UNIVERSITY OF SOUTHAMPTON

Faculty of Engineering and Physical Sciences
School of Electronics and Computer Science

Depth Estimation for Indoor Single Omnidirectional Images

by

Yihong Wu

ORCiD: 0000-0003-3340-2535

Main Supervisor: Hansung Kim
Second Supervisor: Mahesan Niranjan

A thesis for the degree of
Doctor of Philosophy

September 2024

University of Southampton

Abstract

Faculty of Engineering and Physical Sciences
School of Electronics and Computer Science

Doctor of Philosophy

Depth Estimation for Indoor Single Omnidirectional Images

by Yihong Wu

Omnidirectional cameras are becoming popular in various applications owing to their ability to capture the full surrounding scene in one frame. However, depth estimation for an omnidirectional scene is more difficult than for general images due to its different system properties and distortions. Monocular depth estimation for single-view using deep learning can be a good solution, but it requires a large labelled depth dataset with various scenes. Currently published omnidirectional depth datasets cover limited types of scenes and are not suitable for depth estimation for various real-world scenes. In addition, the existing methods are basically data-driven, and the depth estimation process based on deep learning is still a black box. In order to overcome these problems, we first proposed a depth estimation architecture for a single omnidirectional image using domain adaptation, only with limited labelled real-world scenes. With the challenge of getting labelled real-world datasets and stability of the performance, we updated the components of architecture and proposed a reverse-gradient warming-up threshold discriminator (RWTD) to estimate real-world depth maps from synthetic ground truth. It takes labelled synthetic scenes of a source domain and unlabelled real-world scenes of a target domain as inputs to predict the corresponding depth maps. To solve the black-box depth estimation process, we analyse the role of gravity in depth estimation and propose a slicing method based on the gravity direction. Equally crucial to our investigation is the examination of the contributions of different cues to the results of indoor depth estimation. The results show that the four factors of colour, saturation, local texture and shape show different extent contributions, and among them, the shape feature plays a dominant role in the performance of depth estimation. These works present solutions for depth estimation of omnidirectional images in real-world applications and delve into the critical role of gravity alignment, as well as the exploration of how machines perceive depth, providing a foundation for subsequent research.

Contents

List of Figures

List of Tables

Listings

Declaration of Authorship

I declare that this thesis and the work presented in it is my own and has been generated by me as the result of my own original research.

I confirm that:

1. This work was done wholly or mainly while in candidature for a research degree at this University;
2. Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated;
3. Where I have consulted the published work of others, this is always clearly attributed;
4. Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work;
5. I have acknowledged all main sources of help;
6. Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself;
7. Parts of this work have been published as:

   Publications as the First Author:

   - Yihong Wu, Yuwen Heng, Mahesan Niranjan, and Hansung Kim. Sliceformer: Deep dense depth estimation from a single indoor omnidirectional image using a slice-based transformer. In *International Conference on Electronics, Information and Communication (ICEIC)*, January 2024
   - Yihong Wu, Yuwen Heng, Mahesan Niranjan, and Hansung Kim. Depth estimation for a single omnidirectional image with reversed-gradient warming-up thresholds discriminator. In *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5. IEEE, 2023

- Yihong Wu, Yuwen Heng, Mahesan Niranjan, and Hansung Kim. Depth estimation from a single omnidirectional image using domain adaptation. In *European Conference on Visual Media Production (CVMP)*, pages 1–9, 2021

Publications as Co-author:

- Jiaqi Zhou, Yihong Wu, Hwasub Lim, and Hansung Kim. Omnidirectional depth estimation for semantic segmentation. In *International Conference on Electronics, Information and Communication (ICEIC)*, January 2024
- Yuwen Heng, Yihong Wu, Srinandan Dasmahapatra, and Hansung Kim. Enhancing material features using dynamic backward attention on cross-resolution patches. In *British Machine Vision Conference (BMVC)*, page 4, 2022b
- Mona Alawadh, Yihong Wu, Yuwen Heng, Luca Remaggi, Mahesan Niranjan, and Hansung Kim. Room acoustic properties estimation from a single 360° photo. In *2022 30th European Signal Processing Conference (EUSIPCO)*, pages 857–861. IEEE, 2022
- Yuwen Heng, Yihong Wu, Srinandan Dasmahapatra, and Hansung Kim. Cam-segnet: A context-aware dense material segmentation network for sparsely labelled datasets. In *Proceedings of the 17th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP 2022)*, pages 190–201, 2022a

Signed:..... Date:.....

Acknowledgements

I would like to express my deepest gratitude to my supervisors, Dr. Hansung Kim and Professor Mahesan Niranjan, for their invaluable guidance and support throughout my research. Their extensive knowledge and dedication profoundly influenced my work, providing unique insights that were crucial to my development as a researcher.

My heartfelt thanks also go to my colleagues and friends, particularly Yuwen, Ruixiao, and Mona. Yuwen provided invaluable feedback through in-depth discussions of my ideas, Ruixiao broadened my perspective and guided me toward projects in autonomous driving, and Mona played a crucial role in validating critical task designs, such as immersive sound reconstruction. Their contributions, along with those from other friends and colleagues who shared their expertise and advice, have significantly enriched my academic and personal growth.

I am also deeply grateful to the members of the VLC and AIC research groups for their generous assistance, which has been a constant source of warmth and support. Their care has enriched my experience, alleviated much of my stress, and made my PhD journey both enjoyable and memorable.

Finally, I owe my deepest thanks to my family, my steadfast source of strength, motivation, and inspiration. Their love, understanding, sacrifices, and financial backing have provided the solid foundation that made this achievement possible. Their unwavering belief in my potential has been my greatest encouragement.

Definitions and Abbreviations

AI	Artificial Intelligence
AR	Augmented Reality
BiLSTM	Bidirectional Long Short-Term Memory network
BiRNNs	bidirectional recurrent neural networks
CG	Computer Graphics
CKA	Centred Kernel Alignment
CNNs	convolutional neural networks
CRFs	conditional random fields
CVMP	European Conference on Visual Media Production
DAN	Deep adaptation networks
DLL	Domain Label Loss
DLLF	domain label loss factor
ESWA	Expert Systems with Applications
FoV	field-of-view
GAN	generative adversarial network
GRUs	gated recurrent units
GT	ground truth
hLSTM	horizontal BiLSTM
HSV	hue, saturation and luminance value
ICASSP	International Conference on Acoustics, Speech and Signal Processing
ICEIC	International Conference on Electronics, Information, and Communication
LSTM	long short-term memory networks
log10	logarithmic error
MLP	multi-layer perceptron
NLLLoss	NegativeLog-Likelihood Loss
RD	Reverse-gradient Discriminator
rel	absolute relative error
RNNs	Recurrent neural networks
ROS	Random object scaling
rmse	root mean squared error
RWTD	reverse-gradient warming-up threshold discriminator
SI	Scale-invariant

SLAM	simultaneous localisation and mapping
SfM	structure from motion
SSIM	Structural Similarity
ToF	Time-of-flight
ViT	vision transformer
vLSTM	vertical BiLSTM
VR	virtual reality

Chapter 1

Introduction

3D scene reconstruction and representation have been essential tasks in computer vision and robot vision in the past decades. As one of the most important tasks of 3D scene reconstruction, depth estimation predicts the distance between the visible surface and the sensors (Steger et al., 2018). Specifically, it is a pixel-to-pixel level task that involves taking an input RGB image and predicting the distance of each pixel in the image from the camera (shown in Fig. 1.1).

1.1 Depth Estimation from Visual Inputs

1.1.1 Traditional Depth Estimation

Early research on depth estimation focused on methods based on traditional geometry and vision. Stereo vision (Szeliski, 2010) is one of the early depth estimation methods

(A) RGB Sample (B) Corresponding Depth

FIGURE 1.1: Depth Estimation Example. The left image is an RGB image, while the right image is the corresponding pixel-level depth map.

that estimate depth by comparing two or more images taken from different perspectives. The stereo vision method mainly relies on finding corresponding points between images and then calculating the disparity between these points to estimate depth.

Structured light (Szeliski, 2010) is also a popular early method for depth estimation research. This method estimates depth information by projecting known patterns of light into a scene and then observing how those patterns deform due to different depths of the surface. For example, the stripe pattern, a common example of a structured light pattern, is utilised for depth information estimation. This involves projecting horizontal or vertical stripes onto a scene. The stripes deform in response to the surface geometry, thereby reflecting depth information. In addition, there are a variety of structured light methods, such as grid patterns and dot patterns.

Time-of-flight (ToF) (Szeliski, 2010) technology employs a ToF camera that emits a light pulse, typically infrared, onto an object. The light reflects back from the object's surface and is captured by the sensor. The ToF camera measures the total time of the light pulse from transmission to return. Since the speed of light is known, distances can be calculated based on the time it takes the light to travel. By repeating this process for measurements at each point in the scene, the ToF camera can generate a depth map where the value of each pixel represents the distance of the corresponding point from the camera. Different from the stereo vision, ToF method measures distance directly rather than relying on image contrast as stereo vision does. However, this method has the problem of low resolution, and it is sensitive to the properties of the surface of the object and will affect the accuracy of the measurement for the surface of the object with low reflectivity. In addition, this method may encounter the problem of multipath interference in complex environments; that is, the light pulse may be reflected several times before reaching the sensor, thus affecting the accuracy of the measurement.

1.1.2 Depth Estimation from Sensors

Depth sensors are based on the above traditional foundations.

LiDARs work by emitting laser pulses and receiving the reflection and then measuring the duration for these pulses to travel (Collis, 1970; Wandinger, 2005). Based on the time it takes for the light to travel back and forth, the system is able to calculate the distance from the laser to the object. It is often used to accurately measure large areas of three-dimensional space, such as terrain mapping, self-driving cars, drone navigation, and other fields. However, due to complex mechanical components, optical components, and accurate time measurement systems, LiDARs are expensive.

Infrared sensors (Yuzbasioglu and Barshan, 2005) and ultrasonic sensors (Shahira et al., 2019) are also used in the field of depth estimation. For example, Kinect uses structured light and time of fly (Smisek et al., 2013). Specifically, infrared is used to project a

specific pattern into the observed scene, and then the infrared camera captures the deformation of the pattern on the scene object and calculates the depth information of the object by analyzing the deformation. Combined with the time it takes the light pulse to be transmitted to the scene and reflected back to the sensor, depth estimation is made. However, they have deficiencies such as low-resolution (Zioulis et al., 2018), inaccuracy in textureless regions, short sensing range and an expensive reconstruction process (Alhashim and Wonka, 2018).

1.1.3 Deep Learning based Methods

There have been many different approaches for depth estimation from visual inputs, such as using motion parallax in videos (Lei et al., 2015), multi-view geometry from multiple cameras (Steger et al., 2018), and depth cues from a single image (Bhoi, 2019). The stereo or multi-view approaches require more constraints in system configuration, such as camera calibration and synchronisation between cameras. In contrast, the single-view approach has more flexibility in its applications, although estimating depth from single images is a challenge for artificial intelligence (AI). A human can perceive depth even from one eye through various monocular depth cues about the scene, e.g., shadow, motion parallax, relative size, etc., based on prior knowledge and experiences (Howard, 2012). By following this way, AI can take computer vision beyond simple tasks such as object recognition and localisation segmentation to complicated tasks like scene understanding. The comparative experiments have been conducted on various aspects, such as object size and camera pose, to reveal exactly how the network learns depth from a single image input (Dijk and Croon, 2019).

1.2 Applications

Depth estimation has a wide range of applications. It is one of the key technologies for self-driving cars, as it needs to measure and estimate the distance to surrounding cars, pedestrians, and barriers (Luo et al., 2018; Wang et al., 2019; Janai et al., 2020). Augmented reality (AR) applications require depth information of a scene to provide users with an immersive experience and correct spatial perception (Lee et al., 2011; El Jamiy and Marsh, 2019). Depth estimation is also useful in surveillance applications (Lamża et al., 2013; Alphonse and Sriharsha, 2021), indoor navigation (Machkour et al., 2023; Kang et al., 2014) and 3D reconstruction (Kim et al., 2022; Zhang et al., 2020; Pan et al., 2020; Alawadh et al., 2022).

1.2.1 Virtual Reality and Augmented Reality

Depth estimation is used in virtual and augmented reality to achieve a more realistic perception of the scene, which can help improve the user's sense of interacting with the virtual world (Lee et al., 2011; El Jamiy and Marsh, 2019). By obtaining depth information about objects in the scene through depth estimation, virtual and augmented reality systems can more accurately overlay virtual objects into the real world, making these virtual objects more integrated with the real-world environment, thereby improving the user experience.

1.2.2 Autonomous Driving Systems

Depth estimation is one of the key techniques in autonomous driving systems. It helps to identify objects on the road, estimate the distance between pedestrians, vehicles, and obstacles, and create environmental maps (Luo et al., 2018; Wang et al., 2019; Janai et al., 2020). It enables automated driving systems to effectively detect and avoid obstacles and understand road structures to ensure safe driving. It is of great significance for realising the safety, and reliability of autonomous driving.

1.2.3 Indoor navigation

Depth estimation helps indoor navigation systems to navigate within buildings (Machkour et al., 2023). This includes detecting walls, furniture, etc., as well as providing accurate depth information about the current location relative to the target. For example, a household robot vacuum cleaner can avoid obstacles during the cleaning process with the help of depth estimation (Kang et al., 2014).

1.2.4 Surveillance

Depth estimation can also be used for video analysis in surveillance cameras, including target recognition, motion tracking, and behaviour analysis (Lamża et al., 2013). By estimating the depth of the object, the system can more accurately understand the scene and provide a more intelligent monitoring system. With the help of depth estimation, for example, surveillance systems can interpret human actions and identify criminal activity using a single RGB camera (Alphonse and Sriharsha, 2021).

1.2.5 3D Reconstruction

Depth estimation plays an important role in the field of 3D reconstruction, which realises 3D modelling by obtaining depth information from 2D images (Kim et al., 2022). For example, depth estimation can be used to help generate indoor layouts of buildings (Zhang et al., 2020), helping architects visualise designs and make more precise plans in indoor architectural design. In cultural heritage conservation, depth estimation can be used to reconstruct heritage (Pan et al., 2020), which can digitise tangible cultural heritage to ensure their integrity and preservation and provide a powerful tool for research and education. Depth estimation can also help generate initial 3D scenes from a single image (Alawadh et al., 2022), which contributes to game design.

1.3 Indoor Depth Estimation

Indoor depth estimation holds significant practical application value for specific scenarios, such as indoor navigation (Zhong et al., 2004), augmented reality (Sari et al., 2023), and domestic robots (Zhou et al., 2014). These applications often require a precise understanding of indoor spaces and the complexity and variability of indoor environments necessitate dedicated research.

Given the critical importance of indoor depth estimation, where scenes are densely populated with small objects and intricate details within a constrained spatial scale, researchers need to address issues arising from the presence of numerous small objects and details within the scene, predict the scale of the scene (Torralba and Oliva, 2002), and also contend with a smaller dynamic range of lighting and more complex and variable lighting conditions (Miled et al., 2009), such as the effects of natural light coming through windows interacting with indoor artificial light sources.

1.3.1 Omnidiretional Depth Estimation

Methods based on deep learning are predominantly focused on general images (Ming et al., 2021; Mertan et al., 2022). The emergence of efficient spherical cameras and omnidirectional cameras has made the production of 360° content much easier. The 360° content has been widely adopted in fields such as entertainment, and robotics applications, as well as in marketing production, events, and news reporting (Zioulis et al., 2018)

One barrier to depth estimation with a general perspective camera is that the limited field-of-view (FoV) provides only a partial observation of the scene. Observation of the whole surrounding 3D environment requires multiple calibrated and synchronised

sensors. Omnidirectional cameras provide a good solution, as they capture the full surrounding scenes in one image (Kim and Hilton, 2013). Due to the distortion in omnidirectional images, the process of these images is different from that of general perspective images, and there have been several end-to-end models on omnidirectional single image for depth estimation (Zioulis et al., 2018; Wang et al., 2020b). These encoder-decoder models require large labelled datasets containing different scenes for model learning.

1.4 Challenges

1.4.1 Insufficient Real-world Data

The majority of models (Bhat et al., 2021; Alhashim and Wonka, 2018) were developed for general perspective images and the trained models cannot predict depth maps for omnidirectional images. This is because omnidirectional images contain more distortion than general perspective images (shown in Figure 2.2), and this distortion prevents the trained model for general perspective images from correctly estimating omnidirectional depth maps. As for the supervised learning models (Zioulis et al., 2018) for omnidirectional images, they perform well on labelled datasets but poorly on the dataset from another domain because of insufficient varieties of scenes for training (Wu et al., 2021). Currently available real-world datasets, such as Stanford2D3D (Armeni et al., 2017) and Matterport3D (Chang et al., 2017), cover limited scenes, such as office rooms or houses. Generating new data sets and measuring the depth of different scenes can be a solution, but it is difficult to collect a large depth-labelled dataset because a synchronised RGB-D sensor for omnidirectional capture is not generally available. Currently, published omnidirectional depth datasets contain limited types of scenes. Even the largest depth datasets, such as 3D60 (Zioulis et al., 2018) and Pano3D (Albanis et al., 2021), contain similar depth distribution and limited real-world scene types. The real-world scenes with different room-scale or different objects will perform poorly even with a model learned from these large datasets because of the difference between training datasets and real-world testing datasets (Wu et al., 2021). Learning information from scenes that are similar to target scenes and inferring depth maps can be an efficient solution. With this idea, we proposed an architecture based on an encoder-decoder model with domain adaptation, which only requires limited real-world depth maps for training and predicting the depth maps for other real-world scenes.

However, sometimes even limited ground-truth depth maps are difficult to get. Computer Graphics (CG) models can solve this problem as they can easily generate a huge amount of rendered images with corresponding depth from 3D models at a low cost, and users have full control of the synthetic datasets, such as adding objects and changing the scene light (Ren and Lee, 2018). Therefore, it is possible to use CG scenes for

training, and domain adaptation can help map the two different domains to a similar feature space (Ganin and Lempitsky, 2015). Ren and Lee (2018) stated that domain adaptation had been used for a general classification problem between photographic and synthetic imagery, and they used general perspective synthetic images to predict depth maps by using physical property maps, such as depth maps and surface normal maps. Inspired by this work, it is hypothesised that domain adaptation can be utilised for depth estimation of unlabelled real-world scenes by learning from synthetic images.

SunCG dataset from Zioulis et al. (2018) contains 12,863 scenes and various types of indoor scenes. The CG scenes are different from the real-world scenes, though some of them are similar to the real-world scenes. The challenge is that if CG datasets are used directly for training and for predicting the depth of real-world scenes, the models do not perform well due to the gap between CG and real-world scenes. The proposed reverse-gradient warming-up threshold discriminator (RWTD) solves this problem by its components. First, the reverse-gradient layer (Ganin and Lempitsky, 2015) enables similar features of source and target domains to be extracted. Second, the idea of focal loss (Lin et al., 2017) enables the model to focus on learning information for the target domain by focusing on images in the source domain that are similar to that in the target domain and ignoring the more differentiated ones. Third, the constrained increasing domain label losses prevent training loss from becoming too large to crash the training. Besides RWTD, the components of the updated architecture, such as the EfficientNet (Tan and Le, 2019) backbone and transformer encoder, also help improve the performance. EfficientNet integrates width, depth and resolution into a comprehensive task, while vision transformer (Dosovitskiy et al., 2020) can apply the attention to the visual input and learn the global information. Based on these ideas, the new single-view depth estimation architecture is able to better predict real-world scene depth by learning information only from CG datasets without using any real-world depth maps for training.

1.4.2 Unclear Insight

1.4.2.1 Gravity

Gravity as a physical constraint plays an important role on indoor scenes (Sun et al., 2021; Pintore et al., 2021). Due to gravity, the depth distribution of the object will show a certain pattern. Specifically, for objects placed on the ground, the depth from bottom to top is usually from near to far. This is because nearby objects block distant objects, creating a pattern of depth change. Therefore, the depth change in the vertical direction may be regular, that is, it may exhibit some predictability in the direction of gravity. In contrast, the horizontal depth does not have such a rule but shows different depths

according to the indoor scene and objects. Based on these motivations, aligning omnidirectional image acquisition with the gravity direction may benefit the models to learn information from these images.

Despite noticing these things, it is still not sure exactly how much gravity affects depth estimates. Figuring out to what extent gravity alignment impacts indoor depth estimation is still a puzzle that needs examination and analysis.

1.4.2.2 Depth Estimation Contribution Factors

Indoor single-image depth estimation using deep learning encounters a challenge related to the adopted deep neural network structure, characterised by a black-box nature. In other words, the internal workings of the network are opaque and resist easy explication. This ambiguity inhibits researchers from obtaining a detailed comprehension of the processes involved in depth estimation, specifically, the extraction and utilisation of visual cues from images and to what extent these factors contribute to depth estimation. Consequently, a comprehensive understanding of the operational mechanisms of depth estimation models is lacking.

1.5 Contributions

In this thesis, there are four main contributions. The first main contribution is to propose an architecture to estimate the depth field of a single omnidirectional scene image based on generative adversarial network (GAN) and domain adaptation, which is used to obfuscate domain labels in the training process so that different domains can be mapped to similar feature distribution, resulting in the domain-invariant features. The proposed architecture takes the labelled and unlabelled data as the source and target domains, respectively. The goal of the architecture is to predict the depth maps for target domain scenes. This architecture provides a good solution when the limited labelled dataset is available for the source domain data. It was evaluated on existing datasets by limiting the number of ground-truth depth maps to simulate the situation that has limited labels. The result showed that the proposed architecture outperforms a traditional encoder-decoder model by over 10% points in first threshold depth accuracy when the labelled set is very limited.

For the second main contribution, considering that it is sometimes difficult to obtain even a small amount of real-world depth maps that are similar to the target domain scenes, the reverse warming-up threshold discriminator (RWTD) was proposed. It is part of an architecture that has a similar structure to previous architecture but with different components, containing a U-Net encoder-decoder model, transformer encoder, and RWTD. This proposed RWTD contains a reverse-gradient layer, warming-up

threshold and focal weights. Reverse-gradient layer makes the gradient descent direction reverse to make the model unable to recognise the images coming from which domain so that they are mapped into the common feature space. Warming-up threshold prevents the domain label losses from increasing too fast and dominating the training loss function. Focal weights will let the RWTD focus on learning information from scenes that are similar to that in the target domain and ignore the different ones. These make the architecture train with CG images without any real ground-truth depth maps. These are explained in detail in Chapter 4. This proposed architecture solves the problem of insufficient ground-truth depth labels in the target domain and infers the depth of the target domain images by learning information from the source domain. Its performance was evaluated by applying models trained on CG scenes to a real-world public dataset and self-recorded real-world scenes, and the results demonstrate notable stability and exceptional depth accuracy.

For the third contribution, taking into account the important role that gravity plays in artificial scenes, the role of gravity in depth estimation was analysed. On this basis, we propose a model that takes into account gravity alignment. In Chapter 5, the images are divided into vertical and horizontal directions as input to study the difference between gravity alignment directions and non-gravity directions. The results show that the alignment of gravity direction can provide more information for the model to obtain better performance. Based on this research, a model based on the direction of gravity was proposed. It includes an encoder-decoder for extracting image features and a slice-based transformer for dividing extracted features according to gravity direction and predicting the final depth map with attention. These are explained in detail in Chapter 5. This study demonstrates the important role of gravity in depth estimation and indicates that gravity should be considered as a significant physical constraint in future studies. It provides a boost to model performance and interoperability, since many existing models are based on data-driven, and the physical factors that directly affect the model performance are not deeply analysed.

As for the fourth contribution, considering that the existing indoor single-image depth estimation methods are based on data-driven black-box models, a study is proposed based on the split analysis of specific factors affecting depth estimation. In Chapter 6, the relative contributions of the known cues of depth in a single-image depth estimation setting using an indoor scene. This work uses feature extraction techniques to isolate individual features of shape, texture, colour, and saturation to predict depth. The study found that the shape of objects extracted by edge detection contributed more than other objects in the considered indoor setting, while other features also contributed to varying degrees. These insights will help depth estimation models, thereby improving their accuracy and robustness. This decomposition can be used to transform the study and interpretation of powerful models (such as deep neural networks) working

in scene understanding, rather than simply treating estimated performance as black-box function approximators.

In summary, the main contributions are listed as the following:

Contribution A

- A good solution for the problem of single omnidirectional image depth estimation when only a limited labelled set is available for the source scenes.
- A domain adaptation-based architecture for single-view omnidirectional depth estimation.
- A published paper of this work with open-source code (shown in **Declaration of Authorship**).

Contribution B

- A good solution for the problem of single omnidirectional image depth estimation without any real-world ground truth labels.
- A RWTD discriminator that contributes a stable performance for domain adaptation of single-view omnidirectional depth estimation.
- Test and analysis for our own dataset captured in various indoor scenes.
- A published paper of this work with open-source code.

Contribution C

- Analysis of the importance of gravity factor in depth estimation.
- An architecture is introduced for deep dense depth estimation from a single indoor omnidirectional image utilising a slice-based transformer.
- A published paper and a submitted paper of this work with open-source code.

Contribution D

- Development of single-feature isolation techniques.
- Assessment and identification of the varying contributions of colour, saturation, local texture, and shape in depth estimation.
- Discovery of insights that can enhance depth estimation models, improving accuracy and robustness, while offering a novel perspective on deep neural networks in scene analysis.
- Submission of a paper on this work with open-source code available.

1.6 Structure of Thesis

This study is dedicated to advancing the field of deep learning-based depth estimation, with a specific emphasis on its applicability to real-world indoor scenes. The primary objective is to improve the performance of depth estimation in real-world applications and to explore and analyse the factors that contribute to depth estimation.

Chapter 1 introduces the theme and research background of this paper. Chapter 2 comprehensively reviews relevant literature, establishing a theoretical foundation for the following research. In Chapter 3, an innovative approach is introduced for realistic-scene depth estimation utilising a limited real-world dataset. Progressing to Chapter 4, the research addresses the challenge of the dependence on realistic data for real-world depth estimation, and training with synthetic data only. In Chapter 5, considering the physical constraints, the influence of gravity alignment is specifically analysed. Consequently, a model is proposed that considers the alignment of gravity alignment. Chapter 6 delves into the analysis of various factors influencing the depth estimation performance in single images and evaluates their respective contributions. Chapter 7 concludes with a discussion of the broader implications of these findings and potential future works to provide the direction for the subsequent research investigative endeavours. Finally, the Appendix chapter shows supplement materials to the experiment.

Chapter 2

Related Work

2.1 Depth Estimation

Depth estimation is a pixel-level method to measure the distance from the object surface and capturing devices (Steger et al., 2018). Most classic depth estimation methods can only be applied to the constraint scenes, depending on depth cues, such as shadow and vanishing points (Ming et al., 2021).

2.1.1 Stereo Approach

Stereo vision is an interesting topic for humans. Since the beginning of the study of visual perception, humans have realised that we do depth perception based on the disparity between the left and right eyes. It is also an important research direction in computer vision (Barnard and Fischler, 1982; Brown et al., 2003; Seitz et al., 2006).

Stereo matching is a common method of depth estimation that measures depth information for objects in a scene based on two or more images taken from different viewing perspectives. That is, the disparity map of the images is obtained through stereo matching, and the distance between the object and the observer is then calculated.

2.1.1.1 Pixel Matching

Early stereo matching was achieved by sparse correspondence (Bolles, 1993; Ohta and Kanade, 1985; Hsieh et al., 1992). The algorithm first relied on specific methods such as a point of interest detection algorithm or edge detector to identify possible corresponding feature positions in the image. They then matched these features by looking for areas in another image that matched them. This process is usually done by comparing the similarity of a small area in the image, such as corners, edges, or other significant

visual features. They can provide important information about the content of the image. Thus, the matching process is based on these selected, relatively small number of points, rather than each pixel in the image, resulting in a sparse correspondence.

Although sparse correspondences are computationally more efficient, they may not be as accurate as methods based on full-pixel (dense) correspondences.

Dense stereo approach algorithms typically perform the following four steps, or some of them (Scharstein and Szeliski, 2002): matching cost computation, cost aggregation, disparity computation and optimisation, and disparity refinement. In dense matching algorithms, similarity measures play a crucial role in assessing the likelihood of matching by comparing pixel values. This includes match loss functions at the pixel level, such as sums of squared intensity differences and absolute intensity differences, as well as approaches employing more robust techniques, such as truncated quadratics and contaminated Gaussians (Szeliski, 2010).

2.1.1.2 Deep-learning-based Methods

With the advancement of deep learning, models based on deep learning are also being utilised for stereo vision due to their strong capability to extract features and broad generalisation ability (Poggi et al., 2021). Deep-learning-based stereo approaches can be categorised into three types (Poggi et al., 2021):

Stereo Pipeline. Zbontar et al. (2016) describes a method for extracting depth information from corresponding image pairs by training convolutional neural networks to compare image blocks. Its contributions focus on matching cost computation. Specifically, using supervised learning methods, a binary classification dataset containing similar and dissimilar block pairs is constructed. By comparing pairs of small image blocks from left and right images, the neural network learns to determine whether these blocks match. Moreover, the publication of large datasets, such as Freiburg SceneFlow (Mayer et al., 2016), has enabled end-to-end training of stereoscopic networks and led to an increase in new methods.

End-to-end 2D Architecture. Knobelreiter et al. (2017) proposed a model that combined the strengths of convolutional neural networks (CNNs) and conditional random fields (CRFs), taking advantage of both in a unified approach. Multi-task learning is also an effective method for 2D architectures. Yang et al. (2018) proposes a method for binocular stereoscopic images, combined with a parallax estimation method for semantic cues, which improves prediction performance by blending semantic cues. Similarly, Jiang et al. (2019) combines four closely related tasks, such as semantic segmentation

and stereo parallax estimation, based on the motivation that sharing features can make the network more compact and promote better feature representation.

End-to-end 3D Architecture. 3D architecture is able to simulate and understand the 3D structure of a scene more accurately than 2D architecture, although it requires more memory and computing resources. Kendall et al. (2017) proposes an end-to-end deep learning method to estimate the parallax of each pixel from a pair of corresponding images. The architecture explicitly takes geometric information into account by forming cost volume, using 3D convolution to learn how to merge context from data, while the cost column is used to represent the cost or similarity of the potential disparity of each pixel calculated from a pair of stereoscopic images. Specifically, it is a three-dimensional data structure in which two dimensions correspond to the width and height of the image, and the third dimension corresponds to the possible value of parallax. Chang and Chen (2018) proposes a pyramid stereoscopic matching network, which uses global context information to improve the accuracy of stereoscopic image depth estimation through two modules of spatial pyramid pooling and a 3D convolutional neural network.

2.1.1.3 Multi-view Stereo

Stereo depth estimation methods calculate the disparity map between two images for the same scene and leverage stereo matching and triangulation for estimating depth maps (Zbontar et al., 2016). These methods require at least two fixed cameras to capture images (Zhang, 2000), which is expensive and inconvenient. There were significant works for depth estimation from stereo approaches (Rajagopalan et al., 2004; Ha et al., 2016). However, it is difficult to get enough features from images to match when the scene contains less texture (Liu et al., 2019), and they require more data and resources when compared with monocular depth estimation (Bhoi, 2019).

However, Stereo depth estimation relies on finding identical feature points across two or more views. Several factors limit its precision, including the calibration of cameras, the resolution of images, and the geometric constraints of the imaging system. Should these factors be improperly managed, it may result in inaccurate depth estimation. In complex scenarios, such as those involving occlusions, reflections, and transparent objects, stereo-matching becomes significantly more challenging, or even fails, due to the absence of matching points. Moreover, the disparity in the camera views may be minimal for distant objects, leading to an increase in the uncertainty of depth estimation.

