DEPTH ESTIMATION FOR A SINGLE OMNIDIRECTIONAL IMAGE WITH REVERSED-GRADIENT WARMING-UP THRESHOLDS DISCRIMINATOR

Yihong Wu *Yuwen Heng* *Mahesan Niranjan* *Hansung Kim*

Vision, Learning and Control Research Group, School of Electronics and Computer Science
University of Southampton, UK

ABSTRACT

Depth estimation for single image using deep learning requires a large labelled depth dataset with various scenes for training. However, currently published omnidirectional depth datasets cover limited types of scenes and are not suitable for depth estimation for various real-world scenes. With the challenge of labelled real-world datasets generation and stability of the performance, we propose an architecture with the Reverse-gradient Warming-up Threshold Discriminator (RWTD) to estimate real-world depth maps from the 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. Compared with state-of-the-art encoder-decoder models, the proposed architecture shows an 11% points improvement on the testing dataset for depth accuracy.

Index Terms— Depth estimation, domain adaptation

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 [1]. Depth sensors, such as time-of-flight cameras and LiDARs, can generate precise depth maps [2]. However, they have deficiencies such as low-resolution [3], inaccuracy in textureless regions, short sensing range and expensive reconstruction process [4].

One barrier to depth estimation with a normal 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 [5]. There have been several end-to-end models on omnidirectional single image depth estimation for the omnidirectional depth estimation [3, 6]. These encoder-decoder models require large labelled datasets containing different scenes for learning to be able to generally predict depth maps for real-world scenes [7]. However, 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 [3] and Pano3D [8], contains similar depth distribution and limited real-world scene types.

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 [9]. 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 [10]. Inspired by previous works [9, 7], 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.

2. RELATED WORK

Depth estimation. End-to-end neural network based on U-Net was utilised to estimate depth from omnidirectional RGB images [3]. [6] proposed to combine two networks with an equirectangular image and its corresponding cubic projection map to avoid the distortion problem of omnidirectional images. SliceNet [11] uses long short-term memory (LSTM) to represent relationships between vertical slices of equirectangular projections to estimate depth maps. Although these models show good performance with the given labelled datasets, they cannot 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 [7], 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 common feature space [12].

This work was partially supported by the EPSRC Programme Grant Immersive Audio-Visual 3D Scene Reproduction (EP/V03538X/1) and partially by the Korea Institute of Science and Technology (KIST) Institutional Program (Project No. 2E31591)

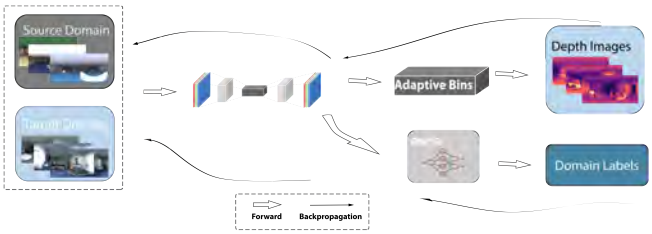


Fig. 1. Overview of Proposed Architecture

Domain adaptation. Domain adaptation is a method to map different domain data into a common feature space. [10] proposed a Generative Adversarial Network (GAN) [13] based model for depth estimation. This model learned from the digital handwriting dataset can recognise a different digital dataset with colourful handwriting images. There is a work for normal perspective images to comprehensively predict depth maps, surface normals, and edge contour maps [9]. Similarly, [7] 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.

From Regression to Classification [14] demonstrated that the regression problem could be transformed into a series of ordinal binary classification tasks (ordinal regression). [15] proposed depth estimation by an ordinal regression network, which divides a depth range into a set of discrete intervals. Each interval represents a threshold with 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. [16] added a transformer encoder to the model based on the work of [15] to predict the adaptive depth intervals of different images rather than fix them, thus obtaining more accurate and smooth depth maps.

3. METHOD

3.1. Proposed Architecture

Figure 1 illustrates the overview of the proposed architecture. It consists of an encoder-decoder model, transformer encoder, and proposed Reverse Warming-up Threshold Discriminator (RWTD).

Encoder-decoder Model. The encoder-decoder model is the U-Net model. For encoder, EfficientNet B5 [17] is used as backbone because of the better performance according to our experimental results for comparing backbone of ResNet [18], EfficientNet, and DenseNet [19]. This is because EfficientNet can integrate width, depth, and resolution into a comprehensive task of the network [17]. For the decoder, we use 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 for the transformer encoder.

Adaptive Bins. Regression-based architectures do not get enough global information for 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. The transformer can help as it considers global information throughout. The predicted depth range can be divided into bins [16], and the final depth estimate is a linear combination of these bins centres. The main body of the adaptive bins block in our architecture is a vision transformer [20] based structural block that divides the depth range of each scene into multiple bins, and the central bin values show the depth adaptively. Following this idea, the depth regression task is transformed into a classified task.

