TRIBHUVAN UNIVERSITY

INSTITUTE OF ENGINEERING

PASHCHIMANCHAL CAMPUS

LAMACHAUR, POKHARA

[Subject Code: EX 707]

A MID TERM REPORT

ON

ADVANCEMENTS IN TEXTURE EXTRACTION AND DEPTH ESTIMATION:TECHNIQUES AND APPLICATION

SUBMITTED BY:

AVISHEK POUDEL	[WRC077BEI009]
BINAYAK SHRESTHA	[WRC077BEI012]
PRAKANDA BHANDARI	[WRC077BEI030]
PRATHAM ADHIAKRI	[WRC077BEI032]

SUBMITTED TO:

DEPARTMENT OF ELECTRONICS AND COMPUTER ENGINEERING

December, 2024

ABSTRACT

Accurate depth estimation is a fundamental challenge in computer vision with significant implications for autonomous navigation, augmented reality, and robotics. Traditional depth sensing methods using stereo cameras or LIDAR systems, although effective, are often cost-prohibitive and complex. This proposal outlines the development of a cost-effective, real-time depth estimation system using RGB images.The core of the system involves capturing RGB images through a camera module interfaced with the Raspberry Pi. Advanced deep learning algorithms will be employed to process these images and estimate depth, utilizing the Raspberry Pi's computational power. The Arduino Nano will control a Visual cue generator and auxiliary components such as a flashlight for illumination in low-light conditions and an IR light to enhance depth estimation accuracy by providing additional visual cues.By combining low-cost hardware with sophisticated depth estimation algorithms, this project aims to provide an accessible alternative to traditional depth sensing technologies. The proposed system's ability to deliver accurate, real-time depth information using a single RGB camera makes it a valuable tool for various applications, from simple obstacle avoidance in robotics to complex interactive environments in augmented reality.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

PBR	*Physically Based Rendering*
CRT	*Convolutional Regression Tree*
CNN	*Convolutional Neural Network*
CRF	*Conditional Random Field*
DC	*Direct Current*
DORN	*Deep Ordinal Regression Network*
DPT	*Dense Prediction Transformer*
GHz	*Giga Hertz*
GPIO	*General Purpose Input/Output*
HDMI	*High-Definition Multimedia Interface*
HTML	*Hypertext Markup Language*
IDE	*Integrated Development Environment*
I2C	*Inter-Integrated Circuit*
JSX	*JavaScript XML*
KB	*Kilo Byte*
LIDAR	*Light Detection and Ranging*
NRF	*Neural Regression Forest*
RGB	*Red Green Blue*
SID	*Spacing Increasing Discretization*
SPI	*Serial Peripheral Interface*
TOF	*Time of Flight*
UART	*Universal Asynchronous Receiver/Transmitter*
UI	*User Interface*
USB	*Universal Serial Bus*
URL	*Universal Resource Locator*

CHAPTER 1

INTRODUCTION

1.1 Background

Depth estimation has long been a critical aspect of computer vision, providing essential information for understanding and interacting with three-dimensional environments. Traditional approaches to depth estimation often rely on TOF systems or active sensors like LIDAR.A TOF sensor measures the time it takes for a light signal to travel to an object and back to determine the object's distance. It emits a light pulse (usually infrared) and calculates the distance based on the travel time of the reflected signal. LIDAR, on the other hand, employs laser pulses to measure the time it takes for the light to travel to an object and back, providing precise distance measurements. While effective, these systems have notable limitations in terms of cost, size, power consumption, and complexity, making them less suitable for many applications. Recent advances in machine learning and computer vision have significantly enhanced the capacity of monocular depth estimation. CNNs and other deep learning models have been trained on large datasets to predict depth maps from single images with remarkable accuracy. These models leverage large amounts of labeled data to learn the intricate patterns and cues that signify depth, making it possible to generate dense and accurate depth maps from RGB images.

1.2 Problem Statement

Accurate depth estimation is a fundamental requirement for a wide range of applications in computer vision and robotics. Traditional depth estimation methods predominantly rely on TOF or LIDAR sensors, which provide reliable depth measurements but come with significant drawbacks including high cost, substantial power consumption, and increased system complexity. Depth estimation, which infers depth from RGB image, presents a more accessible and cost-effective alternative. The approach must infer from visual cues such as texture, shading, perspective, and motion, which can be ambiguous

and complex to interpret.

1.3 Objectives

The objectives of this project is:

- To design and develop an accurate depth estimation system using RGB images.

1.4 Application

The major application of the project are:

- Enhances the ability of robots to understand and navigate human environments safely and effectively.
- Allows systems to detect and avoid obstacles in real-time, preventing collisions and enhancing navigation.
- Provides accurate 3D maps of the environment, crucial for navigation and situational awareness.
- Improved depth perception in AR/VR system and Realistic object interaction and manipulation and enhances immersive experiences by providing accurate spatial information and enabling precise tracking of virtual objects.
- Enables vehicles to accurately perceive and navigate their surroundings, improving safety and efficiency.
- For 3D reconstructions and modeling, it facilitates the creation of detailed and accurate 3D models for various applications.
- Enhances medical imaging by surgical planning, navigation and diagnosis by providing detailed 3D views of anatomical structures, aiding in precise surgical planning and diagnosis.
- Enables the creation of customized and well-fitted prosthetic and orthotic devices through accurate 3D modeling.