2.1.2 Monocular Video Depth Estimation

Video-based monocular depth estimation is defined as using a single lens to obtain video sequences without requiring additional professional and complicated equipment to measure the scene depth (Ming et al., 2021). Video-based monocular depth estimation usually works with simultaneous localisation and mapping (SLAM) and structure from motion (SfM). SLAM is mainly used to solve the problems of robot localisation and map construction when moving in an unknown environment (Szeliski, 2010). Depth estimation contributes as an essential part of SLAM. Visual SLAM mainly adopts a depth camera and visual SLAM scheme based on monocular, binocular and fish-eye cameras. By inputting a series of frames taken in the same scene, SfM outputs the camera pose corresponding to each frame and 3D point cloud in the scene (Szeliski, 2010). With these methods, in a video, multiple frames are used to estimate camera pose changes, and then the distance of the object is calculated by accumulative pose changes.

Monocular video depth estimation does not face the calibration issues present in stereo approaches and requires only one camera, as opposed to stereo vision which necessitates at least two or more cameras. This affords monocular methods advantages in terms of hardware costs and ease of use. However, video depth estimation necessitates the storage of information from successive frames, leading to high memory requirements and computational costs.

2.1.3 Depth Estimation with Deep Learning

Deep learning methods bring significant advantages over traditional methods in depth estimation Bhoi (2019); Ming et al. (2021). It can automatically learn complex feature representations from data without the need for manual feature design, especially for processing image data. In addition, deep learning models such as convolutional neural networks achieve end-to-end learning from input images to deep information, simplifying the processing process and reducing errors. Moreover, the more general depth representations learned by deep learning models through large-scale dataset training significantly improve the generalisation of models.

2.1.3.1 Deep Learning Models

Convolution Neural Network Since AlexNet (Krizhevsky et al., 2012) came out, complex convolutional neural networks supported by GPU computing clusters have been widely used. General convolutional neural networks consist of input layers, hidden layers, and output layers. In computer vision, an input layer is usually taking two-dimensional or three-dimensional features as inputs, such as images (Krizhevsky et al.,

2012) and point clouds (Yan et al., 2018). The hidden layer usually includes the convolution layers, pooling layers, and fully connected layers. Among them, the function of the convolution layer is to extract the features of the input data. It contains multiple convolution cores, and each element of the convolution kernel corresponds to a weight coefficient and a deviation quantity. The main function of a pooling layer is to reduce the spatial dimension of the feature map, therefore reducing the amount of computation, and achieving the location invariance of the feature. The fully connected layer plays the role of feature integration, nonlinear transformation, and classification in the neural network. With the CNN-based backbones, the encoder-decoder of our proposed model helps extract features from RGB image inputs.

Recurrent Neural Network Recurrent neural networks (RNNs) are commonly used to process sequential data. Its basic structure consists of a cyclic unit with the input data of the current time step and the hidden state of the previous time step (Medsker and Jain, 2001). There are many variants of the RNN architecture. Common variants are bidirectional recurrent neural networks (BiRNNs) (Schuster and Paliwal, 1997), long short-term memory networks (LSTM) (Yu et al., 2019), and gated recurrent units (GRUs) (Cho et al., 2014). BiRNN extends the traditional RNNs by consisting of two RNNs that handle the forward and reverse directions of the data. It captures contextual information in a sequence and is often used for tasks that require understanding the entire sequence to make decisions, such as text translation. LSTM can solve the long-term dependence problem of traditional RNNs in processing long sequence data, and it enables the network to learn what information needs to be retained for a long time. The GRU is a simplified structure of the LSTM, which makes the model structure simpler with fewer parameters by merging the forgotten and input gates into one update gate and merging the unit and hidden states. Leveraging the characteristics of RNNs, we employed LSTM to examine the contributions of slices in the direction of gravity and those perpendicular to it towards depth estimation, with details presented in Chapter 5.

Generative Adversarial Network The generative adversarial network (GAN) architecture works by training two neural networks to compete against each other to generate similar data from a given training dataset (Goodfellow et al., 2014). Its main structure consists of a generator and a discriminator. The two structures compete with each other during training, eventually allowing the generator to learn to produce data that cannot be distinguished by the discriminator. For example, GANs can learn data distribution and generate new images that look similar to real handwritten digital images (Goodfellow et al., 2014). These characteristics of GAN make unsupervised domian adaptation available, as shown in Sec. 2.2.

Transformer The Transformer architecture has become a pivotal framework in the field of deep learning. It is implemented through a self-attention mechanism (Vaswani et al., 2017). The input sequence is represented in context by multiple layers of self-attention and feedforward neural network encoders. The decoder then generates the output sequence using the same structure, gradually generating the target sequence through self-attention and feedforward operations on the representation of the encoder. Specifically, the model input is first embedded, and then positional encoding is applied according to Equation (2.1):

$$\begin{aligned} PE_{(pos,2i)} &= \sin\left(pos/10000^{2i/d_{\text{model}}}\right) \\ PE_{(pos,2i+1)} &= \cos\left(pos/10000^{2i/d_{\text{model}}}\right). \end{aligned} \tag{2.1}$$

Subsequently, attention, defined as

$$\text{Attention}(Q,K,V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V, \tag{2.2}$$

is applied, that is, multiplied by three different weight matrices to get Q, K, and V (Equation (2.2)). By calculating the scaled dot-product attention between Q and K and applying the resulting attention score to the value matrix V, the output of self-attention

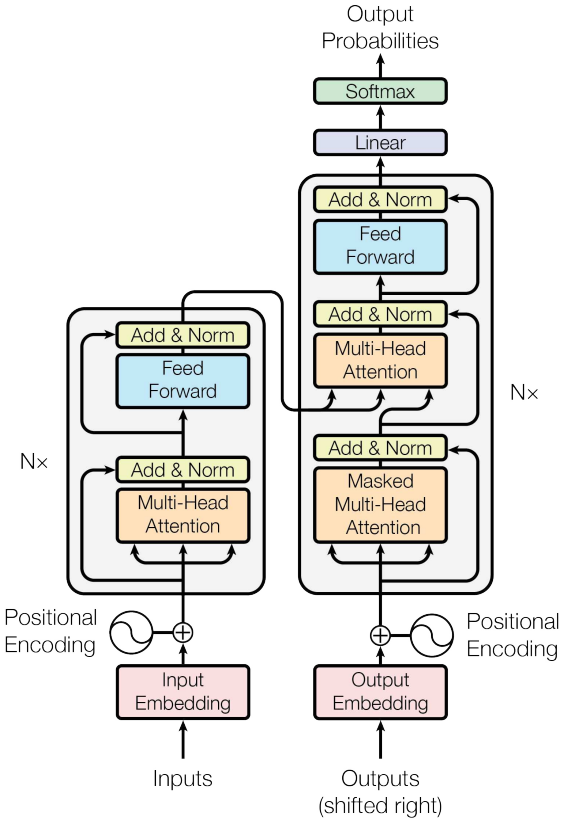


FIGURE 2.1: Transformer Architecture from Vaswani et al. (2017)

is finally obtained. This process is performed multiple times in both the encoder and decoder, which are the core architecture of the transformer. The whole process of encoding and decoding is end-to-end. Compared to RNNS, a transformer can process information in a sequence in parallel (process shown in Figure 2.1).

Subsequently, the transformer is applied to an image classification model, which is called the vision transformer (ViT) (Dosovitskiy et al., 2020). ViT divides the input image into multiple 16×16 patches and then projects each patch into a fixed-length vector into the transformer. The operation of the subsequent encoder is almost the same as in the original transformer.

With similar architecture on ViT, Bhat et al. (2021) introduces the mViT model, employing the transformer encoder for depth estimation predicated on ordinal regression. This approach enhances the model's performance. Inspired by it, we have contemplated such a mechanism within the structure we propose, as shown in Chapter 4.

2.1.3.2 Single-view Depth Estimation

Bhoi (2019) and Mertan et al. (2022) defined the monocular depth estimation as single-view depth estimation. Compared with multi-view and video-based monocular depth estimation, single-view monocular depth estimation takes less computing cost and data because it takes a single frame as input. The common deep learning models for monocular depth estimation are based on convolutional neural networks, recursive neural networks and generative adversarial networks (Ming et al., 2021) as mentioned in Sec. 2.1.3.1.

Single-view depth estimation is often seen as a regression task from an RGB image to a depth map (Fu et al., 2018). Eigen et al. (2014) proposed an end-to-end model concatenating AlexNet-based coarse and fine networks for depth estimation of general perspective images, and the output of the coarse network is concatenated as part of the input of the fine network. This work is the first to use CNN for single-view depth estimation. In order to get higher performance, Alhashim and Wonka (2018) proposed a U-Net end-to-end model with a deeper encoder with DenseNet-169 (Huang et al., 2017) and a shallow decoder to estimate depth maps with RGB images as input. Simultaneously, the encoder is initialised using a network pre-trained on ImageNet (Deng et al., 2009). This approach facilitates the easier transfer of learned features in other fields to depth estimation.

S2DNet (Hambarde and Murala, 2020) estimates a coarse depth map through the coarse depth network (S2DCNet) and then combines the estimated coarse depth map with the input image to further estimate a refined depth map through the fine depth network (S2DFNet). Moreover, S2DFNet incorporates the attention mechanism, where the attention block takes features from both the encoder and the decoder in S2DFNet as input.

(A) general perspective Sample Image (Silberman et al., 2012)

(B) Omnidirectional Sample Image (Zioulis et al., 2018)

FIGURE 2.2: Comparison of Different Types of Images

By assigning weights to each feature, it allows the network to focus on those features crucial for depth estimation, thereby achieving the depth map.

Although many models have good performance with general perspective images (Fu et al., 2018; Alhashim and Wonka, 2018; Hambarde and Murala, 2020; Abuowaida and Chan, 2020), the small field-of-view (FoV) of general perspective images includes limited content (comparison shown in Figure 2.2). In practical applications, only the partial surface depth of the scene can be estimated. This means that getting a complete scene depth map requires multiple estimates of the depth of a scene.

2.1.3.3 Extension to Omnidirectional Images

Different from the general perspective image, an omnidirectional image can get the whole surrounding information in one capture. There have been models focused on the supervised omnidirectional depth estimation task (Zioulis et al., 2018; Wang et al., 2020b), trained with public omnidirectional depth datasets, such as Matterport3D and StanFord2D3D (Zioulis et al., 2018). Similar to the depth estimation of general perspective images, there was an end-to-end neural network based on U-Net to train the omnidirectional RGB images and predict the depth maps (Zioulis et al., 2018). Wang et al. (2020b) proposed to combine two networks with an equirectangular image and its corresponding cubic projection map to avoid the distortion problem of omnidirectional images. Although these models show good performance with the given labelled datasets, they may not perform well for other real-world scenes because the model can only predict certain types of scene depth due to the limited variety of training datasets (Wu et al., 2021), and they need a large number of labelled datasets for training. This different data distribution of different scenes problem can be solved by mapping information in different fields to a feature space (Pan and Yang, 2009). With this motivation, we consider the domain adaptation method in our research.

2.2 Domain Adaptation

For datasets with different distributions, a model trained on one dataset usually does not perform well on another. Domain adaptation can be a solution to map different domain data into a common feature space (Wang and Deng, 2018). In the context of domain adaptation, it is presupposed that there exists a certain degree of distribution discrepancy between the source domain and the target domain, with the two being related yet not entirely identical. It is commonly assumed that the task remains unchanged, meaning that the class labels are shared between the source and target domains (Csurka, 2017; Farahani et al., 2021). Should the two domains be entirely unrelated, domain adaptation may not be applicable.

2.2.1 Domain Adaptation Methods

Domain adaptation based on deep learning can be divided into the following categories (Farahani et al., 2021):

2.2.1.1 Discrepancy-based

The basic idea of discrepancy-based methods is to focus on reducing the difference between the source domain and the target domain, usually by measuring and minimising some kind of statistical difference between the two domains. (Long et al., 2015) proposes deep adaptation networks (DAN) to solve the challenge of domain adaptation problems. It starts with a deep learning model that has already been trained on a large dataset and then tweaks this model through a fine-tuning process so that it can better cope with the new target task. Specifically, DAN achieves this by making the data representations of the source and target domains as similar as possible in a high-dimensional space, allowing the model to efficiently process and compare different data distributions. Since features close to the input layers are more general, while features close to the output layers are more task-specific in deep neural networks, DAN improves feature transferability by reducing differences within the network, especially at the task-specific level, between source and target domains, thereby enhancing the expressiveness of the model on new tasks.

2.2.1.2 Reconstruction-based

The basic idea of reconstruction-based domain adaptation is to learn a universal feature representation that can capture the common features between different domains (Farahani et al., 2021). The model learns this feature representation by reducing reconstruction errors, aiming to maintain information that is important to both domains

while ignoring domain-specific information that is not relevant to the task. For example, for the sentiment classification problem, Glorot et al. (2011) processes the text comment data by adding random noise to the input data and attempting to recover the original data, a process needed to minimise the differences between the input data and its corresponding reconstructed data. After learning these high-level features, a simple linear classifier, such as SVM, is then trained based on these features, whose task is to determine whether a comment is positive or negative based on the extracted features.

2.2.1.3 Adversarial-based

The basic idea of adversarial-based methods is to learn how to generate labels of data in the target domain by using the framework of a generative adversarial network, to achieve domain adaptation. Ganin and Lempitsky (2015) proposed a generative adversarial network (GAN) (Goodfellow et al., 2014) based model for depth estimation. This model learned from the digital handwriting dataset can recognise a different digital dataset with colourful handwriting images. It is a feedforward architecture comprising a deep feature extractor and a category label predictor. Specifically, its operation is facilitated by unsupervised domain adaptation through the addition of a gradient reversal layer linked to the feature extractor and a domain classifier. The gradient reversal layer guides the feature distributions of both domains to evolve towards similarity during the training process. The training procedure aligns with conventional training methods, aiming to minimise the label prediction loss for the source domain, along with the domain classification loss for both the source and target domains.

In summary, the discrepancy-based method may align the marginal distribution, and ignore the inconsistency of the conditional distribution. So its performance is limited by the ability of the feature extraction layer, which easily leads to poor performance. Furthermore, the reconstruction-based method may over-rely on the feature representation of the source domain, which affects the generalisation ability of the target domain. In contrast, although the adversarial-based approach requires expensive computational resources, it can automatically find suitable feature alignment through adversarial training, which can generate more robust, domain-invariant feature representations.

2.2.2 Domain Adaptation for Depth Estimation

Domain adaptation has been used for a general classification problem between photographic and synthetic imagery (Ren and Lee, 2018). There is a work for general images to comprehensively predict depth maps, surface normals, and edge contour maps (Ren and Lee, 2018). Considering that there are domain differences between synthetic

images and real images, it adopts unsupervised feature space domain adaptation techniques based on adversarial learning to reduce such differences. Specifically, domain adaptation is achieved by training a domain classifier to distinguish between synthetic image features and real image features, while optimising the generation network to deceive the domain classifier so that it cannot distinguish between the two feature sources. Similarly, Wu et al. (2021) proposed a domain adaptation-based model for predicting the omnidirectional depth maps with limited labelled data available with two similar domains. This work shows that domain adaptation can work for omnidirectional depth estimation, but it still requires similar real-world scenes for training. Furthermore, Wu et al. (2023) achieved the task of estimating real-world depth maps from a single panoramic image by combining encoder-decoder architecture and reverse-gradient warm-up threshold discriminator. First, the encoder-decoder structure is used to process the input panoramic RGB image and generate the depth feature vector. The adaptation between domains is then achieved by having the model confuse the feature representations of the synthetic data and the real-world data. This allows the model to effectively predict depth maps of real-world scenarios when trained using only synthetic data.

2.3 From Regression to Classification

Niu et al. (2016) demonstrated that the regression problem could be transformed into a series of ordinal multiple binary classification tasks. This work presents a CNN model with multiple output layers to solve the problem of age estimation. In particular, the ordinal regression problem is transformed into a series of binary classification subproblems. The basic idea of this transformation is to break down the entire age prediction task into multiple simple yes-no tasks, each corresponding to a specific age cut-off point that determines whether a person is older than this cut-off point. Therefore, by combining the predictions of all binary classifiers, the age of the sample can be predicted. With this method, the order information of age can be used more effectively and the complexity of the problem can be simplified

Similarly, ordinal regression can also be applied to depth estimation. Figure 2.3 illustrates the process of predicting a pixel's depth based on ordinal regression. Fu et al. (2018) proposed depth estimation by an ordinal regression network, which divides a depth range into a set of discrete intervals. Each interval represents a binary classifier that determines whether it is greater than a particular depth, and the final depth result is the cumulative truth values of these binary classifiers. This work proposes a spacing-increasing discretisation strategy to discretise depth values, convert continuous depth into a series of intervals, and treat deep network learning as an order regression problem. Different from the uniform discretisation strategy, it takes into account that the

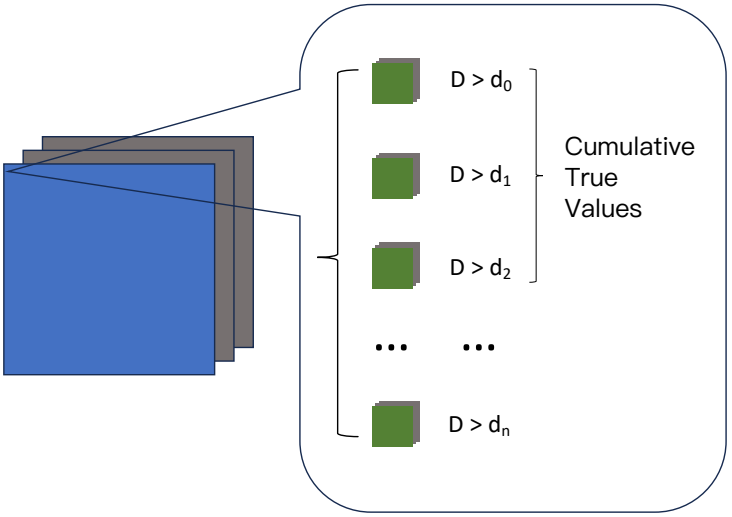


FIGURE 2.3: Process of Ordinal Regression for Depth Estimation. In the prediction of depth maps, the continuous depth values for each pixel are discretised into a sequence of binary classifications, each corresponding to a specific depth interval. The ultimate depth value of the pixel is ascertained by summing the depth interval values represented by all binary classifications deemed to be true. This method allows converting the task of depth estimation into a series of binary classification problems, with each problem determining whether the pixel's depth surpasses a certain threshold, thereby incrementally building accurate depth information.

uncertainty of depth prediction increases with the increase of true depth value, so it allows large errors when predicting larger depth values.

Bhat et al. (2021) added a transformer encoder to the model based on the work of Fu et al. (2018) to predict the adaptive depth intervals of different images rather than fix them, therefore matching the depth distribution of each input image, thus obtaining more accurate depth maps. Finally, instead of directly predicting the centre value of the most likely box as the depth value, the weighted average of the centre value of all boxes and the corresponding softmax score is used to calculate the final depth value, resulting in a smoother predicted depth map compared with that from Fu et al. (2018).

Given the advantages of converting regression to classification, we employed this approach in our suggested architecture. By predicting various depth intervals for different images, we aimed to produce precise and smooth depth maps.

2.4 Gravity Alignment

Sun et al. (2019) uses a method of splitting features into 1D representations with LSTMs (Yu et al., 2019) to estimate 3D room layouts from a single panoramic image. Following Sun et al. (2019), Sun et al. (2021) adopted a similar structure, suggesting that when the vertical axis of an image aligns with the direction of gravity, the information in the

image columns becomes more compressible and representable. By aligning the vertical direction of the image with the gravitational direction, the structural information within the image columns, such as walls and boundaries, is distributed more regularly in space. This alignment can aid models in more effectively capturing and utilising this structural information.

SliceNet realises depth estimation from a single image by segmenting the output features of the encoder-decoder and using BiLSTM for processing (Pintore et al., 2021). Specifically, it introduces a compact representation method that segments a single indoor omnidirectional image into vertical slices. These slices are generated by slicing the output features of the encoder-decoder along the direction of gravity, resulting in a sequence of feature sets. To maintain global information, the slicing operation concatenates across four different resolution levels. The concatenated features from multichannel slicing are then processed through an LSTM to learn the relationships between slices, ultimately predicting the corresponding depth map.

Nevertheless, they only proposed the concept of gravitational alignment but did not conduct a detailed analysis of the specific influence of gravity on the depth estimation results and its contribution throughout the process, and our study addresses this problem. In our work, we deeply analyse the influence and contribution of gravity alignment on performance in different settings in single image depth estimation, such as FoVs, pitch angles, etc.

2.5 Monocular Depth Cues

The studies of human vision have shown that humans can use multiple cues for monocular depth estimation Szeliski (2010); Howard (2002); Lebreton et al. (2014); Reichelt et al. (2010); Saxena et al. (2007); Landy et al. (1995); Kelly (1977). Here are some common depth estimation cues:

2.5.1 Relative Size

Relative size is one of the depth cues (Torralba and Oliva, 2002), visually, as distance increases, the size of the object projected on the retina decreases. Therefore, when two objects are similar in actual size, the smaller object in the image is usually seen as farther away. For example, in an indoor environment, people tend to think that a chair that looks smaller will be farther away from the observer.

2.5.2 Occlusion

Occlusion is a depth cue for depth estimation (Marshall et al., 1996), which describes the relationship between objects before and after, that is, near and far. Visually, when an object partially or completely blocks another object, it is usually interpreted to mean that, to the observer, it is in front of the latter. For example, when there are two cups on a table and one cup partly covers the other, then to the observer, the first cup is in front of the second cup and thus covers the second cup. This is a common depth cue used to determine the relationship between objects.

2.5.3 Linear Perspective

In the realm of visual perception, parallel lines visually converge in the distance (Mulajkar and Gohokar, 2017). For example, on a straight road, the edges of the road appear to intersect in the distance. This linear perspective allows humans to estimate the relationship between objects near and far.

2.5.4 Texture Gradient

Texture is a fine pattern or texture element on the surface of an object. In visual perception, gradient changes due to the texture on the surface of an object as the distance from which it is observed changes. This gradient change can affect human perception of the distance of objects (Gibson, 1950). Specifically, the texture of the surface of an object visually becomes more compact and less with the increase in distance. For example, bricks or grass on the ground will look denser from a distance. This is more obvious in outdoor settings.

2.5.5 Aerial Perspective

When observing objects from afar, distant items seem to have reduced contrast and saturation, and might even appear blurred, due to the scattering and absorption of light by the air in the atmosphere (Mulajkar and Gohokar, 2017). This phenomenon affects the human visual perception of distant objects. Mountains or distant buildings often exhibit this characteristic.

2.5.6 Shading

Lighting conditions have an impact on the degree of light and darkness of objects, which affects the human perception of the depth of the scene (Langer and Bülthoff,

2000). The light and shade changes on the surface of the object, the shape and position of the shadow, etc., can provide important information about the depth and location of the objects. These clues help humans understand the position of objects in three-dimensional space.

These human visual cues are not only related to the objective environment but also to psychophysics. For neural networks, is it also possible to predict depth maps based on such cues? How does the Machine Perceive Depth for Indoor Single Images? In Chapter 6, several features that can be independently separated from RGB images are selected for experiment and analysis.

Chapter 3

Depth Estimation with Limited Real-world Labels

A difficult problem in depth estimation research relates to limitations imposed by available datasets. Owing to substantial expenses, it is often difficult to capture a large number of paired RGB images and corresponding depth maps of different scenes. In this Chapter, we explore a strategy that leverages a limited quantity of real-world data for depth estimation.

The performance of a well-trained model with a large dataset can be poor when it is implemented on an unlabeled different image set. Figure 3.1 shows an example. RectNet model (Zioulis et al., 2018) was trained with Stanford2D3D dataset from Zioulis et al. (2018) and shows high performance on test data from the same dataset (95% for a_1 depth accuracy). However, it showed poor results when it was applied to a different indoor scene image capture in the studio. It is obvious that the estimated depth map includes lots of errors, especially on planar regions such as the walls, ceiling and floor, where a smoother transition of depth field is expected. The model does not get high performance because the existing training dataset covers only a few types of scenes, which leads to an overfitting problem of the model during the training process. In addition, it is much more difficult to generate ground-truth depth maps for omnidirectional images than for general perspective images because there is no omnidirectional depth sensor available. A depth sensor takes a lot of time to scan and capture a high-resolution depth map, and manual depth map generation is also hard due to its image distortion and wide FoV. Therefore, the lack of a training depth label set is a serious problem in single omnidirectional image depth estimation.

In order to overcome the poor performance with a new dataset from a different domain and the difficulty of getting a large number of labelled images from new scenes,

an architecture based on domain adaptation is proposed. By considering the unlabelled target domain RGB images, the proposed architecture outperforms the traditional encoder-decoder model, with only limited labelled images.

3.1 Method

3.1.1 Overview

The overview of the proposed architecture is illustrated in Figure 3.2. The proposed domain adaptation-based architecture not only allows the model to accurately predict the depth of the input RGB images but also cannot distinguish their domain labels, therefore mapping the domains to a common feature space. This architecture leverages the domain adaptation technique for omnidirectional depth estimation with input images from different domains, including unlabelled images. In this architecture, the

FIGURE 3.1: Depth Estimation Result for a Different Real-world Indoor Scene

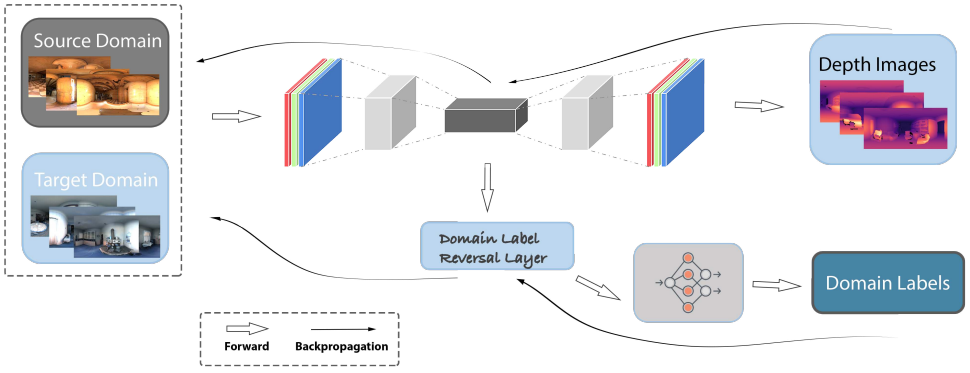


FIGURE 3.2: Overview of Proposed Architecture. It takes omnidirectional RGB images from both source and target domains as inputs and outputs corresponding depth maps. During training, the source domain has omnidirectional RGB images and corresponding depth map labels, while the target domain has only RGB images without labels. The training involves back propagation where the gradients are calculated for each layer from the output towards the input (the solid arrows are shown in the figure). In the testing phase, only the target domain's RGB images are used as input to predict the corresponding depth maps. Specifically, the U-Net network is trained to predict the depth of images of a source domain. The features learnt by the model are constrained by the parallel discriminator branch, which is trained to separate source and target domain images by propagating a reverse gradient through the encoder weights. Therefore, the domains are mapped to the common feature space.

input images are omnidirectional RGB images with given domain labels 1 (source) and 0 (target).

3.1.2 Proposed Architecture

The architecture can be divided into three parts, the encoder, decoder and domain classifier. The end-to-end model was improved from Alhashim and Wonka (2018) by using a ResNet50 backbone, as the experimental results show a better performance with it. This is because the new backbone is not easy to get overfitting, and the fit is achieved using fewer resources and lower complexity.

For the training process, the encoder transforms the RGB images into embedded features, while the decoder predicts the depth maps based on these embedded features. The encoder-decoder is called a depth predictor, and the training process tries to make the loss of the predictor as small as possible. The green part in Figure 3.3 shows the reverse-gradient layer. The domain classifier predicts domain labels based on the reverse features outputted from this reverse-gradient layer and makes gradient descent towards the direction of loss increase. It is used to obfuscate domain labels in the training process so that different domains can be mapped to the common feature space with similar feature distribution, resulting in domain-invariant features.

Therefore, there are two directions of gradient descent during the training process, the loss of the encoder-decoder model is expected to be as low as possible, while the loss of the domain classifier is expected to be as high as possible. By adding a domain classifier to the end-to-end model, it makes the model unable to identify which domain the images come from (Ganin and Lempitsky, 2015). By loading the model with depth-labelled images as the source domain and unlabelled images in the target domain, the model can predict the depth maps of the target domain images.

3.1.3 Loss Function

The training loss function is defined as:

$$\mathcal{L}(\mathcal{G},\mathcal{O}) = \lambda\mathcal{L}_{depth}(\mathcal{G},\mathcal{O}) + \mathcal{L}_{SSIM}(\mathcal{G},\mathcal{O}) \ + \mathcal{L}_{label_s}(\mathcal{G},\mathcal{O}) + \mathcal{L}_{label_t}(\mathcal{G},\mathcal{O}). \tag{3.1}$$

That is, it is a combination of four loss functions (Equation (3.1)), including depth loss, defined as

$$\mathcal{L}_{\text{depth}}(\mathcal{G},\mathcal{O}) = \frac{1}{n}\sum_{p=1}^{n}\left|\mathcal{G}_p - \mathcal{O}_p\right|; \tag{3.2}$$

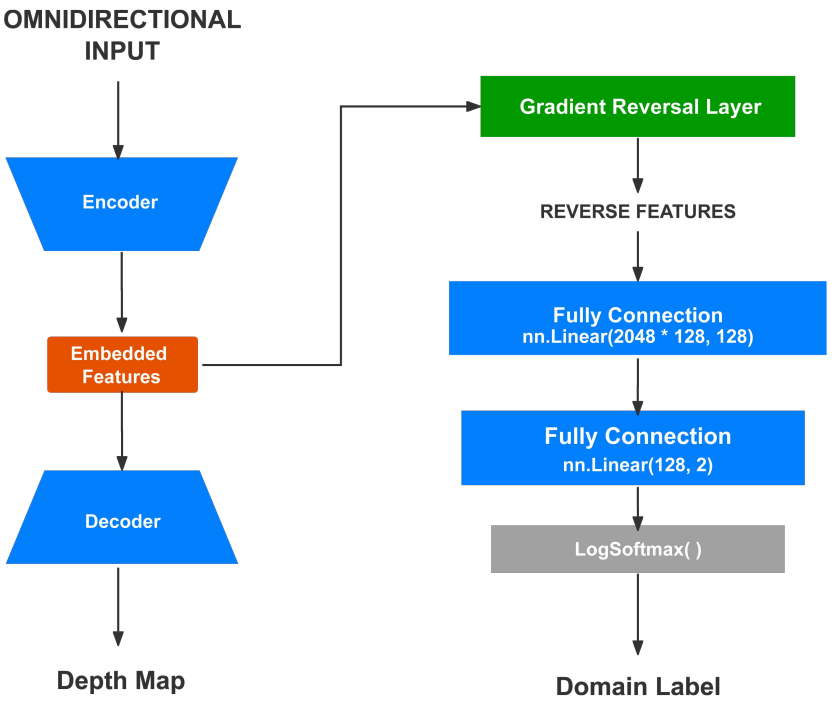


FIGURE 3.3: Structure of domain adaptation

Structural Similarity (SSIM) loss, defined as

$$\mathcal{L}_{\text{SSIM}}(\mathcal{G}, \mathcal{O}) = \frac{1}{n} \sum_{p=1}^{n} \frac{1 - \text{SSIM}(\mathcal{G}_p, \mathcal{O}_p)}{2}; \tag{3.3}$$

and two domain label losses for the source domain and target domain, respectively. $\mathcal{G}$ represents the ground truth depth map and $\mathcal{O}$ represents the depth map of the model output. SSIM loss is useful for comparing the difference between two images as it considers the difference in brightness, contrast and structural similarity (Wang et al., 2004).

λ is a weight parameter and set as 0.1 according to empirical result (Alhashim and Wonka, 2018). $\mathcal{G}$ represents ground truth depth maps, while $\mathcal{O}$ demonstrates the output depth map from the network, and p means the pixel in the image.

The source and target domain label losses, L_{label_s} and L_{label_t}, are calculated with Negative Log-Likelihood Loss (NLLLoss). Note that through the gradient reversal layer, the training process tries to increase the loss of the domain classifier to promote the feature learning of domain indiscriminability. The process is dynamically adjusted and the goal is to reach a balance, rather than unilaterally increasing field losses.

