Depth Estimation on Underwater Omni-directional Images Using a Deep Neural Network

Haofei Kuang[1], Qingwen Xu[1], Sören Schwertfeger[1]

Abstract— In this work, we exploit a depth estimation Fully Convolutional Residual Neural Network (FCRN) for in-air perspective images to estimate the depth of underwater perspective and omni-directional images. We train one conventional and one spherical FCRN for underwater perspective and omni-directional images, respectively. The spherical FCRN is derived from the perspective FCRN via a spherical longitude-latitude mapping. For that, the omni-directional camera is modeled as a sphere, while images captured by it are displayed in the longitude-latitude form. Due to the lack of underwater datasets, we synthesize images in both data-driven and theoretical ways, which are used in training and testing. Finally, experiments are conducted on these synthetic images and results are displayed in both qualitative and quantitative way. The comparison between ground truth and the estimated depth map indicates the effectiveness of our method.

I. INTRODUCTION

Due to the properties of underwater environments, underwater perception is quite different from air. Images captured in underwater case usually look bluish or greenish. Besides, the underwater images are more blurred than that in air captured by the same camera due to turbidity. These reasons increase the difficulty of depth estimation from images. Thus many researchers put effort on the underwater image processing. For example, using dark channel priors is proposed to restore underwater images in [9], [20], inspired by He et al.'s work on removing haze in air [12]. Pfingsthorn et al. implemented underwater image stitching based on spectral methods [24], which are more robust to turbidity than feature based methods. Besides image enhancement, some work focuses on depth estimation. Peng et al. exploited the relationship between depth and blurriness of underwater images to estimate depth [23]. In addition, deep learning was also applied to estimate the depth of underwater images, for example, Li et al. used a convolution neural network (CNN) to generate relative depth, which was then one of the inputs for a color correction network [17].

In addition to normal pin-hole cameras, omni-directional cameras are becoming popular due to their large field of view (FOV). They have been widely used on ground robots [2], [3], [16]. Some research groups also studied omni-directional cameras for underwater use since they provide more information than perspective ones on object detection, localization and mapping. Boult designed an omni-directional video equipment and put it on dolphins to capture data [6]. In [5], Bosch et al. improved on-land omni-directional cameras

[1]Authors are with School of Information Science Technology of ShanghaiTech University <kuanghf, xuqw, soerensch>@shanghaitech.edu.cn

Fig. 1: An underwater omni-directional image[1].

for underwater use and proposed the method for camera calibration.

In this paper, we aim to estimate the depth of underwater omni-directional images. In contrast to on-land scenarios,underwater depth estimation is more challenging due to scattering and absorption effect [8], [23] as mentioned above. In the very beginning, Eigen et.al. [10] proposed a two-stack convolutional neural network to estimate depth from single images. Later, many researchers improved the performance of depth estimation based on deep learning [14], [15], [19]. We try to apply deep learning to estimate depth of omni-directional underwater images in this work to solve the difficulty in underwater scenario. Since deep learning is a data-driven way, a large amount of data is necessary. However, underwater images are hard to collect, especially omni-directional ones. Thus, we generate synthetic datasets based on available in-air datasets to handle this issue.