- Enhances the realism and interactivity of games by providing accurate depth information for virtual environments.
- In photography and cinematography, depth maps can improve the quality of images and videos by enabling advanced effects and accurate focus.
- Security and surveillance with enhanced facial recognition and biometrics
- Enhances the efficiency and precision of automated systems in manufacturing, industrial automation and other industrial applications.
- In telepresence and teleconferencing, it can provide a more immersive and interactive experience by accurately capturing and rendering 3D environments.
- Assists in creating accurate 3D reconstructions of crime scenes, aiding in investigations and analysis.

CHAPTER 2

LITERATURE REVIEW

The quest for the estimation of depth from image has captivated researchers and scientists for ages.Traditional depth estimation methods of image-based depth were Shape from Shading [1], though not monocular,laid the foundation for recovering depth cues from image shading variation. Other methods like Structure from Motion [2], photogrammetiric,Shape from texture[3], based on binocular camera through stereo matching and traingulation to obtain a depth map.

After the advancement of Deep neural networks, multiple papers has been published about using Deep learning as a tool for depth estimation from RGB images. The paper[4] proposed depth estimation from a single 2D color image through a deep neural network. It employed two deep network stacks: one that makes a coarse global prediction based on the entire image and another that refines this prediction locally.

In this paper[5] they introduce an approach of binocular depth estimation method based on deep learning. A new convolutional neural network is designed, which consists of two sub-networks. The first sub-network is a deep network with Siamese branches and 3D convolutional layer, it learns parallax and global information and generates a global depth estimation result in low resolution. The second is a fully convolutional deep network, which reconstructions the depth map to original resolution. The two sub-networks are connected by a pool pyramid.

The paper[6] proposes the use of deep architecture called NRF which combines CNNs with Regression Forest. It achieves robustness by processing a data sample with CRT an ensemble of binary regression trees with CNNs at every node. CNNs at every node of CRT have significantly fewer parameters. For each CNN in the split node, it uses only the RGB input window instead of convolutional outputs from the parent split node; the size of input windows for the split nodes is gradually reduced as we go down the tree along its depth. It does not back-propagate the loss of depth prediction bottom-up rather, compute distinct loss for every CNN in the tree, and then use these losses for parallel training of all CNNs in the tree. With the consideration of neighbouring information it

results in smoother depth maps.

This paper[7] proposes a novel training objective that enables our convolutional neural network to learn to perform single image depth estimation, despite the absence of ground truth depth data. The model uses bilinear sampling to generate images , resulting in a fully (sub-)differentiable training loss. A fully convolutional deep neural network, by posing monocular depth estimation as an image reconstruction problem, can solve the disparity field without requiring ground truth depth. It includes a left-right consistency check to improve the quality of synthesised depth images.

This paper[8] proposes use of synthetic data to train the model for handling adverse weather conditions like rain and night using a method called md4all. Md4all utilise existing successful depth estimation methods for ideal conditions. First it generates complex samples corresponding to normal training ones. They trained the model by guiding it self- or full-supervision by feeding the generated samples and computing the standard losses on the corresponding original images. Doing so enables a single model to recover information across diverse conditions without modifications at inference time. There approach was general and not bound to specific architecture

This paper [9] proposes a SID strategy to discretize depth and recast depth network learning as an ordinal regression problem. By training the network using an ordinary regression loss, it achieves much higher accuracy and faster convergence. It adopts a multi-scale network structure which avoids unnecessary spatial pooling and captures multi-scale information in parallel. Proposed DORN achieves state-of-the-art results on three challenging benchmark and outperforms existing methods by large margin

This survey paper [10] reviews five papers that attempt to solve the depth estimation problem with various techniques including supervised, weakly-supervised,and unsupervised learning techniques. It compare these papers and understand the improvements made over one another. Explores the potential improvements that can aid to better solve this problem. Different papers are: Depth Map Prediction from a Single Image using a Multi-Scale Deep Network[4], it uses multiscale information and also introduced the concept of directly regressing over pixels for depth estimation. They use a special scale-invariant loss to account for scale-dependent error. Multi-Scale Continuous CRFs as Sequential Deep Networks for monocular depth estimation[11],A novel approach for

predicting depth maps from RGB inputs which exploit multi-scale estimations derived from CNN inner layers by fusing them within a CRF framework Structured Attention Guided Convolutional Neural Fields for monocular depth estimation[12]Similar framework, using CNN and feeding extracted multi-scale information into a continuous CRF model. The major addition enforcement of similarity constraints and usage of structured attention model which can automatically regulate amount of information transferred between corresponding features at different scales. Deep Ordinal Regression Network for monocular depth estimation[9], Unsupervised monocular depth estimation with Left-Right Consistency[7]

This paper[13] proposes a simple model with minimum reprojection loss, designed to robustly handle occlusions, a full-resolution multi-scale sampling method that reduces visual artifacts and an auto-masking loss to ignore training pixels that violate camera motion assumptions.

This paper[14] proposes an efficient and lightweight encoder-decoder network architecture and apply network pruning to further reduce computational complexity and latency. It demonstrates that it is possible to achieve similar accuracy as prior work on depth estimation, but at inference speeds that are an order of magnitude faster. State-of-the-art single-view depth estimation algorithms are based on complex deep neural networks that are too slow for real-time inference on an embedded platform, so it was introduced to address the problem of fast depth estimation on embedded systems.

