Received: 23 April 2019 | Accepted: 29 April 2019
DOI: 10.1002/cav.1896

SPECIAL ISSUE PAPER

WILEY

Multitask learning on monocular water images: Surface reconstruction and image synthesis

Xueguang Xie[1,2] | Xiao Zhai[2] | Fei Hou[3,4] | Aimin Hao[2] | Hong Qin[5]

[1]Qingdao Research Institute, Beihang University, Qingdao, China

[2]State Key Laboratory of Virtual Reality Technology and Systems, Beihang University, Beijing, China

[3]State Key Laboratory of Computer Science, Institute of Software, Chinese Academy of Sciences, Beijing, China

[4]University of Chinese Academy of Sciences, Beijing, China

[5]Department of Computer Science, Stony Brook University, Stony Brook, New York

Correspondence
Fei Hou, State Key Laboratory of Computer Science, Institute of Software, Chinese Academy of Sciences, No. 4, Southern Fourth Street, Zhongguancun, Haidian, Beijing 100190, China.
Email: houfei@ios.ac.cn

Hong Qin, Department of Computer Science, Stony Brook University, Stony Brook, NY 11794.
Email: qin@cs.stonybrook.edu

Funding information
National Natural Science Foundation of China, Grant/Award Number: 61872347, 61532002, 61672149, 61672077, and 61872020; National Science Foundation, Grant/Award Number: IIS-1715985 and IIS-1812606; Special Plan for the Development of Distinguished Young Scientists of the Institute of Software, Chinese Academy of Sciences, Grant/Award Number: Y8RC535018; Chinese Academy of Sciences Key Research Program of Frontier Sciences, Grant/Award Number: QYZDY-SSW-JSC041

Abstract

In this paper, we present a new strategy, a joint deep learning architecture, for two classic tasks in computer graphics: water surface reconstruction and water image synthesis. Modeling water surfaces from single images can be regarded as the inverse of image rendering, which converts surface geometries into photorealistic images. On the basis of this fact, we therefore consider these two problems as a cycle image-to-image translation and propose to tackle them together using a pair of neural networks, with the three-dimensional surface geometries being represented as two-dimensional surface normal maps. Furthermore, we also estimate the imaging parameters from the existing water images with a subnetwork to reuse the lighting conditions when synthesizing new images. Experiments demonstrate that our method achieves an accurate reconstruction of surfaces from monocular images efficiently and produces visually plausible new images under variable lighting conditions.

KEYWORDS

image synthesis, multitask learning, water surface reconstruction

Xueguang Xie and Xiao Zhai contributed equally and should be regarded as co-first authors.

1 | INTRODUCTION

Water surface modeling has always been an intriguing but tricky subject in computer graphics and computer vision. Some researchers achieved accurate results with professional capturing devices.[1,2] However, their methods are difficult to reproduce in daily life due to the necessity of delicate setups. Another possible solution is to use the shape-from-shading (SFS) technique to recover water surfaces from videos,[3,4] but this does not handle highlights and occlusions well and brings geometric distortions. Machine learning–based methods[5–8] have shown promising progress in three-dimensional (3D) reconstruction and depth estimation recently. Moreover, some machine learning–based fluid simulators have also been developed.[9,10] Strongly inspired by these most recent works, we advocate seeking novel strategies for water surface reconstruction using the machine learning approach.

Photorealistic rendering of water contents is another complicated topic. Although many successful techniques have been developed in the film industry, they are usually computationally demanding because the intrinsic optical properties of water are sophisticated and could lead to various lighting effects. Kallweit et al.[11] proposed to tackle a similar problem, rendering atmospheric clouds, by synthesizing the high-order scattering with neural networks. Comparably, we also utilize a deep learning method in this paper to achieve the fast synthesizing/rendering of wave images.

Given the apparent inverse relation of the above two goals, we propose a joint network architecture to efficiently handle water surface reconstruction and water rendering together instead of solving them separately (see Figure 2). Conceptually, our multitask learning framework includes a forward surface estimation net and a backward surface rendering net, which is very similar to the work on cycle image-to-image transformation.[12] To reuse the lighting situation from the source images in the rendering process, we further employ an extra subnetwork to encode the imaging parameters in the forward part and exert those parameters on the backward part. The inverse relation of the two parts is exploited to expedite the learning process of each other and improve overall robustness. With the joint network trained, our work makes estimating fluid geometries and editing fluid images/videos possible in real time.