Reverse Gradient Warming-up Threshold Discriminator. As the main contribution of our work, Reverse-gradient Warming-up Threshold Discriminator (RWTD) enables the architecture to predict depth maps without training on real-world ground truths, but only on CG 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. With the idea of reverse-gradient descent [10], the RWTD is trained to be unable to distinguish which domain the feature vectors belong to. In addition, RWTD allows the discriminator to focus on 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 [10, 21, 7], it assigns different weights to different scenes in the training data set during the training process. Therefore, the information learned in the source domain can be applied to predict depth maps of unlabelled scenes from the target domain. Moreover, a further reason why the previous architecture cannot train just on CG pictures and predict depth maps for real-world situations is that the domain label losses will continue to increase and dominate the loss function, hence guiding the whole architecture in the incorrect gradient direction. To address this issue, RWTD employs warming-up thresholds to set constraints on the loss values throughout the training process, and this value is modified based on the training epoch to ensure the optimal performance of the whole architecture (details shown in Sec. 3.2.1).

3.2. Loss Function

The loss function combines the dense depth loss [22], the ChamferLoss [16], and Domain Label Loss (DLL) (Equation 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. More details about

the dense depth loss and Chamfer Loss can be found on in the GitHub page introduced in Sec. 3.3.

$$L(GT, Output) = \alpha L_{dense}(GT, Output) + \beta L_{Chamfer}(GT, Output) + \theta(L_{label_s}(GT, Output) + L_{label_t}(GT, Output)) \quad (1)$$

3.2.1. Domain Label Losses

The source and target domain images are labelled with domain labels 1 and 0, respectively. DLL function calculates the loss values between the original domain label and the output domain label from the discriminator. Inspired by focal loss [23, 21], RWTD was designed to solve the low-performance problem caused by imbalanced data in the image domain. The proposed discriminator can ignore the easily distinguished samples and increase the weight of the samples that are difficult to distinguish (Equation 2 and 3). $thres$ is the RWTD threshold factor, and p is the model's estimated probability for the class, while d is the domain label.

$$RWTD(p) = -f(p)\log(p), \quad f(p) = (1-p)^{\gamma}, \quad p = max(p, thres) \quad (2)$$

$$p = \begin{cases} p & \text{if } d = 1 \\ 1-p & \text{if } d = 0 \end{cases} \quad (3)$$

As shown in Equation 4, the threshold probability $thres$ decreases according to the epoch number during the training process. From experiments with preliminary architecture [7], it can be observed that the architecture does not perform well because 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.

$$thres = \begin{cases} 1 \times 10^{-4} \times 10^{-epoch} & \text{if } p \geq 1 \times 10^{-24} \\ 1 \times 10^{-24} & \text{otherwise} \end{cases} \quad (4)$$

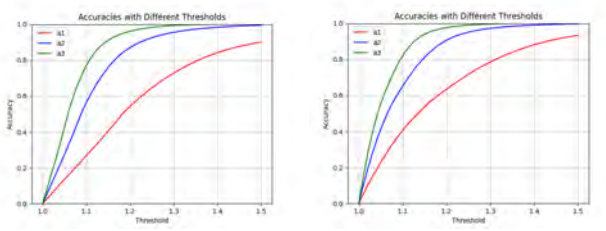
3.3. Implementation and Evaluation Metrics

In order to support the research in this field, the code and implementation details of this work were made available at `https://github.com/MinisculeDust/RWTD`. The implementation details and supplement materials are also shown on GitHub page.

4. EXPERIMENTS

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 on GitHub) for the CG 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 [4], AdaBins [16] and SliceNet [11] were set as 1×10^{-5}, 3×10^{-4} and 1×10^{-3} respectively after doing the similar experiments.



(a) AdaBins (b) Proposed Architecture

Fig. 2. 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.

Table 1 shows that the proposed architecture outperformed the state-of-the-art (SOTA) models [4, 16, 3, 11] 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 [3]. 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.

Fig. 2 shows the performance of two trained models with different thresholds 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 [22, 4, 3, 16]. 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.

4.2. Comparison with DA

Table 2 shows the results with different discriminators. We were unable to obtain a result with the weak-alignment discriminator [21] because the probability value during training

Table 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 [4]	50.35±1.55	81.8±1.49	95.24±0.62	0.255±0.007	0.973±0.019	0.118±0.004
	AdaBins [16]	63.03±4.27	90.32±1.83	97.7±0.53	0.25±0.025	0.699±0.058	0.091±0.008
	RectNet [3]	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 [11]	59.63±4.27	88.11±3.82	97.8±0.70	0.26±0.029	0.624±0.051	0.096±0.009
	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 [4]	50.56±0.32	78.6±0.57	92.52±0.32	0.271±0.003	1.098±0.007	0.123±0.001
	AdaBins [16]	69.42±5.68	90.71±1.67	97.29±0.43	0.227±0.03	0.641±0.034	0.083±0.01
	RectNet [3]	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 [11]	57.91±6.23	86.87±1.67	96.17±0.64	0.281±0.028	0.734±0.044	0.103±0.009
	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**