Structured light is widely used in the field of depth estimation and shape reconstruction techniques. In this paper[15], they actively utilized motion blur, which they refer to as a light flow, to estimate depth. Analysis reveals that minimum two light flows, which are retrieved from two projected patterns on the object, are required for depth estimation. To retrieve two light flows at the same time, two sets of parallel line patterns are illuminated from two video projectors and the size of motion blur of each line is precisely measured. By analyzing the light flows, i.e. lengths of the blurs, scene depth information is estimated.

CHAPTER 3

RELATED THEORY

3.1 Hardware

The Arduino Nano is a small, breadboard-friendly microcontroller board based on the ATmega328. It is equipped with 30 male I/O headers which can be programmed using the Arduino Software IDE. It can communicate with a computer and other microcontrollers.It consists of flash memory of 32KB capacity, of which 2 KB is used for Bootloader.It supports I2C, SPI and UART.

3.1.1 Raspberry Pi 4 Model B

The Raspberry Pi is a credit card-sized single-board computer that provides a powerful processing platform in a compact form factor. It is equipped with a range of input/output interfaces, including USB ports, HDMI, Ethernet, GPIO, and a camera interface. The processor speed of Raspberry Pi 4 Model B is 1.5 GHz. It comes with onboard wireless networking and Bluetooth .

3.1.2 Motor driver

The L293 and L293D are integrated circuits that can be used to control a variety of motors, including DC motors, stepper motors, and solenoids. They can provide up to 1 amp of current per channel, and they are designed to operate with a wide range of supply voltages (from 4.5 volts to 36 volts). The L293D is a lower-current version of the L293, but it is still capable of providing up to 600 mA of current per channel. Both devices have built-in electrostatic discharge protection and high-noise-immunity inputs, which makes them resistant to damage from electrical spikes and noise. They also have output clamp diodes, which help to suppress inductive transients. The L293 and L293D are designed for use in a variety of applications, including robotics, automation, and automotive systems.[16]

3.1.3 Motor

Motors also known as actuators are fundamental components in robotics, providing the necessary actuation, precision, and control for robot movement and manipulation. With the help of motors, the system can be moved to different locations using a precise control system.

3.2 Software

3.2.1 PyTorch

PyTorch is an open source machine learning framework based on the Python programming language and the Torch library. Torch is an open source machine learning library used for creating deep neural networks and is written in the Lua scripting language. It's one of the preferred platforms for deep learning research. The framework is built to speed up the process between research prototyping and deployment.

3.2.2 Flask

Flask is a lightweight and easy-to-use micro web framework for Python, designed to facilitate quick web development while being flexible enough for complex applications. As a micro framework, Flask provides the essential tools needed to build web applications without mandating specific libraries or components, focusing on simplicity and ease of use. Its flexibility allows developers to choose and integrate various libraries and tools, making it highly customizable. Key features of Flask include a built-in development server and debugger, which streamline testing and debugging processes. It uses Jinja2 as its templating engine, enabling the creation of dynamic HTML pages with powerful features. Flask's intuitive URL routing system maps URLs to Python functions, simplifying endpoint management. [17]

3.2.3 React

React is a popular JavaScript library developed by Facebook for building user interfaces, particularly single-page applications. It allows developers to create large web

applications that can update and render efficiently in response to data changes. React is component-based, meaning the UI is divided into reusable components, each managing its own state and rendering logic. This modular approach promotes code reusability and easier maintenance.

3.3 Visual Cues

3.3.1 Stereo Vision (Binocular Disparity)

Using two images captured from slightly different viewpoints (like human eyes), depth can be estimated by calculating the disparity between corresponding points in the two images. Disparity refers to the difference in the relative position of an object in the two images, which is inversely proportional to the object's distance [18] .

3.3.2 Motion Parallax

Motion parallax is the optical change of the visual field of an observer which results from a change of the observer's viewing position [19]. When an observer or the camera moves, nearby objects appear to move faster across the field of view than distant objects. This relative motion provides cues about the depth and distance of objects [20].

3.3.3 Structured Light

Structured lighting for depth estimation involves projecting a known pattern (like grids or stripes) onto a scene. The pattern's deformation when viewed from a different angle is analyzed to determine the depth and shape of objects.Thus, structured lighting technique is based on projecting a light pattern and viewing the illuminated scene from one or more points of view. By comparing the projected and observed patterns, a depth map is generated [21]. This technique is widely used in 3D scanning, robotics, and computer vision for accurate and detailed depth measurements.

3.3.4 Active Illumination

Active illumination for depth estimation involves projecting light, such as laser or infrared, onto a scene and measuring the reflected light to determine depth [22]. This method provides precise depth information and is commonly used in applications like autonomous vehicles, 3D mapping, and augmented reality.

3.3.5 Texture Gradient

The density of texture patterns changes with distance, textures appear denser and smaller as they get further away. This gradient helps infer the relative depth of surfaces and objects in the image.

3.3.6 Occlusion (Interposition)

When one object overlaps or covers another, the occluded object is perceived to be farther away. This cue provides relative depth information between overlapping objects. Shading and Shadows:

Variations in light and shadow on objects provide information about their shape and depth. The position and length of shadows also help in estimating the spatial relationship and distance of objects.

