

TRIBHUVAN UNIVERSITY

INSTITUTE OF ENGINEERING

PASHCHIMANCHALCAMPUS

LAMACHAUR, POKHARA

[Subject Code: EX 707]

A PROGRESS REPORT

ON

**Texture and Depth Estimation**

**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**

[**ABSTRACT i**](#_Toc20268)

[**TABLE OF CONTENTS ii**](#_Toc20269)

[**LIST OF FIGURES iv**](#_Toc20270)

[**LIST OF TABLES v**](#_Toc20271)

[**LIST OF ABBREVIATIONS vi**](#_Toc20272)

[**1 INTRODUCTION 1**](#_Toc20273)

[1.1 Background 1](#_Toc20274)

[1.2 Problem Statement 1](#_Toc20275)

[1.3 Objectives 2](#_Toc20276)

[1.4 Application 2](#_Toc20277)

[**2 LITERATURE REVIEW 4**](#_Toc20278)

[**3 METHODOLOGY 1**](#_Toc20279)

[3.1 System Design 1](#_Toc20280)

[3.1.1 Hardware Design 1](#_Toc20281)

[3.1.2 Software Design 1](#_Toc20282)

[3.2 Model Training Pipelines 3](#_Toc20285)

[3.2.1 Supervised Learning 3](#_Toc20286)

[3.2.2 Multiscale Learning 3](#_Toc20287)

[**4 EXPECTED OUTPUT AND ANALYSIS 7**](#_Toc20301)

[4.1 Expected output 7](#_Toc20302)

[4.1.1 Budget analysis 7](#_Toc20303)

[4.1.2 Gantt Chart 8](#_Toc20304)

[**5 RESULT AND DISCUSSION 10**](#_Toc20308)

[**5.1 WORK ACCOMPLISHED 10**](#_Toc20309)

[5.1.1 PHASE I: Exploratory Data Analysis (EDA) 7](#_Toc20303)

[5.1.2 PHASE II: Architecture Model 7](#_Toc20303)

[5.1.3 PHASE III: Image Analysis and Processing 7](#_Toc20303)

[5.1.4 PHASE IV: Depth Map Estimation Techniques 7](#_Toc20303)

[**5.2 WORK TO BE DONE 10**](#_Toc20309)

[**6 CONCLUSION 10**](#_Toc20308)

**REFERENCES** 14

# 

# LIST OF FIGURES

|  |  |
| --- | --- |
| 3.1 Arduino Nano . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | 1 |
| 3.2 Raspberry Pi 4B . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | 2 |
| 3.3 Servo Motor . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .. . | 3 |
| 3.4 DC Motor . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | 3 |
|  |  |
|  |  |
|  |  |
| 4.4 Multiscale learning . . . . . . . . . . . . . . . . . . . . . . . . . . . . | 9 |
| 5.1 Gantt chart . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | 11 |

# LIST OF TABLES

5.1 Budget analysis . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 10

# LIST OF ABBREVIATIONS

|  |  |
| --- | --- |
| *ADC* | *Analog to Digital Converter* |
| *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], photogrammetric Shape from texture [3], based on binocular camera through stereo matching and triangulation to obtain a depth map.

After the advancement of Deep neural networks, multiple papers have 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 neighboring 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 synthesized 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 utilize 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. Their 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 compares 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 projection 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

# 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.

## 3.1 System Design

### 3.1.1 Hardware design

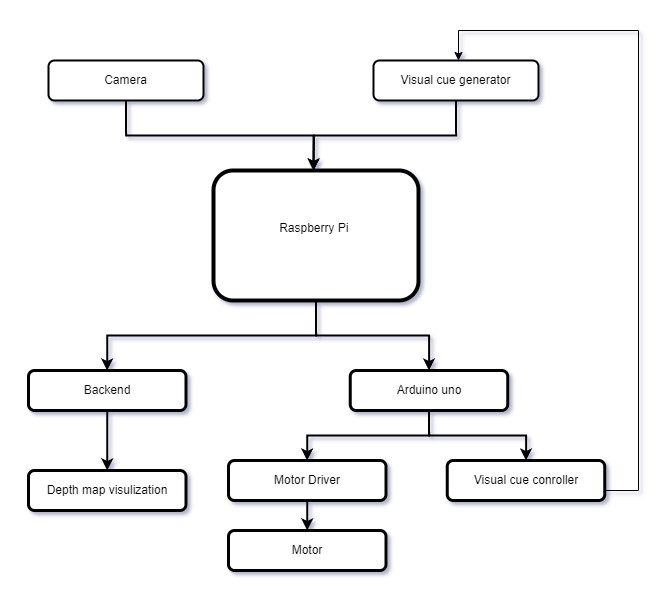


Figure 4.1: Hardware block diagram

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.