Another challenge is the serious distortion of omni-directional images as shown in Figure 1. For that, we learn from approaches that work for in-air images. Gehrig rectified the region of interest of omni-directional images to perspective images for the specific task of object detection[11]. Omni-directional images are undistorted into longitude-latitude coordinates in [18]. Zhao et al. used a geodesic grid when extracting features on the omni-directional images [29]. In [1], [21], the omni-directional camera is modeled as a sphere and the omni-directional images are projected to the bounding cubic of the sphere, so that the image processing algorithms of perspective images, which are captured by pin-hole cameras, can be applied to the sub-images of each cubic side. In this work, we describe the omni-directional images in longitude-latitude coordinates and build a mapping between the longitude-latitude coordinate and the tangent space of the spherical camera model, then apply the mapping in deep neural networks as mentioned in [7], [28].
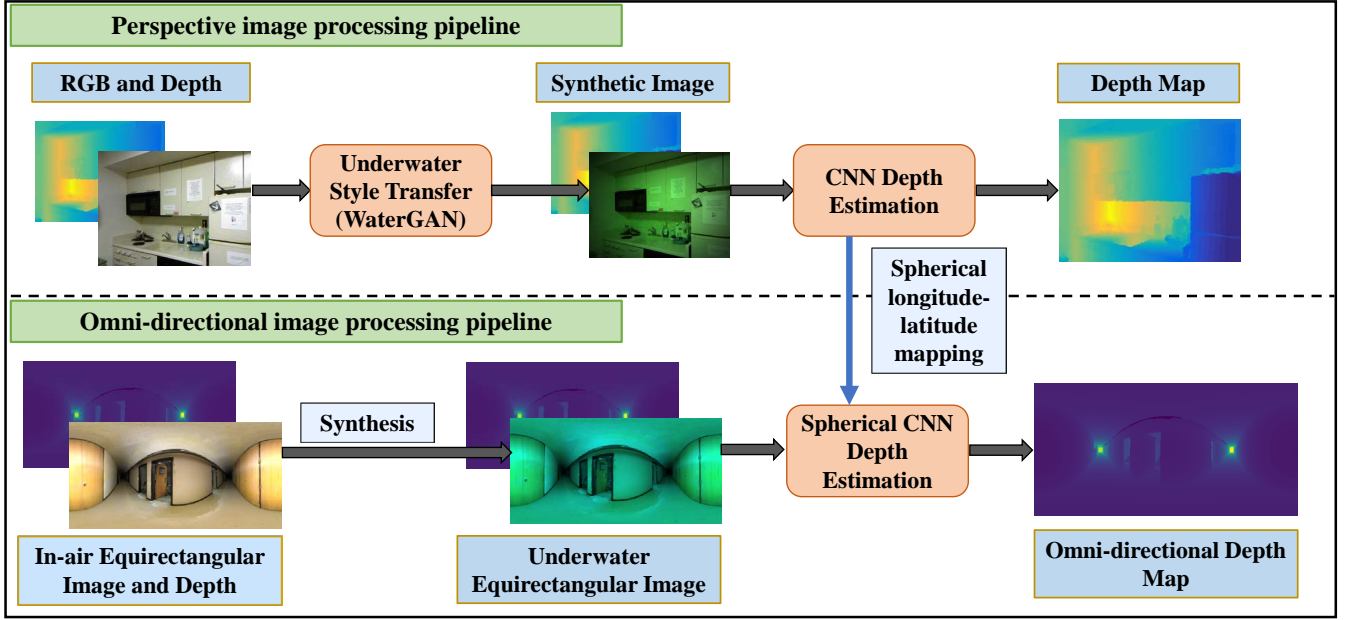
Fig. 2: Perspective and omni-directional image processing training pipeline.

This paper is about transferring a neural network for in-air depth estimation to the underwater case. In Section II we introduce the pipeline of estimating depth on underwater images. Afterwards, all networks used in the work are explained in Section III, where we also show approaches to the challenges of dataset generation and undistortion. In Section IV, we show the experimental details and analyze the results. Finally, we conclude this work in Section V.

II. SYSTEM OVERVIEW

Applying deep learning, we conduct this work based on two known neural networks: WaterGAN [17] and FCRN [15]. WaterGAN uses a generative adversarial network to transfer in-air perspective images to underwater tunes; FCRN has been proven to work well in depth estimation for both RGB and RGBD images. Furthermore, we use ideas presented in [7] to convert the networks to work with omni-directional images.

The pipeline of this work is shown in Figure 2. The upper pipeline describes the training on perspective images. Firstly, RGB and depth images are transmitted into a style transfer network (WaterGAN). Then the synthetic underwater images and the corresponding depth images are the input of the conventional CNN (FCRN). After training, the network outputs the estimated depth map. The lower row shows the similar training process on omni-directional images, where we modify the conventional FCRN to spherical FCRN based on the similar idea introduced in [7], [28]. Since WaterGAN does not support style transfer on omni-directional images, we distort the in-air omni-directional images to underwater tunes by decreasing values in red channel and blurring images according to depth as mentioned in [9]. Then the synthetic underwater omni-images are used as input to the spherical CNN network to estimate the omni-directional
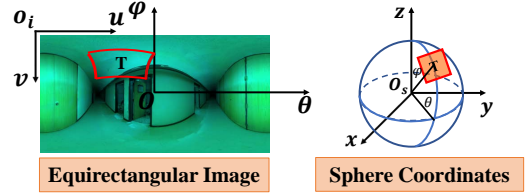

Fig. 3: Mapping between longitude-latitude and spherical coordinates.

depth map, where the standard convolution and pooling is replaced with spherical ones.