3.3.7 Aerial Perspective (Atmospheric Perspective)

Distant objects often appear hazier and less distinct due to atmospheric scattering of light. This effect can be used to gauge the depth of objects based on their clarity and color saturation.

3.3.8 Relative Size

When the size of familiar objects is known, their relative sizes in the image can be used to infer their distances. Larger objects are perceived as closer, while smaller ones are seen as farther away.

3.3.9 Known Object Size (Size Constancy)

Using the actual size of known objects to infer depth. If an object appears smaller than its known size, it is inferred to be further away.

3.3.10 Linear Perspective

Parallel lines appear to converge as they recede into the distance, meeting at a vanishing point. This convergence provides cues about the depth and distance of objects along those lines.

3.3.11 Defocus Blur (Depth from Defocus)

Objects at different distances from the camera have different amounts of blur due to the depth of field effect. The amount of defocus blur can be used to estimate the distance of objects from the camera.

CHAPTER 4

METHODOLOGY

The project follows an Iterative Development Process, allowing for continuous refinement and adaptation based on learning from each phase of the project. Different statistical and mathematical tools are used to establish clear decision points throughout the project lifecycle and define key artifacts (both input and output) to guide the project's progression and ensure alignment with project goals.

4.1 System Design

4.1.1 Hardware design

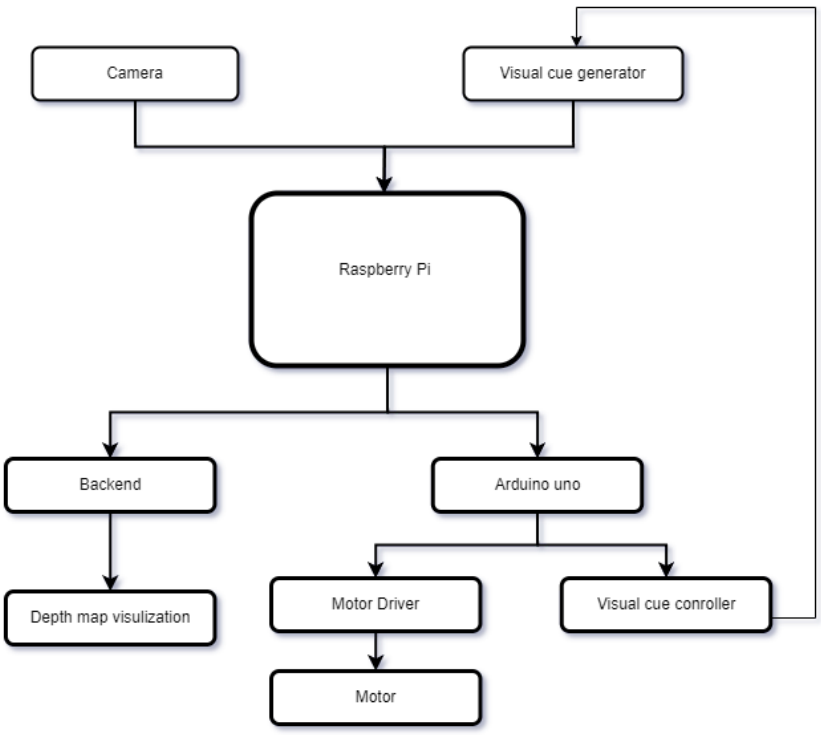


Figure 4.1: Hardware block diagram

The project uses a Raspberry Pi as the brain.Visual Cue generator generates cues that help in understanding how far away objects are. An Arduino Uno microcontroller steps

in to control the Visual cue generator, and motor driver, which in turn powers a motor.The Raspberry Pi processes images captured by camera and sends them to the backend through Wi-Fi for visualization.

4.1.2 Software design

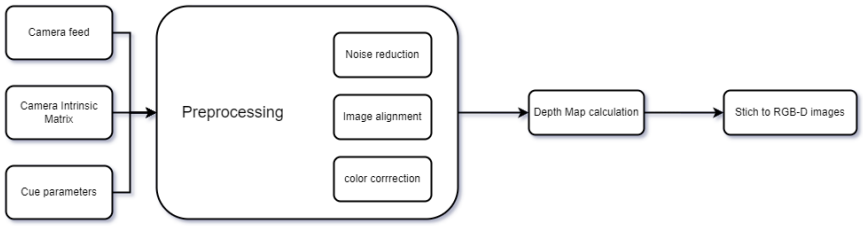


Figure 4.2: Software block diagram

The software for this project tackles the video feed from the camera and turns it into a depth map showing how far away things are. First, the program cleans up the video, getting rid of any noise. If multiple pictures are needed, the software align them up perfectly. Then the model figures out depth from the video.This might involve comparing slightly different pictures. Once it has depth information, the program combines it back with the original color picture to create a special image with both color and depth data. There might be some extra steps to fine-tune the picture and account for the camera's quirks.