The main contributions of this paper are

1. a method to simultaneously estimate water surface geometries and the imaging parameters from a single image,
2. an image synthesizing function to apply lighting conditions on water surfaces without the physics-based modeling, and
3. a joint learning paradigm to solve the correlative 3D reconstruction and image synthesizing tasks on fluids in conjunction.

2 | RELATED WORK

The topic of this paper is closely related to fluid surface reconstruction, the rendering of fluid contents, and image-to-image translation. In this section, we briefly review them in the following categories.

Fluid surface reconstruction is challenging due to the fact that fluids are often transparent and are hard to capture directly. Some researchers employed camera arrays to capture the fluid surface through its optical properties.[1,2] The constraints on consistency or physics[13,14] can also be exploited for this purpose using optimization-based methods.

These methods are relatively accurate but difficult to carry out without expensive devices or delicate setups. To overcome the drawback, some works have focused on surface estimation from monocular videos via SFS. Li et al.[4] and Yu and Quan[3] acquired water surface from videos combining SFS with the shallow water model and the Stokes wave model correspondingly. Similarly, Eckert et al.[15] reconstructed the density and motion of smoke based on monocular videos by resorting to physical constraints. Such methods are, overall, easier to proceed, but the results are also relatively inaccurate and error prone, especially when dealing with surface occlusions and highlights. Recently, Xie et al.[9] have taken advantage of machine learning to reconstruct high-resolution fluid details from low-resolution velocities or vorticities. In this paper, we seek a new strategy to reconstruct water geometry from a monocular RGB image through deep learning.

Photorealistic rendering of fluids dates back to the work of Fournier and Reeves.[16] Tessendorf[17] presented a sophisticated lighting model for a realistic reproduction of ocean waves, whereas Ashikhmin et al.[18] and Premoze and Ashikhmin[19] presented a light transport approach for the complex lighting effects of the ocean. However, the traditional physically based rendering is time consuming and impractical for real-time applications. Hu et al.[20] and Schneider and Westermann[21] studied real-time water surface rendering, whereas Bruneton et al.[22] accelerated the rendering by hardware and incorporated more sophisticated lighting models, such as reflection, refraction, and the Fresnel term.

To avoid the computationally demanding requirements, the most recent research studies have adopted data-driven approaches to synthesize images. Kallweit et al.[11] proposed a neural network framework to render the atmospheric clouds. Nalbach et al.[23] used convolutional neural networks to synthesize ambient occlusion, illumination, and other effects in screen space. Kato et al.[24] also adopted a neural 3D mesh renderer in which a simple ambient light and directional light is used without shading. These machine learning methods achieved plausible results much more efficiently than traditional renders. Similarly, in this paper, we propose a network to function as a renderer given surface geometries and lighting conditions.

Image-to-image translation has made plentiful progress since the seminal work by Goodfellow et al.[25] Isola et al.[26] proposed the pix2pix framework, which uses the conditional generative adversarial network (cGAN)[27] to learn a mapping from input to output images. However, this method relied on paired data for supervised learning. To avoid this prerequisite, CycleGAN,[12] DiscoGAN,[28] and DualGAN[29] were designed following the cycle consistency for training using unpaired data. These series of methods have been proven effective in various tasks, such as collection style transfer, object transfiguration, season transfer, and generating photographs from sketches.[30] Motivated by them, we learn the two-way mapping from water images and their corresponding surface geometries, with forward mapping reconstructing the water surfaces and backward mapping synthesizing photorealistic images of waves. Moreover, we include a subnetwork in our framework to extract and reuse the lighting conditions from existing images.

3 | METHOD

In this paper, we propose a deep learning method to reconstruct water surfaces and synthesize water images. A multi-task cycle network based on image-to-image transformation is designed for these two tasks. This section introduces the network architecture and details the involved formulation in our method.

3.1 | Method overview

Our model, illustrated in Figure 1, is an end-to-end cycle deep learning framework. The two ends are respectively the RGB water images and the normal maps with imaging parameters, including the viewpoint, light position, etc. The whole cycle network consists of a forward estimation network E and a backward rendering network R. Specifically, the estimation net E, taking a single color image as input, produces a normal map and the imaging parameters accordingly; the backward net R, which is essentially a renderer, synthesizes the water image according to the input normal map and the imaging condition.