III. NEURAL NETWORK TRANSFER

In this section, we introduce the two main neural networks used to solve the challenges: distortion removal, dataset generation and depth estimation.

A. Distortion Removal

With the large FOV, omni-directional images suffer from serious distortions. We do not intend to remove all the distortion from the longitude-latitude rectangular image, but provide corrected pixel coordinates in the convolution neural network (CNN). In the convolution and pooling layer of CNN, a square kernel is used to slide over the image, which is not applicable on the equirectangular image, due to the distortion. Thus we exploit the sphere to model the omni-directional camera and build the mapping between the spherical surface $S(\phi, \theta)$ and equirectangular image $I(u, v)$. Then the image projected to the tangent space $T(x, y)$ of S can be considered with no distortion [1]. As shown in Figure 3, the mapping between $I(u, v)$ and $S(\phi, \theta)$ can be
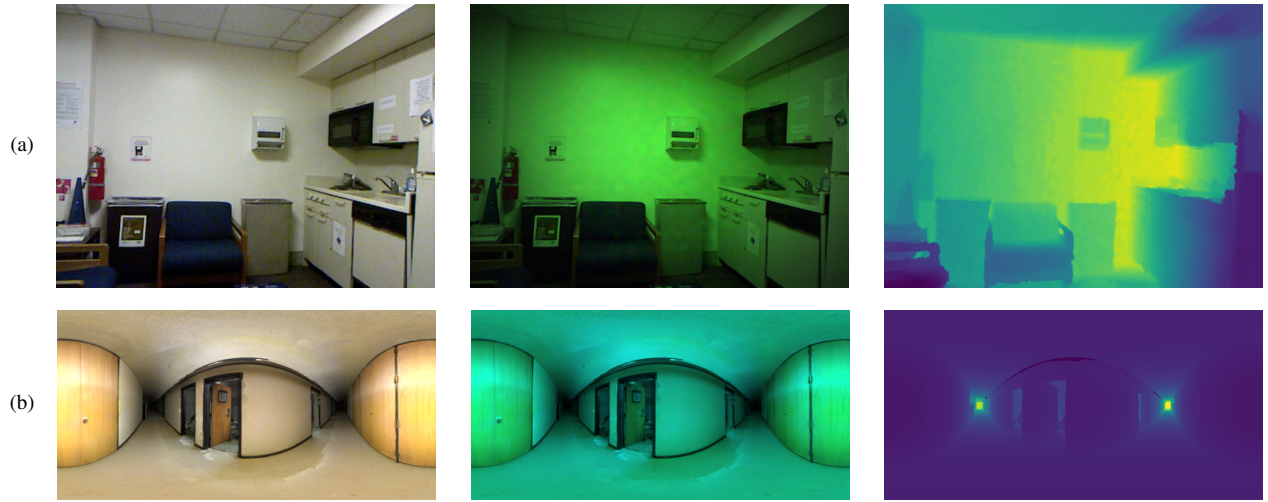
Fig. 4: The examples from the NYU-v1 dataset and the 360D dataset. (a) is from the NYU-v1 dataset and (b) is from the 360D dataset. First column are RGB images from each dataset. Second column show the synthetic RGB images in underwater style. The last column shows the ground truth depth map.

described as

$$\phi = (\frac{H}{2} - v)\frac{\pi}{H} \tag{1a}$$

$$\theta = (u - \frac{W}{2})\frac{2\pi}{W} \tag{1b}$$

where H and W are the height and width of equirectangular image $I(u,v)$. Besides, the mapping between the tangent space $T(x,y)$ and spherical coordinates $S(\phi,\theta)$ can be described by gnomonic projection. Thus, when calculating the kernel coordinates during convolution and pooling, we only need to calculate the relative coordinates to center on $T(x,y)$. Extending from the 3×3 kernel grid from [7], the relative coordinates of $n \times n$ kernel can be described as $\left[(\alpha x_{ij}, \beta y_{ij}) \right]_{n \times n}$ when the kernel center coordinates is $(0,0)$, where

$$x_{ij} = \tan(|i|\,\Delta_\theta) \tag{2a}$$

$$y_{ij} = \frac{\tan(|j|\,\Delta_\phi)}{\cos(|i|\,\Delta_\theta)} \tag{2b}$$

, α and β are the symbols consistent with the relative coordinates to center. Afterwards, we project these pixels (x,y) on tangent plane $T(x,y)$ to $I(u,v)$ by inverse gnomonic projection.