Table 2. Effect of discriminator

Model	$a_1 \uparrow$	$a_2 \uparrow$	$a_3 \uparrow$	rel $\downarrow$	rms $\downarrow$	log10 $\downarrow$
Unsupervised DA [7]	26.2	50.7	68.1	0.855	1.720	0.235
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**

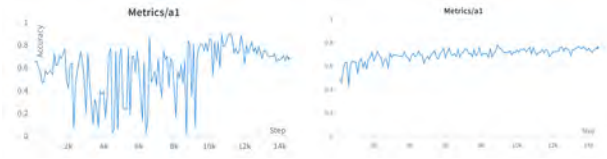
was too low. The model from [10] cannot work well with the task from CG scenes to real-world scenes because of the dominant DLL.

4.3. Stability

The proposed model not only outperforms the SOTA models, such as AdaBins but also performs more stable than them. Fig. 3 shows the comparison of the stability of models. 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.

4.4. Ablation Study

For individual component analysis of the proposed architecture, the ablation studies are conducted on the SunCG and Stanford2D3D datasets. Table 3 shows to what extent different components contributed to the proposed architecture. This table shows the accuracy of the proposed model is improved by about 9% points compared with the structure containing encoder-decoder only with the help of RWTD, and about 4% points accuracy improvement than that with reverse-gradient discriminator (RD).



(a) Stanford2D3D Area5

(b) Stanford2D3D Area6

Fig. 3. Stability comparison of a_1 accuracy (Left: AdaBins; Right: Proposed method)

Table 3. 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 $\downarrow$
Encoder-decoder model only	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**

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 CG 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.

6. REFERENCES

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

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

[3] Nikolaos Zioulis, Antonis Karakottas, Dimitrios Zarpalas, and Petros Daras, "Omnidepth: Dense depth estimation for indoors spherical panoramas," in *Proc. ECCV*, 2018, pp. 448–465.

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

[5] Hansung Kim and Adrian Hilton, "3d scene reconstruction from multiple spherical stereo pairs," *IJCV*, vol. 104, no. 1, pp. 94–116, 2013.

[6] 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 *Proc. CVPR*, 2020, pp. 462–471.

[7] Yihong Wu, Yuwen Heng, Mahesan Niranjan, and Hansung Kim, "Depth estimation from a single omnidirectional image using domain adaptation," in *Proc. CVMP*, 2021, pp. 1–9.

[8] 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 *Proc. CVPR*, 2021, pp. 3727–3737.

[9] Zhongzheng Ren and Yong Jae Lee, "Cross-domain self-supervised multi-task feature learning using synthetic imagery," in *Proc. CVPR*, 2018, pp. 762–771.

[10] Yaroslav Ganin and Victor Lempitsky, "Unsupervised domain adaptation by backpropagation," in *Proc. ICML*, 2015, pp. 1180–1189.

[11] 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 *Proc. CVPR*, 2021, pp. 11536–11545.

[12] Sinno Jialin Pan and Qiang Yang, "A survey on transfer learning," *IEEE Transactions on knowledge and data engineering*, vol. 22, no. 10, pp. 1345–1359, 2009.

[13] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio, "Generative adversarial nets," in *Proc. NeurIPS*. 2014, vol. 27, Curran Associates, Inc.

[14] Zhenxing Niu, Mo Zhou, Le Wang, Xinbo Gao, and Gang Hua, "Ordinal regression with multiple output cnn for age estimation," in *Proc. CVPR*, 2016, pp. 4920–4928.

[15] Huan Fu, Mingming Gong, Chaohui Wang, Kayhan Batmanghelich, and Dacheng Tao, "Deep ordinal regression network for monocular depth estimation," in *Proc. CVPR*, 2018, pp. 2002–2011.

[16] Shariq Farooq Bhat, Ibraheem Alhashim, and Peter Wonka, "Adabins: Depth estimation using adaptive bins," in *Proc. CVPR*, 2021, pp. 4009–4018.

[17] Mingxing Tan and Quoc Le, "Efficientnet: Rethinking model scaling for convolutional neural networks," in *Proc. ICML*, 2019, pp. 6105–6114.

[18] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, "Deep residual learning for image recognition," in *Proc. CVPR*, 2016, pp. 770–778.

[19] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger, "Densely connected convolutional networks," in *Proc. CVPR*, 2017, pp. 4700–4708.

[20] 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.

[21] Kuniaki Saito, Yoshitaka Ushiku, Tatsuya Harada, and Kate Saenko, "Strong-weak distribution alignment for adaptive object detection," in *Proc. CVPR*, 2019, pp. 6956–6965.

[22] David Eigen, Christian Puhrsch, and Rob Fergus, "Depth map prediction from a single image using a multi-scale deep network," *Proc. NeurIPS*, vol. 27, 2014.

[23] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár, "Focal loss for dense object detection," in *Proc. ICCV*, 2017, pp. 2980–2988.