3.2 Evaluation Metrics

In order to quantify and accurately describe the performance of the model, the six metrics about accuracy and loss of models are often used as evaluation indicators as they are all correlated to the performance of models (Eigen et al., 2014; Alhashim and Wonka, 2018; Zioulis et al., 2018; Bhat et al., 2021). In this section, these six evaluation metrics will be introduced: a_1, a_2, a_3, rel, rms, and log_{10}.

3.2.1 Accuracy Metrics

For comparing the performance of the models, three accuracies were used with thresholds 1.25, 1.25^2, and 1.25^3 (a_1, a_2, a_3) (Eigen et al., 2014; Alhashim and Wonka, 2018; Zioulis et al., 2018; Bhat et al., 2021), defined as

$$\max\left(\frac{\mathcal{G}_p}{\mathcal{O}_p}, \frac{\mathcal{O}_p}{\mathcal{G}_p}\right) = \delta < \tau, \quad \tau = a_1, a_2, a_3. \tag{3.4}$$

Different thresholds correspond to the varying sensitivity of the accuracy metric to predicted depth maps and ground truth depth maps. Shown in Equation (3.4), p represents the pixels on depth maps. It shows the differences by comparing the ground truth depth maps with the output depth maps of the models. Every pixel on a predicted depth map and corresponding ground truth will be accumulated, and the percentage of all points smaller than this threshold in the number of the total pixels is defined as first-threshold accuracy, second-threshold accuracy and third-threshold accuracy. The larger the accuracy values, the better the performance (marked as ↑).

3.2.2 Error Metrics

There are three error metrics to evaluate the models: Abs Relative Difference, defined as

$$\text{rel} = \frac{1}{|\mathcal{T}|}\sum_{p=1}^{\mathcal{T}}\frac{|\mathcal{G}_p - \mathcal{O}_p|}{\mathcal{O}_p}, \tag{3.5}$$

Linear RMSE, defined as

$$\text{rms} = \sqrt{\frac{1}{|\mathcal{T}|}\sum_{p=1}^{\mathcal{T}}|\mathcal{G}_p - \mathcal{O}_p|^2}, \tag{3.6}$$

and Log10 RMSE, defined as

$$\text{log10} = \frac{1}{\mathcal{T}}\sum_{p=1}^{\mathcal{T}}\left|\log_{10}(\mathcal{G}p) - \log_{10}(\mathcal{O}_p)\right|, \tag{3.7}$$

as referenced in Eigen et al. (2014); Alhashim and Wonka (2018). They are shown as *rel*, *rms*, and log_{10}, respectively, in the result tables. $\mathcal{T}$ represents the total number of pixels in an image. *rel* is a traditional method that is mainly used to evaluate the relative size of the prediction error per pixel. Linear RMSE calculates the square root of the mean of the square of the difference between the predicted and true values, so it is more sensitive to large prediction errors, while Log10 RMSE reduces the impact of large distance errors. The smaller the loss values, the better the performance (marked as $\downarrow$).

3.3 Implementation

In this section, the proposed architecture is trained and tested at the pixel level and regarded as a depth regression problem. In order to prove that the depth prediction proposed in this research can be used for unlabelled omnidirectional images, the architecture for omnidirectional image depth estimation with a house-scene-based dataset and an office-scene-based dataset from 3D60 dataset (Zioulis et al., 2018) is implemented.

The proposed architecture and models are trained on NVIDIA RTX 3090, with 24GB of CUDA memory.

3.3.1 Data Exploration

3D60 dataset (Zioulis et al., 2018) was released with three omnidirectional image datasets, including Matterport3D, Standford 2D3D, and SUNCG. SUNCG is a computer graphic dataset, while Matterport3D and Standford 2D3D are real-world captures. Figure 3.4 shows some samples of the Matterport3D dataset, presenting house scenes, while the Stanford2D3D dataset demonstrates the scenes in office rooms dataset in Figure 3.5.

It should be noted that these sets contain outliers even though they are used as the ground truth. They were captured by RGB cameras and LiDAR sensors and then were synthesised. These sensors have limitations of scanning density and also false (or missing) depth in transparent or reflective surface areas. Due to these hardware limitations, there are some missing depth areas. These pixels are recorded as 1,000,000 meters and marked as outliers (Zioulis et al., 2018). There are also false depth regions, such as the area behind glass or windows, and it is difficult to filter them out.

Stanford2D3D dataset contains 898 images that are divided into six parts as they are taken in 6 different office buildings. Among them, Area1 with 190 images is selected as a training dataset in the experiments. For data preprocessing, the scenes that contain more than 5% of outliers were removed. After that, the source domain of Stanford2D3D Area1 contains 128 images. The Matterport3D dataset contained 1280 images. One

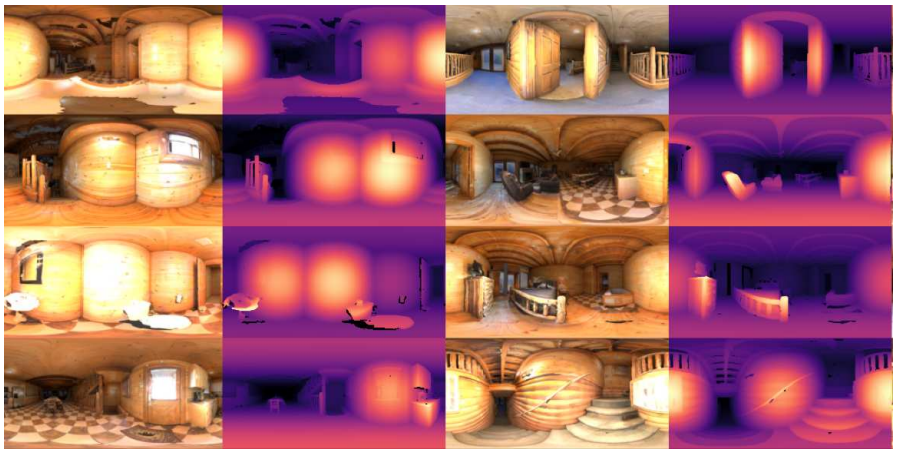


FIGURE 3.4: Samples of Matterport3D. The left is original RGB image and right is its corresponding depth map. In the depth map, the brightness represents its depth (the brighter, the closer)

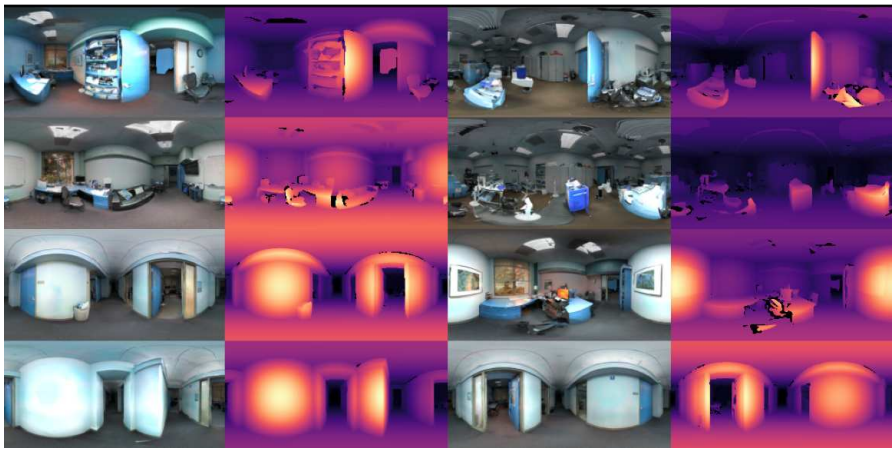


FIGURE 3.5: Samples of StanFord2D3D

area (88 images after removing scenes containing more than 5% outliers) of this house-scene dataset is chosen for the target domain, called 'Matterport3D Area2'. As the 2D projection of 3D scenes, equirectangular images contain obvious distortion. Compared with the middle part of the horizontal direction, the top and bottom parts show an exaggerated distortion.

The distribution of depth maps shows that depth in the scene is between 0.5 metres and 10 metres, and very few areas are above 10 metres. In order to compensate for the inherent problem with the loss terms (Ummenhofer et al., 2017; Huang et al., 2018), the maximum distance of depth maps is set as 10 meters and normalised all depth fields considering the reciprocal of the depth (Alhashim and Wonka, 2018). As shown in

$$\mathcal{N}_d = \mathcal{M}_d / \mathcal{D}_{max}, \tag{3.8}$$

$\mathcal{N}_d$ denotes the result after depth normalisation, while $\mathcal{M}_d$ represents the depth map, and $\mathcal{D}_{max}$ denotes the maximum depth. .

These datasets were acquired in different circumstances with different cameras but with some similar objects, such as doors and chairs.

3.3.2 Data Augmentation

In the training process, due to the convolution, the accuracy of the edge part may be affected by the padding. We assumed that the performance of the middle parts would be better than the edges and wanted to see if the edges could sometimes be moved to the middle. Based on this hypothesis, this experiment was conducted by separating the equirectangular images into several chunks and shifting the chunks to check whether image shift data augmentation helps for a small dataset. Experiment details can be checked in Appendix A.

However, this data augmentation method only improves less than 1% point of first-threshold accuracy but makes the computing cost four times. Therefore, in the later experiments, this data augmentation method will not be used because of the cost of using it, although it tempts the model to slightly improve the performance.

3.3.3 Implementation Details

In the experiments, the input image resolution was 256×512, and the batch size was 16. The learning rate was set as 0.0001, and the number of the epoch was set as 100. The Adam optimiser was adopted, with parameter $\beta 1 = 0.9$, $\beta 2 = 0.999$.

There is no crop of any part of the input images, even though they contained missing points or outliers due to correction preprocessing. There is also no crop of the output images before computing the accuracies and losses. This is because, in practice, the image contains different amounts of outliers, which affects the output of the model to some extent. In addition, the purpose of this work is not to simply improve the accuracy of the predicted depth map but to verify that the proposed semi-supervised architecture based on the domain adaptation method can outperform the traditional supervised model with limited labelled data.

3.4 Experiments

3.4.1 Baseline

In order to simulate the situation of limited labelled images in different scene types, the performance of depth estimation was evaluated according to the size of the labelled training dataset. The proposed architecture trained the end-to-end models with

TABLE 3.1: Performance of Models with Different Sizes of Dataset. All models were tested on the Matterport3D Area2 dataset. The first row shows the upper-bound performance of RestNet50 when there is no domain gap. The training dataset is the whole Matterport3D (except Area2).

Model	Training Dataset	$a_1 \uparrow$	$a_2 \uparrow$	$a_3 \uparrow$	$rel \downarrow$	$rms \downarrow$	$log_{10} \downarrow$
ResNet50 Backbone Encoder-decoder	Whole Matterport3D	0.8996	0.9774	0.9918	0.1039	0.9017	0.0442
ResNet50 Backbone Encoder-decoder	Whole Stanford2D3D Area1 (128 images)	0.6576	0.8986	0.9585	0.1918	1.8300	0.0908
	40% Stanford2D3D Area1 (51 images)	0.6494	0.8871	0.9587	0.2077	1.9669	0.0935
	20% Stanford2D3D Area1 (25 images)	0.6135	0.8394	0.9390	0.2376	2.2732	0.1033
Proposed Architecture	Whole Stanford2D3D Area1 (128 images)	0.7259	0.8994	0.9557	0.2189	1.7223	0.0839
	40% Stanford2D3D Area1 (51 images)	0.7191	0.9063	0.9546	0.1805	1.6025	0.0827
	20% Stanford2D3D Area1 (25 images)	0.7181	0.9252	0.9709	0.1871	1.5431	0.0799

Stanford2D3D Area1 as the training dataset and Matterport Area2 as the testing set. With a gradual reduction of the proportion of the training set randomly, the scenario is simulated in which a limited amount of data is used to train and predict depth maps of unlabelled RGB images.

Table 3.1 shows the upperbound performance of RestNet50 when there is no domain gap. In addition , for cross-domain task, it shows that the accuracy of estimated depth by the ResNet50 backbone encoder-decoder model decreased to 61.35% of first thresholding accuracy when only 20% of the training set (25 images) were used.

3.4.2 Domain Adaptation

For the experiments, the source domain is Stanford2D3D Area1, and the testing dataset is Matterport3D Area2. Table 3.1 shows the output of the proposed domain adaptation architecture with decreasing number of labelled training images. Overall the proposed architecture shows higher accuracy of depth estimation than the baseline method. One more important observation is that the proposed method kept a similar level of performance (72.59% to 71.81% for a_1) when the number of the labelled training set was reduced, while the performance of the baseline method decreased from 65.76% to 61.35%. Other metrics also showed similar performance even though the size of the training set had been decreased. They sometimes showed even slightly better performance with less training set (source domain). It may be because the training has been less biased to the source domain.

Obtained results of the proposed architecture with 20% data are shown in Figure 3.6. The first and second ones are close to the ground truth though they are a bit blurry. It can be observed in the third image that if a wall has some patterns, it influences the depth map and make it bumpy. The fourth one shows some depth errors around the stairs region, but even the ground truth map also has errors in the region. The fifth and sixth ones show errors in the window regions as they are transparent.

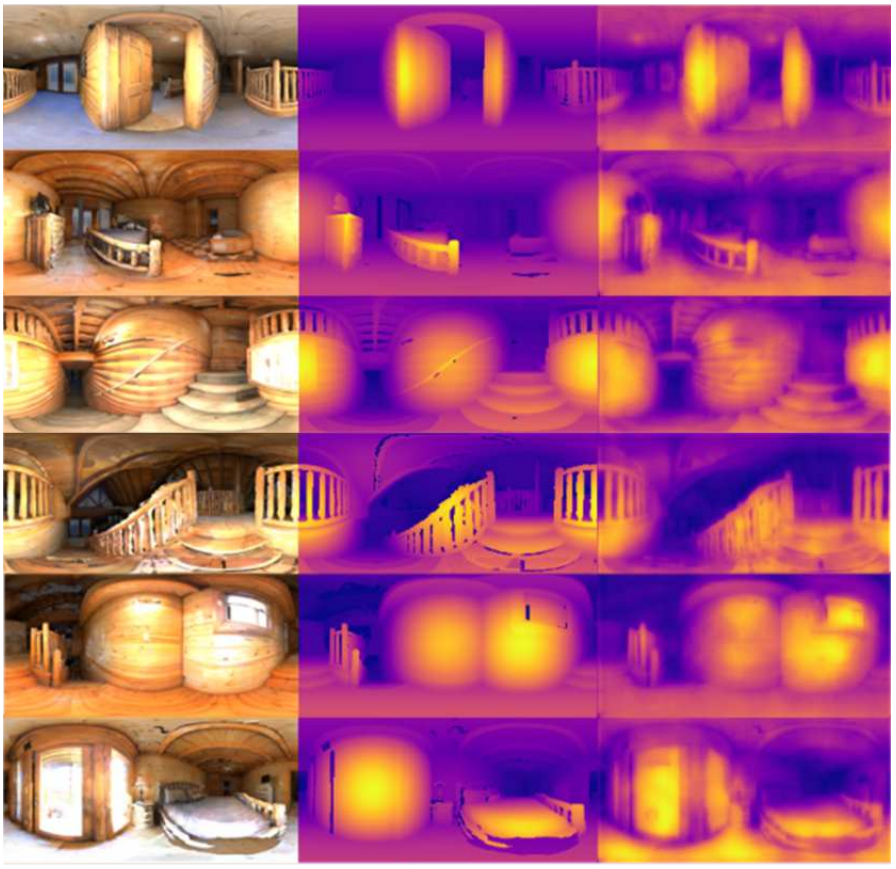


FIGURE 3.6: Depth Estimation Results with the Proposed Domain Adaptation Architecture. (Left: Original image, Middle: Ground-truth depth map, Right: Estimated depth map)

The proposed method was also tested with our own dataset captured in various indoor scenes: studio, corridors, and building reception areas. They were captured with Spheron VR[1] and Ricoh Theta S[2] omnidirectional cameras. Figure 3.7 shows the comparison of depth estimation results of the proposed domain adaptation architecture against the encoder-decoder architecture. Only subjective evaluation can be provided as their ground-truth depth maps are not available. The test scenes are different from the training set, and the proposed method predicted roughly accurate depth maps for the test images. It can be observed that the output generated by domain adaptation architecture has a smoother texture on the object with the same depth plane in the real world. The estimated depth by the proposed model with domain adaptation is closer to the real distance.

In conclusion, the results show that the performance of the proposed architecture outperforms the traditional end-to-end models when the labelled omnidirectional images are limited.

[1] https://www.spheron.com/home.html
[2] https://theta360.com/uk/about/theta/s.html

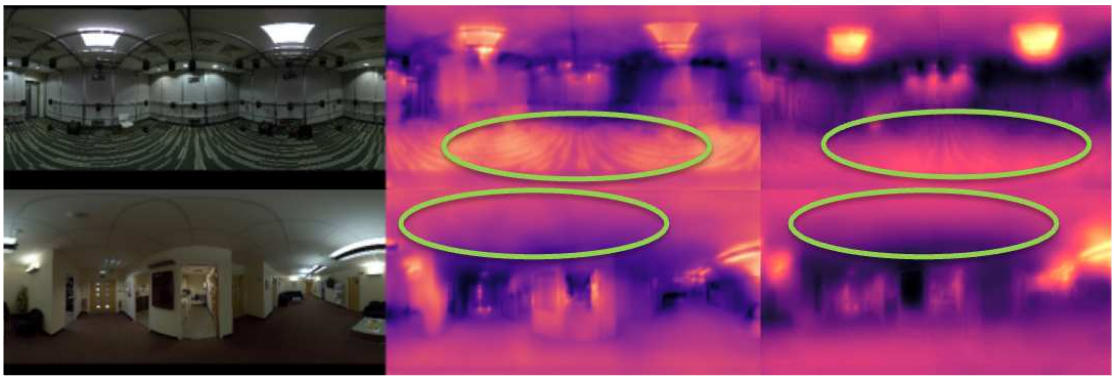


FIGURE 3.7: Depth Estimation Results on Own Dataset. (Left: Original image, Middle: Depth map by the encoder-decoder model, Right: Depth map by the proposed domain adaptation model)

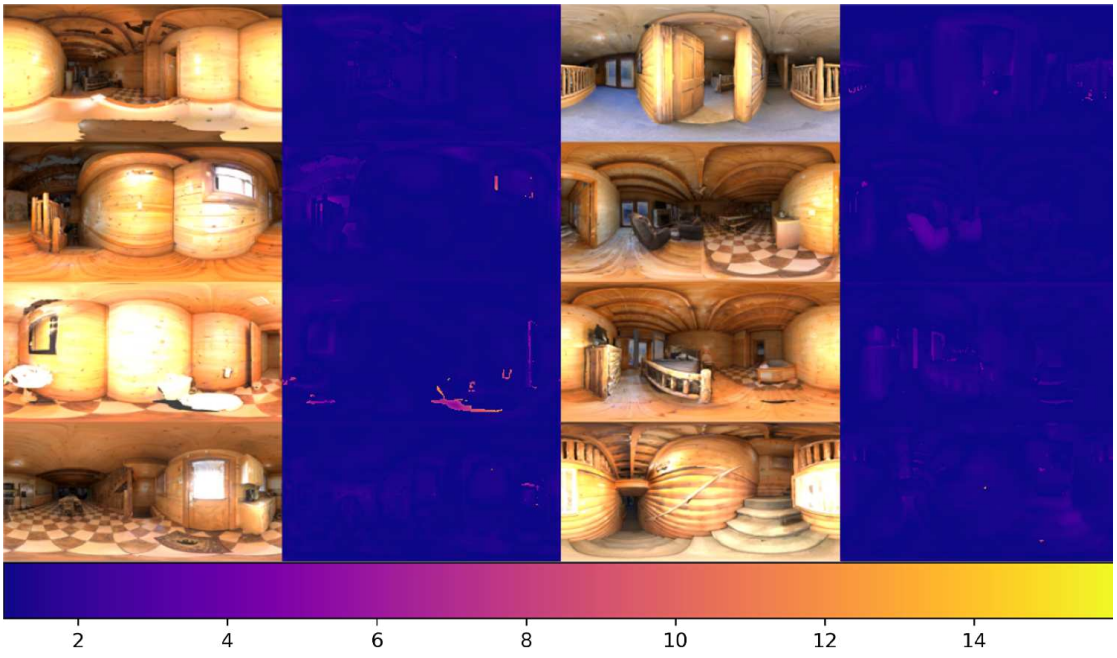
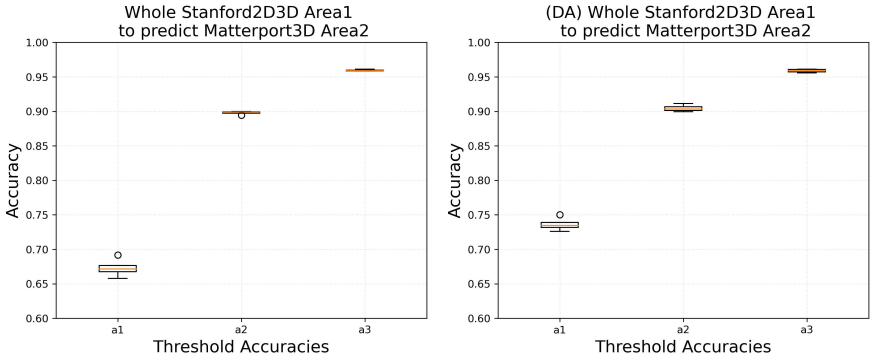


FIGURE 3.8: δ Maps of the Proposed Domain Adaptation based Architecture

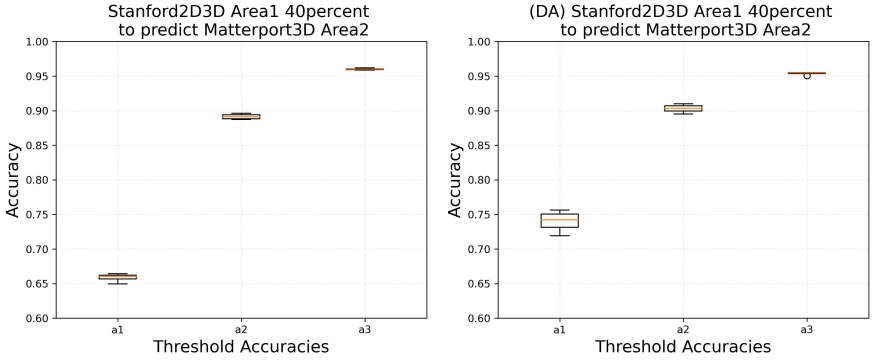
3.4.3 Error Analysis and Discussion

To further analyse the performance of the model, the δ maps of several samples are demonstrated, representing the difference of output depth maps against the ground-truth depth map in the form of a heat map. The δ map in Figure 3.8 shows errors calculated by the first thresholding accuracy evaluating formula mentioned in Section 3.2. It can be observed that the performance is generally good. Some areas have tremendous δ values because ground-truth depth maps contain outliers. As mentioned in Section 3.3.1, these outlier values are marked as 1,000,000.

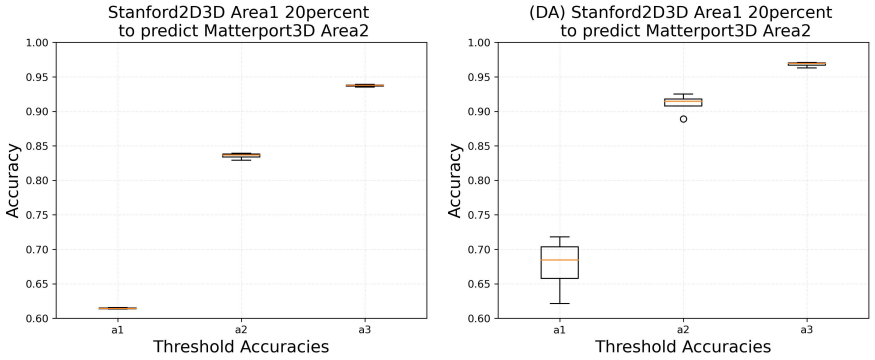
For the uncertainty of the results, Figure 3.9 demonstrates the encoder-decoder model and domain adaptation architecture's performance with different sizes of the source



(A) Whole Data



(B) 40%



(C) 20%

FIGURE 3.9: Different Threshold Accuracies of Depth Estimation under Different Dataset Sizes. Uncertainty in estimates displayed as boxplots. Each of A, B and C shows results without (left) and with(right) domain adaptation.

domain, respectively. Each box contains five values, representing the accuracies on the epoch 80, 85, 90, 95 and 100. It can be observed that although the stability of domain adaptation is not as good as the traditional end-to-end model when the dataset is small, the accuracy is significantly higher than the traditional model.

As previously mentioned, the ground-truth depth maps for training have incomplete regions due to hardware limitations. These false labels may cause the wrong prediction of the model. Figure 3.10 shows an example showing serious depth errors in the regions with large glass walls. If we consider those glasses as a solid structure, the depth map should show planar depth at the locations of the walls. Even if we ignore

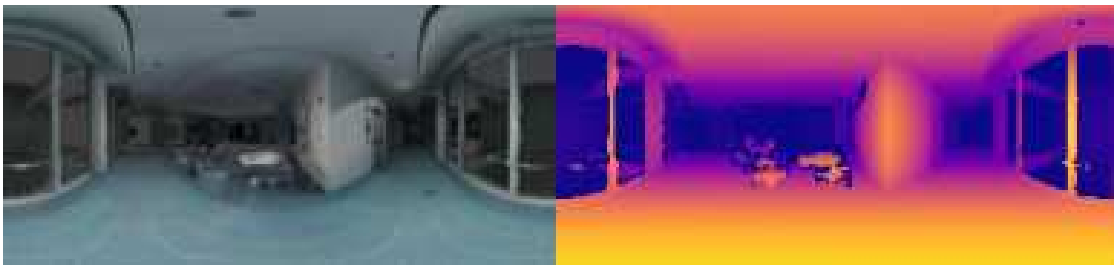


FIGURE 3.10: Example of False Ground Truth. (Left: Original image, Right: Given depth labels)

glasses, considering the limitation of the sensors, the ground truth for the regions beyond the glasses is still wrong. Most depth sensors, including LiDAR, cannot properly detect and measure transparent or reflective surfaces. This is another reason for the low accuracy of the proposed model, as those wrong depths were also considered as ground-truth for training and even for evaluation.

For practical applications which need to detect even glasses and mirrors, additional modalities, such as acoustic sensors, can be considered to overcome these problems (Kim et al., 2020). However, these issues were not considered in this work, and the research focus was to verify the application and efficiency of domain adaptation in the field of omnidirectional images. Enhancing the performance of the model itself was not the primary focus of this work.

3.5 Conclusion

Traditional deep learning-based single-image indoor depth estimation relies on supervised learning and tends to overfit a specific dataset, resulting in a lack of generalisability. To address the issue of poor model performance across different domains in depth estimation, the architecture with domain adaptation is proposed to predict scene depth for unlabelled omnidirectional image sets when the labelled training sets are limited. The experiments show that the performance of domain adaptation architecture outperforms the traditional end-to-end model for omnidirectional depth estimation in the situation of a limited number and variety of data. Furthermore, this performance shows that an end-to-end model with domain adaptation can predict the reasonably good quality of depth maps for the omnidirectional images in a different scene without labels. This result means that the work creates a potential direction for depth estimation of unlabelled omnidirectional scenes with limited labelled data. This work has been published at the ACM SIGGRAPH European Conference on Visual Media Production (CVMP).

However, for the proposed architecture, a similar labelled dataset is still necessary for training. In practice, it is usually difficult to find a suitable labelled dataset. Therefore,

in the next chapter, a new architecture is proposed which can be trained with synthetic datasets instead of real-world ground-truth sets, considering a situation in which no real-world label is available.

Chapter 4

From Simulation to Reality: Depth Estimation with Synthetic Data

As mentioned in the previous chapter, it is often a challenge to find real-world annotated datasets that are similar to a specific target dataset.

4.1 Motivation

Currently, existing omnidirectional depth datasets contain limited types of scenes. Even the largest depth datasets, such as 3D60 (Zioulis et al., 2018) and Pano3D (Albanis et al., 2021), contain similar depth distributions and limited real-world scene types. Computer Graphics (CG) models can solve this problem as they can easily generate a huge amount of rendered synthetic images with corresponding depth from 3D models at a low cost, and users have full control of the synthetic datasets, such as adding objects and changing the scene light (Ren and Lee, 2018). Moreover, synthetic datasets tend to be more abundant and cover a wider range of scenarios compared to real-world datasets. Therefore, it is possible to use synthetic scenes for training and domain adaptation can help map the two different domains to a similar feature space (Ganin and Lempitsky, 2015). Inspired by previous works (Ren and Lee, 2018; Wu et al., 2021), we hypothesised that learning only from synthetic images can help estimate depth maps for unlabelled omnidirectional real-world scenes and proposed the architecture with both better performance and stability.

Building on this hypothesis, leveraging the data from synthetic datasets to learn and predict valuable information about depth maps in real-world scenes could provide significant assistance for depth estimation tasks in real-world settings. Given these, an architecture without using any real-world labels for predicting depth maps in real-world scenes will be introduced.

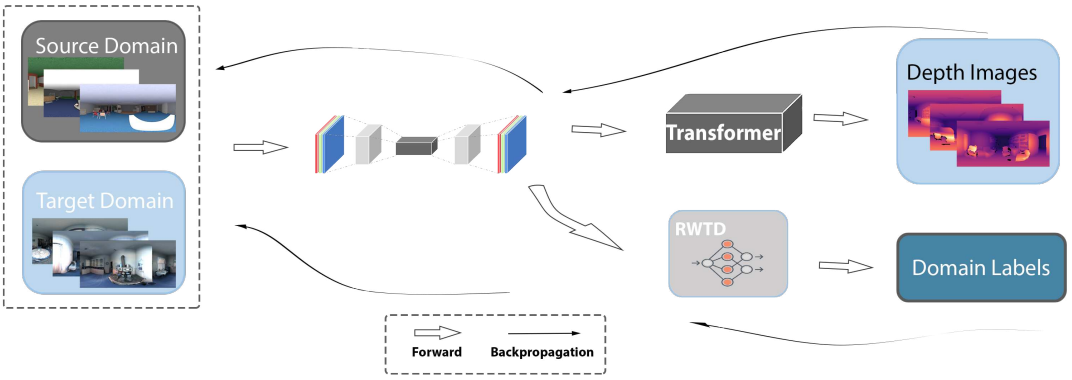


FIGURE 4.1: Overview of Proposed Architecture. It has a similar architecture but includes different modules compared to the previous architecture in Chapter 3, including an encoder-decoder model, a transformer encoder, and the proposed reverse warming-up threshold discriminator.

4.2 Method

4.2.1 Overview

Figure 4.1 illustrates the overview of the proposed architecture. In general, the structure is similar to that in the last chapter. It belongs to an unsupervised domain adaptation method that can predict the depth of unlabelled scenes, using the domain adaptation method to estimate the depth of input scenes for both source domain images with labels and unlabelled target domain images. However, the components of this architecture are different.

For this new model, the input images are omnidirectional RGB images with their domain labels (1 for the source domain and 0 for the target domain). These inputs will go into the encoder-decoder model and output the embedding features. The transformer encoder module will then take these features as input and estimate the corresponding depth maps. The features will also enter the reverse warming-up threshold discriminator (RWTD) and then predict the domain labels.

4.2.2 Proposed Architecture

The architecture consists of updated components, such as an encoder-decoder model, transformer encoder, and proposed RWTD.