B. Synthetic Dataset

Even though there are some released underwater datasets, they cannot provide hundred thousands of images for the training of network. Motivated by [17], we augment in-air images in underwater style to synthesize adequate underwater images for training. As mentioned in Sec II, perspective images are augmented in a data-driven way, i.e. use Water-GAN[2] to transfer in-air images to underwater ones based on given underwater samples. And omni-directional images are distorted in color space and depth channel. The red light

disappears firstly in the ocean due to its short wavelength so that the underwater image often looks blue or green as introduced in [26]. Besides, the underwater object becomes blurred when it gets far away from the camera owing to the attenuation of direct light and backscattering effect [9], [23]. Thus the in-air images can be converted to underwater style approximately by reducing red component of the images and blurring the images according to pixels' depth. This can be implemented in the following:

$$I_w(u,v,'r') = \gamma * I_a(u,v,'r') \tag{3a}$$

$$I_w(u,v) = GaussBlur(I_a(u,v), f(d(u,v))); \tag{3b}$$

, where I_w and I_a are underwater and in-air images, $0 < \gamma < 1$ is the attenuation factor of the red component, $I(u,v,'r')$ represents the red channel of an image and $f(d(u,v))$ is the kernel size of Gaussian blur depend on pixel (u,v) whose depth is $d(u,v)$. The synthetic perspective and omni-directional image samples are shown in Figure 4a and 4b respectively. Both underwater samples looks greenish than in-air ones.

C. Depth Estimation

The core idea of spherical CNN is introduced in both [7] and [28]. Since codes of both works have not been released, we modify a conventional CNN, FCRN[3], to realize our spherical CNN based on partial details from these works. The structure of standard and spherical CNN is shown in Figure 5. We use conventional FCRN to estimate depth for underwater perspective images and spherical FCRN for omni-directional ones.

In the conventional FCRN, we use ResNet-50 [13] as feature extraction layer (encoding layer) and up-project [15] as up-sampling (decoding) layer. Besides, the unpooling
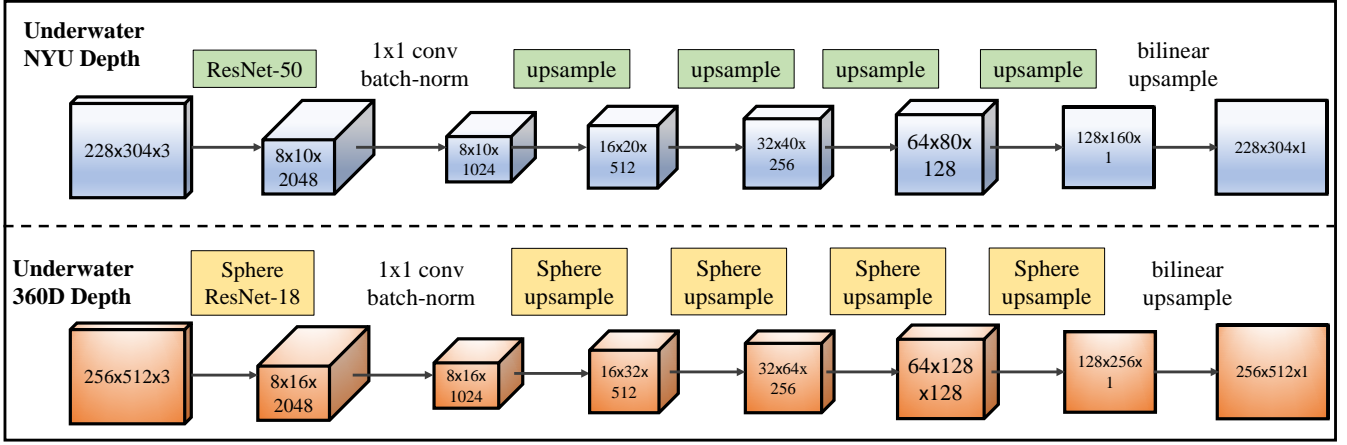
Fig. 5: The architecture of depth estimation networks for perspective and omni-directional images. We train a ResNet-50 FCRN model for the underwater NYU dataset. We are then using a Sphere ResNet-18 architecture to train the model with underwater 360D datasets to reduce the memory and training time.

layer of FCRN is replaced with deconvolution layer in order to simplify the calculation as reported in [22]. In the spherical FCRN, the convolution and pooling layers are replaced with spherical convolution and pooling to build a SphereResNet model. In other words, we use the approach described in Sec III-A to calculate the corresponding pixels on omni-directional images with given pixels and square kernel. Then these corresponding pixels are used for spherical convolution and pooling calculation. Besides, ResNet-18 is the alternative to ResNet-50 in SphereResNet to reduce the consumption of memory and training time.