### 3.1.2 Software design

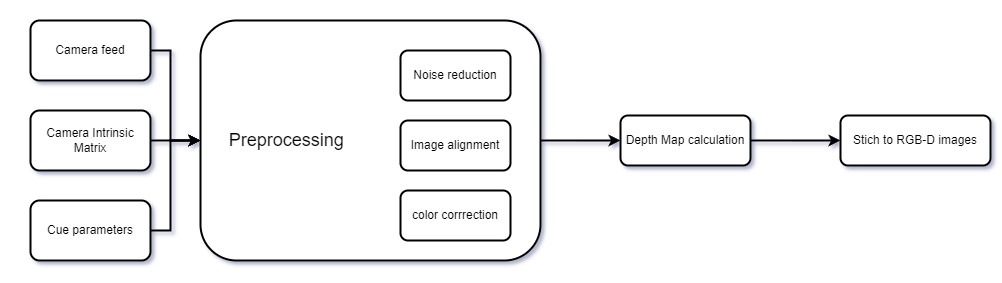


Figure 4.2: Software block diagram

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.

## 3.2 Model Training Pipelines

### 3.2.1 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 ResNet50 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

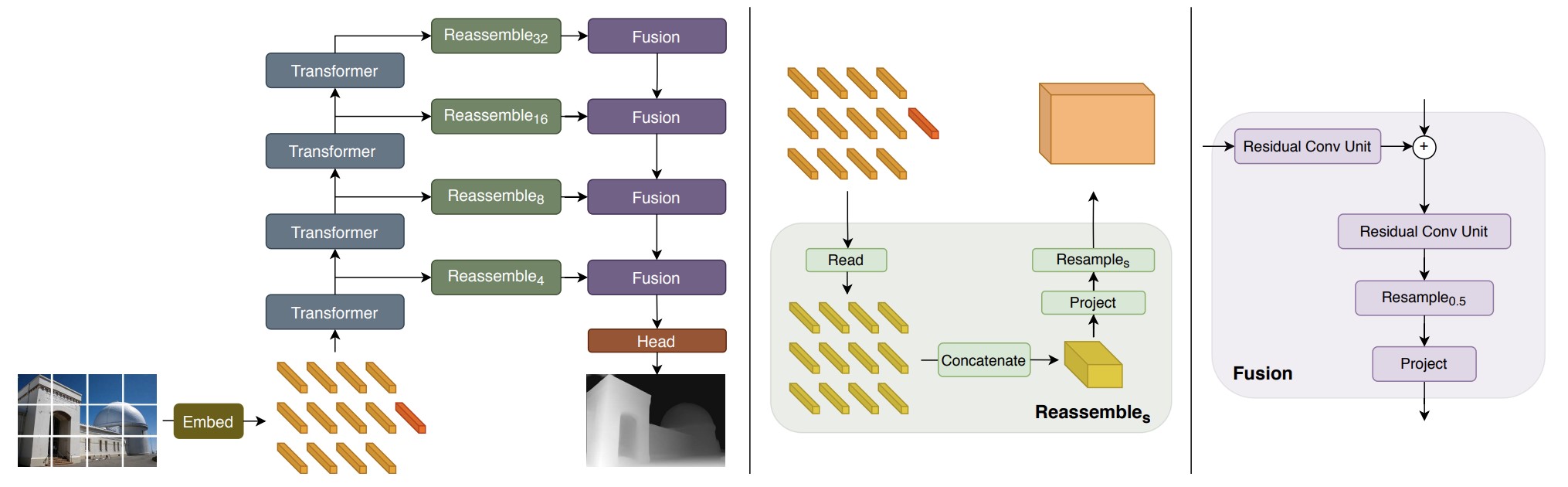


Figure 4.3: Supervised learning of DPT transformer [23]

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

### 3.2.2 Multiscale learning

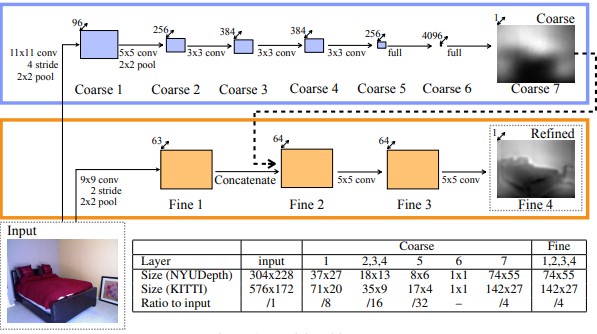


Figure 4.4: Multiscale learning [24]

The described method involves a global, coarse-scale network with five convolutions and max-pooling layers, followed by two fully connected layers for feature extraction. Input images are down sampled by a factor of 2, and the final output is at 1/4-resolution of this down sampled 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. [24]

**CHAPTER 5**

# **RESULT AND DISCUSSION**

## **5.1 WORK ACCOMPLISHED:**

### 5.1.1 PHASE I: Exploratory Data Analysis (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:

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

### 5.1.2 PHASE II: Architecture Model

.