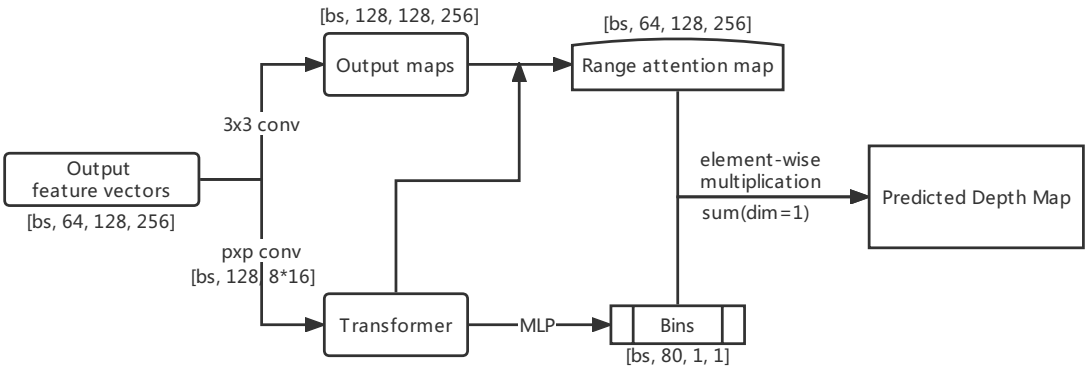


FIGURE 4.2: Process of Transformer Encoder.

4.2.2.1 Encoder-decoder Model

The encoder-decoder model is the U-Net model. For encoder, EfficientNet B5 (Tan and Le, 2019) is used as backbone because of the better performance according to the experimental results for comparing backbone of ResNet (He et al., 2016), EfficientNet, and DenseNet (Huang et al., 2017). This is because the main idea of EfficientNet is to integrate width, depth and resolution into a comprehensive task of the network (Tan and Le, 2019). For the decoder, following Alhashim and Wonka (2018), the architecture uses a shallow decoder that contains two convolution layers and four bilinear upsampling layers. The encoder-decoder model takes omnidirectional RGB images as inputs and outputs corresponding feature vectors.

4.2.2.2 Transformer Encoder

Ordinal regression is suggested to be used for monocular depth estimation task (Fu et al., 2018; Bhat et al., 2021). Regression-based architectures do not do enough global analysis of the output values because a limitation of the convolution layer is that they process global information only when the tensor reaches low spatial resolution or near the bottleneck. Therefore, Bhat et al. (2021) suggests dividing the predicted depth range into bins, whose width changes with each image, and the final depth estimate is a linear combination of these bins centres. Following this, the depth regression task is transformed into a classified task. The main body (Figure. 4.2) of the transformer encoder is a vision transformer (Dosovitskiy et al., 2020) based structural block that divides the depth range into multiple bins, and the central value of each bin shows the depth adaptively.

The encoder-decoder output goes to two branches. One is to convolute with a 3×3 kernel to get the 'output maps'. The other goes through an embedding convolution layer and generates patch embeddings as input to the transformer encoder. The transformer

encoder will then output two branches: Range attention maps and depth range with bins.

Range attention maps. The outputs from the transformer encoder pixel-wise dot the convolution result between 'output maps' and finally, produce range attention maps.

Depth range with bins. This depth range shows how the depth interval of the scene is divided into bins. The output depth range is generated by a multi-layer perceptron (MLP).

To get the combined information between local and global features, this depth range with bins is then combined with the range attention map by element-wise multiplication and the sum of pixel values according to the channel direction. With this transformer encoder, a final depth map is predicted as a combination of range attention maps and normalised bin centres, enabling the model to estimate accurate and smooth depth maps.

4.2.2.3 Reverse Gradient Warming-up Threshold Discriminator

As the main contribution of our work, the RWTD makes the architecture able to predict the depth maps without training with real-world ground truths, but only with the synthetic dataset.

The discriminator in the proposed architecture is to classify output feature vectors of the encoder-decoder model from the source domain or target domain. Similarly, there is a reverse-gradient layer (Ganin and Lempitsky, 2015). With its help, the RWTD learns not to recognise where the feature vectors are from which domain. Therefore, there are also two gradient descent directions during training. In addition, RWTD allows the discriminator to increase the weight given to similar images while ignoring the differentiated ones from the source and target domains with the increase of epoch number. In this way, compared with the previous GAN-based domain adaptation methods (Ganin and Lempitsky, 2015; Saito et al., 2019; Wu et al., 2021), it can make the predicted depth distribution similar to the ground truth of the target domain and get better results.

Based on this, the information learned in the source domain can be applied to predict depth maps of unlabelled scenes from the target domain. Moreover, the main reason that previous architecture in Chapter 3 cannot train only with synthetic images and predict depth maps for real-world scenes is that the domain label losses will keep going up and dominate the loss function and then guide the whole architecture to a wrong gradient direction. To solve that problem, RWTD has a warming-up threshold to set constraints on the loss values during the training process, and this value will be changed according to the training epoch to make the whole architecture perform well.

4.2.3 Loss Function

The loss function combines the dense depth loss, the ChamferLoss, and Domain Label Loss (DLL), which is defined as

$$\mathcal{L}_{(}\mathcal{G},\mathcal{O}) = \alpha\mathcal{L}_{dense}(\mathcal{G},\mathcal{O}) + \beta\mathcal{L}_{Chamfer}(\mathcal{G},\mathcal{O}) \ + \theta(\mathcal{L}_{label_s}(\mathcal{G},\mathcal{O}) + \mathcal{L}_{label_t}(\mathcal{G},\mathcal{O})). \quad (4.1)$$

Shown in Equation (4.1), α and β represent the factor of dense depth and ChamferLoss, respectively. θ represents domain label loss factor (DLLF), and it controls the influence of DLL. These factors balance the weight of different losses and lead to the good performance of the proposed architecture.

The dense loss function has changed with the updated architecture because scale-invariant loss can better help model training. In addition, because the regression problem has been changed to ordinal regression, using ChamferLoss can encourage bin centres to be as close to the value of ground-truth depth maps as possible.

4.2.3.1 Dense Depth Loss

Scale-invariant (SI) Loss (Eigen et al., 2014) is used for the dense depth loss function. In contrast to the square variance error, which usually measures the difference between two images, SI Loss does not depend on the scale of the images. Following the SILoss from Bhat et al. (2021), shown in Equation defined below:

$$\mathcal{L}_{SI} = 10\sqrt{\frac{1}{\mathcal{T}}\sum_{p=1}^{\mathcal{T}}(\log(\mathcal{O}_p) - \log(\mathcal{G}_p))^2 - \frac{0.85}{\mathcal{T}^2}\left(\sum_{p=1}^{\mathcal{T}}(\log(\mathcal{O}_p) - \log(\mathcal{G}_p))\right)^2}. \quad (4.2)$$

$\mathcal{T}$ denotes the number of pixels.

4.2.3.2 Chamfer Loss

In order to shrink the gap between bin centres and ground truth depth values, the chamfer loss function (Bhat et al., 2021) is used, which uses bi-directional chamfer losses as a regularised item, defined as below:

$$\mathcal{L}_{bins} = \sum_{x\in\mathcal{G}}\min_{y\in c(\mathbf{b})}\|x-y\|^2 + \sum_{y\in c(\mathbf{b})}\min_{x\in\mathcal{G}}\|x-y\|^2. \quad (4.3)$$

In the training process, the distance between the predicted bin centres to each pixel on the ground truth and the distance between the ground truth and each pixel on the predicted bin centres were added and reduced to make the bin centres close to the depth values of ground truths while making the rest to be far from these depth values.

Shown in 4.3, $c(\mathbf{b})$ denotes the bin centres, while GT represents all pixels on a ground truth depth map.

4.2.3.3 Domain Label Losses

The source and target domain images are labelled with domain labels 1 and 0, respectively. The Domain Label Loss (DLL) function calculates the loss values between the original domain label and the output domain label from the discriminator. Inspired by focal loss (Lin et al., 2017; Saito et al., 2019), RWTD can solve the low-performance problem caused by imbalanced data in the image domain. For example, for a classification task with two image datasets which include a mixture of easy and difficult images to be classified, it can focus on difficult-to-classify data and ignore the easy-to-classify images. The proposed discriminator can ignore the easily distinguished samples and increase the weight of the samples that are difficult to distinguish. Equations are defined as

$$RWTD\left(q\right) = -f\left(q\right)\log\left(q\right), \quad f\left(q\right) = \left(1-q\right)^{\gamma}, \quad q = \max(q, \text{thres}) \tag{4.4}$$

and

$$q = \begin{cases} q & \text{if } d{=}1 \\ 1-q & \text{if } d{=}0 \end{cases} \tag{4.5}$$

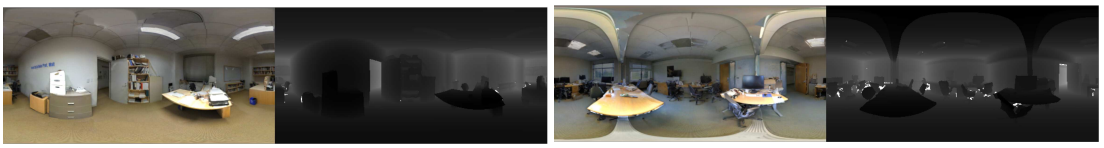
thres is the RWTD threshold factor. Its main idea is to reduce the loss contribution of those samples that are correctly classified so that the model pays more attention to those samples that are difficult to classify correctly during training.

As defined in

$$thres = \begin{cases} 1 \times 10^{-4} \times 10^{-epoch} & \text{if } q \geq 1 \times 10^{-24} \\ 1 \times 10^{-24} & \text{otherwise} \end{cases}, \tag{4.6}$$

the threshold probability *thres* decreases according to the epoch number during the training process. From preliminary experiments, it can be observed that the architecture does not perform well if the DLL increases at the beginning, as the model does not learn enough information from the source domain. In addition, with the unconstrained increasing DLL, the domain loss will lead in the wrong direction, only focusing on making the model unable to recognise the image coming from which domain. Therefore, this loss will dominate the loss function and causes poor performance. RWTD will solve this problem by constraining the loss values.

(A) SunCG



(B) Stanford2D3D

FIGURE 4.3: Sample Images from the Datasets

4.3 Implementation and Evaluation Metrics

The proposed architecture is trained on an NVIDIA RTX 3090 GPU with 24GB CUDA memory.

4.3.1 Data Exploration

The computer graphic (CG) images are from SunCG (Zioulis et al., 2018), and two different office-scene-based real-world 360 image datasets of two different buildings from Stanford2D3D (Zioulis et al., 2018). These datasets contain 512 $\times$ 256 resolution RGB images of indoor scenes with corresponding depth maps in metres. The real-world ground truth depth maps contain outliers caused by missing depth pixels. In order to improve the training efficiency, the scenes containing over 5% outliers are removed. After the pre-processing, SunCG contains 2319 scenes. Stanford2D3D area5 contains 82 scenes, and area6 contains 132 scenes.

Figure 4.3a shows sampled scenes of the SunCG dataset. They illustrate RGB images and corresponding depth maps of indoor scenes that are simulated and rendered by computers. SunCG contains different scenes that cover a variety of objects that might exist in the real world, such as beds, ladders, fans, etc. However, there are some differences between these synthetic and real-world scenes, including the textures and colours of the scenes. Figure 4.3b shows the real-world scenes from the Stanford2D3D dataset, which are taken in different buildings. It also includes many images, but they are from very limited kinds of scenes.

4.3.2 Implementation Details

All of the experiments receive inputs with the resolution of 512 $\times$ 256. Similar to previous tasks (Bhat et al., 2021; Alhashim and Wonka, 2018; Zioulis et al., 2018), in order

to save the GPU memory, the output resolution is set as 256×128. The batch size is 8, with the learning rate of 3.57×10^{-4} and 50 epochs. The Adam-optimizer (Kingma and Ba, 2014) is adopted, with the parameters $\beta 1 = 0.9$ and $\beta 2 = 0.999$. Both the dense loss factor α and Chamfer loss β are set as 0.1, and DLLF δ is set as 0.01 (Equation (4.1)), while γ in Equation (4.5) is set as 2 in the experiments.

4.3.3 Evaluation Metrics

The evaluation method is to use six metrics that have been mentioned in Section 3.2.

4.4 Experiments

In this section, the proposed architecture will be trained and tested with synthetic scenes as the source domain and real-world scenes as the target domain.

4.4.1 Performance

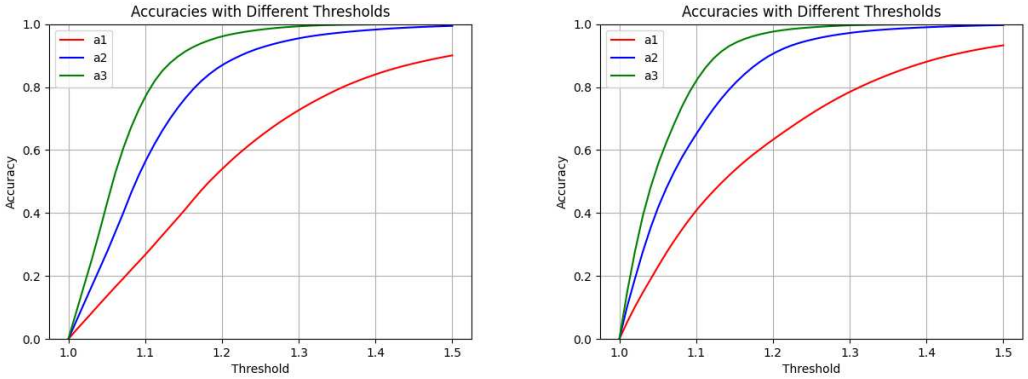
For a fair comparison, we considered different depth estimation models and compared our architecture with the best of them. The default hyperparameters were tried, but the model showed poor performance. This is because the model was becoming overfitting (see Appendix) for the synthetic dataset and performed poorly on the real-world dataset. Finally, by doing experiments with different learning rates from 1×10^{-7} to 0.1, an appropriate learning rate of 1×10^{-6} for RectNet was found to get better performance. Learning rates of U-Net Model (Alhashim and Wonka, 2018), AdaBins (Bhat et al., 2021) and SliceNet (Pintore et al., 2021) were set as 1×10^{-5}, 3×10^{-4} and 1×10^{-3} respectively after doing the similar experiments.

Table 4.1 shows that the proposed architecture outperformed the state-of-the-art (SOTA) models (Alhashim and Wonka, 2018; Bhat et al., 2021; Zioulis et al., 2018; Pintore et al., 2021) with different testing datasets, with 11% and 3% points improvement. They were trained with the SunCG dataset and tested with the Stanford2D3D testing dataset (area5) and the Stanford2D3D area6 dataset (Zioulis et al., 2018).

The proposed architecture outperforms other methods for two reasons: First, it estimates the depth by information from both the source and target domains rather than directly applying what is learned from the source domain to the target domain. Second, the architecture assigns different weights to different scenes in the source domain according to their similarity to that in the target domain during training. Thus, the proposed architecture can focus on learning scenes similar to the target domain.

TABLE 4.1: Performance Comparisons of Baseline and Proposed Architecture

Testing dataset	Model	$a_1 \uparrow$	$a_2 \uparrow$	$a_3 \uparrow$	rel $\downarrow$	rms $\downarrow$	log10 $\downarrow$
area5	Alhashim and Wonka (Alhashim and Wonka, 2018)	50.35±1.55	81.8±1.49	95.24±0.62	0.255±0.007	0.973±0.019	0.118±0.004
	RectNet (Zioulis et al., 2018)	61.04±0.86	85.81±0.49	96.23±0.21	0.216±0.002	0.926±0.009	0.098±0.001
	SliceNet Pintore et al. (2021)	59.63±4.27	88.11±3.82	97.8±0.70	0.26±0.029	0.624±0.051	0.096±0.009
	AdaBins (Bhat et al., 2021)	63.03±4.27	90.32±1.83	97.7±0.53	0.25±0.025	0.699±0.058	0.091±0.008
	Ours	**74.08±2.37**	**95.81±0.63**	**99.21±0.2**	**0.18±0.009**	**0.543±0.042**	**0.069±0.003**
area6	Alhashim and Wonka (Alhashim and Wonka, 2018)	50.56±0.32	78.6±0.57	92.52±0.32	0.271±0.003	1.098±0.007	0.123±0.001
	RectNet (Zioulis et al., 2018)	55.34±1.16	82.14±1.33	93.49±0.52	0.263±0.003	1.096±0.008	0.113±0.003
	SliceNet (Pintore et al., 2021)	57.91±6.23	86.87±1.67	96.17±0.64	0.281±0.028	0.734±0.044	0.103±0.009
	AdaBins (Bhat et al., 2021)	69.42±5.68	90.71±1.67	97.29±0.43	0.227±0.03	0.641±0.034	0.083±0.01
	Ours	**72.33±1.77**	**93.38±0.35**	**98.22±0.14**	**0.197±0.009**	**0.595±0.013**	**0.075±0.003**



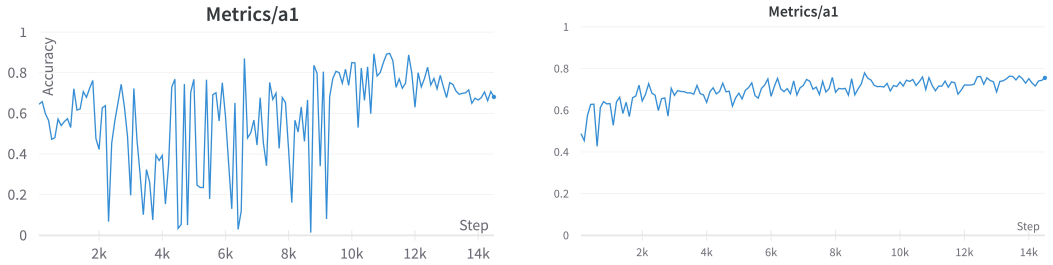
(A) AdaBins (B) Proposed Architecture

FIGURE 4.4: Accuracies with Different Thresholds. Each graph contains three accuracy curves that are used to evaluate the performance of the model with different thresholds, and these curves show the accuracies with $thresholds$, $thresholds^2$ and $thresholds^3$, respectively.

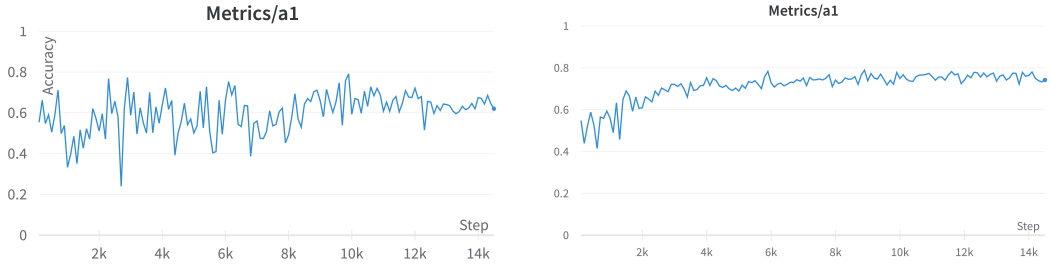
Figure. 4.4 shows the performance of two trained models with different threshold accuracies on Stanford2D3D area5. The threshold ranges from 1.0 to 1.5. For evaluation methods in this paper, 1.25 is used as the threshold according to (Eigen et al., 2014; Alhashim and Wonka, 2018; Zioulis et al., 2018; Bhat et al., 2021). This figure shows that the proposed architecture has more obvious advantages when the threshold is low, and it can show a significant competitive advantage in a more stringent evaluation condition.

4.4.2 Stability

The proposed model not only outperforms the SOTA models, such as AdaBins but also performs more stable than them. Figure. 4.5 shows the comparison of the stability of models.



(A) Stanford2D3D Area5



(B) Stanford2D3D Area6

FIGURE 4.5: Stability Comparison of First-threshold Accuracy (Left: AdaBins; Right: Proposed method)

The performance of models is evaluated with testing data every 100 batches. It can be observed that the test results of the AdaBins model fluctuated significantly during the training process, while the proposed structure is more stable than it. This is because the training dataset contains different types of scenes. When a batch of training data containing scenes is significantly different from the testing dataset, the model performance suddenly deteriorates. In the proposed structure, RWTD assigns different weights to different scenes in the training data set during the training process. It assigns high weights to scenes with high similarity while ignoring scenes from source and target domains with low similarity as much as possible, which leads to a stable performance.

Stability is essential in practical applications. After fifty epochs of training, AdaBins model converges to a smaller accuracy because it is overfitted. If we train only 20 or 30 Epochs, AdaBins method can occasionally achieve high accuracy, such as 80%, but it also fluctuates wildly. This means that the model cannot be used in practical applications because the results are uncertain, and its performance may be very good or very bad. In practice, it is difficult to know how many epochs a model should train when there are no labels. In contrast, the proposed architecture can maintain a high and stable performance.

TABLE 4.2: Investigation on the Effect of Each Component in the Proposed Architecture

Model	$a_1 \uparrow$	$a_2 \uparrow$	$a_3 \uparrow$	rel $\downarrow$	rms $\downarrow$	log10 $\downarrow$
without RD	65.15±4.05	91.13±1.59	97.71±0.53	0.24±0.025	0.683±0.055	0.087±0.008
with RD	69.68±5.43	94.57±1.95	99.03±0.4	0.199±0.025	0.565±0.068	0.075±0.008
with RWTD (Ours)	**74.08±2.37**	**95.81±0.63**	**99.21±0.2**	**0.18±0.009**	**0.543±0.042**	**0.069±0.003**

TABLE 4.3: Effect of Discriminator

Model	$a_1 \uparrow$	$a_2 \uparrow$	$a_3 \uparrow$	rel $\downarrow$	rms $\downarrow$	log10 $\downarrow$
Unsupervised Domain Adaptation (Ganin and Lempitsky, 2015)	26.2	50.7	68.1	0.855	1.720	0.235
Omnidirectional Domain Adaptation (Wu et al., 2021)	60.22±4.85	85.72±3.84	96.25±1.48	0.213±0.016	0.961±0.05	0.097±0.01
RWTD (Ours)	**74.08±2.37**	**95.81±0.63**	**99.21±0.2**	**0.18±0.009**	**0.543±0.042**	**0.069±0.003**

4.4.3 Ablation Study

4.4.3.1 Comparison of Different Components

For individual component analysis of the proposed architecture, the ablation studies are conducted on the SunCG and Stanford2D3D datasets.

Table 4.2 shows to what extent different components contributed to the proposed architecture. The architecture is evaluated with and without a discriminator. The fixed threshold discriminator was also compared, which had the same threshold as RWTD's final threshold (1×10^{-24}), called Reverse-gradient Discriminator (RD). As can be seen from the results of this table, with the help of RWTD, the accuracy of the proposed model is improved by about 9% points of a_1 accuracy compared with the structure containing encoder-decoder only, and about 4% points accuracy improvement than that with RD. The results of error metrics also show this trend.

4.4.3.2 Comparison with other Domain Adaptation Methods

Table 4.3 shows the results with different discriminators. The model from Ganin and Lempitsky (2015) cannot work well with the task from synthetic scenes to real-world scenes because of the dominant DLL. The training loss is the combination of Chamfer-Loss, SI loss and DLL. If the DLL increases dramatically at the beginning, these losses will dominate the training loss and guide the model learning in the wrong direction. This will make the discriminator unable to learn enough information from the source domain and cannot recognise the images that come from which domains.

In contrast, the proposed model makes the loss of domain labels able to increase but not dominate the loss values. For example, when the epoch is 0, DLL is 0, which provides the model with an opportunity to learn enough information from the source domain. It also keeps the direction of gradient descent from being far away from the direction of

FIGURE 4.6: Performance on a New Real-world Dataset (left: RGB images of scenes, middle: AdaBins, right: proposed architecture)

learning information of predicting depth maps only from the source domain. In each subsequent epoch, the architecture gradually increases the threshold of DLL so that the model can continuously learn information from the source domain and predict the depth map of the target domain.

4.4.4 Performance on New Dataset

Figure. 4.6 shows real-world images captured in a building with an off-the-shelf omni-directional camera. Though there is no ground-truth depth data, it can be observed that the estimated depth maps show the correct depth of the scenes with smooth changes within objects. Compared with the results from AdaBins, the proposed method shows better depth estimation for the planar ceiling regions in all test images. As AdaBins are affected by textures and lighting conditions, the saturated areas by the lighting in the ceiling show the wrong depth, while the proposed model produced smooth and planar ceiling regions learned from scenes in the source domain. For the same reason, we can see that the result of AdaBins shows discontinuous depth fields around shadow regions (e.g., the red box in Figure. 4.6). The proposed model recognised the wall and predicted a continuous and smooth depth map. In addition, AdaBins failed to predict the depth of the door (e.g., blue box in Figure. 4.6). The middle part of the door should be smooth,

but the depth map of AdaBins indicates that it is closer to the camera. Compared with it, the proposed architecture could infer a relatively smooth and planar depth for the door.

4.5 Conclusion

Existing encoder-decoder models are often incapable of reliably predicting depth maps for unlabeled real-world situations due to the lack of labelled dataset types and the difficulties of getting real-world depth maps. In this paper, we proposed to use a synthetic dataset to estimate real-world depth maps since they span a variety of scene types and are easy to acquire. A domain adaptation-based architecture with RWTD is proposed in order to address the gap between synthetic images and real-world images. It shows significantly better stability and 11% points higher accuracy than SOTA encoder-decoder models. This research makes it feasible to predict omnidirectional depth maps for real-world scenarios using a labelled dataset of synthetic images. This work has been published at the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP).

Thus far, we have addressed a challenge in a depth estimation application, enabling the model to learn from synthetic data and apply it to real-world depth estimation. However, this approach still simply uses deep learning models to take RGB images and their corresponding depth maps as learning samples without further considering what factors affect the performance of depth estimation. Based on these considerations, we aim to delve deeper into depth estimation. In the following chapter, we seek to consider physical constraint and explore its role in depth estimation, as well as propose a corresponding model.

Chapter 5

SliceFormer: Depth Estimation considering Gravity

In this research, we tackle the task of estimating depth from a single indoor omnidirectional image. Our experimental evaluation substantiates the significant impact of gravity on artificially constructed indoor environments. Building on this foundation, we process the input from the equirectangular projection by dividing it into vertical slices. These slices are then utilised as patch embeddings for the transformer encoder, a strategy designed to predict an equirectangular depth map. Our architecture is evaluated against state-of-the-art models using real-world datasets, namely Stanford2D3D and Matterport3D, demonstrating its superior performance. These results underscore the significance of our gravity-aligned approach for depth estimation in omnidirectional images, especially in man-made settings.

Since gravity has been described as an important factor in previous studies (Sun et al., 2021; Pintore et al., 2021), in this study, we sought to analyse the contribution of gravity to depth estimation. With the results and analysis in Sec. 5.3.4, we propose employing a slice-based representation for depth estimation in single indoor omnidirectional images, based on the assumption that omnidirectional images are taken by a camera placed on a horizontal-ground plane (Pintore et al., 2021; Sun et al., 2021) since most off-the-shelf cameras provide automatic alignment as their internal function. 360° cameras come with several lenses and tools like gyroscopes to automatically fix their position, or with automatic adjustment methods (Jung et al., 2017), making sure photos stay straight up and down. General cameras might also have a level or gyroscope but mainly depend on the user taking the picture to change the angle and frame based on the purpose of a certain look or creation. On the other hand, 360° cameras aim to capture everything around, auto-straightening to prevent viewers from feeling dizzy or lost.

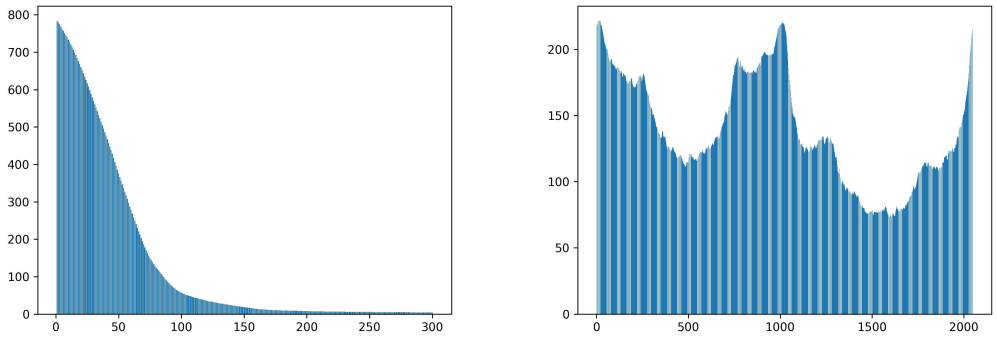
FIGURE 5.1: The Example Scene Showing How the Depth of an Object Changes along the Direction of Gravity.

5.1 Motivation

The force of gravity plays a pivotal role in shaping various vertical and horizontal components (Sun et al., 2021; Pintore et al., 2021). In the direction of gravity, due to the effect of gravity, the depth distribution of objects will show certain rules. For example, as shown in Fig.5.1, for an object placed on the ground directly in front, the depth from bottom to top is usually from near to far. The reason is that nearby objects will block distant objects. This means that depth changes in the vertical direction may be regular. In contrast, the depth in the horizontal direction does not have such a rule but shows different depths. Consequently, it is hypothesised that aligning omnidirectional image acquisition with the gravitational vector enables easier learning for models from these images, utilising features precisely oriented with gravity.

Fig.5.2 confirms this hypothesis, presenting the depth distribution in relation to gravity and the horizontal direction, based on the 360° KITTI dataset. In the right image, the protrusion is due to the fact that in the 360° KITTI dataset, the directions of left, centre, and right are roads without obstacles, as shown in Fig.5.3. However, compared to outdoor scenes, indoor omnidirectional scenes exhibit a different impact on depth distribution due to room structure, such as floors and ceilings typically being closer to the camera. Given this difference, is this depth distribution pattern also observed indoors? This question will be explored through further experimental research and analysis.

Based on these, we assume that the continuity of the information in the picture along the direction of gravity is more important for depth estimation compared with horizontal information.



(A) Vertical depth distribution (B) Horizontal depth distribution

FIGURE 5.2: Depth Distribution with 360° KITTI Dataset (de La Garanderie et al., 2018). The left image shows the depth distribution from top to bottom (vertical), while the right image displays the depth distribution from left to right (horizontal). The x-axis represents pixels from top to bottom, and from left to right of input images, respectively. The y-axis represents the average depth in metres.

FIGURE 5.3: Sample from 360° KITTI

5.2 Method

5.2.1 Gravity

For general images, determining the direction of gravity can be challenging due to unknown extrinsic camera parameters. However, as outlined in Introduction, the direction of gravity is readily identifiable in omnidirectional images, where the vertical axis corresponds to this direction.

To investigate the continuous impact of images on depth estimation along different directions (specifically, the gravity direction and the horizontal direction), we have opted to employ the bidirectional long short-term memory network (BiLSTM) model instead of the CNN for depth estimation purposes.

Specifically, shown in Fig.5.4, the vertical BiLSTM (vLSTM) segments the image into vertical slices, aligning with the gravitational direction, and these slices are then concatenated into a linear sequence, which serves as the model's input. These inputs are fed into a bidirectional BiLSTM (Siami-Namini et al., 2019) with a hidden layer size of 128 units. The bidirectional structure of the BiLSTM enables the network to simultaneously process both forward and reverse information in the sequence. Subsequently, the tensor output by the BiLSTM is reshaped and then passed through subsequent fully

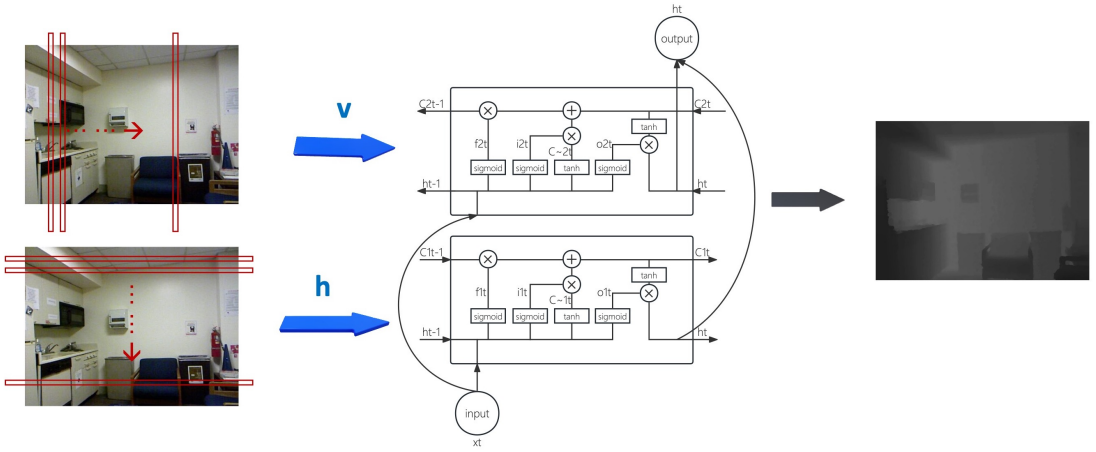


FIGURE 5.4: Pipeline for Analysing Gravity. Two types of inputs are utilised: sequences created from slices along the direction of gravity (vertical) and slices along the direction perpendicular to gravity (horizontal) are concatenated and each is used as inputs for a BiLSTM, which then predicts the corresponding depth maps, respectively.