In addition, the mean absolute error $\mathcal{L}_1$ is used as default metric for its simplicity and performance, when operating optimization via Stochastic Gradient Descent (SGD) and back-propagation (BP). Moreover, we rescale the input images to meet the resolution consistency in [28] that the angle per pixel of both perspective and omni-directional images should be the same.

IV. EXPERIMENTS AND RESULTS

We perform the proposed method with synthetic datasets in the experiment. All networks: WaterGAN, FCRN and spherical FCRN are trained on a Titan V with 12G memory. The in-air dataset **NYUv1** [27] and real underwater images **MHL** [17] are used as the input of WaterGAN to synthesize underwater style **NYUv1** (**UW-NYU**).Thanks to the robustness of WaterGAN, we maintain the settings with learning rate 0.0002, batch size 64 and training epoch 25.

Both depth estimation networks are implemented with Py-Torch[4]. In the perspective images depth estimation, we train the FCRN model with the batch size of 16 on **UW-NYU**. The number of total training epochs is 30, the start learning rate is 0.01 and reduced 20% every 5 epochs; weight decay is $1e^{-4}$ for regularization and momentum is 0.9. Afterwards, we build spherical FCRN based on the conventional one and
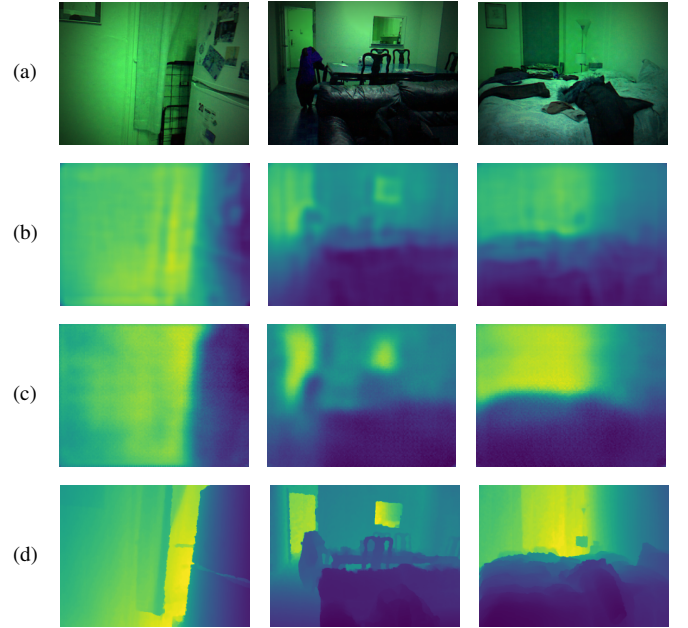
---

[4] https://pytorch.org/

Fig. 6: The experimental results of underwater NYU dataset. (a) are RGB images from the testing set. (b) are the predicted depth of Eigen et al.'s [10]; (c) are the predicted depth maps of ours; (d) are the ground truth depth maps.

the hyper-parameter are the same as FCRN. Finally, we use the synthetic underwater omni-directional images to train and validate the spherical FCRN.

In addition to show the advantage of our network, we also compare the performance of our network with Eigen et al.'s [10] on the perspective images.