The detailed overview of our network is illustrated in Figure 2. Both the forward net E and the backward net R contain an encoder–transformation–decoder structure for image-to-image translation. Additionally, to reuse the lighting conditions of existing images, a Sub-Net 2 for extracting imaging parameters is included in E. This subnetwork shares features from the encoding block and captures the exact imaging parameters in a one-dimensional vector through a convolution layer, a pooling layer, and two linear layers. To conveniently insert the extracted lighting conditions into R without further interpreting the imaging parameters, we regard the two-dimensional (2D) feature before the linear layers as the representation of imaging parameters. This 2D feature can be appended directly to the intermediate feature of R due to the compatible dimensionality. E and R can be used together or individually. However, due to the extraction and application of imaging parameters, the training process of E net R would influence each other. Therefore, the whole network needs to be trained jointly.

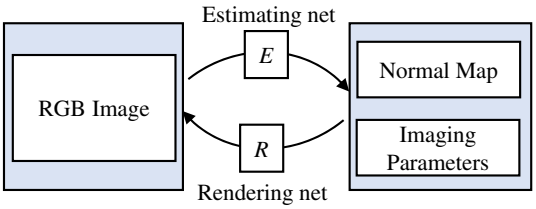

FIGURE 1 The sketch of our cycle network

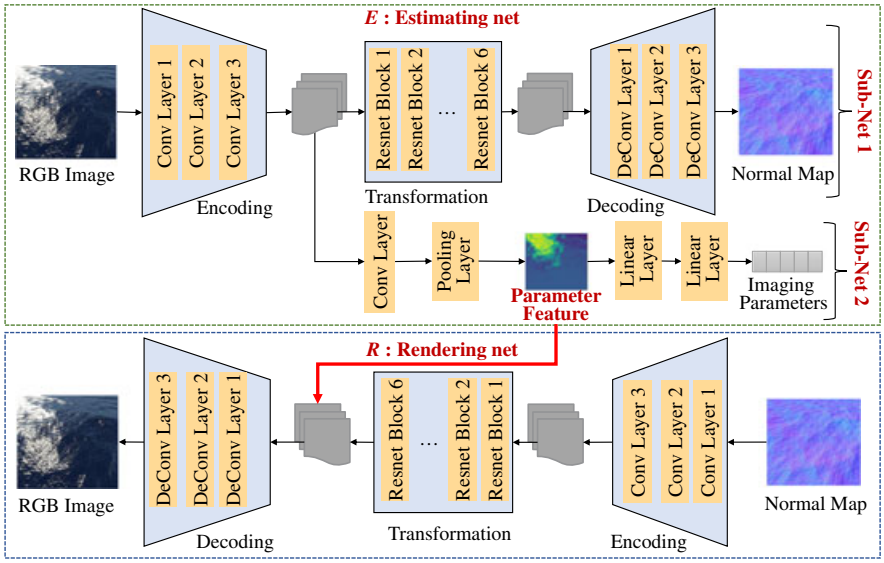

FIGURE 2 Network architecture. Our network includes a forward estimating net E and a backward rendering net R, both based on the encoder–transformation–decoder structure. The estimating net E also includes a Sub-Net 2 for extracting imaging parameters to reuse the lighting conditions in the backward rendering pass of net R

3.2 | Formulation

Our goal is to learn the two-way mapping functions between two domains, namely, a set of RGB images $\{c_i\}$ and a set of normal maps $\{n_i\}$ with imaging parameters $\{p_i\}$. To achieve this, we design appropriate loss functions to measure the performance of the learned mappings.

The generative adversarial network proposed by Goodfellow et al.[25] contains two components, the generator and the discriminator, where the generator mimics the images from the provided data set, whereas the discriminator tries to spot the generated fakes. Through competing against each other, the generator could learn the intrinsic distribution and gives improving results. Essentially, the discriminator plays the part of the loss function in the training process. However, this configuration guarantees no convergence toward the desired direction and could lead to unpredictable results. In contrast, the cGAN is proposed to learn a mapping from the observed image and random noise vector to the target, which solves the above problem. In addition, cGANs penalize the joint configuration of the output, whereas the L_1 and L_2 losses treat the output space as "unstructured" in the sense that each output pixel is considered conditionally independent from all others. Hence, we build our model based on cGAN.