FIGURE 5.5: Slices along the Direction of Gravity.

connected layers to obtain the predicted depth map. This process is analogous to the model learning continuous vertical features from the image. Similarly, a horizontal BiLSTM (hLSTM) extends an image along the horizontal axis.

5.2.2 SliceFormer

In indoor spaces, gravity serves as a crucial factor influencing both vertical and horizontal elements to different extents, which typically have distinct characteristics (Pintore et al., 2021). Consequently, aligning the omnidirectional image acquisition with the gravitational vector allows for the direct manipulation of these gravitational-aligned spatial features. This approach results in producing flattened and compact sequences of slices from the omnidirectional image, where each slice encapsulates a portion of the scene information, as illustrated in Fig.5.5.

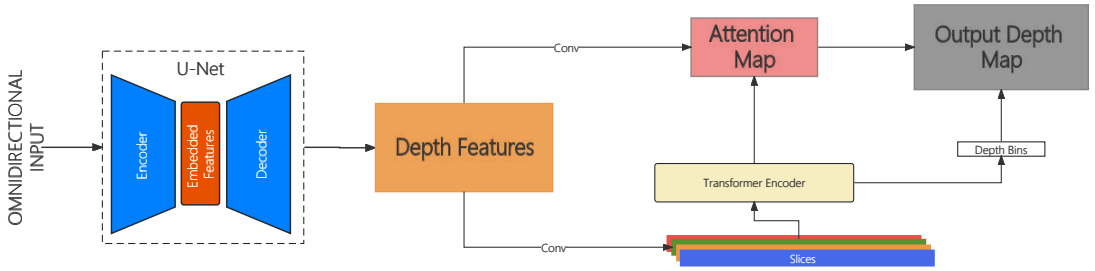


FIGURE 5.6: Overview of Proposed Architecture

5.2.2.1 Overview

Fig.5.6 illustrates the pipeline of our proposed architecture, which consists of two primary components: a U-Net shape encoder-decoder network and a slice-based transformer. The encoder-decoder captures and processes image features, while the transformer, with the proposed novel slice-based approach, facilitates effective depth map generation.

Encoder-decoder In our research, we employed a U-Net shape encoder-decoder architecture. The choice of the U-Net architecture was motivated by its remarkable performance across various image-processing tasks. This architectural design facilitates the comprehensive information of image features at varying scales, enabling the model to incorporate contextual information inherent in the images effectively. Specifically, the encoder component is tasked with capturing global image characteristics, while the decoder component is dedicated to the reconstruction of detailed information pertaining to depth maps. To bolster the performance of the encoder, the EfficientNet B5 (Tan and Le, 2019; Bhat et al., 2021) is adopted as encoder backbone architecture because it systematically expands the width, depth, and resolution of the network.

Slice-based Transformer Our significant contribution involves introducing the concept of slice-based patch embedding, a novel approach compared to the square patches used in traditional vision transformers (Wang et al., 2021). Our model segments the decoded features into slices as the transformer's input. With the encoder-decoder producing an image feature of dimensions [bs, 1, H, W], we create slices each having a height of H and a width of 1 along the vertical axis, which are suitable for patch embedding.

The initial one-dimensional features generated by the transformer undergo ReLU activation through a multilayer perceptron (MLP) head (Bhat et al., 2021), resulting in a 100-dimensional vector of depth bins, which is subsequently normalised. The remaining portion is processed by 1×1 convolution kernels, which are convolved with the

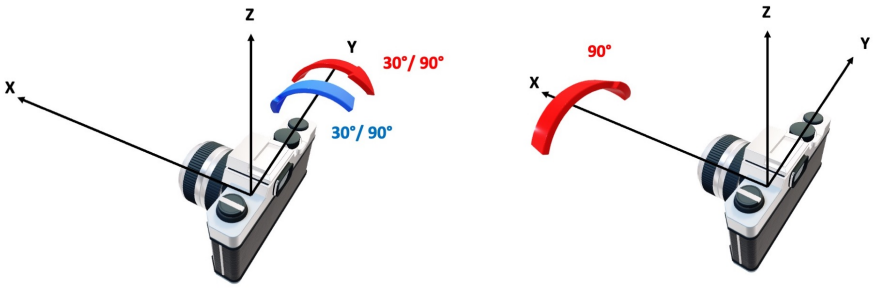


FIGURE 5.7: Representation of Different Rotation Orientation.

decoded features to produce an attention map. This attention map, in conjunction with the depth bins, is utilised to derive the final depth map.

5.2.2.2 Loss Function

Mean squared error (MSE) loss is employed to compute the loss between predicted depth maps and ground truth depth maps. This loss function adeptly accommodates data of varying scales, thereby enhancing the model's robustness and generalisation capabilities. Simultaneously, we utilise the chamfer loss (Bhat et al., 2021) to promote centres in depth bins, aligning their distribution with that of the ground truth depth maps. The loss function is defined as

$$\mathcal{L}(\mathcal{G}, \mathcal{O}) = \alpha \mathcal{L}_{\text{MSE}}(\mathcal{G}, \mathcal{O}) + \beta \mathcal{L}_{\text{Chamfer}}(\mathcal{G}, \mathcal{O}). \tag{5.1}$$

As illustrated in Equation (5.1), the parameters α and β are assigned values of 1 and 0.1, respectively, to ensure they are on a similar contribution scale.

5.3 Experiments

5.3.1 Datasets

The 3D60 dataset (Zioulis et al., 2018) was employed for training purposes. This dataset includes real-world data from Standford2D3D (Armeni et al., 2017) and Matterport3D (Chang et al., 2017), consisting of equirectangular RGB images along with corresponding depth maps, aligned to the gravity direction. To ensure data quality, scenes with more than 5% outliers were removed. Consequently, the Standford2D3D training set was reduced to 645 images, while the test set remained at 82 images. In the case of the Matterport3D dataset, the training set consisted of 2075 images, and the testing dataset comprised 1144 images.

Instead of utilising conventional perspective datasets directly, we opted to employ the tangent projection derived from the 360 dataset for a more robust comparative analysis. This decision was made to ensure a more stringent evaluation and fair comparison of our approach. For example, the NYU dataset (Silberman et al., 2012) employs relative depth measurements instead of absolute depth, and due to its hand-held nature when the data was collected, not all images within the NYU dataset exhibit perfect alignment with gravity. This particular dataset introduces additional variables and uncertainties to our experimentation.

Based on this, three types of general perspective projection datasets are generated from Stanford2D3D by using bilinear interpolation to check whether FoVs will change the contribution of gravity. The input equirectangular projections were transformed into perspective images with a 90-degree, 60-degree, and 45-degree field of view angle, respectively, resulting in output images of 256×256 pixels through perspective projection.

For general perspective images with different angles, shown in Fig.5.7, we set the X-axis as the roll axis, the Y-axis as the pitch axis, and the Z-axis as the yaw axis. The datasets with random pitch angles of -30° to 30°, and -90° to 90°, marking them as 'v30' and 'v90' are generated, respectively. The dataset with the random roll rotation angles between 0° to 90° was marked as 'roll90'.

5.3.2 Implementation

The proposed architecture and models are trained on NVIDIA RTX3090, with 24GB of CUDA memory.

5.3.3 Evaluation Metrics

Six commonly utilised metrics in prior depth estimation studies are a_1, a_2, a_3, $\log_{10}$, *rel*, and *rmse*. Three accuracy metrics are assessed using the accuracy thresholds 1.25, 1.25^2, and 1.25^3 to evaluate performance across various sensitivity levels. Larger values of these accuracy metrics indicate better model performance. Three different loss functions are employed to assess the model's robustness: *absolute relative error*, *linear rmse* and log_{10} *rmse*. For experiment results in the following tables, these are denoted as *rel*, *rmse* and log_{10}, respectively. The *rel* is a typical approach for quantifying regression errors, measuring the relative difference between the predicted depth value and the true depth value. The *rmse* metric helps highlight the impact of significant distance errors by squaring the differences between the output of the model and the ground truth depth map, whereas log_{10} mitigates the effect of a small number of outliers by presenting the error in relative form through logarithmic transformations. Smaller values of these loss metrics indicate better model performance.

5.3.4 Gravity

In this section, to investigate the contribution of gravity alignment to the depth estimation of a single indoor image, we analyse the depth distribution in vertical and horizontal directions. In order to further analyse the difference between the two directions, we trained the data of different perspectives, different FoVs, and different rotation angles according to horizontal and vertical modes respectively and analysed their performance. Each training is conducted five times, and the means and standard deviations are calculated.

5.3.4.1 Experiment Results and Analysis

For these experiments, since the dataset is aligned, the vertical direction is the direction of gravity. Table 5.1, 5.2, 5.3, 5.4, and 5.5 show the performance of different datasets and different modes for the same dataset. Compared to these results, the experimental results indicate that when there is a discrepancy between the vertical and horizontal depth distributions (shown in Fig.5.8), similar discrepancies emerge in performance, as shown in the above tables.

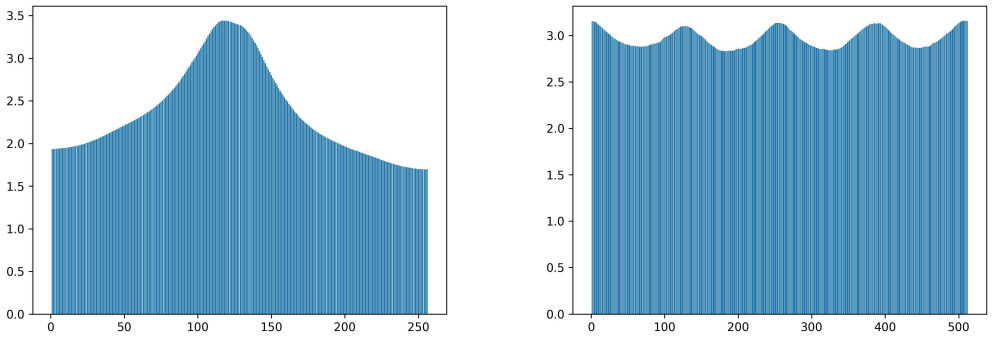
Depth Distribution The depth distribution was visually evaluated to compare the difference between vertical and horizontal directions. Fig.5.8a and Fig.5.8b illustrate distinct variations in depth distribution between the vertical and horizontal directions, respectively. Vertically, there is a discernible pattern of transitioning from near to far and then from far to near, whereas horizontally, the distribution appears to exhibit a greater degree of randomness, except for showing higher depth in the four corners of the room.

TABLE 5.1: Performance of Different Datasets with Equirectangular Projections.

Dataset	Mode	$a_1 \uparrow$	$a_2 \uparrow$	$a_3 \uparrow$	$log_{10} \downarrow$	$rel \downarrow$	$rmse \downarrow$
Matterport3D	vertical	84.99 ± 0.07	96.45 ± 0.02	98.89 ± 0.01	0.0538 ± 0.0001	0.1322 ± 0.0004	0.4672 ± 0.0009
	horizontal	73.14 ± 0.72	92.34 ± 0.55	97.33 ± 0.33	0.0775 ± 0.0019	0.1985 ± 0.0068	0.622 ± 0.0192
Stanford2D3D	horizontal	90.38 ± 0.52	97.94 ± 0.09	99.41 ± 0.04	0.0434 ± 0.0012	0.1056 ± 0.0034	0.3483 ± 0.0087
	vertical	79.5 ± 0.55	95.38 ± 0.17	98.61 ± 0.03	0.0615 ± 0.0008	0.1475 ± 0.0018	0.525 ± 0.0062

Original Equirectangular Projections With equirectangular projections of Matterport3D, Table 5.1 shows that the performance of hLSTM is worse compared with them. For example, It shows about a 13% points drop in a_1 metric compared with the performance of vLSTM and hLSTM.

In the case of Stanford2D3D in Table 5.1, the outcomes for continuous data in the gravitational direction closely align with those achieved by the vLSTM model, exhibiting a roughly 10% increase in a_1 when compared to the hLSTM model.



(A) Vertical Depth Distribution (B) Horizontal Depth Distribution

FIGURE 5.8: Depth Distributions along Different Directions. The figures show the depth distribution of the Stanford2D3D dataset in the vertical and horizontal directions. The y-axis represents the average depth (unit in meters). The left figure represents the vertical direction, and the x-axis represents pixels from top to bottom of input images. The right figure represents the horizontal direction, and the x-axis represents from left to right of input images.

TABLE 5.2: Performance of Half-equirectangular Projections with the Matterport3D Dataset.

Dataset	Mode	$a_1 \uparrow$	$a_2 \uparrow$	$a_3 \uparrow$	$log_{10} \downarrow$	$rel \downarrow$	$rmse \downarrow$
Half	vertical	84.08 ± 0.4	96.09 ± 0.13	98.77 ± 0.02	0.0558 ± 0.0009	0.1371 ± 0.0025	0.4626 ± 0.0033
	horizontal	69.98 ± 0.89	90.75 ± 0.49	96.58 ± 0.27	0.0838 ± 0.0019	0.2176 ± 0.0061	0.6698 ± 0.0181
Half Rotate	vertical	70.12 ± 0.54	90.78 ± 0.31	96.52 ± 0.17	0.0839 ± 0.0012	0.2192 ± 0.0039	0.6763 ± 0.0128
	horizontal	84.33 ± 0.09	96.08 ± 0.03	98.73 ± 0.02	0.0554 ± 0.0001	0.1358 ± 0.0004	0.4661 ± 0.0015

TABLE 5.3: Performance of Half-equirectangular Projections with the Stanford2D3D Dataset.

Dataset	Mode	$a_1 \uparrow$	$a_2 \uparrow$	$a_3 \uparrow$	$log_{10} \downarrow$	$rel \downarrow$	$rmse \downarrow$
Half	vertical	90.83 ± 0.43	98.11 ± 0.1	99.51 ± 0.06	0.0427 ± 0.002	0.1033 ± 0.0049	0.334 ± 0.0091
	horizontal	77.77 ± 0.5	95.13 ± 0.08	98.58 ± 0.06	0.0644 ± 0.0006	0.1528 ± 0.0014	0.5316 ± 0.0061
Half Rotate	vertical	77.81 ± 0.3	95.25 ± 0.13	98.71 ± 0.05	0.0639 ± 0.0003	0.1496 ± 0.0013	0.5123 ± 0.002
	horizontal	90.67 ± 0.57	98.24 ± 0.16	99.54 ± 0.06	0.0426 ± 0.0007	0.1025 ± 0.0024	0.3167 ± 0.0104

Half Original Equirectangular Projections In order to maintain uniformity in the number of consecutive pixels during expansion in both directions, the original image was bisected, with only the left half utilised for training and testing. Since an equirectangular projection represents a 360° view, it is of no significance which part we choose after the crop. We therefore choose the left half of each equirectangular projection. Given the original image's height-to-width ratio of 1:2, the cropping process results in a new ratio of 1:1. Consequently, this approach ensures consistency in the number of consecutive pixels during BiLSTM training and testing in both directions.

It can be observed in Table 5.2 that the performance of vLSTM exhibits approximately a 14% a_1 accuracy improvement compared to the hLSTM with Matterport3D dataset. Similarly, Table 5.3 shows that the performance of vLSTM exhibits approximately a 14% a_1 accuracy improvement compared to the hLSTM with Stanford2D3D dataset. Errors

metrics also show the trend, with about 0.03 improvement for log_{10}, 0.08 for *rel*, and about 0.2 for *rmse*.

Rotate Half Original Equirectangular Projections After 90° rotation, the performance between vertical and horizontal inputs has been switched. As shown in Table 5.2, the results indicated that the hLSTM achieved approximately 14% higher a_1 accuracy than the vLSTM on the Matterport3D dataset. In contrast to the performance with images before rotation, the vertical and horizontal BiLSTMs exhibit reversed priorities, and the error metrics also show a reversed trend. Table 5.3 demonstrates the same trend with Stanford2D3D dataset.

These observations demonstrate that the information conveyed by consecutive pixels in the two directions is genuinely distinct.

General Perspective These images are generated from the Stanford2D3D dataset and inherently align with the direction of gravity unless the roll angles are adjusted.

TABLE 5.4: Performance of Different FoVs with General Perspective Stanford2D3D.

Dataset	Mode	$a_1 \uparrow$	$a_2 \uparrow$	$a_3 \uparrow$	$log_{10} \downarrow$	$rel \downarrow$	$rmse \downarrow$
FoV 90°	vertical	87.79 ± 0.1	97.28 ± 0.05	99.2 ± 0.02	0.0497 ± 0.0001	0.1217 ± 0.0004	0.4028 ± 0.0009
	horizontal	76.01 ± 0.41	94.22 ± 0.06	98.15 ± 0.05	0.0708 ± 0.0006	0.1677 ± 0.0016	0.6128 ± 0.0031
FoV 60°	vertical	84.97 ± 0.3	96.87 ± 0.06	99.05 ± 0.02	0.0559 ± 0.0005	0.1345 ± 0.0011	0.4436 ± 0.0038
	horizontal	76.88 ± 0.29	93.92 ± 0.09	98.1 ± 0.05	0.0696 ± 0.0003	0.164 ± 0.0007	0.6326 ± 0.0033
FoV 45°	vertical	84.78 ± 0.57	97.22 ± 0.08	99.1 ± 0.01	0.0561 ± 0.001	0.1336 ± 0.0019	0.4224 ± 0.0049
	horizontal	78.87 ± 0.24	94.37 ± 0.11	98.15 ± 0.04	0.0665 ± 0.0003	0.1591 ± 0.0009	0.6062 ± 0.004

FoV Even when general perspective projections serve as input data, it is observed that the vLSTMs show better performance compared to the hLSTM, although the difference decreases from about 10% to 7% for 45°, 60° and 90° FoVs (shown in Table 5.4). Different FoVs still show the same trend in both accuracy metrics and error metrics that the performance of vertical mode is better than horizontal mode. Therefore, FoV is not the main factor influencing vertical and horizontal differences in depth estimation.

Pitch Angle We project the projections obtained after randomly rotating the camera from -30 degrees to 30 degrees and -90 to 90 degrees in the pitch direction as input to the model with a 60-degree field-of-view. As can be seen from Table 5.5, the performance of vLSTM is still better than that of hLSTM with about 7% and 6% points difference of a_1 accuracy, respectively.

Random Roll Rotation We set the FoV to 60 degrees, randomly rotate the camera's posture in the vertical direction (from -90 to 90 degrees), and consider the roll rotation

TABLE 5.5: Performance of Different Angles with General Perspective Stanford2D3D with FoV 60°. As stated in Sec. 5.3.1, the first variable v corresponds to random pitch angles, and *rot* refers to roll angles, with their respective angle values following each notation.

Dataset	Mode	$a_1 \uparrow$	$a_2 \uparrow$	$a_3 \uparrow$	$log_{10} \downarrow$	$rel \downarrow$	$rmse \downarrow$
v30°	vertical	83.47 ± 0.26	96.49 ± 0.07	98.99 ± 0.02	0.0578 ± 0.0005	0.1403 ± 0.0013	0.4506 ± 0.0039
	horizontal	76.01 ± 0.2	93.72 ± 0.04	97.97 ± 0.02	0.0714 ± 0.0003	0.1705 ± 0.0005	0.5987 ± 0.0019
v90°	vertical	79.22 ± 0.45	96.66 ± 0.14	99.33 ± 0.02	0.0621 ± 0.0005	0.1422 ± 0.001	0.3876 ± 0.0027
	horizontal	73.6 ± 0.13	93.42 ± 0.07	98.01 ± 0.02	0.0722 ± 0.0003	0.1646 ± 0.001	0.4877 ± 0.0014
v90° roll90°	vertical	77.6 ± 0.15	95.03 ± 0.12	98.64 ± 0.02	0.0664 ± 0.0003	0.1573 ± 0.0008	0.428 ± 0.0024
	horizontal	77.48 ± 0.27	95.31 ± 0.09	98.72 ± 0.04	0.0662 ± 0.0004	0.1565 ± 0.001	0.4274 ± 0.0031

from 0-90 degrees to simulate the random pictures with a normal camera. Table 5.5 shows that the performance of vLSTM and hLSTM become similar. This is because the random rotation prevents the scene from strictly aligning with the direction of gravity.

5.3.4.2 Summary and Discussion

In general, without the change in the direction of gravity involved, we can see that the performance shows the same trend, that is, the performance with input in the vertical direction is better than that in the horizontal direction. These results clearly show the importance of gravity alignment in depth estimation.

Previous research has utilised gravity alignment but has not explicitly analysed the role of gravity, thereby presenting a gap in the literature (Sun et al., 2019, 2021; Pintore et al., 2021). Our study furnishes substantial supportive evidence and a foundational understanding of the role of gravity alignment in depth estimation. For future endeavours, researchers could potentially enrich the model's informational yield by focusing on the insights provided by vertical orientation, thereby enhancing the model's performance and generalisation.

5.3.5 SliceFormer

From the above experiments, it can be seen that gravity plays an important role in depth estimation. The information of the direction aligned with gravity provides more depth cues and shows better performance when used as input to the BiLSTM model.

5.3.5.1 Results and Analysis

We compared the proposed architecture with UResNet in OmniDepth (Zioulis et al., 2018), HQM (Alhashim and Wonka, 2018), SliceNet (Pintore et al., 2021) and AdaBins (Bhat et al., 2021) models. Table 5.6 and Table 5.7 show the performance of different

TABLE 5.6: Depth Estimation Performance with Matterport3D

Model	$a_1 \uparrow$	$a_2 \uparrow$	$a_3 \uparrow$	$log_{10} \downarrow$	$rel \downarrow$	$rmse \downarrow$
OmniDepth (Zioulis et al., 2018)	87.37	97.24	99.14	0.0493	0.1211	0.4279
HQM (Alhashim and Wonka, 2018)	89.00	97.91	99.41	0.0460	0.1117	0.3915
SliceNet (Pintore et al., 2021)	86.25	97.73	99.31	0.0576	0.1285	0.4285
AdaBins (Bhat et al., 2021)	90.66	98.44	99.55	0.0425	0.1021	0.3659
Ours	**90.76**	**98.61**	**99.64**	**0.0421**	**0.1002**	**0.3607**

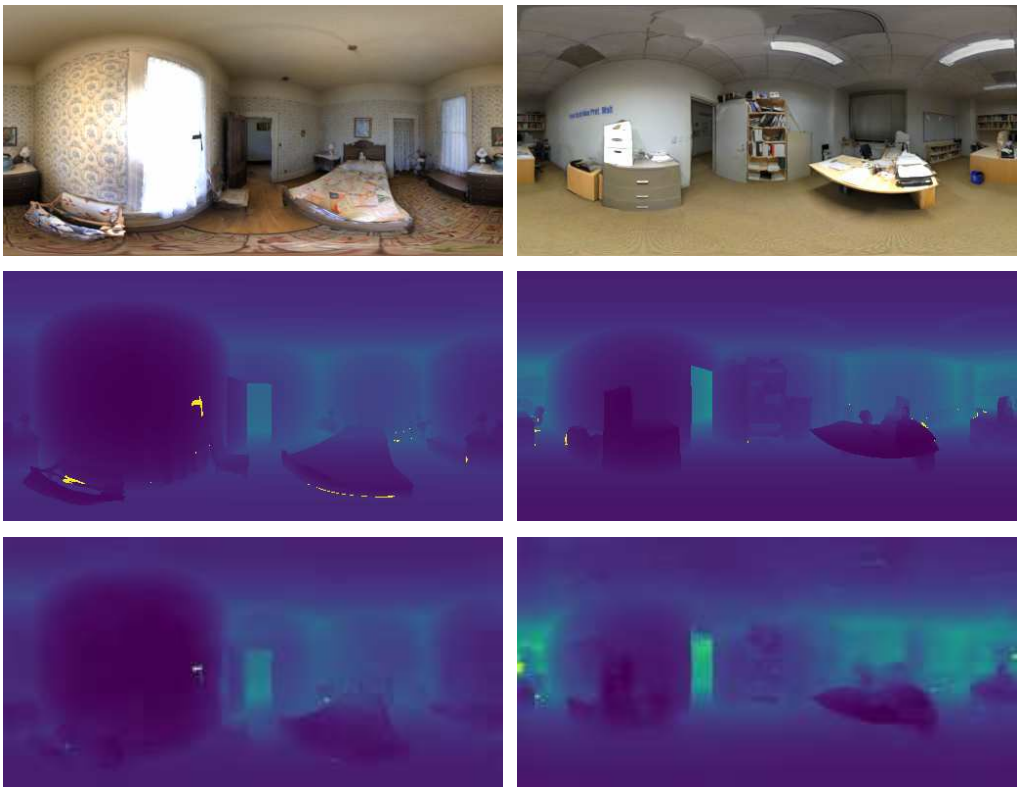
TABLE 5.7: Depth Estimation Performance with Stanford2D3D

Model	$a_1 \uparrow$	$a_2 \uparrow$	$a_3 \uparrow$	$log_{10} \downarrow$	$rel \downarrow$	$rmse \downarrow$
OmniDepth (Zioulis et al., 2018)	92.42	**98.63**	99.54	0.0360	0.0838	**0.3392**
HQM (Alhashim and Wonka, 2018)	90.17	98.32	**99.57**	0.0440	0.1042	0.3485
SliceNet (Pintore et al., 2021)	88.61	98.40	99.56	0.0562	0.1220	0.3757
AdaBins (Bhat et al., 2021)	93.15	98.43	99.36	0.0334	0.0780	0.3853
Ours	**93.43**	98.55	99.38	**0.0333**	**0.0770**	0.3772

models on the Matterport3D and Stanford2D3D datasets, respectively. The proposed model suppresses other state-of-the-art models in performance on the Matterport3D dataset, although without demonstrating an absolute advantage on the Stanford2D3D dataset. This is because the Stanford2D3D dataset is small and easy to learn so the performance of different models is similar. In addition, although other models such as OmniDepth and HQM, slightly outperformed on $a_{2,3}$ and *rmse* on the Stanford2D3D dataset, respectively, it is crucial to highlight that our approach demonstrates superior performance under the rigorous a_1 accuracy metric. The relative strengths of OmniDepth and HQM in $a_{2,3}$ and *rmse* potentially stem from their insensitivity to a limited set of outliers. This highlights the superior and stringent accuracy of our model, underscoring its precision-centric advantages in performance. Therefore, for comprehensive scene depth estimation tasks, our model offers distinct advantages.

In addition, Fig.5.9 presents the qualitative outcomes of SliceFormer on the Matterport3D and Stanford2D3D datasets. As evident from the figure, the model offers highly accurate depth estimations. Moreover, in regions of the ground truth depth map containing outliers, the model provides precise estimations that align with the actual scene conditions. Heatmaps in Fig.5.10 are obtained by subtracting the ground truth depth map and the predicted depth map, respectively. It is observed that except for the parts with outliers, the error is low for the remaining sections, as indicated by the black colour. They substantiate the analysis for Table 5.6 and 5.7, demonstrating that the proposed architecture yields accurate depth estimations overall.

In order to further understand the contribution of the slice module, we compared the slice module with the traditional square patch (16×16) method for comparison. It can be seen from the experiments in Table 5.8 that the model based on slice has better performance than that with traditional square patches. Our model performs better than the traditional square patch under more rigorous and sensitive metrics. For example, there



(A) Matterport3D Dataset (B) Stanford2D3D Dataset

FIGURE 5.9: Qualitative Results. The top row illustrates the original RGB images, the middle row displays the ground truth depth maps, and the bottom row showcases the depth maps predicted by SliceFormer.

(A) Heatmap of Matterport3D (B) Heatmap of Stanford2D3D

FIGURE 5.10: Heatmaps for Different Datasets. The output depth and ground truth depth of Matterport3D and Stanford2D3D in Fig.5.9 are respectively subtracted.

are 0.7% and 1% points a_1 accuracy improvement in Matterport3D and Stanford2D3D, respectively.

5.4 Conclusion

In this study, we have analysed the importance of gravity in depth estimation. The experimental results show that gravity direction alignment plays a positive role in depth

TABLE 5.8: Comparison of performance between traditional square patches as transformer inputs and gravity-aligned slices as inputs.

Dataset	Model	$a_1 \uparrow$	$a_2 \uparrow$	$a_3 \uparrow$	$log_{10} \downarrow$	$rel \downarrow$	$rmse \downarrow$
Matterport3D	Traditional Patches	90.01	98.452	**99.57**	0.0433	0.1034	0.3677
	Gravity Alignment	**90.79**	**98.459**	99.55	**0.0419**	**0.1004**	**0.3636**
Stanford2D3D	Traditional Patches	92.73	98.48	**99.46**	0.0344	0.0808	**0.3707**
	Gravity Alignment	**93.75**	**98.51**	99.39	**0.0324**	**0.0752**	0.3765

estimation, regardless of the input pictures with different FoVs or pitch rotations. Considering the alignment of gravity, we introduced an innovative architecture for deep dense depth estimation from a single indoor omnidirectional image, utilising a slice-based transformer approach. This method begins with feature extraction from the input image using a U-Net-shaped encoder-decoder model, followed by partitioning these features into gravity-aligned slices for patch embedding. The transformer then generates a 100-dimensional vector representing depth bins, which, when combined with the attention map, produces the final depth map. This architecture has been rigorously tested on real-world indoor omnidirectional datasets, demonstrating superior performance compared to current state-of-the-art methods. The notable aspect of our approach is its utilisation of gravitational direction for depth estimation, which aims to contribute to the broader discourse in single-image depth estimation, potentially offering new insights for depth estimation in indoor environments.

Future work could explore the contribution of gravity and its challenges in different application scenarios, such as depth estimation for outdoor scenes. In addition, it will be a valuable research direction to study the internal mechanism of how the model uses the information of gravity alignment for depth estimation. This may not only improve the internal mechanisms, transparency, and credibility of depth estimation models, but also open up new perspectives and methodologies for the development of depth estimation techniques.

Chapter 6

How does the Machine Perceive Depth for Indoor Single Images with CNN?

As mentioned before, depth estimation from a single image is a challenging problem in computer vision because binocular disparity or motion information is absent. Whereas impressive performances have been reported in this area recently using end-to-end trained deep neural architectures, as to what cues in the images that are being exploited by these black box systems is hard to know. To this end, in this work, we quantify the relative contributions of the known cues of depth in a single-image depth estimation setting using an indoor scene dataset. Our work uses feature extraction techniques to relate the single features of shape, texture, colour and saturation, taken in isolation, to predict depth. We find that the shape of objects extracted by edge detection substantially contributes more than others in the indoor setting considered, while the other features also have contributions in varying degrees. These insights will help optimise depth estimation models, boosting their accuracy and robustness. They promise to broaden the practical applications of vision-based depth estimation.

Our investigation reveals that, even for general images, there is a lack of detailed work analysing the impact of various independent features in deep learning-based single-image depth estimation for indoor settings. Furthermore, it is obvious that factors such as colour and texture have a general influence on visual perception, and this situation remains consistent across both general and omnidirectional images. Additionally, the dataset for general images is significantly larger than that for omnidirectional images, offering a more universally applicable set of conclusions. Considering these factors, we have chosen to conduct our experiments using the NYU (Silberman et al., 2012) dataset in this chapter.