A. Data Augmentation

To strengthen the robustness of the network, we conduct data augmentation on input images, including scale, rotation, flips, color jitter and normalization, similar to the procedure
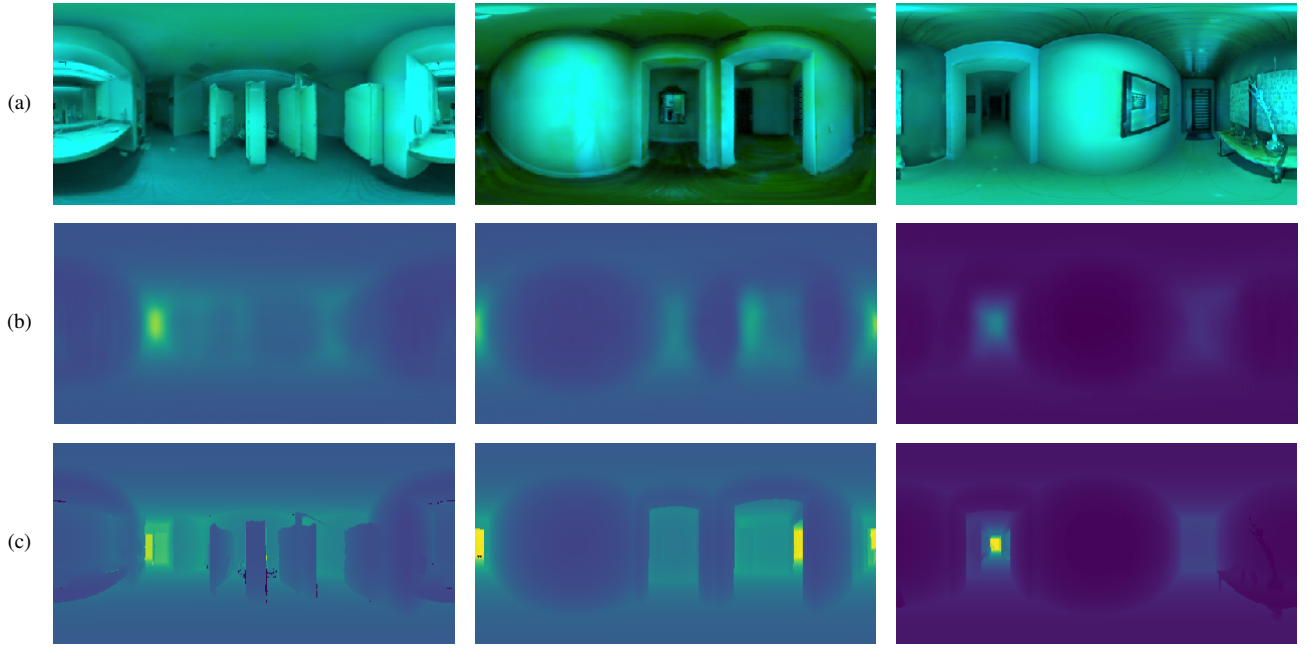
Fig. 7: The experimental results of underwater 360D dataset. (a) are RGB images from the testing set. (b) are the predicted depth maps; (c) are the ground truth depth maps.

in [22]. After augmentation, we crop the images from the center to keep the consistency of the input images.

B. Error Metrics

We also take the same metrics in [22] to evaluate our network. Here we address the metrics again:

- RMSE: root mean squared error
- MAE: mean absolute error
- REL: mean absolute relative error
- $\delta_1 = \frac{\#(\{\hat{y}_i : \max\{\frac{\hat{y}_i}{y_i}, \frac{y_i}{\hat{y}_i}\} < 1.25^i\})}{\#(\{y_i\})}$ (y_i, $\hat{y}_i$ are respectively the ground truth and the prediction) : percentage of predicted pixels whose depth error is smaller than a threshold. The higher δ_1 is, the better prediction is.

C. Results

1) Perspective Images: The experiment result of the depth prediction of **UW-NYU** dataset is shown in Figure 6, and the precision evaluation is described in the second row of Table I. The estimated depth map is close to ground truth except some details. Small RMSE, MAE and REL, together with δ_1 beyond 0.9 of ours in Table I indicates the result of the predict depth map achieve a high performance on the testing dataset, which is much better than Eigen et al.'s [10]. Besides, the average prediction time of each image is about $0.02s$.

2) Omnidirectional Images: Figure 7 shows the experimental results of the spherical FCRN, which still predicts depth correctly for most pixels. The quantitative results in the third row of Table I also shows that the result is acceptable. However, RMSE, MAE and REL of spherical FCRN are higher than that of conventional FCRN and δ_1 decreases a little in Table I, which indicates that the performance is not