In our formulation, the forward network E and the backward network R are two generators, and their corresponding discriminators D_E and D_R are used for binary classifying the generated images. Therefore, we have $\hat{n}_i, \hat{p}_i, \hat{p}_i^{2D} = E(c_i)$, and $\hat{c}_i = R(n_i, \hat{p}_i^{2D})$, where $\hat{c}_i, \hat{n}_i, \hat{p}_i$ represent the inferred c_i, n_i, p_i through networks and $\hat{p}_i^{2D}$ is the aforementioned 2D feature. The subscript i is omitted hereinafter for brevity.

Normal loss. The discriminator D_E is used to measure the high-level consistency between the ground truth n and the generated normal $\hat{n}$. At the same time, the per-pixel cosine similarity, which is the dot product[31] when the vectors are normalized to the unit l_2-norm, is also evaluated. The normal loss L_N is therefore defined as

$$\begin{aligned} L_{\cos}(E) &= \mathbb{E}_{n,\hat{n}}[1 - \cos(n, \hat{n})], \\ L_{\text{cGAN}}(E, D_E) &= \mathbb{E}_{c,\hat{n}}[\log(1 - D_E(c, \hat{n}))] + \mathbb{E}_{c,n}[\log D_E(c, n)], \\ L_N &= \lambda L_{\cos}(E) + L_{\text{cGAN}}(E, D_E), \end{aligned} \tag{1}$$

where λ is the weight of $L_{\cos}(E)$.

Image loss. The generated images $\hat{c}$ need to be as approximative as possible to the ground truth c and to exhibit the same distribution of the domains given training samples. However, it is well known that the L_1 and L_2 losses are prone to produce blurry results on image generation problems.[26] Although these losses fail to encourage high-frequency crispness, they are able to capture the low frequencies efficiently in most cases. In contrast, a generative adversarial network discriminator is capable of modeling high-frequency structures accurately but at a high computation cost. To optimize our networks, we follow the work of Isola et al.,[26] which relies on the cGAN discriminator D_R and an L_1 term to enforce

high-frequency and low-frequency correctness, respectively. The image loss L_I is therefore defined as

$$\begin{aligned} L_1(R) &= \mathbb{E}_{c,\hat{c}}[\|c-\hat{c}\|_1], \\ L_{\text{cGAN}}(R,D_R) &= \mathbb{E}_{n,c}[\log D_R(n,c)] + \mathbb{E}_{n,\hat{c}}[\log(1-D_R(n,\hat{c}))], \\ L_I &= \alpha L_1(R) + L_{\text{cGAN}}(R,D_R), \end{aligned} \tag{2}$$

where α is the weight of $L_1(R)$.

Lighting loss. The estimated imaging parameters should be consistent with the ground truth. For easy problems where this is the case, we do not need a loss function more complicated than L_1 to enforce coherence. Thus, the lighting loss L_{Param} can be defined as

$$L_{\text{Param}} = \mathbb{E}_{p,\hat{p}}[\|p-\hat{p}\|_1]. \tag{3}$$

Although the estimated $\hat{p}$ is not involved in the synthesis directly, it enforces the 2D feature $\hat{p}^{2D}$ to encode the essential imaging information, including the viewpoint and light position.

Universality loss. The parameter feature $\hat{p}^{2D}$ applied to the rendering net should be independent of surface geometry. To ensure this, in every training pass, $\hat{p}^{2D}$ is applied to render an extra normal map n' that has not been exposed to the forward net E into image $\hat{c}' = R(n', \hat{p}^{2D})$. The universality loss L'_I is employed, comparing $\hat{c}'$ with the ground-truth image c', as

$$\begin{aligned} L'_1(R) &= \mathbb{E}_{c',\hat{c}'}[\|c'-\hat{c}'\|_1], \\ L'_{\text{cGAN}}(R,D_R) &= \mathbb{E}'_{n',c}[\log D_R(n',c')] + \mathbb{E}_{n',\hat{c}'}[\log(1-D_R(n',\hat{c}'))], \\ L'_I &= \beta L'_1(R) + L'_{\text{cGAN}}(R,D_R), \end{aligned} \tag{4}$$

where β is the weight of $L'_1(R)$.