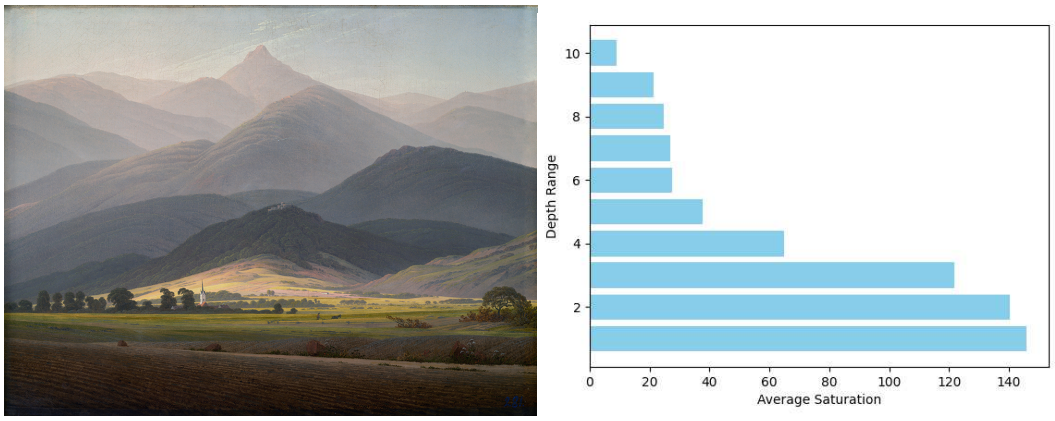
6.1 Motivation

Depth estimation, predicting the distance from an object's surface to the camera, is a key task in the field of computer vision. It plays a crucial role in many applications, such as 3D reconstruction (Alawadh et al., 2022), autonomous driving (Janai et al., 2020; Wang et al., 2019), virtual reality (VR) (Dickson et al., 2021), augmented reality (AR) (Lee et al., 2011), etc. The goal of single-image depth estimation based on deep learning is to infer the depth value of each pixel by analysing the scene information in a single image.

Due to the ill-posed problem of single-image depth estimation, there is a fundamental need to move towards a scene understanding of objects in images so that various characteristics of objects can cue depth information. Recent work on single-image depth estimation using end-to-end trained deep neural network models shows that such cues are collectively learnable and satisfactory depth estimation can be achieved (Eigen et al., 2014; Alhashim and Wonka, 2018; Bhat et al., 2021; Wu et al., 2023). However, the black-box nature of such models prohibits the understanding of what cues are exploited in single-image depth estimation. The mechanism behind single-image depth estimation based on 2D images in neural networks is still not clearly explained, and the extent to which these models can approximate the human capability of monocular depth perception remains uncertain.

Building on this gap in understanding, and inspired by causality analysis (Liu et al., 2022), we aimed to investigate the factors that influence depth estimation. This work will pave the way for the development of versatile models applicable to a broader spectrum of depth estimation tasks, moving beyond reliance solely on data-driven approaches. Research has shown that single-image depth cues in 2D images include phenomena such as blurring, shading and brightness (Swain, 1997). This paper investigated and analysed the factors that influence machine-based single-image depth estimation. To provide a comprehensive understanding, we investigated the roles of the factors relevant to object recognition (Ge et al., 2022), such as colour and texture, in the context of single-image depth estimation. However, many factors are interrelated and cannot be independently segregated. In the context of scenarios where it is possible to directly and independently extract features from 2D images, we have considered colour, saturation, texture and shape, and each of them holds significant relevance in image processing, exerting varying degrees of impact on the overall outcome.

Colour. Colour is recognised by the perception and interpretation of different wavelengths of light by the eye (Grzybowski and Kupidura-Majewski, 2019). The visual information humans gather heavily relies on the presence of colour (Neitz and Neitz, 2000). Colour helps humans recognise and remember objects faster (Gegenfurtner and Rieger, 2000). Nevertheless, when defining colour, it is critical to recognise that RGB images do not only represent a singular colour but also include various elements in



(A) Nature Scene Image (B) Saturation Distribution

FIGURE 6.1: Saturation Analysis for a Nature Scene

addition to colour, such as shape and texture. To isolate the pure colour information, we utilised a phase scrambling approach (Ge et al., 2022), which effectively separates the colour from these additional attributes.

Saturation. The second feature of interest is saturation. Saturation refers to the purity or intensity of a colour. For instance, high saturation indicates a more vivid and pure colour, while low saturation suggests a lighter or more desaturated colour with a hint of grey. Aerial perspective, within the domain of remote viewing, refers to the impact of the atmosphere on the visual depiction of an object. For instance, in Figure 6.1a, a nature photograph is displayed. We evenly split images into ten rows, and the average saturation values have been calculated for each row, as depicted in Figure 6.1b. As the object moves away from the camera, it can be observed that the saturation decreases. Building on this observation, saturation serves as a depth cue for outdoor single-image depth estimation. We aimed to investigate the utility of saturation as a depth cue in indoor scenes.

Texture. In computer vision, texture is defined by repetitive patterns with varying intensities present in an image(Tuceryan and Jain, 1993). Prior research has found that textures are important when influencing a human's perception of distance (Rowland, 1999), with specific regions in the brain having been found to be activated when exposed to varying textures (Puce et al., 1996). Therefore, we also sought to independently extract the features pertaining to texture and assess their impact on depth estimation.

Shape. A shape is generally considered to be a graphical representation of an object or its external borders, contours or external surfaces. Acquiring precise boundaries of objects in the 3D world based solely on 2D images is challenging. To simplify this process, we defined the shape feature as the edge graph, which corresponds to a greyscale map generated using an edge detection algorithm designed to preserve the object's boundaries. Edges are regarded as one of the primary cues essential for the human

visual system (Farid et al., 2013). Edge graphs usually represent geometric structures or boundaries between objects. For depth estimation tasks, the geometric features of an object are crucial to inferring its depth. The geometric structure aids depth estimation algorithms in capturing the shapes and relationships between objects (Jin et al., 2020), with edge maps providing supplementary geometric information. Edge detectors can analyse pixel gradients in different areas of the image, thereby assisting in the estimation of the relative distances between objects.

6.2 Background

Interpretability within deep learning is attracting significant and growing interest. Interest in Convolutional Neural Networks (CNNs) and Visual Transformers has been rapidly increasing lately, particularly concerning their interpretability. A study revealed that CNN models trained on ImageNet exhibit a heightened sensitivity to texture information (Geirhos et al., 2018). In addition, recent research has undertaken a comparison of the attributes between CNNs and Transformers across various layers (Raghu et al., 2021) by using Centred Kernel Alignment (CKA) (Cortes et al., 2012; Kornblith et al., 2019). According to their claims, the transformer allows the early gathering of global information in contrast to CNNs. This results in a robust propagation of features from lower to higher layers in the network. Nevertheless, the primary focus of these enquiries remains centred on model analysis.

In the realm of human depth perception, substantial work has been conducted to investigate cues such as position in the image, texture density and focus blur (Gibson, 1950; Cutting and Vishton, 1995). Existing works have demonstrated various methods for indoor single-image depth estimation that exhibit good performance (Eigen et al., 2014; Bhat et al., 2021). Despite this, an analysis of their operations is still lacking. To the best of our knowledge, there has been no analysis of the contributions of different types of depth cues specifically for deep learning-based depth estimation in indoor single-image scenarios. In the two most relevant prior studies to our work, (Hu et al., 2019) conducts attribution analysis to identify pixels that contribute most significantly to the final depth map. However, these methods can only offer insights into the low-level workings of CNNs. The analysis in Dijk and Croon (2019) was primarily confined to specific objects situated in outdoor environments, such as animals and vehicles on roadways. In contrast, in our study, we focused on colour, saturation, texture and shape, taking into account that our target application pertains to indoor scenes and requires the extraction of these cues from a single image.

Approaches for emulating the human capacity for gauging depth from an indoor scene still encounter gaps in knowledge. The objective of this paper is to unveil how neural networks extract depth-related information from a single indoor image to attain a more

(A) Original RGB (B) Corresponding GT Depth

FIGURE 6.2: A Sample from the NYU Dataset

profound comprehension of the disparities between monocular visual depth estimation and the depth perception exhibited by humans.

Simultaneously, our work offers a foundational framework to facilitate subsequent investigations into assessing the interdependence among pertinent variables in the realm of depth estimation. A prior exploration delved into the causal interplay within 3D reconstruction, deconstructing elements like perspective and depth while also attempting to substantiate the interlinkages amid diverse variables (Liu et al., 2022). Nonetheless, the model expounded upon in this enquiry operates on the assumption that the object is symmetric (Wu et al., 2020). Through an autoencoder mechanism, it internally dissects the input image into manifold components, as opposed to explicitly extracting a corresponding viewpoint, depth and related insights from the RGB image. In our study, we exclusively extracted various factors from RGB images while carefully investigating the significance of these factors within the context of depth estimation. Our work is set to further enable causal analyses in the field of depth estimation, paving the way for future advancements in comprehending the causality of depth estimation.

6.3 Method

In this section, we consider four cues for single-image depth estimation. In order to compare the appearance of these four different features, we use the same sample in this section. Figure 6.2 shows the original RGB image and its corresponding ground truth (GT) depth from the NYU dataset (Silberman et al., 2012).

6.3.1 Colour

Figure 6.3 illustrates the relationship between the distribution of original RGB three-channel values and the depth maps. The pixels on original RGB images are primarily concentrated between 0 and 100 in the corresponding grey-scale depth maps. The

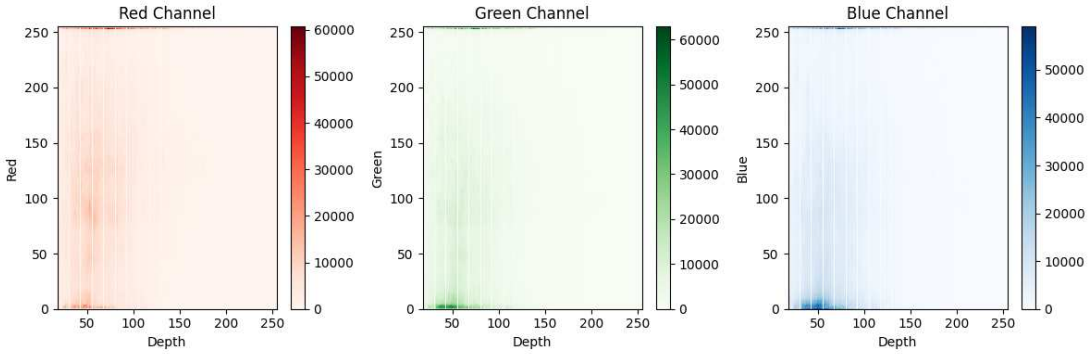


FIGURE 6.3: Heat Map of the Relationship between the Distribution of Original RGB Three-channel Values and the Depth Map. The horizontal axis represents the depth range, while the vertical axis corresponds to the pixel count of the R, G and B channels within the respective depth ranges. The colour bar values represent the pixel counts for three respective channels from 500 images randomly selected from the NYU dataset.

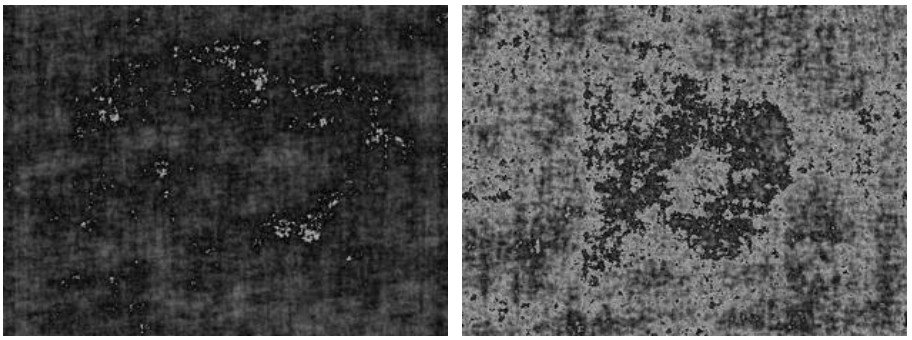
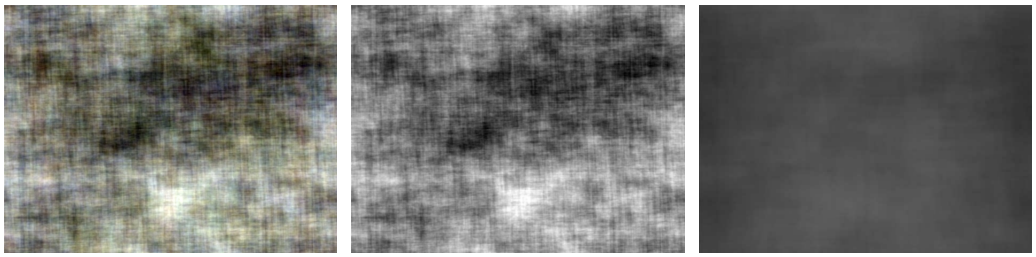


FIGURE 6.4: Phase Scrambled H Map and Corresponding Depth Map of Figure 6.2

heat map reveals that the values of R, G and B pixels are similarly distributed across a specific depth range. This shows that the factors affecting depth are not significantly related to the distribution of pixels on the RGB channel. More details are shown in the Appendix.

Hue from the hue, saturation and luminance value (HSV) colour space can be an expression of colour. However, hue values represent the projection of the RGB colour space onto a non-linear chroma angle (Szeliski, 2010). If an output pixel value falls outside the valid range, it necessitates remapping to bring it within the specified range. The chroma angle represents a non-linear trajectory within a continuous, uninterrupted space. Here, starting at 0 degrees is the same as coming full circle to 360 degrees. However, when we apply this idea to an image, like with H maps, the smooth flow is interrupted, creating a series of separated points instead. Figure 6.4 illustrates the images and corresponding depth maps resulting from the phase scrambling and remapping process applied to the H map from Figure 6.2, which are mapped back to specific intervals. Some discontinuous blocks can be observed in this figure. Therefore, we did not consider utilising the hue maps as the colour feature.



(A) RGB Image Phase Scrambling

(B) Gray-scale Image Phase Scrambling

(C) Depth Phase Scrambling

FIGURE 6.5: Phase Scrambling Results of Figure 6.2

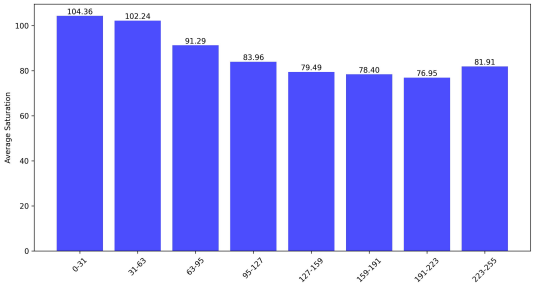


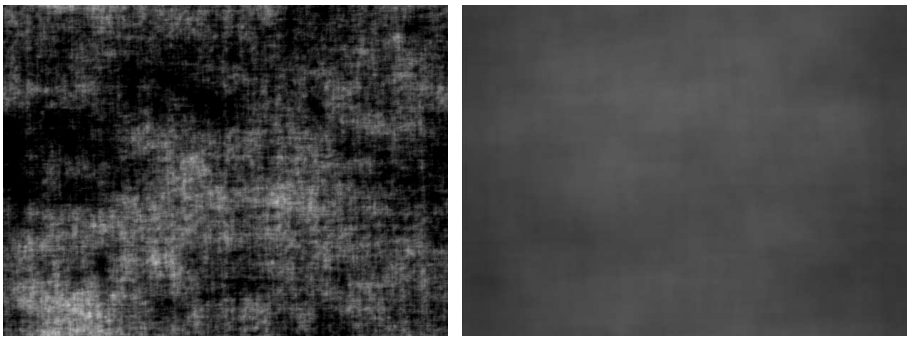
FIGURE 6.6: Average Saturation at Different Depth Intervals for Indoor Scenes (NYU dataset)

To examine the contribution of colour to depth, we performed phase scrambling (Ge et al., 2022) on original RGB images (details shown in Appendix B) and their corresponding depth maps to remove influences from shapes, textures and other geometric features. The resulting dataset was labelled "RGB Phase Scrambled". Nevertheless, even after the phase scrambling, the outcome still retains the brightness information, making it not purely a colour feature. Subsequently, these outputs were converted to greyscale, effectively removing the colour information, and the resulting dataset was labelled as "Greyscale Phase Scrambled". To illustrate the role of colour, a comparison of these two phase-scrambled features is conducted.

6.3.2 Saturation

We investigated whether saturation varies at different depths in indoor scenes. We partitioned this depth range 0-255 in the NYU dataset into eight segments and then calculated the average saturation for each by converting RGB to HSV colour space and extracting the saturation values. Figure 6.6 shows the average saturation of the NYU dataset in different depth ranges. Based on the observations, it appears that saturation may have less influence on the results for indoor scenes, different from the result for outdoor scenes shown in Figure 6.1.

Nevertheless, we intend to further investigate the extent to which this subtle difference can affect depth estimation. In addition, as mentioned above, human depth perception



(A) Saturation Feature (B) Saturation Depth

FIGURE 6.7: Saturation with Phase Scrambling of Figure 6.2

can be influenced by saturation. To assess the specific contribution of saturation, we extracted the saturation feature for experimentation independently.

To extract the features pertaining to saturation, we started by converting the RGB colour space to the HSV colour space and then extracting the saturation maps. Subsequently, these saturation maps are subjected to phase scrambling to eliminate features such as shape, texture and other visual characteristics.

As shown in

$$V \leftarrow \max(R, G, B), \tag{6.1}$$

for each pixel, the V maps are obtained by taking the maximum value among the RGB channels. Subsequently, the saturation feature is obtained based on phase scrambling from S maps, as shown in

$$S \leftarrow \begin{cases} \frac{V-\min(R,G,B)}{V} & \text{if } V \neq 0 \\ 0 & \text{otherwise} \end{cases}. \tag{6.2}$$

As depicted in Figure 6.7, Figure 6.7a illustrates the saturation feature, with Figure 6.7b displaying its corresponding depth map.

6.3.3 Local Texture

The preference for local textures over global textures stems from the fact that the extraction of global textures includes the consideration of additional factors, including shape and other features. To mitigate the influence of other factors and preserve the texture, the images were segmented into patches and shuffled, thus eliminating global information such as shapes, since this information introduces more features than just textures.

(A) Shape with Sobel (B) Shape with Canny (C) Depth

FIGURE 6.8: Shape Maps from Figure 6.2 and Corresponding Ground Truth Depth

6.3.4 Shape

The boundaries of an object define its precise outline, marking the separation between the object and its immediate environment. Edge maps are generated through the analysis of gradient variations in image pixel values and identify changes in these values. Although edge maps do not always faithfully represent real object boundaries, when dealing with a single 2D image, they offer an efficient means of simulating object shapes. This feature has been defined as 'shape' for the subsequent experiment.

As shown in Figure 6.8, we utilised the Canny operator instead of the Sobel operator because the latter will find the gradient in the x and y directions, reflecting the differential changes of pixels (Szeliski, 2010). Therefore, not only the shape feature is included when using the Sobel operator, but some texture information may also be introduced.

6.4 Experiments

As mentioned above, we considered four factors that may contribute to depth estimation: colour, saturation, local texture and shape. All of these features were trained using the baseline model, and the obtained results were analysed.

6.4.1 Data

We used the NYU dataset (Silberman et al., 2012), which serves as a widely employed dataset in computer vision, particularly for depth estimation research. Comprising images from diverse indoor scenes, it encompasses a variety of objects and furniture. The size of the NYU dataset enhances the representativeness of model training and evaluation. It is derived from 464 scenes in three cities. The resolution of the images is 640×480. 10% of the data is split as the testing set.

TABLE 6.1: Depth Estimation Performance with Different Inputs

Features	$a_1 \uparrow$	$a_2 \uparrow$	$a_3 \uparrow$	$log_{10} \downarrow$	$rel \downarrow$	$rmse \downarrow$
Original RGB Images	98.13 ± 0.0013	99.64 ± 0.0003	99.9 ± 0.0001	0.0176 ± 0.0001	0.0413 ± 0.0008	0.0174 ± 0.0002
RGB Phase Scrambled	43.5 ± 0.0067	72.64 ± 0.0068	87.47 ± 0.0042	0.1498 ± 0.0021	0.4754 ± 0.017	0.113 ± 0.0019
Grayscale Phase Scrambled	36.13 ± 0.0316	64.09 ± 0.041	81.65 ± 0.033	0.1769 ± 0.0143	0.5627 ± 0.0585	0.1364 ± 0.0149
Saturation	36.9 ± 0.0094	65.35 ± 0.0113	82.86 ± 0.0065	0.1718 ± 0.0026	0.5321 ± 0.0208	0.1296 ± 0.0015
Local Texture	49.95 ± 0.0286	77.88 ± 0.0228	90.83 ± 0.0114	0.1276 ± 0.0068	0.3187 ± 0.0215	0.1065 ± 0.0047
Shape	96.46 ± 0.0003	99.12 ± 0.0002	99.71 ± 0.0002	0.0235 ± 0.0001	0.0556 ± 0.0004	0.0224 ± 0.0001

6.4.2 Model

The UNet architecture is preferred for deep learning-based depth estimation due to its comprehensive design, adept at gathering context and integrating features across different scales (Bhat et al., 2021; Wu et al., 2023; Alhashim and Wonka, 2018; Eigen et al., 2014). This preference is rooted in UNet's feature pyramid structure and efficient reuse of features, enhancing depth estimation by capturing diverse scale information while preserving detail. Our experiments demonstrate that employing ResNet50 as the backbone is sufficient for model convergence on our dataset. Subsequently, we utilised the U-Net network with ResNet50 as the backbone in the following experiment.

Note that we did not compare SOTA models because the focus of our work was not on the performance of the models, but rather on the contribution of various features to indoor single-image depth estimation based on a stable baseline model.

6.4.3 Evaluation Metrics

The same as that in Sec. 3.2, we utilised six metrics commonly used in the field of depth estimation, which include three accuracy metrics and three error metrics. The accuracy metrics are distinguished by thresholds at 1.25, 1.25^2 and 1.25^3, each reflecting different levels of tolerance for deviation from the true values. Higher values of these accuracy metrics indicate better model performance. For error metrics, the absolute relative error (*rel*) quantifies the average deviation of predicted values from the actual values. The root mean squared error (*rmse*) can amplify the effect of outliers by taking the square root of the average of the squared deviations from the ground truth, and the logarithmic error (log_{10}) metric mitigates the impact of outliers by applying a logarithmic scale to the error values. Lower values of these error metrics signify superior model performance.

6.4.4 Experiments and Analysis

Quantitative results are shown in Table 6.1, presenting the performance of depth estimation using different input features, evaluated by several metrics (details shown in Sec.6.4.3). Original RGB images performed the best with high accuracy (a_1, a_2, a_3) and

(A) RGB Image (B) GT Depth (C) Estimated Depth

FIGURE 6.9: Depth Estimation with a Colour Feature Input. The left and middle images are the original RGB image and the corresponding ground truth depth map, respectively. The image on the right depicts the estimated depth map, which is the result of the model's output after inverse phase scrambling, employing the colour feature as the input.

low error (log_{10}, *rel*, *rmse*). Phase-scrambled RGB and greyscale images, along with saturation inputs, showed significantly worse performance, with greyscale being the least accurate. Inputs of local texture had moderate accuracy and error rates, while shape features performed close to the original RGB images.

6.4.4.1 Colour

To evaluate the contribution of the colour feature, we trained the model with phase-scrambled RGB images. Figure 6.9 displays the original RGB image, ground truth depth map and the estimated depth map, the latter of which has been reconstructed from the scrambled image using a pre-stored random matrix. As aligned to the low accuracy indicated in Table 6.1, it is hard to recognise the original scene structure from the estimated depth.

To simulate scenarios where the model output differs from the ground truth, we added Gaussian noise (mean = 0, std = 25) to the phase scrambled image. Figure 11 shows examples of our phase scrambled image with added Gaussian noise and their corresponding reconstructions. Figure 6.10 shows the outcomes of introducing Gaussian noise to the phase-scrambled image, followed by its restoration using the pre-stored random matrix. As we can see, despite the introduction of noise through phase scrambling, this noise does not affect the shape and position of objects in the recovered images. The performance in Figure 6.9c can be attributed to the poor performance of the model when provided with colour phase-scrambled input.

Furthermore, by comparing the respective performances of "RGB Phase Scrambled" and "Grayscale Phase Scrambled" inputs as shown in Table 6.1, it can be observed that, after excluding the contribution of brightness, colour has a limited impact on depth estimation.

(A) Noise for Whole Image (B) Noise in Central Region

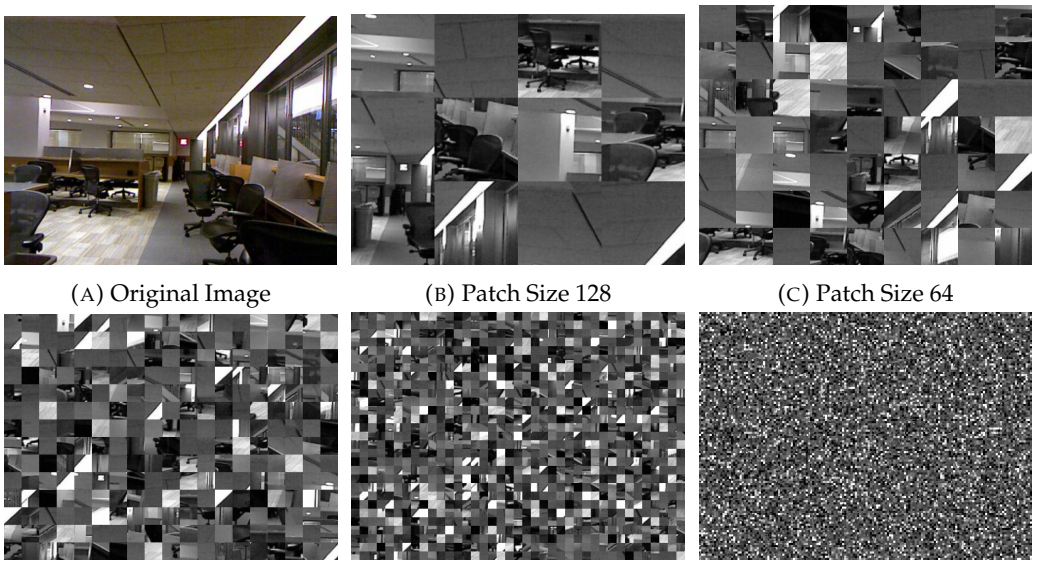
FIGURE 6.10: Noised with Phase Scrambled Images. Figure 6.10a illustrates the outcome of applying Gaussian noise to the entire image and subsequently restoring it, while Figure 6.10b depicts the results of adding noise and restoring only the central area, where both the length and width are half of the original image's dimensions.

(A) Saturation Map (B) GT Depth (C) Estimated Depth

FIGURE 6.11: Depth Estimation with a Saturation Feature Input. The left and middle images are the original RGB image and the corresponding ground truth depth map, respectively. The image on the right depicts the estimated depth map, which is the result of the model's output after inverse phase scrambling, employing a saturation map as the input.

6.4.4.2 Saturation

We trained the baseline model with saturation maps as the input and evaluated the contribution of the saturation feature. Figure 6.11 illustrates the saturation map, corresponding ground truth depth and the output from the restoration process. Similarly, due to the poor performance, the restored output lacks discernible features such as object contours.



(A) Original Image (B) Patch Size 128 (C) Patch Size 64

(D) Patch Size 32 (E) Patch Size 16 (F) Patch Size 4

FIGURE 6.12: Local Texture with Different Patch Sizes of a Random Sample

Table 6.1 shows that the a_1 is about 37%. Although saturation contributed to estimating the depth of the indoor scene, its contribution was minor. Saturation exhibits lower accuracies compared to other features except for greyscale phase scrambled input. Furthermore, error metrics substantiate this observation. The *rel* stands at 0.5321, while the root mean square error (*rmse*) is shown as 0.1296. Therefore, using saturation as a measure for depth estimation clearly introduces a significant error compared to the true depth values. Despite its poor performance, saturation still plays a role in assessing depth in indoor scenes. This highlights that saturation can provide some depth cues in certain contexts, although it comes with a higher error margin.

6.4.4.3 Local Texture

Variations in the field of view and resolution will impact the size of the patch used to extract local textures. The optimal patch size should align with the specific dataset and scene scale employed. Figure 6.12 illustrates the paths with varying patch dimensions. As shown in Figure 6.12b, when we use a large patch size of 128, the texture itself is not isolated because the shape and context information still present in the patches. As the patch size is decreased, the shape of the objects in the image becomes less apparent and, therefore, the textures present in the image are increasingly segregated. As demonstrated in Figure 6.12e, for the dataset we used, the 16×16 patch size is particularly well-suited for texture extraction while minimising the influence of other features (e.g. shape). This size is large enough to restrict shape details but not so small as to be impractical, unlike the 4×4 patch depicted in Figure 6.12f.

TABLE 6.2: Performance with Different Patch Sizes

Size	$a_1 \uparrow$	$a_2 \uparrow$	$a_3 \uparrow$	$log_{10} \downarrow$	$rel \downarrow$	$rmse \downarrow$
4	40.97 ± 0.0037	69.6 ± 0.003	85.99 ± 0.0018	0.1523 ± 0.0008	0.3812 ± 0.0026	0.1239 ± 0.0007
16	49.95 ± 0.0286	77.88 ± 0.0228	90.83 ± 0.0114	0.1276 ± 0.0068	0.3187 ± 0.0215	0.1065 ± 0.0047
32	53.27 ± 0.042	80.64 ± 0.0253	92.46 ± 0.0104	0.1185 ± 0.0087	0.3012 ± 0.0266	0.1009 ± 0.0094
64	74.22 ± 0.0166	92.48 ± 0.0122	97.51 ± 0.0057	0.0747 ± 0.0039	0.1885 ± 0.0199	0.0629 ± 0.002
128	93.12 ± 0.0066	98.47 ± 0.0019	99.5 ± 0.0007	0.0358 ± 0.0016	0.0863 ± 0.0039	0.0338 ± 0.0015

(A) Gray-scale Image (B) GT Depth (C) Estimated Depth

FIGURE 6.13: Depth Estimation with a Local Texture Input with Patch Size 16×16. The left and middle images are the original RGB image and the corresponding ground truth depth map, respectively. The image on the right depicts the estimated depth map, which is the result of the model's output after inverse shuffle by using a pre-stored random matrix, employing a local texture feature as input.

Table 6.2 shows the performance of texture inputs (shuffle patches) in different patch sizes. As can be seen in the figure, the accuracy rate gradually increases with the increase of patch sizes in height. This is because a larger patch contains more information besides the texture, such as the shape of the object.

A sample of the original greyscale image, along with the corresponding depth map and the estimated depth map, is displayed in Figure 6.13, demonstrating the results of the model trained with local texture inputs. To focus on local textures during training, Figure 6.13a and Figure 6.13b are split into 16×16 patches and these patches are shuffled using the same random matrix to eliminate global scene information, such as object shapes. As shown in Figure 6.13c, the estimated depth map only provides a coarse approximation of the scene's depth, distinguishing between nearer and farther areas but failing to capture the precise depth details.

The local texture appears as a minor factor in depth estimation, yielding an a_1 accuracy of a mere 50% in Table 6.1. Error metrics also show this trend although they are slightly better than the colour and saturation features. This issue happens because the position changes to the patches weaken their connections, thereby making it harder for the model to understand objects and context. This proposition finds corroboration in the robust performance observed upon deploying the shape feature as the input data source in Sec 6.4.4.4.

(A) Shape Map (B) GT Depth (C) Estimated Depth

FIGURE 6.14: Depth Estimation with a Shape Feature Input. The left and middle images are the original RGB image and corresponding ground truth depth map. The right is the estimated depth map from the model trained with shape maps.

6.4.4.4 Shape

As we noted in Table 6.1, the shape comes across as the most dominant feature in these experiments, significantly outperforming other cues taken in isolation. We suggest this is because the dataset contains indoor scenes of objects such as furniture with accurately extractable edges whose relative orientations and geometric forms can serve as powerful cues as seen in Figure 6.14 (b and c).

The outcome aligns with the finding presented in (Hu et al., 2019), suggesting that CNNs are capable of deducing the depth map using merely a limited subset of pixels from the input image. This hypothesis aligns with human perceptual abilities, which allow for the extraction of approximate distance assessments from images that depict geometric shapes.

