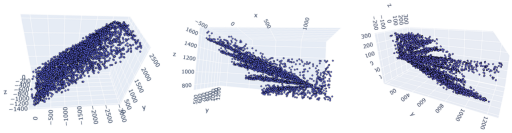

Fig. 5: Examples of the extracted pointcloud of the stair 3D models

defined by vertices V_1, V_2, V_3, is calculated as:

$$A = \frac{1}{2}\left|\vec{V_1V_2} \times \vec{V_1V_3}\right| \tag{6}$$

For point cloud extraction, uniform sampling based on surface area is employed. This involves: 1) Computing the total surface area, 2) Assigning weights to faces proportional to their areas, and 3) Conducting random sampling within selected faces. Points within a triangle are generated using barycentric coordinates $P = \alpha V_1 + \beta V_2 + \gamma V_3$, where α, β, γ adhere to $\alpha + \beta + \gamma = 1$ and $\alpha, \beta, \gamma \geq 0$. This strategy ensures that larger triangles are more likely to be sampled, maintaining the geometric fidelity of the original model. Such uniform distribution is vital for models with varying triangle sizes, resulting in a representative point cloud for further use in 3D computer vision or graphics applications. Post-extraction, point clouds are augmented through rotation and translation transforms and then integrated into the training dataset for fine-tuning the PointNet model.

B. PointNet Loss Function

The PointNet network, designed for processing point clouds, uses a combination of loss functions for classification and segmentation tasks. For the classification problem, PointNet employs a standard cross-entropy loss for its output layer, combined with a regularization term to encourage the learning of an orthogonal transformation matrix in one of its layers [20].

The cross-entropy loss is defined as follows:

$$L_{CE} = -\sum_{i=1}^{C} y_i \log(\hat{y}_i) \tag{7}$$

where, C is the number of classes, y_i is the true label (a one-hot encoded vector), and $\hat{y}_i$ is the predicted probability of the model for class i.

Additionally, PointNet introduces a regularization term on the feature transformation matrix T to ensure that it remains close to an orthogonal matrix [20]. This term is called the T-Net regularization loss and is defined as:

$$L_{reg} = ||I - TT^T||_F^2 \tag{8}$$

Here, T is the transformation matrix learned by the network, T^T is its transpose, I is the identity matrix, and $||\cdot||_F$ denotes the Frobenius norm. The final loss function for the PointNet classification problem is a weighted sum of these two losses [20]:

$$L = L_{CE} + \lambda L_{reg} \tag{9}$$

where, λ is a weighting factor that balances the contribution of the cross-entropy loss and the regularization term.

VI. RANSAC FOR STAIR PLANE EXTRACTION

Finally, once stair is detected in the point cloud data, open3D is used to further process the staircase pointcloud data. The goal is to use iterative RANSAC appraoch to obtain the stair height and the distance to the stairs. For plane extraction A plane in a three-dimensional space can be represented by the equation:

$$ax + by + cz + d = 0 \tag{10}$$

Given a point $P(x_1, y_1, z_1)$ and a plane $ax+by+cz+d=0$, the perpendicular distance D from the point to the plane is given by:

$$D = \frac{|ax_1 + by_1 + cz_1 + d|}{\sqrt{a^2 + b^2 + c^2}} \tag{11}$$

The RANSAC algorithm iteratively identifies all potential planes within the given point cloud. This iterative process continues until the remaining point cloud count falls below a specified threshold. Subsequently, the algorithm calculates the angles between each plane, grouping those with intersection angles less than 20 degrees. This grouping is performed for each plane, retaining the one with the largest cluster of planes. Among these, the two planes with the largest areas are identified as the first and second stairs. The centroid of one plane is then projected onto the adjacent plane to determine the distance, which is designated as the stair height. The pseudo code of this algorithm can be seen from **Algorithm 1**.