4.2 Model Training Pipelines

4.2.1 Supervised learning

The input image is transformed into tokens either by extracting non-overlapping patches followed by a linear projection of their flattened representation or by applying a ResNet-50 feature extractor. The image embedding is augmented with a positional embedding and a patch-independent readout token is added. The tokens are passed through multiple transformer stages. We reassemble tokens from different stages into an image-like representation at multiple resolutions. Fusion modules progressively fuse and upsample the representations to generate a fine-grained prediction. Center: Overview of the Reassembles operation. Tokens are assembled into feature maps with 1 s the spatial

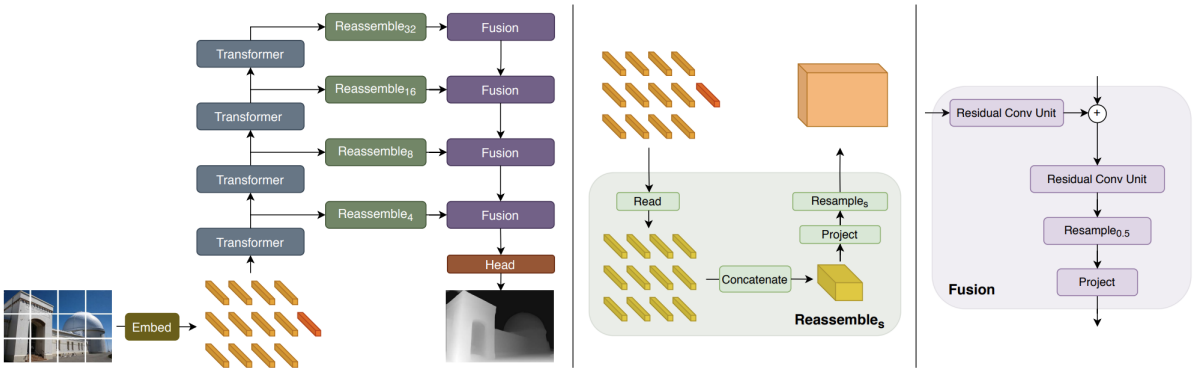
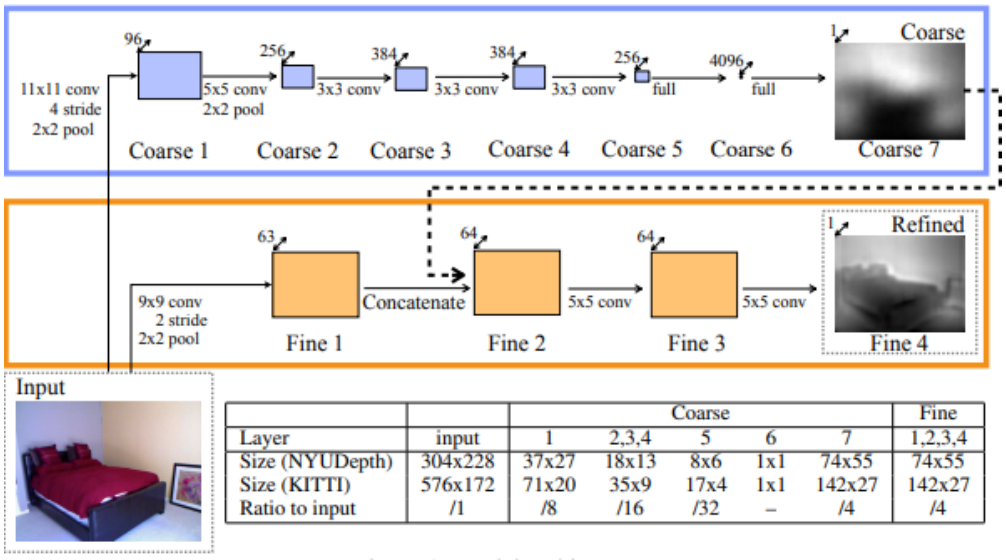


Figure 4.3: Supervised learning of DPT transformer [23]

resolution of the input image. Right: Fusion blocks combine features using residual convolutional units and upsample the feature maps.[23]

4.2.2 Multiscale learning



		Coarse					Fine
Layer	input	1	2,3,4	5	6	7	1,2,3,4
Size (NYUDepth)	304x228	37x27	18x13	8x6	1x1	74x55	74x55
Size (KITTI)	576x172	71x20	35x9	17x4	1x1	142x27	142x27
Ratio to input	/1	/8	/16	/32	–	/4	/4

Figure 4.4: Multiscale learning [**?**]

The described method involves a global, coarse-scale network with five convolution and max-pooling layers, followed by two fully connected layers for feature extraction. Input images are downsampled by a factor of 2, and the final output is at 1/4-resolution of this downsampled input. This output corresponds to a center crop, retaining most of the input image while losing a small border area due to the initial layer of the fine-scale network and image transformations.

CHAPTER 5

WORK ACCOMPLISHED

5.1 EDA

In the initial phase of our project, we focused on performing Exploratory Data Analysis (EDA) on various publicly available datasets for monocular RGB image depth estimation. The purpose of this analysis was to gain a deeper understanding of the datasets, including their characteristics, quality, and relevance to the task. Key steps in the EDA process included:

- Dataset Evaluation: We examined multiple datasets, focusing on aspects such as resolution, diversity, and annotation accuracy.
- Data Distribution Analysis: We studied the depth distribution, identifying trends, anomalies, and outliers that could impact model performance.
- Visualization: Various visualization techniques were employed to inspect the image-depth correlations and validate the dataset's suitability for our objectives.