6.4.4.5 Generalisation

In light of the fact that models using shape maps as input exhibit performance approximating that of models employing original RGB images as the input, we have assessed the generalisation capacity of shape models trained with shape maps on the NYU dataset. We applied it to a diverse set of indoor environments from a different dataset (Quattoni and Torralba, 2009) that includes kitchens, bedrooms, bathrooms and various other scenes. The performance of the shape model is depicted in Figure 6.15, illustrating its ability to predict depth maps even for scenes from a different domain, and the performance is similar to that of the original RGB model. However, shape maps, as input for depth estimation, still have their limitations. For instance, in the fourth-row images in Figure 6.15, the sink only has partial edges, leading to poor depth prediction. Additional results are presented in the Appendix.

Shape maps require significantly less memory storage compared to original RGB images, while still providing comparable performance. In a similar vein, event cameras

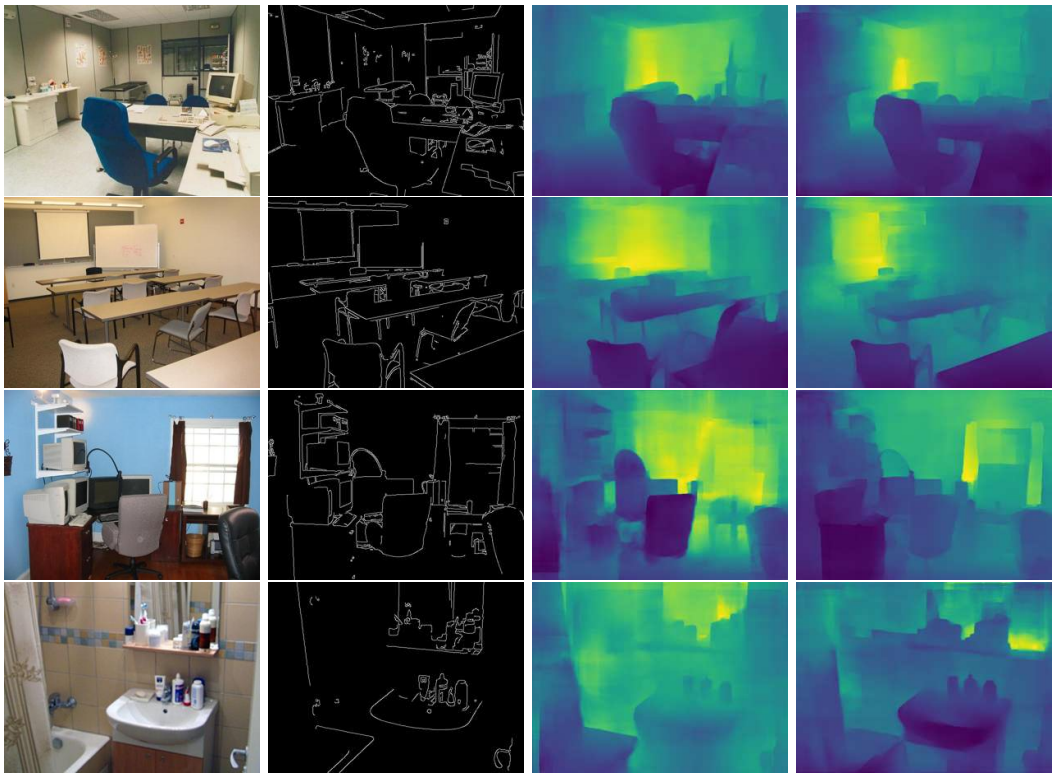


FIGURE 6.15: Performance of Shape ONLY Model with New Indoor Scenes from other Domains. The left column displays original RGB scene images, the second column presents corresponding edge maps and the third column showcases the results generated by the pre-trained shape-input model. The right column exhibits the outcomes produced by the pre-trained original-RGB-image-input model.

are designed to only detect rapid changes in pixel intensity (Rebecq et al., 2018; Scheerlinck et al., 2020) which often occur at the edges of objects or where there is texture variation, which is similar to a shape map. Moreover, event cameras have previously been applied in the field of depth estimation (Gallego et al., 2020). Given these considerations, our research may offer supporting evidence for the application of event cameras in single-image depth estimation.

In addition to this, our work will also contribute to 3D sketch reconstruction. A sketch input is a hand-drawn image that is primarily used to capture the basic shape and key features of an object (Lun et al., 2017; Wang et al., 2020a). It shows the overall outline and main features of the object, and this contour information is linked to the concept of 'shape' discussed in our work. For example, Lun et al. (2017) convert sketches into multi-view 2D images that capture the surface depth and normals, then fuse these into a 3D point cloud and eventually convert them into a polygonal mesh, and finally generate detailed 3D models from simple line sketches.

Therefore, our study may also provide supporting evidence for 3D sketch reconstruction.

6.4.4.6 Discussion

Different types of input data have varying effects on the performance of depth estimation. Comparative analysis of diverse evaluation metrics clearly highlights the superior role of shape information in the depth estimation task. Colour, saturation and local texture collectively enhance the indoor scene depth estimation, although the influence of colour and saturation appears relatively circumscribed.

For phase-scrambled and local texture inputs, human vision finds it difficult to interpret images when their phase information is scrambled or only shuffled patches are present. In contrast, machines are adept at using these inputs to predict depth maps. Given that the models can output corresponding depth maps when employing these as inputs, their performance, albeit not optimal, is still noteworthy compared to the ability of humans.

6.5 Limitations

Throughout our study, we sought to isolate each feature we were evaluating. However, it is difficult to entirely isolate individual features. For instance, during the extraction of shape features, the edge detector might inadvertently capture some texture information.

6.6 Conclusion

In this work, we have sought to decouple and quantify the relative contributions of various depth cues in single-image depth estimation. Whereas good results have been demonstrated in the literature by the end-to-end training of deep neural network models to achieve this task, ours is the first attempt to understand the degree to which some known cues of depth contribute when taken in isolation. Our results show that, in a dataset of indoor scenes, shape extracted by edge detection is relatively the most significant contributor, while other cues (colour, saturation and texture) also play a role. In achieving these conclusions, this work sought to carefully design feature extraction techniques that aimed to isolate a single feature from the other known ones, which is non-trivial. We speculate (and this is the subject of our current research) that, on different depth inference problems (e.g. outdoor scenes), the relative contributions of texture and saturation are likely to play a greater role. This kind of decomposition which we have extracted can serve to shift research more in the direction of understanding and explaining how powerful models, such as deep neural networks, work in scene understanding as opposed to simply offering estimation performance as black-box function approximators.

Chapter 7

Conclusions and Future Work

This section presents the conclusions of my PhD research (sections 7.1 to 7.4) and suggests possible avenues for further research.

In this study, four contributions to this field are introduced. It mainly focuses on depth estimation for deep-learning-based indoor single omnidirectional images. By addressing the challenges of limited real-world labelling and the utilisation of the synthetic dataset, the model can be applied to real-world scenes without labels. In the in-depth study of contributing factors to depth estimation, the physical limitation of gravity is considered and analysed, and based on this, a model based on gravity alignment is proposed. Further, isolated depth clues are studied and analysed to gain a deeper understanding of their contributions to explore the insight of depth estimates.

7.1 Contribution A: Depth Estimation with Limited Real-world Labels

Due to the insufficient types of labelled datasets and the difficulty of obtaining real-world depth maps, existing encoder-decoder models trained with another dataset are usually unable to accurately predict depth maps for real-world scenes in the target domain. To solve the problem of limited omnidirectional depth maps for real-world scenes, the depth estimation architecture with domain adaptation is proposed to predict scene depth for unlabelled omnidirectional images with only limited real-world ground truth depth maps. The experiments show that the performance of domain adaptation architecture outperforms the traditional end-to-end model for omnidirectional depth estimation in the situation of a limited number and variety of data.

7.2 Contribution B: Real-world Depth Estimation from a Synthetic Dataset

CG datasets can be used for real-world depth map estimation as they cover various types of scenes and are easier to obtain compared with real-world scenes. A method that uses CG scenes and does not use any real-world depth maps was considered for training. Because of the gap between CG images and real-world images, the previous contribution method does not perform well, as it only considers making the discriminator not recognise the features come from which domain and finally crashes the training process. To solve this problem, a discriminator named RWTD is proposed with an updated architecture containing different components compared with the previous method. It shows significantly better stability and about 11% points higher accuracy than state-of-the-art encoder-decoder models. This result means that the work provides an effective solution for depth estimation learning from CG scenes and can be applied to real-world scenes.

7.3 Contribution C: Depth Estimation considering Gravity

In this research, the role of gravity in predicting depth maps has been examined. The results show that, under the influence of gravity, the depth distribution of the object will show a certain rule in the gravity direction, and the input image information along the gravity direction shows better performance in depth estimation than the horizontal information. By accounting for the orientation of gravity, we introduced a novel framework for accurately estimating depth from a single omnidirectional indoor image, through an approach that leverages slice-based transformers. The proposed framework has undergone thorough validation on two real-world, indoor omnidirectional datasets, where it has proven to outperform existing leading-edge methods. A key feature of this methodology is its reliance on the direction of gravity to infer depth, aiming to enrich the ongoing conversation around depth estimation from single images and potentially introduce novel perspectives for assessing depth in indoor settings.

7.4 Contribution D: Depth Insight

In this study, the objective was to separate and measure the individual effects of various depth cues on the estimation of depth from a single-view perspective. While previous research has successfully utilised deep learning models trained end-to-end to accomplish this task, our research represents the initial effort to dissect the extent to which certain recognised depth indicators contribute independently. Our findings indicate

that, within a dataset composed of indoor environments, the shape information obtained through edge detection stands out as the most impactful factor, although other elements (such as colour, saturation, and texture) also have their importance. To arrive at these insights, we developed methods for feature extraction that were designed to isolate one particular feature from the others. This work could pave the way for future research to focus more on elucidating and interpreting the mechanisms behind sophisticated models like deep neural networks in the context of scene comprehension rather than solely evaluating their performance as black-box function approximations.

7.5 Future Works

In this section, four potential paths for future investigation are suggested, offering prospective researchers valuable guidance. These encompass:

- **Depth Range:** Although the proposed methods can solve the real-world depth estimation problem on the basis of synthetic images, the varying range of depths for different scenarios remains a challenging problem. Future work can focus on solving problems of different depth ranges in different scenes.
- **Extend to Scene Understanding:** The importance of gravity in artificial scenes has been highlighted in Chapter 5. Future research directions can be further investigated on this basis: How exactly does gravity affect depth estimation? This can be combined with scene understanding for further study.
- **Extend to Comparison with Outdoor Scenes:** Although my PhD research is about indoor scenes, comparing the depth cues of indoor scenes to their performance in outdoor scenes could be an interesting and valuable direction.
- **Explore Causality:** The contribution of isolated features to indoor single image depth estimation is analysed in Chapter 6. Whether there is a causal relationship between these features is a worthy research direction.

7.5.1 Depth Range

Random object scaling (ROS) (Yang et al., 2021), as a 3D object enhancement technique, can improve the generalisation ability of the 3D detection model in the target domain by randomly scaling 3D objects to reduce the bias of the source domain. Motivated by it, one possible solution is to randomly shrink or enlarge the size of the room layout so that the model can generalize scenes of different depths (Shown in Figure 7.1).

However, direct scaling solves the problem of the room, but the objects in the room may have the wrong sizes. For example, the chair in the first scene in Figure 7.2 should

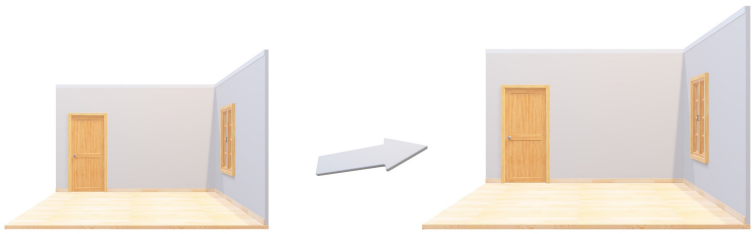


FIGURE 7.1: Depth Range for Considering Different Layout Sizes

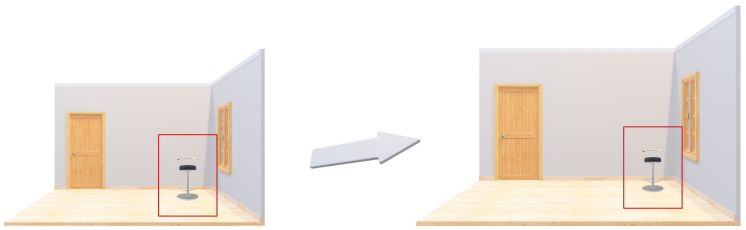


FIGURE 7.2: Depth Range: Consider Object Sizes

remain the same size as the chair in the second scene after being shrunk instead of being enlarged with the scene. Based on this problem, a possible solution is to add datasets of different scene sizes to the existing synthetic dataset and use the 3D engine to generate scenes with different layouts to improve the generalization of the model.

7.5.2 Extend to Scene Understanding

The supporting relation under the action of gravity can be used as effective prior knowledge, which can assist the depth estimation model to analyse the scene structure better. Figure 7.3 shows an example of the supporting relationship between the teacup, laptop and desk. When training a depth estimation model, using the concept of object support as a constraint or regularization can guide the model to learn depth information that is more consistent with the physical world. Moreover, it can infer the relative depth of an object by learning to recognise different support planes. This prior knowledge can improve the prediction accuracy of the relative position and depth relationship between objects. A feasible approach is to combine semantically segmented datasets with the same scenes.

7.5.3 Extend to Comparison with Outdoor Scenes

As mentioned in Chapter 6, saturation plays an obvious role in outdoor natural scenery scenes, while its role in indoor scenes is limited. Do other features like colour and texture make a different contribution to outdoor scenes?

FIGURE 7.3: Supported Object Example

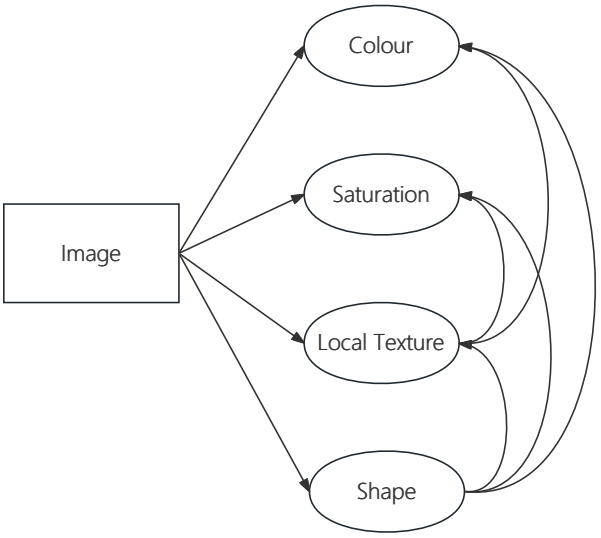


FIGURE 7.4: Causality

For this outdoor scenario, further research could consider using kITTI (Geiger et al., 2013), Waymo (Sun et al., 2020), and nuScenes (Caesar et al., 2020) for experimentation and validation. Although these datasets are mainly about autonomous driving, they also include a lot of outdoor scenes like grass, trees, and houses.

7.5.4 Explore Causality

Chapter 6 analyses the contribution of isolated features to depth estimation in different degrees. Exploring the dependencies and causality (shown in Figure 7.4) among different features can make the model better adapt to new scenarios and data, instead of just fitting the training data in a data-driven mode. Considering causality may help optimise the performance of a model, especially in cases involving complex data distribution and multimodal problems.

References

Suhaila FA Abuowaida and Huah Yong Chan. Improved deep learning architecture for depth estimation from single image. *Jordanian Journal of Computers and Information Technology (JJCIT)*, 6(04):434–445, 2020.

Mona Alawadh, Yihong Wu, Yuwen Heng, Luca Remaggi, Mahesan Niranjan, and Hansung Kim. Room acoustic properties estimation from a single 360° photo. In *2022 30th European Signal Processing Conference (EUSIPCO)*, pages 857–861. IEEE, 2022.

Georgios Albanis, Nikolaos Zioulis, Petros Drakoulis, Vasileios Gkitsas, Vladimiros Sterzentsenko, Federico Alvarez, Dimitrios Zarpalas, and Petros Daras. Pano3d: A holistic benchmark and a solid baseline for 360deg depth estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3727–3737, 2021.

Ibraheem Alhashim and Peter Wonka. High quality monocular depth estimation via transfer learning. *arXiv preprint arXiv:1812.11941*, 2018.

PJA Alphonse and KV Sriharsha. Depth estimation from a single rgb image using target foreground and background scene variations. *Computers & Electrical Engineering*, 94: 107349, 2021.

I. Armeni, A. Sax, A. R. Zamir, and S. Savarese. Joint 2D-3D-Semantic Data for Indoor Scene Understanding. *ArXiv e-prints*, February 2017.

Stephen T Barnard and Martin A Fischler. Computational stereo. *ACM Computing Surveys (CSUR)*, 14(4):553–572, 1982.

Shariq Farooq Bhat, Ibraheem Alhashim, and Peter Wonka. Adabins: Depth estimation using adaptive bins. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4009–4018, 2021.

Amlaan Bhoi. Monocular depth estimation: A survey. *arXiv preprint arXiv:1901.09402*, 2019.

Robert C Bolles. The jisct stereo evaluation. In *Proc. of Image Understanding Workshop, 1993*, 1993.

Myron Z Brown, Darius Burschka, and Gregory D Hager. Advances in computational stereo. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(8):993–1008, 2003.

Holger Caesar, Varun Bankiti, Alex H Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multimodal dataset for autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11621–11631, 2020.

Angel Chang, Angela Dai, Thomas Funkhouser, Maciej Halber, Matthias Niessner, Manolis Savva, Shuran Song, Andy Zeng, and Yinda Zhang. Matterport3d: Learning from rgb-d data in indoor environments. *arXiv preprint arXiv:1709.06158*, 2017.

Jia-Ren Chang and Yong-Sheng Chen. Pyramid stereo matching network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5410–5418, 2018.

Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*, 2014.

RTH Collis. Lidar. *Applied optics*, 9(8):1782–1788, 1970.

Corinna Cortes, Mehryar Mohri, and Afshin Rostamizadeh. Algorithms for learning kernels based on centered alignment. *The Journal of Machine Learning Research*, 13(1): 795–828, 2012.

Gabriela Csurka. Domain adaptation for visual applications: A comprehensive survey. *arXiv preprint arXiv:1702.05374*, 2017.

James E Cutting and Peter M Vishton. Perceiving layout and knowing distances: The integration, relative potency, and contextual use of different information about depth. In *Perception of space and motion*, pages 69–117. Elsevier, 1995.

Greire Payen de La Garanderie, Amir Atapour Abarghouei, and Toby P Breckon. Eliminating the blind spot: Adapting 3d object detection and monocular depth estimation to 360 panoramic imagery. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 789–807, 2018.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255. Ieee, 2009.

Anthony Dickson, Alistair Knott, and Stefanie Zollmann. Benchmarking monocular depth estimation models for vr content creation from a user perspective. In *2021 36th*

International Conference on Image and Vision Computing New Zealand (IVCNZ), pages 1–6. IEEE, 2021.

Tom van Dijk and Guido de Croon. How do neural networks see depth in single images? In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2183–2191, 2019.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

David Eigen, Christian Puhrsch, and Rob Fergus. Depth map prediction from a single image using a multi-scale deep network. *Advances in Neural Information Processing Systems*, 27, 2014.

Fatima El Jamiy and Ronald Marsh. Distance estimation in virtual reality and augmented reality: A survey. In *2019 IEEE International Conference on Electro Information Technology (EIT)*, pages 063–068. IEEE, 2019.

Abolfazl Farahani, Sahar Voghoei, Khaled Rasheed, and Hamid R Arabnia. A brief review of domain adaptation. *Advances in Data Science and Information Engineering: Proceedings from ICDATA 2020 and IKE 2020*, pages 877–894, 2021.

Muhammad Shahid Farid, Maurizio Lucenteforte, and Marco Grangetto. Edges shape enforcement for visual enhancement of depth image based rendering. In *2013 IEEE 15th International Workshop on Multimedia Signal Processing (MMSP)*, pages 406–411. IEEE, 2013.

Huan Fu, Mingming Gong, Chaohui Wang, Kayhan Batmanghelich, and Dacheng Tao. Deep ordinal regression network for monocular depth estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2002–2011, 2018.

Guillermo Gallego, Tobi Delbrück, Garrick Orchard, Chiara Bartolozzi, Brian Taba, Andrea Censi, Stefan Leutenegger, Andrew J Davison, Jörg Conradt, Kostas Daniilidis, et al. Event-based vision: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(1):154–180, 2020.

Yaroslav Ganin and Victor Lempitsky. Unsupervised domain adaptation by backpropagation. In *International Conference on Machine Learning*, pages 1180–1189. PMLR, 2015.

Yunhao Ge, Yao Xiao, Zhi Xu, Xingrui Wang, and Laurent Itti. Contributions of shape, texture, and color in visual recognition. In *European Conference on Computer Vision*, pages 369–386. Springer, 2022.

Karl R Gegenfurtner and Jochem Rieger. Sensory and cognitive contributions of color to the recognition of natural scenes. *Current Biology*, 10(13):805–808, 2000.

Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. *The International Journal of Robotics Research*, 32(11):1231–1237, 2013.

Robert Geirhos, Patricia Rubisch, Claudio Michaelis, Matthias Bethge, Felix A Wichmann, and Wieland Brendel. Imagenet-trained cnns are biased towards texture; increasing shape bias improves accuracy and robustness. *arXiv preprint arXiv:1811.12231*, 2018.

James J Gibson. *The perception of the visual world.* Houghton Mifflin, 1950.

Xavier Glorot, Antoine Bordes, and Yoshua Bengio. Domain adaptation for large-scale sentiment classification: A deep learning approach. In *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, pages 513–520, 2011.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 27. Curran Associates, Inc., 2014. URL https://proceedings.neurips.cc/paper/2014/file/5ca3e9b122f61f8f06494c97b1afccf3-Paper.pdf.

Andrzej Grzybowski and Konrad Kupidura-Majewski. What is color and how it is perceived? *Clinics in dermatology*, 37(5):392–401, 2019.

Hyowon Ha, Sunghoon Im, Jaesik Park, Hae-Gon Jeon, and In So Kweon. High-quality depth from uncalibrated small motion clip. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5413–5421, 2016.

Praful Hambarde and Subrahmanyam Murala. S2dnet: Depth estimation from single image and sparse samples. *IEEE Transactions on Computational Imaging*, 6:806–817, 2020.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016.

Yuwen Heng, Yihong Wu, Srinandan Dasmahapatra, and Hansung Kim. Cam-segnet: A context-aware dense material segmentation network for sparsely labelled datasets. In *Proceedings of the 17th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP 2022)*, pages 190–201, 2022a.

Yuwen Heng, Yihong Wu, Srinandan Dasmahapatra, and Hansung Kim. Enhancing material features using dynamic backward attention on cross-resolution patches. In *British Machine Vision Conference (BMVC)*, page 4, 2022b.

Ian P Howard. *Seeing in depth, Vol. 1: Basic mechanisms.* University of Toronto Press, 2002.

Ian P Howard. *Perceiving in depth, volume 1: basic mechanisms*. Oxford University Press, 2012.

Yuan C Hsieh, David M McKeown, and Frederic P Perlant. Performance evaluation of scene registration and stereo matching for artographic feature extraction. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 14(02):214–238, 1992.

Junjie Hu, Yan Zhang, and Takayuki Okatani. Visualization of convolutional neural networks for monocular depth estimation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3869–3878, 2019.

Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4700–4708, 2017.

Po-Han Huang, Kevin Matzen, Johannes Kopf, Narendra Ahuja, and Jia-Bin Huang. Deepmvs: Learning multi-view stereopsis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2821–2830, 2018.

Joel Janai, Fatma Güney, Aseem Behl, Andreas Geiger, et al. Computer vision for autonomous vehicles: Problems, datasets and state of the art. *Foundations and Trends® in Computer Graphics and Vision*, 12(1–3):1–308, 2020.

Huaizu Jiang, Deqing Sun, Varun Jampani, Zhaoyang Lv, Erik Learned-Miller, and Jan Kautz. Sense: A shared encoder network for scene-flow estimation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3195–3204, 2019.

Lei Jin, Yanyu Xu, Jia Zheng, Junfei Zhang, Rui Tang, Shugong Xu, Jingyi Yu, and Shenghua Gao. Geometric structure based and regularized depth estimation from 360 indoor imagery. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 889–898, 2020.

Jinwoong Jung, Beomseok Kim, Joon-Young Lee, Byungmoon Kim, and Seungyong Lee. Robust upright adjustment of 360 spherical panoramas. *The Visual Computer*, 33: 737–747, 2017.

Mun-Cheon Kang, Kwang-Shik Kim, Dong-Ki Noh, Jong-Woo Han, and Sung-Jea Ko. A robust obstacle detection method for robotic vacuum cleaners. *IEEE Transactions on Consumer Electronics*, 60(4):587–595, 2014.

DH Kelly. Visual contrast sensitivity. *Optica Acta: International Journal of Optics*, 24(2): 107–129, 1977.

Alex Kendall, Hayk Martirosyan, Saumitro Dasgupta, Peter Henry, Ryan Kennedy, Abraham Bachrach, and Adam Bry. End-to-end learning of geometry and context for deep stereo regression. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 66–75, 2017.

Hansung Kim and Adrian Hilton. 3d scene reconstruction from multiple spherical stereo pairs. *International Journal of Computer Vision*, 104(1):94–116, 2013.

Hansung Kim, Luca Remaggi, Sam Fowler, Philip Jackson, and Adrian Hilton. Acoustic room modelling using 360 stereo cameras. *IEEE Transactions on Multimedia*, 2020.

Hansung Kim, Luca Remaggi, Aloisio Dourado, Teofilo de Campos, Philip JB Jackson, and Adrian Hilton. Immersive audio-visual scene reproduction using semantic scene reconstruction from 360 cameras. *Virtual Reality*, pages 1–16, 2022.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Patrick Knobelreiter, Christian Reinbacher, Alexander Shekhovtsov, and Thomas Pock. End-to-end training of hybrid cnn-crf models for stereo. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2339–2348, 2017.

Simon Kornblith, Mohammad Norouzi, Honglak Lee, and Geoffrey Hinton. Similarity of neural network representations revisited. In *International Conference on Machine Learning*, pages 3519–3529. PMLR, 2019.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in Neural Information Processing Systems*, 25, 2012.

Aleksander Lamża, Zygmunt Wróbel, and Andrzej Dziech. Depth estimation in image sequences in single-camera video surveillance systems. In *International Conference on Multimedia Communications, Services and Security*, pages 121–129. Springer, 2013.

Michael S Landy, Laurence T Maloney, Elizabeth B Johnston, and Mark Young. Measurement and modeling of depth cue combination: in defense of weak fusion. *Vision Research*, 35(3):389–412, 1995.

M. Langer and H. Bülthoff. Depth discrimination from shading under diffuse lighting. *Perception*, 29:649 – 660, 2000. .

Pierre Lebreton, Alexander Raake, Marcus Barkowsky, and Patrick Le Callet. Measuring perceived depth in natural images and study of its relation with monocular and binocular depth cues. In *Stereoscopic Displays and Applications XXV*, volume 9011, pages 82–92. SPIE, 2014.

Wonwoo Lee, Nohyoung Park, and Woontack Woo. Depth-assisted real-time 3d object detection for augmented reality. In *ICAT*, volume 11, pages 126–132, 2011.

Jianjun Lei, Jianying Liu, Hailong Zhang, Zhouye Gu, Nam Ling, and Chunping Hou. Motion and structure information based adaptive weighted depth video estimation. *IEEE Transactions on Broadcasting*, 61(3):416–424, 2015. .

Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2980–2988, 2017.

Fei Liu, Shubo Zhou, Yunlong Wang, Guangqi Hou, Zhenan Sun, and Tieniu Tan. Binocular light-field: Imaging theory and occlusion-robust depth perception application. *IEEE Transactions on Image Processing*, 29:1628–1640, 2019.

Weiyang Liu, Zhen Liu, Liam Paull, Adrian Weller, and Bernhard Schölkopf. Structural causal 3d reconstruction. In *European Conference on Computer Vision*, pages 140–159. Springer, 2022.

Mingsheng Long, Yue Cao, Jianmin Wang, and Michael Jordan. Learning transferable features with deep adaptation networks. In *International Conference on Machine Learning*, pages 97–105. PMLR, 2015.

Zhaoliang Lun, Matheus Gadelha, Evangelos Kalogerakis, Subhransu Maji, and Rui Wang. 3d shape reconstruction from sketches via multi-view convolutional networks. In *2017 International Conference on 3D Vision (3DV)*, pages 67–77. IEEE, 2017.

Yue Luo, Jimmy Ren, Mude Lin, Jiahao Pang, Wenxiu Sun, Hongsheng Li, and Liang Lin. Single view stereo matching. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 155–163, 2018.

Zakariae Machkour, Daniel Ortiz-Arroyo, and Petar Durdevic. Monocular based navigation system for autonomous ground robots using multiple deep learning models. *International Journal of Computational Intelligence Systems*, 16(1):79, 2023.

Jonathan A Marshall, Christina A Burbeck, Dan Ariely, Jannick P Rolland, and Kevin E Martin. Occlusion edge blur: a cue to relative visual depth. *JOSA A*, 13(4):681–688, 1996.

Nikolaus Mayer, Eddy Ilg, Philip Hausser, Philipp Fischer, Daniel Cremers, Alexey Dosovitskiy, and Thomas Brox. A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4040–4048, 2016.

Larry R Medsker and LC Jain. Recurrent neural networks. *Design and Applications*, 5 (64-67):2, 2001.

Alican Mertan, Damien Jade Duff, and Gozde Unal. Single image depth estimation: An overview. *Digital Signal Processing*, page 103441, 2022.

Wided Miled, Jean-Christophe Pesquet, and Michel Parent. A convex optimization approach for depth estimation under illumination variation. *IEEE Transactions on Image Processing*, 18(4):813–830, 2009.

Yue Ming, Xuyang Meng, Chunxiao Fan, and Hui Yu. Deep learning for monocular depth estimation: A review. *Neurocomputing*, 438:14–33, 2021.

R. Mulajkar and V. Gohokar. Development of methodology for extraction of depth for 2d-to-3d conversion. *2017 Second International Conference on Electrical, Computer and Communication Technologies (ICECCT)*, pages 1–5, 2017. .

Maureen Neitz and Jay Neitz. Molecular genetics of color vision and color vision defects. *Archives of ophthalmology*, 118(5):691–700, 2000.

Zhenxing Niu, Mo Zhou, Le Wang, Xinbo Gao, and Gang Hua. Ordinal regression with multiple output cnn for age estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4920–4928, 2016.

Yuichi Ohta and Takeo Kanade. Stereo by intra-and inter-scanline search using dynamic programming. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 139–154, 1985.

J Pan, L Li, H Yamaguchi, K Hasegawa, FI Thufail, Brahmantara, and S Tanaka. Fused 3d transparent visualization for large-scale cultural heritage using deep learning-based monocular reconstruction. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 2:989–996, 2020.

Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, 22(10):1345–1359, 2009.

Giovanni Pintore, Marco Agus, Eva Almansa, Jens Schneider, and Enrico Gobbetti. Slicenet: deep dense depth estimation from a single indoor panorama using a slice-based representation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11536–11545, 2021.

Matteo Poggi, Fabio Tosi, Konstantinos Batsos, Philippos Mordohai, and Stefano Mattoccia. On the synergies between machine learning and binocular stereo for depth estimation from images: a survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(9):5314–5334, 2021.

Aina Puce, Truett Allison, Maryam Asgari, John C Gore, and Gregory McCarthy. Differential sensitivity of human visual cortex to faces, letterstrings, and textures: a functional magnetic resonance imaging study. *Journal of Neuroscience*, 16(16):5205–5215, 1996.

Ariadna Quattoni and Antonio Torralba. Recognizing indoor scenes. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 413–420. IEEE, 2009.