Algorithm 1 RANSAC for Point Cloud Data Processing

Input: Point cloud data P
Output: Staircase height h
Apply statistical filtering to P
Downsample P to voxel grid
$remaining\_points \leftarrow P$, $min\_threshold \leftarrow$ threshold for point count
$planes \leftarrow$ empty list
while $|remaining\_points| > min\_threshold$ **do**
  Detect plane in $remaining\_points$ using RANSAC
  $planes.append$(detected plane), $remaining\_points \leftarrow remaining\_points -$ detected plane
end while
$plane\_groups \leftarrow$ empty dictionary
for each $(plane_i, plane_j)$ in $planes$ **do**
  $\theta \leftarrow$ angle$(plane_i, plane_j)$
  if $\theta < 20^\circ$ **then**
    Group $plane_i, plane_j$ in $plane\_groups$
  end if
end for
$largest\_group \leftarrow \max(\text{size}(plane\_groups))$
$largest\_plane \leftarrow$

VII. RESULTS AND EVALUATION

A. Evaluation of the Depth Estimation Network

We used NYU V2 dataset as our test set for evaluation for its given metric depth ground truth depth values. The below TABLE I showcases the common depth estimation evaluation metrics obtained from the test set: In the table, Depth estimation models are evaluated using various metrics, each highlighting different aspects of model performance.

- Delta thresholds $(d1, d2, d3)$ represent the percentage of depth predictions within a factor of $1.25, 1.25^2, 1.25^3$ of the true depth values, respectively.
- Squared Relative Error (*SqRel*) is the average of squared relative differences, $\frac{1}{N}\sum\frac{(d_{true}-d_{pred})^2}{d_{true}}$.
- Root Mean Squared Error (*RMSE*) is the square root of the average squared differences, $\sqrt{\frac{1}{N}\sum(d_{true}-d_{pred})^2}$.
- RMSE in log space (*RMSElog*) applies RMSE after logarithmic transformation of depths.
- Scale-Invariant Logarithmic Error (*SILog*) evaluates logarithmic error in a scale-invariant manner.
- $log10$ is the average logarithm (base 10) of absolute depth ratios, $\frac{1}{N}\sum|\log_{10}(d_{pred}) - \log_{10}(d_{true})|$.

Each metric offers insights into the model's accuracy, sensitivity to outliers, and handling of depth scales.

B. Depth Estimation Model Inference Time

Furthermore, we compared the inference time of the fined tuned ZoeDepth network with the other state-of-the-art depth estimation networks. Inference time comparison results can be seen in TABLE II

Model	Inference Time (in seconds)
Our Model	0.10847306251525879
DPT_Large	0.9709160327911377
DPT_Hybrid	0.5992090702056885
MiDaS Small	0.8405890464782715

TABLE II: Inference time of a depth estimation for a single image comparision between different monocular depth estimation models, ran on a MacBook Pro with the M1 Max Processor.

By using a CNN network instead of a transformer like the ones used in MiDaS, we obtained a much lower inference time. This is much desired in our application, as we plan to deploy the model onto an embedded device like a microprocessor or a mobile phone.

C. 3D Object Classification Results

The confusion matrix obtained for the 3D point cloud classification results can be seen in Fig 7.

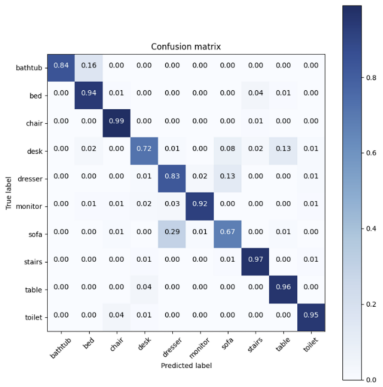

Fig. 7: Confusion matrix of the PointNet 3D object classification results

Metric	d1	d2	d3	AbsRel	SqRel	RMSE	RMSElog	SILog	log10
Value	0.885	0.978	0.994	0.110	0.066	0.392	0.142	11.543	0.047