Figure 5.1: Images of object captured under different lighting condition

5.2 Architecture

During the development process, several architectures were evaluated to determine the most effective design for generating accurate PBR maps. Among these, two architectures stood out for their performance and scalability:

5.2.1 ConvNeXt Encoder with Hourglass Decoder

Overview: This architecture utilizes the *ConvNeXt* encoder, a convolutional neural network backbone renowned for its efficiency and capability in feature extraction, in combination with an *hourglass decoder*. The hourglass decoder is particularly suited for tasks requiring detailed spatial understanding, such as image-to-image translation and PBR map generation.

Strengths:

- High spatial resolution in the output due to the symmetric structure of the hourglass decoder.
- Effective handling of hierarchical feature extraction from images.

Challenges:

- Inconsistent performance on more complex map types, such as metallic and ambient occlusion.
- Computationally more expensive when scaling to larger datasets or higher resolutions.

5.2.2 DINOv2 Encoder with RAFTDepthNormalDPT5 Decoder

Overview: This architecture combines the *DINOv2* encoder, a transformer-based model known for its ability to capture global and contextual features, with the *RAFTDepthNormalDPT5 decoder*, which is specifically optimized for generating depth and normal maps.

Strengths:

- Superior performance in generating depth and normal maps due to its transformer-based design and attention mechanisms.
- Better generalization across diverse datasets, making it ideal for handling varied features in PBR maps.

- Scalable and robust for transfer learning tasks.

Challenges:

- High computational resource requirements during training due to the transformer-heavy architecture.
- Further optimization needed for roughness, metallic, and ambient occlusion maps to fully leverage the encoder's capabilities.

5.2.3 Rationale for Selection

While both architectures demonstrated significant strengths, the *DINOv2 + RAFTDepth-NormalDPT5* configuration was selected as the primary model for the following reasons:

- Its ability to capture global contextual information results in more accurate and consistent map generation.
- The flexibility of the architecture facilitates better integration with transfer learning techniques.
- Preliminary results indicate superior performance on complex datasets and a wider range of map types.

By focusing on refining this architecture and addressing its current limitations, the project aims to achieve highly accurate and efficient PBR map generation.

5.3 Image Analysis and Processing

After completing the EDA and choosing the best architecture, we proceeded to image analysis and processing to enhance the data and extract features that contribute to accurate depth estimation. During this phase, we explored several techniques, and Wavelet Analysis and FFT (Fast Fourier Transform) Analysis emerged as the most effective methods for our application.

- Wavelet Analysis

  Wavelet analysis involves breaking an image into smaller components, called wavelets, which are localized in both time (or space) and frequency. This method allows us to analyze an image at multiple scales and resolutions. Wavelets are particularly useful for capturing fine details, such as edges and textures. By decomposing an image into wavelet coefficients, we were able to highlight important features while reducing noise.

- FFT Analysis

  FFT (Fast Fourier Transform) analysis converts the spatial representation of an image into its frequency components. This technique helps identify patterns and repetitive structures in the image by analyzing its frequency spectrum. High-frequency components correspond to fine details, while low-frequency components capture the broader structures. FFT analysis enabled us to emphasize essential features in the image while suppressing irrelevant information, further improving the depth estimation process. These techniques provided valuable insights and improvements in our preprocessing pipeline, contributing to the robustness of our depth estimation approach

- Local Binary Pattern (LBP) Transform

  The *Local Binary Pattern* (LBP) transform is a widely used method for texture analysis and feature extraction in image processing. It is simple, computationally efficient, and highly effective for capturing local texture information. The LBP transform works by comparing the intensity of a central pixel in a local neighborhood with its surrounding pixels. For each surrounding pixel:

  - If the intensity of the neighboring pixel is greater than or equal to the central pixel, it is assigned a value of 1.
  - Otherwise, it is assigned a value of 0.

  These binary values form a binary pattern (usually an 8-bit binary number for a 3×3 neighborhood), which is then converted into a decimal value to represent the texture feature.

Figure 5.2: Texture extraction by LBP methods (left) and by Wavelet method (right)

Figure 5.3: Depth Map

Figure 5.4: Normal Map

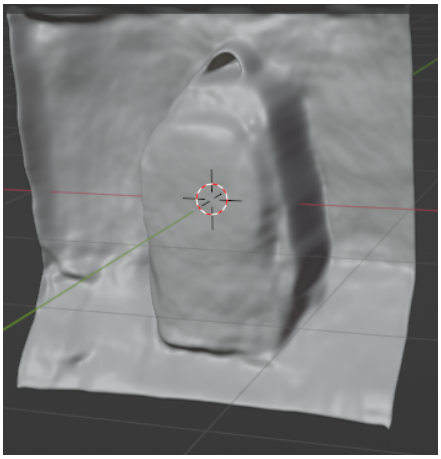


Figure 5.5: 3D Mesh of Bag

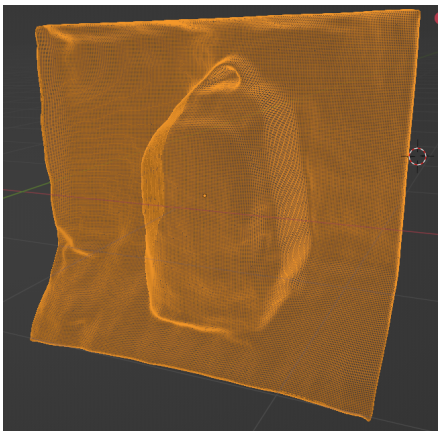


Figure 5.6: Wireframe of Mesh

Figure 5.7: Photorealistic Textured 3D Model

CHAPTER 6

WORK TO BE ACCOMPLISHED

6.1 Stable Architecture for Accurate PBR Map Extraction

A robust architecture has been developed to extract Physically Based Rendering (PBR) maps accurately. The finalized model architecture incorporates *DINOv2* as the encoder and *RAFTDecoderDPT5* for decoding features. The system leverages publicly available datasets and datasets generated in-house using standardized techniques to ensure quality and consistency. While progress has been made in transfer learning for depth and normal maps, further training is required for the following PBR maps:

- Roughness Map
- Metallic Map
- Ambient Occlusion Map

6.2 Training on Remaining PBR Maps

The current focus is on training the model to accurately generate the roughness, metallic, and ambient occlusion maps. This involves augmenting the dataset to include high-quality samples for these map types and fine-tuning the architecture to handle their specific features effectively.

6.3 Enhanced Dataset Generation and Analysis

To improve model performance, it is essential to generate a more accurate and detailed dataset. This includes refining the process for dataset creation and ensuring the inclusion of diverse and complex examples. A detailed analysis will be conducted to identify areas of improvement and validate the dataset's utility.

6.4 Comparative Study of Model Output

A comprehensive study comparing the outputs generated by our model with those from existing methods will be undertaken. This involves both qualitative and quantitative evaluation of the results to assess improvements and areas where our model excels or lags.

6.5 Developing a Stable Mesh Mapping Method

To make the generated PBR maps practical for 3D applications, a stable and efficient method for mapping these textures onto 3D meshes must be developed. This will require exploring and implementing algorithms that streamline the integration process while maintaining the quality of the visual output.

6.6 Budget analysis

Components	**Quantity**	**Unit Cost**	**Total Cost**
Raspberry Pi	1	NRs. 11000	NRs. 11000
Arduino Nano	1	NRs. 800	NRs. 800
Camera	2	NRs. 1500	NRs. 3000
Flashlight	1	NRs. 400	NRs. 400
Motor Driver	1	NRs. 300	NRs. 300
Motor	4	NRs. 250	NRs. 1000
Miscellaneous	-	-	NRs. 1750
Total	-	-	**NRs. 18250**