Maithra Raghu, Thomas Unterthiner, Simon Kornblith, Chiyuan Zhang, and Alexey Dosovitskiy. Do vision transformers see like convolutional neural networks? *Advances in Neural Information Processing Systems*, 34:12116–12128, 2021.

AN Rajagopalan, Subhasis Chaudhuri, and Uma Mudenagudi. Depth estimation and image restoration using defocused stereo pairs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(11):1521–1525, 2004.

Henri Rebecq, Daniel Gehrig, and Davide Scaramuzza. Esim: an open event camera simulator. In *Conference on robot learning*, pages 969–982. PMLR, 2018.

Stephan Reichelt, Ralf Häussler, Gerald Fütterer, and Norbert Leister. Depth cues in human visual perception and their realization in 3d displays. In *Three-Dimensional Imaging, Visualization, and Display 2010 and Display Technologies and Applications for Defense, Security, and Avionics IV*, volume 7690, pages 92–103. SpIE, 2010.

Zhongzheng Ren and Yong Jae Lee. Cross-domain self-supervised multi-task feature learning using synthetic imagery. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 762–771, 2018.

James H Rowland. *The effects of texture on distance estimation in synthetic environments*. PhD thesis, Monterey, California; Naval Postgraduate School, 1999.

Kuniaki Saito, Yoshitaka Ushiku, Tatsuya Harada, and Kate Saenko. Strong-weak distribution alignment for adaptive object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6956–6965, 2019.

Irawati Nurmala Sari, Weiwei Du, et al. Depth map estimation of single-view image using smartphone camera for a 3-dimension image generation in augmented reality. In *2023 Sixth International Symposium on Computer, Consumer and Control (IS3C)*, pages 167–170. IEEE, 2023.

Ashutosh Saxena, Jamie Schulte, Andrew Y Ng, et al. Depth estimation using monocular and stereo cues. In *IJCAI*, volume 7, pages 2197–2203, 2007.

Daniel Scharstein and Richard Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *International Journal of Computer Vision*, 47: 7–42, 2002.

Cedric Scheerlinck, Henri Rebecq, Daniel Gehrig, Nick Barnes, Robert Mahony, and Davide Scaramuzza. Fast image reconstruction with an event camera. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 156–163, 2020.

Mike Schuster and Kuldip K Paliwal. Bidirectional recurrent neural networks. *IEEE transactions on Signal Processing*, 45(11):2673–2681, 1997.

Steven M Seitz, Brian Curless, James Diebel, Daniel Scharstein, and Richard Szeliski. A comparison and evaluation of multi-view stereo reconstruction algorithms. In *2006 IEEE computer society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, volume 1, pages 519–528. IEEE, 2006.

KC Shahira, Sagar Tripathy, and A Lijiya. Obstacle detection, depth estimation and warning system for visually impaired people. In *TENCON 2019-2019 IEEE Region 10 Conference (TENCON)*, pages 863–868. IEEE, 2019.

Sima Siami-Namini, Neda Tavakoli, and Akbar Siami Namin. The performance of lstm and bilstm in forecasting time series. In *2019 IEEE International Conference on Big Data (Big Data)*, pages 3285–3292. IEEE, 2019.

Nathan Silberman, Derek Hoiem, Pushmeet Kohli, and Rob Fergus. Indoor segmentation and support inference from rgbd images. In *Computer Vision–ECCV 2012: 12th European Conference on Computer Vision, Florence, Italy, October 7-13, 2012, Proceedings, Part V 12*, pages 746–760. Springer, 2012.

Jan Smisek, Michal Jancosek, and Tomas Pajdla. 3d with kinect. *Consumer Depth Cameras for Computer Vision: Research Topics and Applications*, pages 3–25, 2013.

Carsten Steger, Markus Ulrich, and Christian Wiedemann. *Machine vision algorithms and applications*. John Wiley & Sons, 2018.

Cheng Sun, Chi-Wei Hsiao, Min Sun, and Hwann-Tzong Chen. Horizonnet: Learning room layout with 1d representation and pano stretch data augmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1047–1056, 2019.

Cheng Sun, Min Sun, and Hwann-Tzong Chen. Hohonet: 360 indoor holistic understanding with latent horizontal features. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2573–2582, 2021.

Pei Sun, Henrik Kretzschmar, Xerxes Dotiwalla, Aurelien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou, Yuning Chai, Benjamin Caine, et al. Scalability in perception for autonomous driving: Waymo open dataset. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2446–2454, 2020.

Cassandra T Swain. Integration of monocular cues to create depth effect. In *1997 IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 4, pages 2745–2748. IEEE, 1997.

Richard Szeliski. *Computer vision: algorithms and applications*. Springer Science & Business Media, 2010.

Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International Conference on Machine Learning*, pages 6105–6114. PMLR, 2019.

Antonio Torralba and Aude Oliva. Depth estimation from image structure. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(9):1226–1238, 2002.

Mihran Tuceryan and Anil K Jain. Texture analysis. *Handbook of Pattern Recognition and Computer Vision*, pages 235–276, 1993.

Benjamin Ummenhofer, Huizhong Zhou, Jonas Uhrig, Nikolaus Mayer, Eddy Ilg, Alexey Dosovitskiy, and Thomas Brox. Demon: Depth and motion network for learning monocular stereo. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5038–5047, 2017.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in Neural Information Processing Systems*, 30, 2017.

Ulla Wandinger. Introduction to lidar. In *Lidar: Range-resolved Optical Remote Sensing of the Atmosphere*, pages 1–18. Springer, 2005.

Fei Wang, Yu Yang, Baoquan Zhao, Junkun Jiang, Teng Zhou, Dazi Jiang, and Tie Cai. Deep 3d shape reconstruction from single-view sketch image. In *2020 8th International Conference on Digital Home (ICDH)*, pages 184–189. IEEE, 2020a.

Fu-En Wang, Yu-Hsuan Yeh, Min Sun, Wei-Chen Chiu, and Yi-Hsuan Tsai. Bifuse: Monocular 360 depth estimation via bi-projection fusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 462–471, 2020b.

Mei Wang and Weihong Deng. Deep visual domain adaptation: A survey. *Neurocomputing*, 312:135–153, 2018.

Yan Wang, Wei-Lun Chao, Divyansh Garg, Bharath Hariharan, Mark Campbell, and Kilian Q Weinberger. Pseudo-lidar from visual depth estimation: Bridging the gap in 3d object detection for autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8445–8453, 2019.

Yulin Wang, Rui Huang, Shiji Song, Zeyi Huang, and Gao Huang. Not all images are worth 16x16 words: Dynamic transformers for efficient image recognition. *Advances in Neural Information Processing Systems*, 34:11960–11973, 2021.

Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, 2004.

Shangzhe Wu, Christian Rupprecht, and Andrea Vedaldi. Unsupervised learning of probably symmetric deformable 3d objects from images in the wild. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1–10, 2020.

Yihong Wu, Yuwen Heng, Mahesan Niranjan, and Hansung Kim. Depth estimation from a single omnidirectional image using domain adaptation. In *European Conference on Visual Media Production (CVMP)*, pages 1–9, 2021.

Yihong Wu, Yuwen Heng, Mahesan Niranjan, and Hansung Kim. Depth estimation for a single omnidirectional image with reversed-gradient warming-up thresholds discriminator. In *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5. IEEE, 2023.

Yihong Wu, Yuwen Heng, Mahesan Niranjan, and Hansung Kim. Sliceformer: Deep dense depth estimation from a single indoor omnidirectional image using a slice-based transformer. In *International Conference on Electronics, Information and Communication (ICEIC)*, January 2024.

Yan Yan, Yuxing Mao, and Bo Li. Second: Sparsely embedded convolutional detection. *Sensors*, 18(10):3337, 2018.

Guorun Yang, Hengshuang Zhao, Jianping Shi, Zhidong Deng, and Jiaya Jia. Segstereo: Exploiting semantic information for disparity estimation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 636–651, 2018.

Jihan Yang, Shaoshuai Shi, Zhe Wang, Hongsheng Li, and Xiaojuan Qi. St3d: Self-training for unsupervised domain adaptation on 3d object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10368–10378, 2021.

Yong Yu, Xiaosheng Si, Changhua Hu, and Jianxun Zhang. A review of recurrent neural networks: Lstm cells and network architectures. *Neural Computation*, 31(7):1235–1270, 2019.

C Yuzbasioglu and Billur Barshan. A new method for range estimation using simple infrared sensors. In *2005 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1066–1071. IEEE, 2005.

Jure Zbontar, Yann LeCun, et al. Stereo matching by training a convolutional neural network to compare image patches. *J. Mach. Learn. Res.*, 17(1):2287–2318, 2016.

Weidong Zhang, Wei Zhang, and Yinda Zhang. Geolayout: Geometry driven room layout estimation based on depth maps of planes. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XVI 16*, pages 632–648. Springer, 2020.

Zhengyou Zhang. A flexible new technique for camera calibration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(11):1330–1334, 2000.

Zhiguang Zhong, Jianqiang Yi, Dongbin Zhao, Yiping Hong, and Xinzheng Li. Robust depth estimation for mobile robot navigation. In *2004 International Conference on Intelligent Mechatronics and Automation, 2004. Proceedings.*, pages 970–975. IEEE, 2004.

Jiaqi Zhou, Yihong Wu, Hwasub Lim, and Hansung Kim. Omnidirectional depth estimation for semantic segmentation. In *International Conference on Electronics, Information and Communication (ICEIC)*, January 2024.

Yimin Zhou, Guolai Jiang, Guoqing Xu, Xinyu Wu, and Ludovic Krundel. Kinect depth image based door detection for autonomous indoor navigation. In *The 23rd IEEE International Symposium on Robot and Human Interactive Communication*, pages 147–152. IEEE, 2014.

Nikolaos Zioulis, Antonis Karakottas, Dimitrios Zarpalas, and Petros Daras. Omnidepth: Dense depth estimation for indoors spherical panoramas. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 448–465, 2018.

Appendix A

Experiment with Data Augmentation

The data augmentation was checked by separating the equirectangular images into several chunks and shifting the chunks (shown in Figure A.1) and the performance was checked by training the model 5 times and calculating the mean values. Table A.1 shows a small improvement after doing data augmentation with Stanford2D3D Area1 and random 20% Standford2D3D Area1 training dataset and tested on Matterport Area2 dataset, which is less than 1% point.

In order to check how many separated chunks led to the best performance, the model was trained with different chunks (from 2 to 10) by randomly shifting the chunks. They were trained five times for each specific number of chunks, and each row on Table A.2 shows the average values of 5 times training results. Figure A.2 shows the boxplots for six evaluation metrics. The results show that splitting the images into four chunks performs the best.

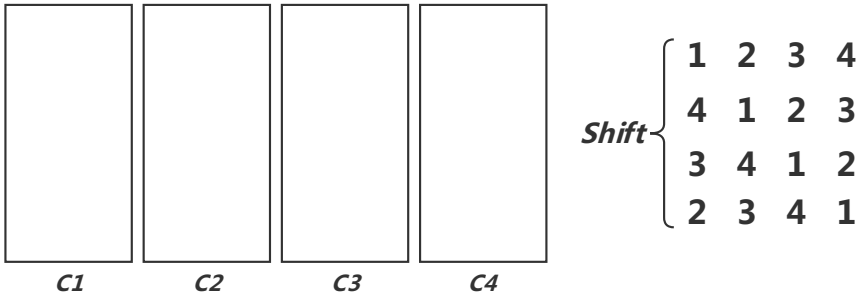


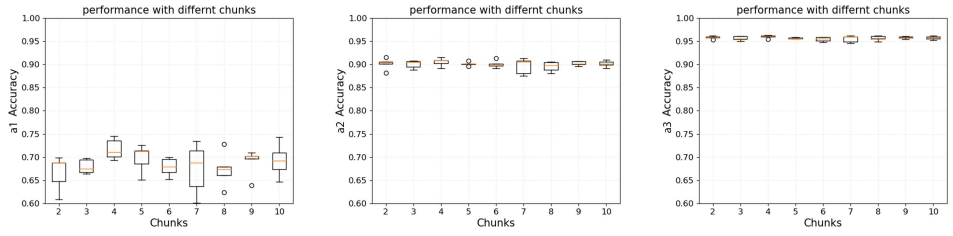
FIGURE A.1: The Process of Data Augmentation

TABLE A.1: Performance comparisons with and without Data Augmentation

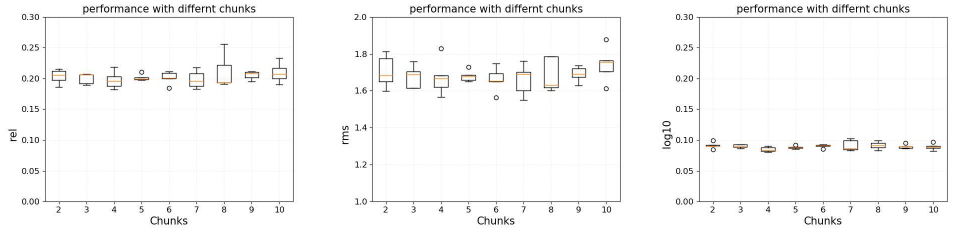
Testing dataset	Model	$a1 \uparrow$	$a2 \uparrow$	$a3 \uparrow$	$rel \downarrow$	$rms \downarrow$	$log10 \downarrow$
Stanford2D3D Area1	without Data Augmentation	76.89	94.66	97.71	0.1575	1.5052	0.0717
20% Stanford2D3D Area1		**69.25**	**90.22**	**95.91**	**0.2011**	**1.8052**	**0.0883**
Stanford2D3D Area1	with Data Augmentation	77.55	94.64	97.68	0.1554	1.4914	0.0708
20% Stanford2D3D Area1		**70.55**	**90.94**	**96.04**	**0.1925**	**1.7453**	**0.0856**

TABLE A.2: Performance with different chunks (trained on 20% Stanford2D3D area1 and tested on Matterport Area2)

Chunks	$a1 \uparrow$	$a2 \uparrow$	$a3 \uparrow$	$rel \downarrow$	$rms \downarrow$	$log10 \downarrow$
2	66.62	90.16	95.80	0.2034	1.7039	0.0914
3	67.94	90.01	95.60	0.2003	1.6749	0.0899
4	**71.72**	**90.52**	**95.98**	**0.1974**	**1.6722**	**0.0849**
5	69.77	90.11	95.62	0.2012	1.6795	0.0882
6	67.88	89.99	95.40	0.2009	1.6615	0.0902
7	67.51	89.63	95.49	0.1986	1.6607	0.091
8	67.33	89.52	95.62	0.2111	1.7374	0.0912
9	68.97	90.19	95.80	0.2052	1.6900	0.0895
10	69.29	90.12	95.72	0.2095	1.7422	0.089



(A) Accuracy



(B) Loss

FIGURE A.2: Boxplots for Data Augmentation

Appendix B

Supplement Materials for DepthInsight

B.1 Phase Scrambling

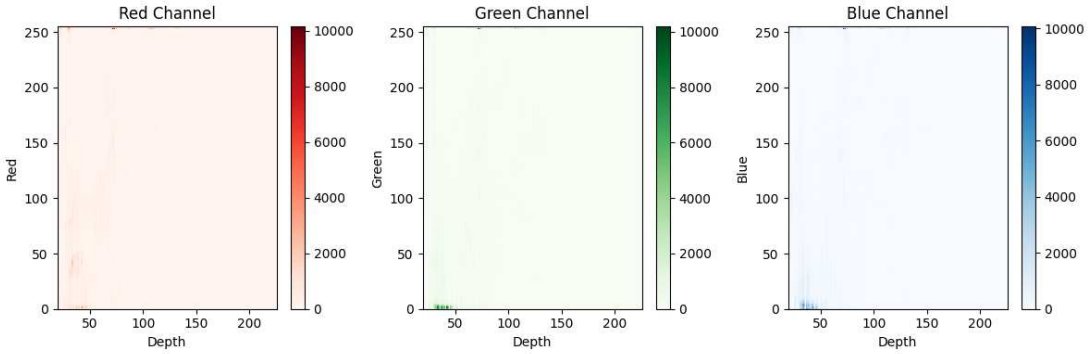
```
imFourier = fft2(input)
Amp = abs(imFourier)
Phase = angle(imFourier)
Phase = Phase + RandomPhase
imScrambled = ifft2(Amp * exp(1j * Phase))
imScrambled = GetRealPart(imScrambled)
```

LISTING B.1: Pseudocode of Phase Scrambling (Ge et al., 2022)

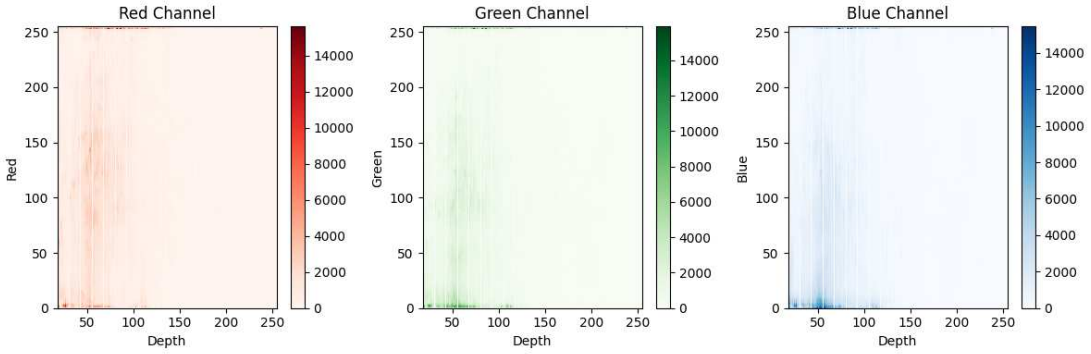
List B.1 presents the pseudo-code for the phase scrambling process.

B.2 Colour

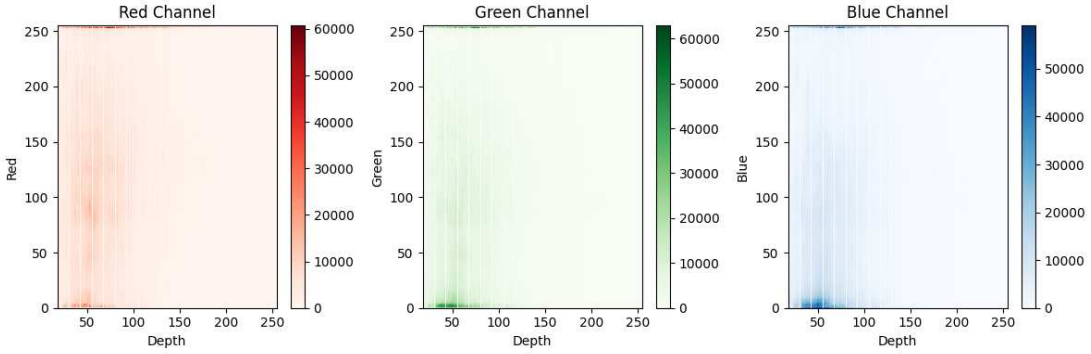
Figure B.1 represents the results of the accumulated values obtained from datasets of 50, 100, and 500 randomly sampled images. The depth range is depicted on the horizontal axis, while the vertical axis indicates the number of pixels in the R, G, and B channels corresponding to the specific depth ranges. These images show that the factors affecting depth are not significantly related to the distribution of pixels on the RGB channel.



(A) 10 Images



(B) 100 Images



(C) 500 Images

FIGURE B.1: Heatmap of the Relationship between the Distribution of RGB Three-channel Values and the Depth Map. The horizontal axis represents the depth range, while the vertical axis corresponds to the pixel count of the R, G, and B channels within the respective depth ranges. The colour bar values represent the pixel counts for three respective channels from different numbers of images randomly selected from the NYU dataset.

B.3 Saturation

Figure B.2 illustrates saturation maps with different saturation values, while Figure B.3 displays various RGB images with different saturation values alongside their corresponding model performance. It can be observed that the model's performance does not exhibit a strong sensitivity to different saturation values. As the saturation values

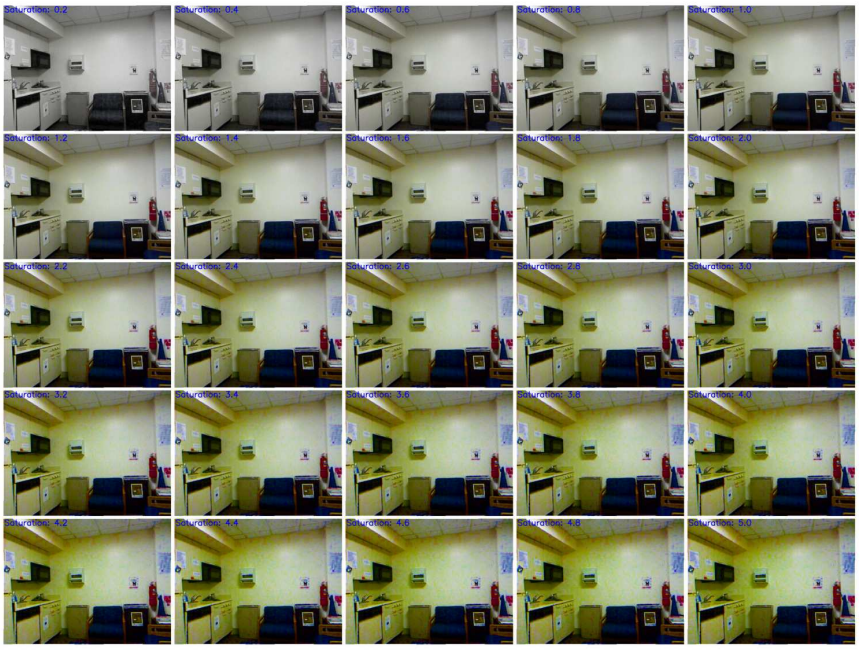


FIGURE B.2: Saturation Maps with Different Saturation Values

increase, there is a slight decline in performance. We hypothesised that this decline is due to the presence of more noise in images with high saturation values, as depicted in Figure B.2, which negatively impacts the model's performance (shown in Sec. B.6).

Note that during training, the channel order is BGR. However, for the sake of convenience in checking, the images have been converted to RGB channel order.

B.4 Shape

Figure B.4 illustrates the RGB images alongside their respective shape maps, as well as the output depth maps generated by the trained model using these inputs. Despite the substantial disparity in information content between the RGB images and shape maps, their contributions to depth estimation appear to be similar.

B.5 Contrast

Due to the inclusion of shape, shading, and other information, Contrast cannot be extracted independently. The adopted method involves utilising a trained model and

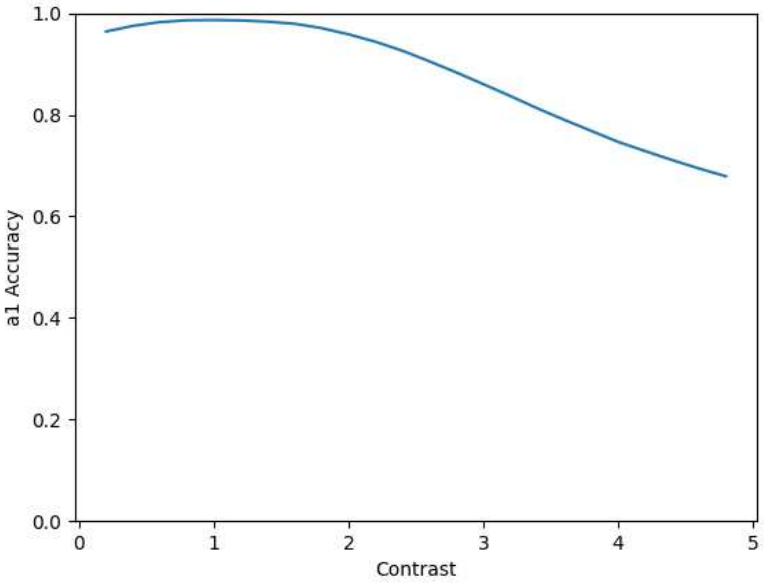


FIGURE B.3: Different Saturation RGB Images and Model Performance

incrementally adjusting the contrast of the test set images during the reasoning process. This enables observation of the performance of the model's depth estimation and facilitates analysis.

Figure B.5 shows images with different contrast values ranging from 0.2 to 5.

Figure B.6 illustrates that when the contrast remains relatively stable compared to the original image, such as within the range of 0.6-1.6, we observed minimal changes in performance. This observation leads us to suspect that the narrower depth range typically found in indoor scenes may contribute to this phenomenon, as the variations within this small range might not be noticeable.

Considering the contrast formula, output = saturate $(src * alpha + beta)$, excessive or insufficient contrast values can result in a loss of picture details, leading to a significant decline in performance.

B.6 Discussion

However, Figure B.2 and Figure B.5, show that these approaches merely appeared to mirror the acquired knowledge of the data-driven model. The model attained its optimal performance when presented with input data characterised by the same levels of original saturation and contrast as those found in the training dataset.

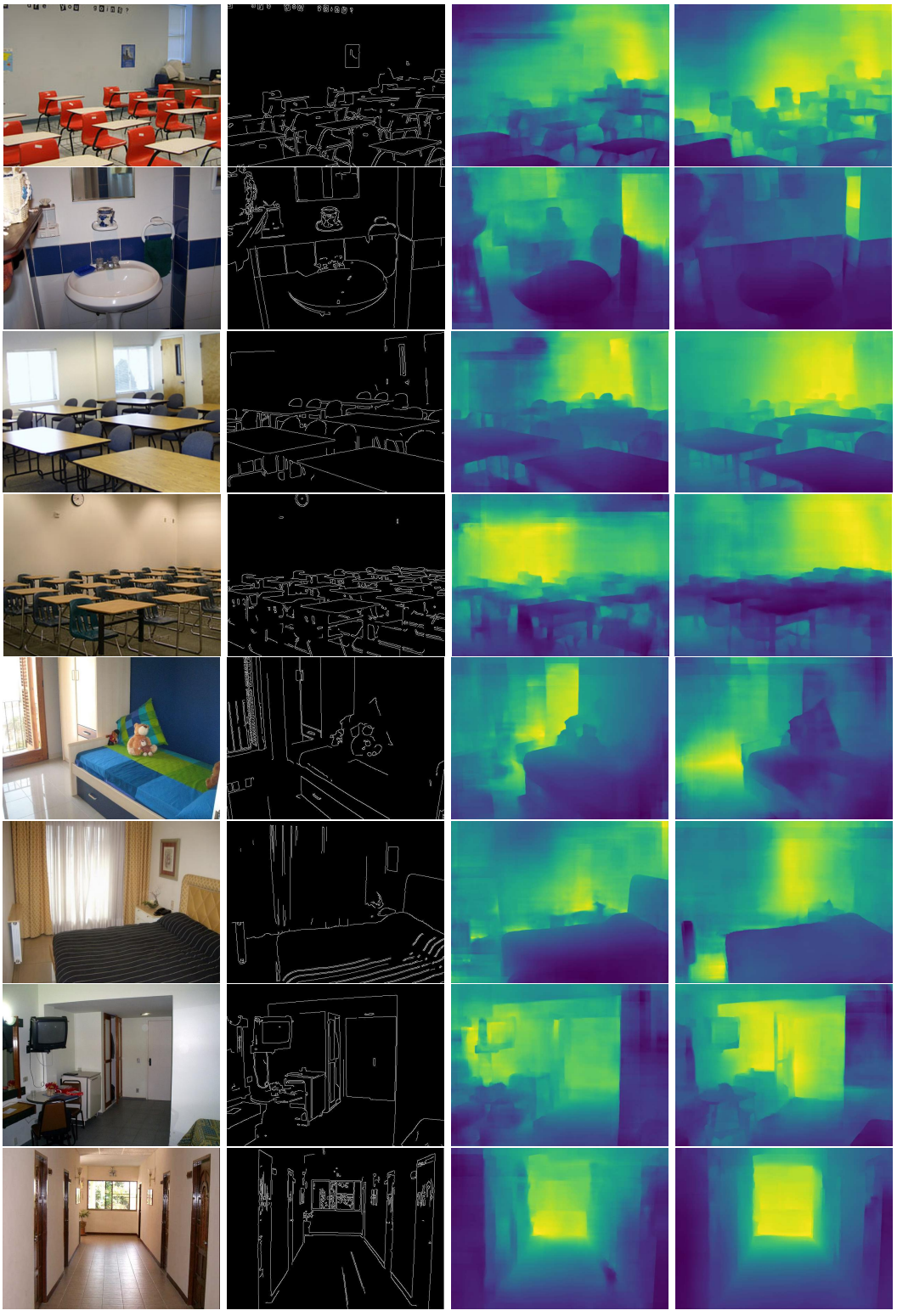


FIGURE B.4: Performance of Shape ONLY model with New Indoor Scenes from other Domains. The left column displays RGB scene images, the second column presents corresponding edge maps, and the third column showcases the results generated by the pre-trained shape-input model. The right column exhibits the outcomes produced by the pre-trained RGB-input model.

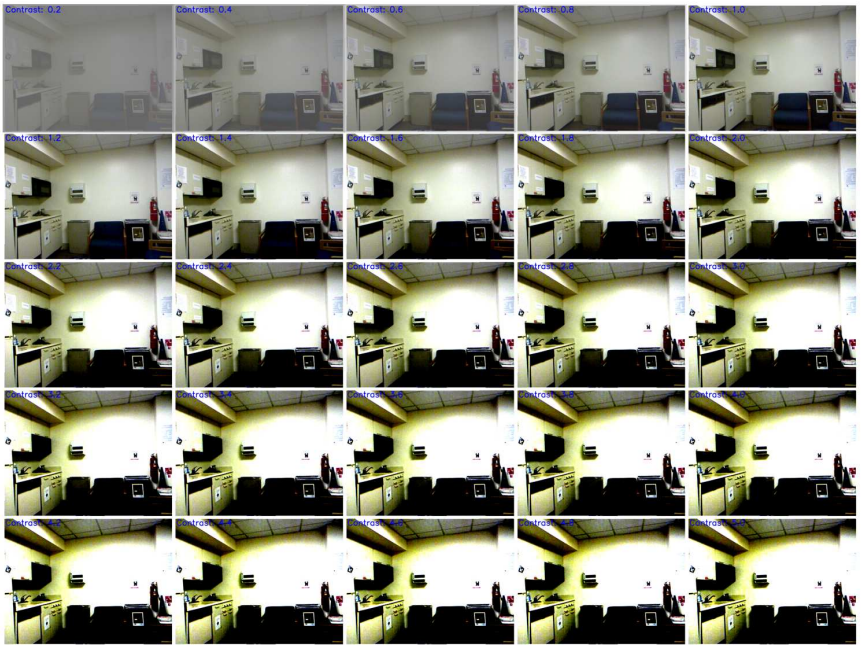


FIGURE B.5: Contrast Maps with Different Contrast Values

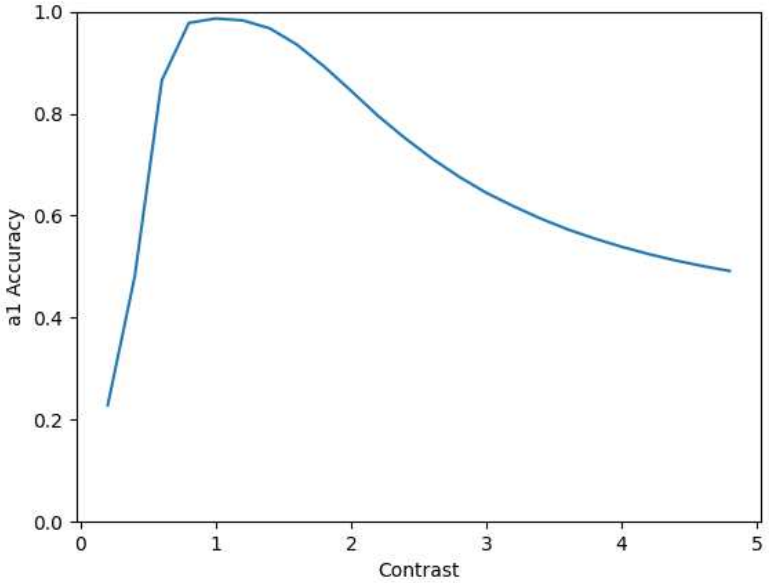


FIGURE B.6: Different Contrast RGB Images and Model Performance