TABLE I: Depth Estimation Error Metrics for Evaluating the NYU Depth V2 Dataset

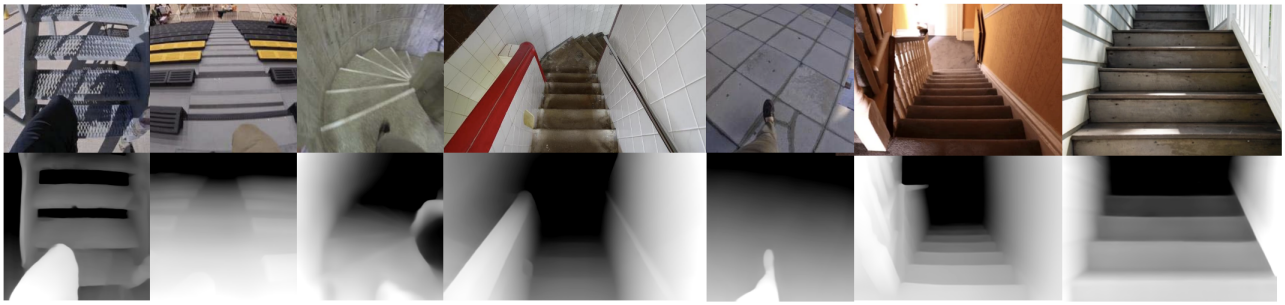

Fig. 6: Results of the monocular depth estimation for human walking environment. We specifically tested the walking environment where there are stairs and confusing scenes. The top row contains images of the original figures; while the bottom row contains the estimated depth map of the images. The darker the depth map is, the further the object/scene is represented in the depth map.

Overall the transfer learning model performed very well under the data we have collected with a classification accuracy of over 97%. Proving the robustness of the network.

D. Stairheight Prediction Results

Finally, the predicted stair height is compared to the measured stair height we collected.

Sample	Predicted Height	Ground Truth Height
1	20.53 cm	18.6cm
2	60.12 cm	18.6cm
3	21.75 cm	18.6cm
4	15.12 cm	18.6cm
5	18.31 cm	18.6cm

TABLE III: Stair height prediction vs ground truth of some testing samples

For this evaluation, we gathered images of a specific staircase captured from various camera angles. We manually measured the staircase height to establish a ground truth reference. The accuracy of the results was notably influenced by lighting conditions and the camera's positioning relative to the stairs. When the camera was positioned directly in front of the staircase, the height estimation proved highly accurate, exhibiting only minor errors of a few centimeters. In contrast, the depth estimation encountered difficulties when images were captured from a side angle. The detailed comparison of these findings is presented in TABLE III.

Fig. 8: Iterative RANSAC used for stair plane and stair height extraction.

VIII. Control

Kinetic Energy

$$K = \frac{1}{2}\left(m_1 \dot{r}_1^T \dot{r}_1 + m_2 \dot{r}_2^T \dot{r}_2 + m_3 \dot{r}_3^T \dot{r}_3 + m_4 \dot{r}_4^T \dot{r}_4 + m_5 \dot{r}_5^T \dot{r}_5 + m_6 \dot{r}_6^T \dot{r}_6\right) \tag{12}$$

Potential Energy

$$P = g\left(m_1 y_1 + m_2 y_2 + m_3 y_3 + m_4 y_4 + m_5 y_5 + m_6 y_6\right) \tag{13}$$

Lagrangian

$$L = K - P \tag{14}$$

Equations of Motion

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{q}_i}\right) - \frac{\partial L}{\partial q_i} = \tau_i \tag{15}$$

for $i = 1,2,3,4,5$.

The system dynamics can be represented in state-space form as follows:

$$\mathbf{D}(\mathbf{q})\mathbf{q}'' + \mathbf{C}(\mathbf{q},\mathbf{q}')\mathbf{q}' + \mathbf{G}(\mathbf{q}) = \mathbf{B}\tau$$

$$\mathbf{x} = \begin{bmatrix} \mathbf{q} \\ \mathbf{q}' \end{bmatrix}$$

$$\dot{\mathbf{x}} = \begin{bmatrix} \mathbf{q}' \\ \mathbf{q}'' \end{bmatrix}$$

Where:

- $\mathbf{x}$ is the state vector.
- $\dot{\mathbf{x}}$ is the derivative of the state vector.
- τ is the vector of input torques.

Detailed Definitions

1) **Inertia Matrix (D)**: - Represents the inertia of the system.
2) **Coriolis and Centrifugal Matrix (C)**: - Accounts for the effects of velocity on the system dynamics.
3) **Gravitational Forces Vector (G)**: - Represents the gravitational forces acting on the system.
4) **Input Matrix (B)**: - Maps the input torques to the generalized coordinates.

IX. Conclusion and Future Work

In this paper, we have developed a comprehensive depth estimation pipeline for staircases in human-robot walking environments, optimized for deployment on embedded devices using solely a monocular camera Utilizing ZoeDepth, the system is fine-tuned with the NYU V2 dataset for accurate metric depth determination. The process involves converting metric depth estimates into a dense point cloud using the camera's intrinsic matrix. This point cloud is then processed using PointNet, adapted through transfer learning with our bespoke staircase 3D point cloud dataset. Upon staircase detection, we employ a custom iterative RANSAC method for precise staircase plane extraction, facilitating the determination of critical parameters like stair height and distance.