(a) RGB image   (b) Depth map

(c) Predicted depth of Eigen et al.'s   (d) Predicted depth of ours
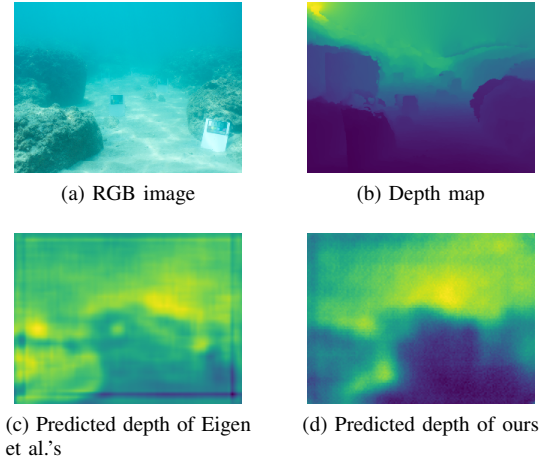
Fig. 8: We test the trained FCRN model on a real underwater image which is proposed by Berman et al. [4]. (a) and (b) are the input RGB image and depth map respectively. (c) is the predicted map of Eigen et al.'s [10]. (d) is the predicted map of ours.

as good as FCRN. The main reason is that we replace the ResNet-50 with ResNet-18 in spherical FCRN since the input omni-directional image is too big and our hardware cannot support training these images with ResNet-50.

3) Testing on Real Data: In order to verify the feasibility of our method, we test our model with some real underwater data collected by Berman et al. [4]. Figure 8 shows a sample of the testing results, where the predicted depth is not correct in some pixels. However, our method still performs better

Input	Model	RMSE	MAE	REL	δ_1	t_gpu(s)
Perspective	ResNet-50	0.162	0.117	0.098	0.914	0.0201
	Eigen et al.	0.235	0.184	0.148	0.806	0.005
Omnidirectinal	SphereResNet-18	0.604	0.362	0.172	0.711	0.0145

TABLE I: The error metrics of each model. t_gpu means the average operation time on each image on GPU.

than Eigen et al.'s. One possible reason could be the style of this image sets is different from our training dataset. The former looks blueish while the latter looks greenish. Another possible reason is the difference between camera models. The largest depth of training dataset is 10 meters while the the longest distance of Figure 8b is more than 15 meters.

D. Discussion

The above experimental results show that the conventional and spherical FCRN achieve good results on synthetic underwater perspective and omni-directional images, respectively. However, it does not show good performance when testing on a real underwater dataset. The main possible reason is that the style of the real underwater dataset and the synthetic dataset are too different. To overcome this challenge, we will enlarge the diversity of the training dataset in our future work, for example mixing images captured from different devices. Moreover, we will try to synthesize underwater images using accurate image formulation model [25]. On the other hand, we also plan to collect more real underwater omni-directional images to validate the spherical FCRN model. In addition, we will put more effort on transferring conventional FCRN to spherical FCRN without retraining, to save computation time and resources, which is reported feasible in [28].

V. CONCLUSION

We trained two depth estimation networks: FCRN for underwater perspective images and spherical FCRN for underwater omni-directional images, which are based on state-of-art depth estimation networks (FCRN) for in-air perspective images. Due to the lack of datasets, we synthesize underwater perspective images in a data-driven way and omni-directional images according to theoretical analysis. The test results on these synthetic images show that conventional and spherical FCRN can estimate depth map correctly on most pixels for synthetic underwater perspective and omni-directional images, respectively. In addition, we tested the trained FCRN model on real underwater images, which didn't give good results. To improve on that, we will increase the robustness of our network by collecting and more real underwater data and fine-tuning the estimation network based on the real data in our future work.

REFERENCES

[1] Juan David Adarve and Robert Mahony. Spherepix: A data structure for spherical image processing. *IEEE Robotics and Automation Letters*, 2(2):483–490, 2017.

[2] Antonis A Argyros, Kostas E Bekris, Stelios C Orphanoudakis, and Lydia E Kavraki. Robot homing by exploiting panoramic vision. *Autonomous Robots*, 19(1):7–25, 2005.

[3] Ryad Benosman, S Kang, and Olivier Faugeras. *Panoramic vision*. Springer-Verlag New York, Berlin, Heidelberg, 2000.

[4] Dana Berman, Deborah Levy, Shai Avidan, and Tali Treibitz. Underwater single image color restoration using haze-lines and a new quantitative dataset. *arXiv preprint arXiv:1811.01343*, 2018.

[5] Josep Bosch, Nuno Gracias, Pere Ridao, and David Ribas. Omnidirectional underwater camera design and calibration. *Sensors*, 15(3):6033–6065, 2015.

[6] Terry Boult. Dove: Dolphin omni-directional video equipment. In *Proc. Int. Conf. Robotics & Autom*, pages 214–220, 2000.