.

.

.

.

.

.

.

.

### 5.1.3 PHASE III: 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

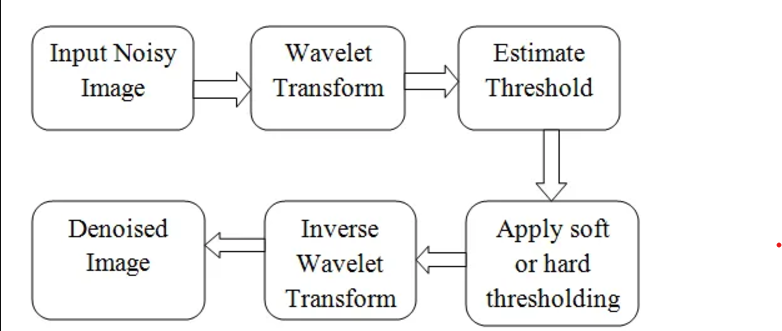
Wavelet analysis involves breaking an image into smaller components, called wavelets, which are localized in both time (or space) and frequency. This method allows us to analyze an image at multiple scales and resolutions. Wavelets are particularly useful for capturing fine details, such as edges and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

Figure 5.1: Block diagram of wavelet denoising technique

* 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

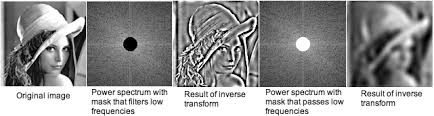


Figure 5.2: FFT Transformation

### 5.1.4 PHASE IV: Depth Map Estimation Techniques

After performing image analysis, **Wavelet Analysis** and **FFT Analysis** demonstrated their effectiveness in enhancing image features for depth estimation. Leveraging these insights, we proceeded to estimate depth for images with varying characteristics using different methods:

1. **Images with Roughness: PyWavelet**

For images containing rough surfaces and textures, we utilized **PyWavelet**, a Python-based wavelet analysis library, to achieve precise depth estimation. Rough surfaces are characterized by high-frequency details and irregular patterns. Wavelet transformations excel in analyzing such features due to their multi-scale nature.

* + **Process:** The images were decomposed into wavelet coefficients, allowing the algorithm to focus on detailed structures. These coefficients were then used to estimate depth accurately, ensuring that the complex textures were properly represented.

1. **Images with Metallic Objects: SAM Model**

Metallic objects in images presented unique challenges due to their reflective properties and sharp highlights. The **SAM (Segment Anything Model)** proved to be highly effective in addressing these issues. Metallic objects often exhibit strong reflections that can distort depth perception. SAM, a robust segmentation model, isolates these objects and analyzes their reflective properties.

* + **Process:** Using SAM, metallic regions in the image were segmented and processed independently. The model incorporated information about light reflection and surface curvature to generate accurate depth maps for these regions.

1. **Ambient Occlusion Images**

For images where ambient occlusion effects were prominent, we developed a two-step approach to generate accurate depth and occlusion maps:

* + **Step 1: Depth Map Generation:** The depth map of the image was generated using standard depth estimation techniques. This provided a foundational understanding of object placement and surface orientation.
  + **Step 2: Ambient Occlusion Map Construction:** By analyzing the depth map, we identified regions with reduced light exposure (shaded areas). These regions were mapped into an ambient occlusion map, highlighting areas where objects block light from reaching surfaces. This technique accurately captured subtle depth variations and enhanced realism.

## **5.1 WORK TO BE DONE :**

.

.

.

.

.

.

.

.

.

**CHAPTER 6**

# **CONCLUSION**

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 selfsupervised 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

1. [Online]. Available: https://flask.palletsprojects.com/en/3.0.x/
2. 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.

1. 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.

1. 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.
2. J. Salvi, J. Pages, and J. Batlle, “Pattern codification strategies in structured light systems,” *Pattern recognition*, vol. 37, no. 4, pp. 827–849, 2004.
3. 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. 13532–13546, 4 2019. [Online]. Available: https://opg.optica.org/oe/abstract.cfm?URI=oe-27-9-13532
4. 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. 12179–12188.
5. 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.