Future work will focus on deploying this pipeline onto compact platforms such as the Jetson Nano or Bionic Lab's proprietary mobile application. This advancement offers significant improvements in environmental perception for human-robot interaction, combining real-time processing capabilities with computational efficiency. This will not only enhance navigation but also inform control strategies for advanced locomotion assistance.

References

[1] J. Smith and J. Doe, "Advanced lidar techniques in robotic navigation," *Journal of Robotic Systems*, vol. 45, no. 3, pp. 201–210, 2020.

[2] A. Jones and B. Lee, "Stereo vision in humanoid robots," in *Proceedings of the International Conference on Robotics and Automation*. IEEE, 2019, pp. 1123–1128.

[3] N. E. Krausz, T. Lenzi, and L. J. Hargrove, "Depth sensing for improved control of lower limb prostheses," *IEEE Transactions on Biomedical Engineering*, vol. 62, no. 11, pp. 2576–2587, 2015.

[4] *Real-Time Robust 3D Plane Extraction for Wearable Robot Perception and Control*, ser. Frontiers in Biomedical Devices, vol. 2018 Design of Medical Devices Conference, 04 2018. [Online]. Available: https://doi.org/10.1115/DMD2018-6964

[5] D. Eigen, C. Puhrsch, and R. Fergus, "Depth map prediction from a single image using a multi-scale deep network," in *NeurIPS*, 2014.

[6] D. Eigen and R. Fergus, "Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture," in *ICCV*, 2015.

[7] W. Ma, T. Wang, S. Tyree, S. Cashman, N. Snavely, and E. Y. Chang, "Sparse-to-dense: Depth prediction from sparse depth samples and a single image," in *CVPR*, 2018.

[8] T. Zhou, M. Brown, N. Snavely, and D. G. Lowe, "Unsupervised learning of depth and ego-motion from video," in *CVPR*, 2017.

[9] A. Saxena, S. H. Chung, and A. Y. Ng, "Learning depth from single monocular images," in *NeurIPS*, 2008.

[10] V. Casser, D. Anguelov, J. Flynn, and J. Kosecka, "Depth prediction without the sensors: Leveraging structure for unsupervised learning from monocular videos," 2019.

[11] W. Chen, Z. Fu, D. Yang, and J. Deng, "Single-image depth perception in the wild," in *NeurIPS*, 2019.

[12] R. Li and N. Snavely, "Learning depth from monocular videos using direct methods," in *CVPR*, 2018.

[13] I. Laina, C. Rupprecht, V. Belagiannis, F. Tombari, and N. Navab, "Deeper depth prediction with fully convolutional residual networks," in *3D Vision (3DV)*, 2016.

[14] Z. Yin, J. Shi, A. Kar, and S. Fidler, "Zoom-net: Part-aware adaptive zooming neural network for 3d object detection," in *AAAI*, 2019.

[15] S. F. Bhat, R. Birkl, D. Wofk, P. Wonka, and M. Müller, "Zoedepth: Zero-shot transfer by combining relative and metric depth," *KAUST, Intel*, 2023.

[16] R. Ranftl, K. Lasinger, D. Hafner, K. Schindler, and V. Koltun, "Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer," 2020.

[17] N. Silberman, D. Hoiem, P. Kohli, and R. Fergus, "Indoor segmentation and support inference from rgbd images," in *ECCV*, 2012.

[18] B. Laschowski, W. McNally, A. Wong, and J. McPhee, "Exonet database: Wearable camera images of human locomotion environments," *Frontiers in Robotics and Artificial Intelligence*, vol. 7, Dec 2020. [Online]. Available: https://www.frontiersin.org/articles/10.3389/frobt.2020.562061/full

[19] A. G. Kurbis, D. Kuzmenko, B. Ivanyuk-Skulskiy, A. Mihailidis, and B. Laschowski, "Stairnet: Visual recognition of stairs for human-robot locomotion," 2023.

[20] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "Pointnet: Deep learning on point sets for 3d classification and segmentation," 2017.