Full objective. Our full objective is

$$L(E,R,D_E,D_R) = \gamma L_{\text{Param}} + L_N + L_I + L'_I, \tag{5}$$

where γ controls the importance of lighting loss. We aim to solve the following optimization to train the network and achieve the optimal results E^* and R^*:

$$E^*, R^* = \arg\min_{E,R}\max_{D_E,D_R} L(E,R,D_E,D_R). \tag{6}$$

In all the experiments of this paper, λ, α, β, and γ involved in our equations are set to 10.

4 | EXPERIMENT

To evaluate the proposed method, we implemented the network and trained it with simulated data. This section documents the details of the data set generation, the network setup, and the training procedure.

Generation of data sets. Considering the fact that collecting paired water images and surface geometries in high definition is difficult in real life, we use an existing fluid simulator and renderer[22] to produce data demanded as the ground truth. Our data set includes all inputs for training purposes, namely, the images of water surfaces, the corresponding surface normal maps, and the imaging parameters, captured from various camera viewpoints and lighting directions. The imaging parameters are simplified to a five-dimension vector, with three dimensions for the lighting direction, one dimension for viewpoint height, and one dimension for viewing angle. In some circumstances, the water surface is not enough to fill the entire image, and sky appears in the images. In this case, we mask the sky instead of cropping the images. Figure 3 enumerates some typical examples in our data set. Our data set has 450 paired items, among which 400 are randomly selected as the training set and the rest as the test set.

Network setup. Our forward Sub-Net 1 and backward network contains three convolutions, several residual blocks, and three deconvolutions. The forward Sub-Net 2 following the encoding block includes a convolution layer, an average pooling layer, and two linear layers. The details of the network are demonstrated in Figure 4. All generators and discriminators use modules of the form convolution-InstanceNorm-ReLu, except for the last layers that uses tanh activation. We use six Resnet blocks for 256 × 256 training images. For the discriminator networks, we use 70 × 70 PatchGANs,[12,26] which distinguish real or fake images using 70 × 70 patches instead of the full 256 × 256 image. Such a patch-level discriminator has fewer parameters to train and could be easily applied to arbitrarily sized images in a fully convolutional fashion.

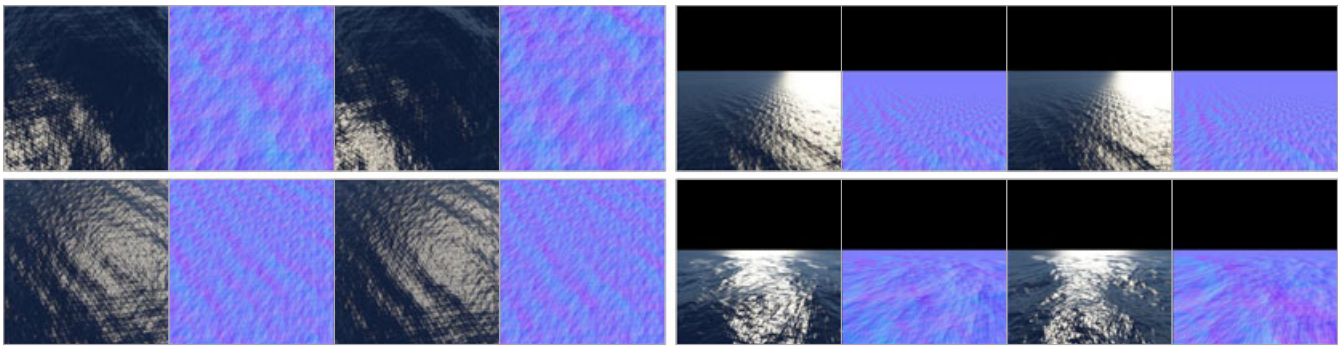

FIGURE 3 Data examples. Each item consists of two pairs of the water images and the corresponding surface normal maps with the same imaging parameters. In case that sky appears in the image, we mask the sky instead of cropping the image