[7] Benjamin Coors, Alexandru Paul Condurache, and Andreas Geiger. Spherenet: Learning spherical representations for detection and classification in omnidirectional images. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 518–533, 2018.

[8] Paul Drews, Erickson Nascimento, Filipe Moraes, Silvia Botelho, and Mario Campos. Transmission estimation in underwater single images. In *Proceedings of the IEEE international conference on computer vision workshops*, pages 825–830, 2013.

[9] Paulo LJ Drews, Erickson R Nascimento, Silvia SC Botelho, and Mario Fernando Montenegro Campos. Underwater depth estimation and image restoration based on single images. *IEEE computer graphics and applications*, 36(2):24–35, 2016.

[10] David Eigen, Christian Puhrsch, and Rob Fergus. Depth map prediction from a single image using a multi-scale deep network. In *Advances in neural information processing systems*, pages 2366–2374, 2014.

[11] Stefan K Gehrig. Large-field-of-view stereo for automotive applications. In *Proc. of OmniVis*, volume 1, 2005.

[12] Kaiming He, Jian Sun, and Xiaoou Tang. Single image haze removal using dark channel prior. *IEEE transactions on pattern analysis and machine intelligence*, 33(12):2341–2353, 2011.

[13] 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.

[14] Yevhen Kuznietsov, Jorg Stuckler, and Bastian Leibe. Semi-supervised deep learning for monocular depth map prediction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6647–6655, 2017.

[15] Iro Laina, Christian Rupprecht, Vasileios Belagiannis, Federico Tombari, and Nassir Navab. Deeper depth prediction with fully convolutional residual networks. In *2016 Fourth International Conference on 3D Vision (3DV)*, pages 239–248. IEEE, 2016.

[16] Thomas Lemaire and Simon Lacroix. Slam with panoramic vision. *Journal of Field Robotics*, 24(1-2):91–111, 2007.

[17] Jie Li, Katherine A Skinner, Ryan M Eustice, and Matthew Johnson-Roberson. Watergan: unsupervised generative network to enable real-time color correction of monocular underwater images. *IEEE Robotics and Automation Letters*, 3(1):387–394, 2018.

[18] Shigang Li. Binocular spherical stereo. *IEEE Transactions on intelligent transportation systems*, 9(4):589–600, 2008.

[19] Fayao Liu, Chunhua Shen, and Guosheng Lin. Deep convolutional neural fields for depth estimation from a single image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5162–5170, 2015.

[20] Tomasz Łuczyński and Andreas Birk. Underwater image haze removal with an underwater-ready dark channel prior. In *OCEANS 2017-Anchorage*, pages 1–6. IEEE, 2017.

[21] Chuiwen Ma, Liang Shi, Hanlu Huang, and Mengyuan Yan. 3d reconstruction from full-view fisheye camera. *arXiv preprint arXiv:1506.06273*, 2015.

[22] Fangchang Mal and Sertac Karaman. Sparse-to-dense: Depth prediction from sparse depth samples and a single image. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1–8. IEEE, 2018.

[23] Yan-Tsung Peng, Xiangyun Zhao, and Pamela C Cosman. Single underwater image enhancement using depth estimation based on blurriness. In *2015 IEEE International Conference on Image Processing (ICIP)*, pages 4952–4956. IEEE, 2015.

[24] Max Pfingsthorn, Andreas Birk, Sören Schwertfeger, Heiko Bülow, and Kaustubh Pathak. Maximum likelihood mapping with spectral image registration. In *2010 IEEE International Conference on Robotics and Automation*, pages 4282–4287. IEEE, 2010.

[25] Monika Roznere and Alberto Quattrini Li. Real-time model-based image color correction for underwater robots. *CoRR*, abs/1904.06437, 2019.

[26] Fatih Porikli Saeed Anwar, Chongyi Li. Deep underwater image enhancement. 2018.

[27] Nathan Silberman and Rob Fergus. Indoor scene segmentation using a structured light sensor. In *2011 IEEE international conference on computer vision workshops (ICCV workshops)*, pages 601–608. IEEE, 2011.

[28] Keisuke Tateno, Nassir Navab, and Federico Tombari. Distortion-aware convolutional filters for dense prediction in panoramic images. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 707–722, 2018.

[29] Qiang Zhao, Wei Feng, Liang Wan, and Jiawan Zhang. Sphorb: A fast and robust binary feature on the sphere. *International journal of computer vision*, 113(2):143–159, 2015.