Table 6.1: Budget analysis

6.7 Gantt chart

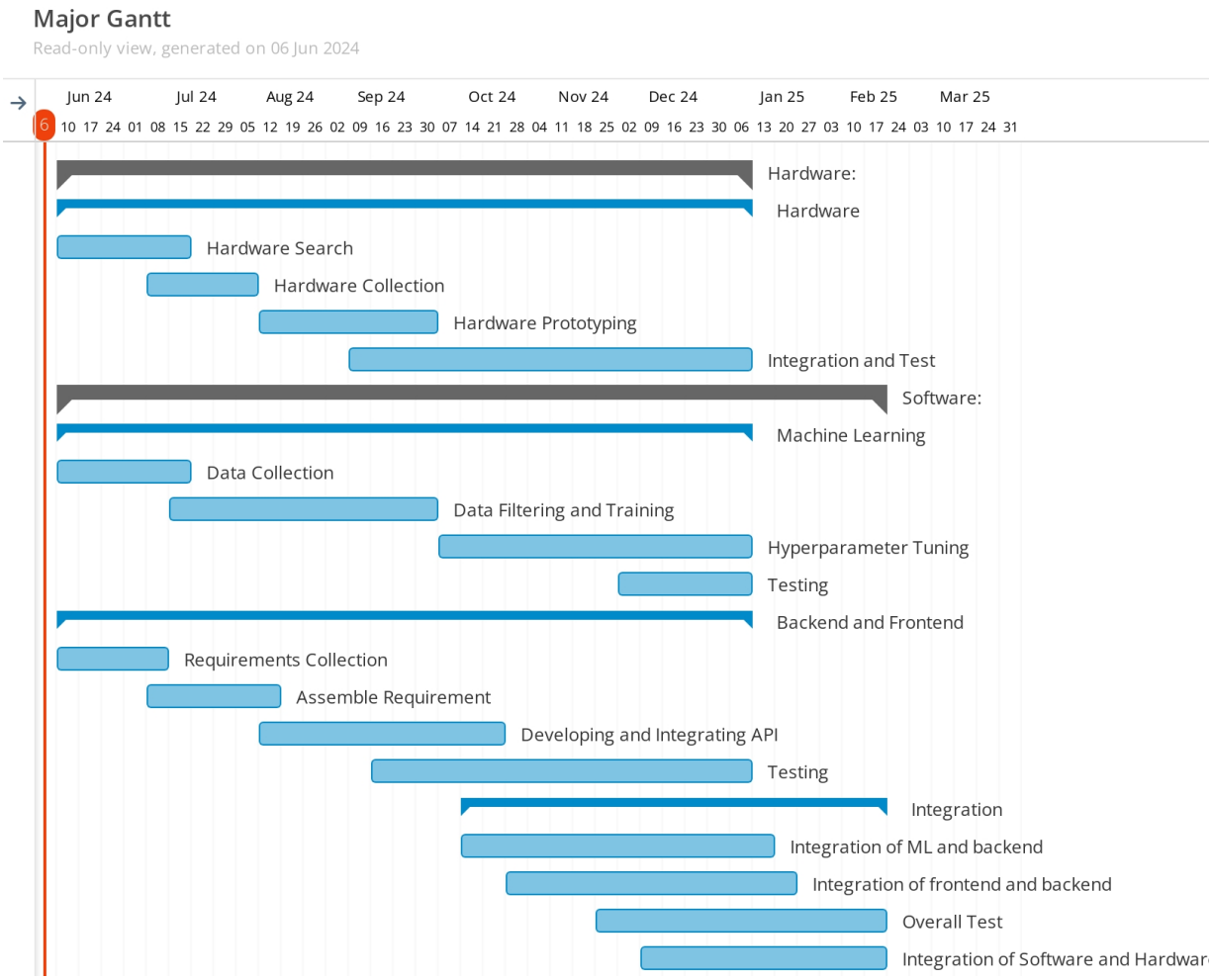


Figure 6.1: Gantt chart

REFERENCES

[1] M. J. Brooks and B. K. Horn, "Shape and source from shading," 1985.

[2] D. Marr, "A photogrammetric method for determining the shape of a nonrigid object from a sequence of photographs," 1975.

[3] K.-i. Kanatani and T.-C. Chou, "Shape from texture: General principle," *Artificial Intelligence*, vol. 38, no. 1, pp. 1–48, 1989.

[4] D. Eigen, C. Puhrsch, and R. Fergus, "Depth map prediction from a single image using a multi-scale deep network," *Advances in neural information processing systems*, vol. 27, 2014.

[5] G. Liu, G. Jiang, R. Xiong, and Y. Ou, "Binocular depth estimation using convolutional neural network with siamese branches," in *2019 IEEE International Conference on Robotics and Biomimetics (ROBIO)*. IEEE, 2019, pp. 1717–1722.

[6] A. Roy and S. Todorovic, "Monocular depth estimation using neural regression forest," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 5506–5514.

[7] C. Godard, O. Mac Aodha, and G. J. Brostow, "Unsupervised monocular depth estimation with left-right consistency," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 270–279.

[8] S. Gasperini, N. Morbitzer, H. Jung, N. Navab, and F. Tombari, "Robust monocular depth estimation under challenging conditions," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2023, pp. 8177–8186.

[9] H. Fu, M. Gong, C. Wang, K. Batmanghelich, and D. Tao, "Deep ordinal regression network for monocular depth estimation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 2002–2011.

[10] D. Wang, Z. Liu, S. Shao, X. Wu, W. Chen, and Z. Li, "Monocular depth estimation: A survey," in *IECON 2023-49th Annual Conference of the IEEE Industrial Electronics Society*. IEEE, 2023, pp. 1–7.

[11] D. Xu, E. Ricci, W. Ouyang, X. Wang, and N. Sebe, "Multi-scale continuous crfs as sequential deep networks for monocular depth estimation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 5354–5362.

[12] D. Xu, W. Wang, H. Tang, H. Liu, N. Sebe, and E. Ricci, "Structured attention guided convolutional neural fields for monocular depth estimation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 3917–3925.

[13] C. Godard, O. Mac Aodha, M. Firman, and G. J. Brostow, "Digging into self-supervised monocular depth estimation," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2019, pp. 3828–3838.

[14] D. Wofk, F. Ma, T.-J. Yang, S. Karaman, and V. Sze, "Fastdepth: Fast monocular depth estimation on embedded systems," in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 6101–6108.

[15] R. Furukawa, R. Sagawa, and H. Kawasaki, "Depth estimation using structured light flow–analysis of projected pattern flow on an object's surface," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 4640–4648.

[16] "L293x quadruple half-h drivers." [Online]. Available: https://www.ti.com/product/L293D

[17] [Online]. Available: https://flask.palletsprojects.com/en/3.0.x/

[18] W. E. L. Grimson, "A computer implementation of a theory of human stereo vision," *Philosophical Transactions of the Royal Society of London. B, Biological Sciences*, vol. 292, no. 1058, pp. 217–253, 1981.

[19] E. J. Gibson, J. J. Gibson, O. W. Smith, and H. Flock, "Motion parallax as a determinant of perceived depth." *Journal of experimental psychology*, vol. 58, no. 1, p. 40, 1959.

[20] M. Nawrot and K. Stroyan, "The motion/pursuit law for visual depth perception from motion parallax," *Vision research*, vol. 49, no. 15, pp. 1969–1978, 2009.

[21] J. Salvi, J. Pages, and J. Batlle, "Pattern codification strategies in structured light systems," *Pattern recognition*, vol. 37, no. 4, pp. 827–849, 2004.

[22] Z. Cai, X. Liu, G. Pedrini, W. Osten, and X. Peng, "Accurate depth estimation in structured light fields," *Opt. Express*, vol. 27, no. 9, pp. 13 532–13 546, 4 2019. [Online]. Available: https://opg.optica.org/oe/abstract.cfm?URI=oe-27-9-13532

[23] R. Ranftl, A. Bochkovskiy, and V. Koltun, "Vision transformers for dense prediction," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2021, pp. 12 179–12 188.