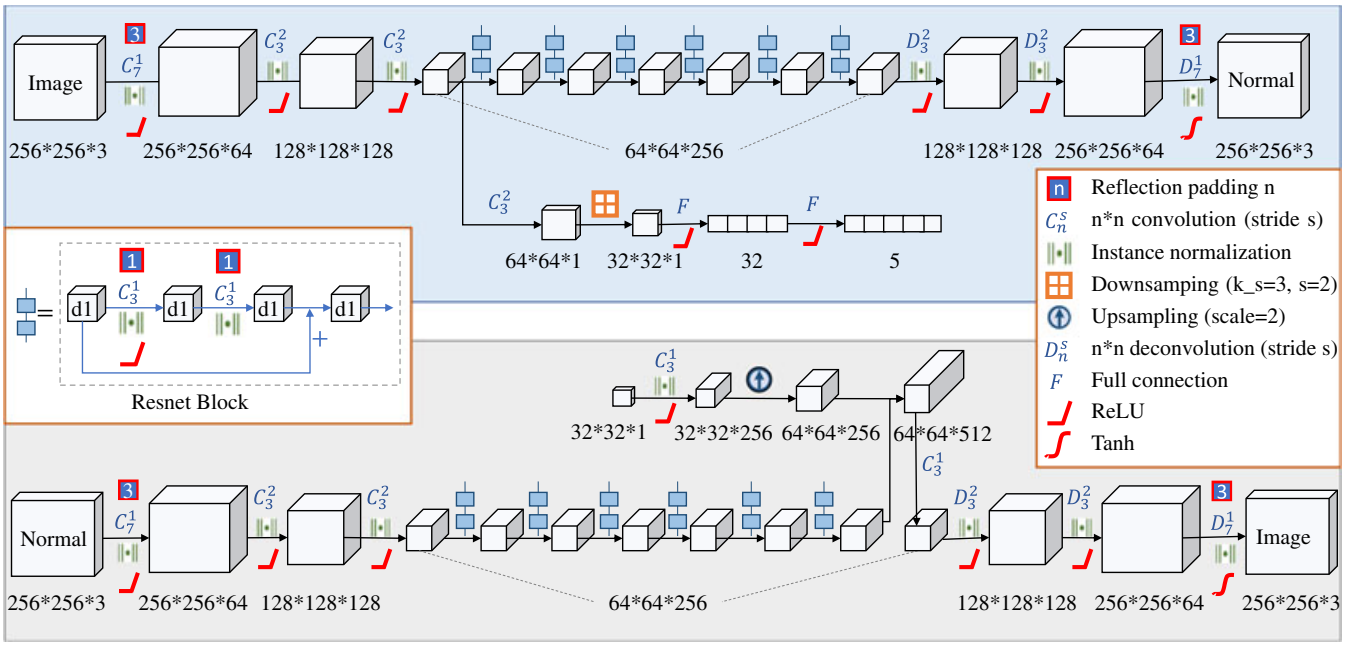

FIGURE 4 A full description of the networks. The part on top and the part on the bottom are the forward network and the backward network, respectively. On the left are the details of the Resnet block

Training procedure. If the joint network is trained from scratch, the parameter features produced by the forward network E could be pretty random noise, which is not much meaningful for the backward network R. Therefore, we split the training process into two phases. First, the E net was trained with L_N on its own for 50 epochs; afterward, the joint network was trained with $L(E, R, D_E, D_R)$ for another 200 epochs. We alternated the gradient descent steps on the discriminators and the generators, which is commonly seen for training GANs. During this process, the Adam solver with a batch size of 1 was applied. All networks were trained with a constant learning rate of 0.0004 until epoch 50 of the joint network training (phase 2), after which the learning rate was linearly dropped to zero over the final 150 epochs.

5 | RESULTS

This section and the Supplementary Video provide the results of our method and the comparisons against the baseline methods, including pix2pix[26] and CycleGAN.[12] The influences of several loss terms are also studied by comparing the full loss function versus its variants. Additionally, we provide examples of reconstructing real-life water photos with our network.

TABLE 1 The time statistics of training and inferring

Method	pix2pix (Reconstruction)	pix2pix (Synthesis)	CycleGAN	Our method
Training (hr)	3.2	4.0	11.5	6.3
Testing (ms)	5.9	6.2	19.6	16.7

We implemented the network using Python 3.6 with pyTorch 0.4.0 and performed the training/testing for this paper using a Windows 10 personal computer with an Intel Core i7-6700 CPU, an NVIDIA GeForce GTX1080 GPU, CUDA 9.0, and cuDNN 7 installed. The time for training and inferring is listed in Table 1.

Water surface reconstruction. Several examples of surface reconstruction using the previous pix2pix, CycleGAN, and our method can be seen in Figure 5, with the 3D surface reconstructed from normals using the work of Agrawal et al.[32] Simultaneously, 3D results reconstructed from SFS are shown as well. Our method achieves slightly better results than pix2pix (see Supplementary Video) since the cosine similarity $L_{\cos}$ is more suitable to evaluate normal maps than the L_1 loss in pix2pix, whereas CycleGAN fails to render satisfactory results and handles the sky mask erroneously due to the lack of supervision. SFS achieves the worst results in all methods, especially in the highlights.

We measure the performance of predicted normals with the same metrics as in the work of Eigen and Fergus[33]: the mean and median angle from the ground truth across all pixels, as well as the percentage of vectors whose angle falls within three thresholds. We also quantize the 3D surface errors with diversified metrics.[31] The results are shown in Table 2. Our model performs similarly or slightly better than pix2pix and substantially outperforms CycleGAN.

Water image synthesis. We apply our backward network and previous works, pix2pix and CycleGAN, to synthesize water images on the test data set. The results are shown in Figure 6. Again, our method is able to accurately reproduce the scene, whereas CycleGAN fails to generate correct results. Moreover, pix2pix also produces the water geometries,

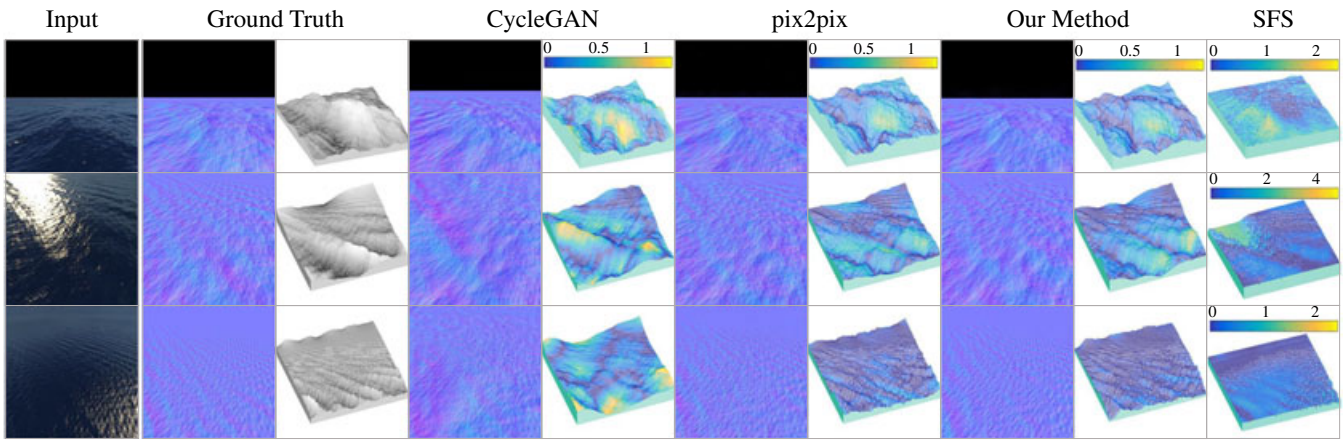

FIGURE 5 A comparison of the water surface reconstruction (including the surface normal and the surface geometry). The reconstructed surfaces are color coded according to the error from ground truth. In this comparison, our approach and pix2pix achieve comparably accurate results, whereas CycleGAN fails to offer satisfactory reconstruction. The shape-from-shading (SFS) method has the least accurate results, especially in the highlights

TABLE 2 The quality measurements of reconstructed normals and three-dimensional (3D) surface against the ground truth

Method	Normals					3D surface		
	Angle distance (°)		Within $t°$ deg. (%)			Height distance		
	Mean	Median	5°	10°	15°	abs rel	sqr rel	RMSE
SFS	–	–	–	–	–	0.573	0.335	0.397
CycleGAN	13.69	9.74	26.0	50.1	70.9	0.485	0.188	0.266
pix2pix	7.17	5.99	41.1	72.2	89.9	0.214	0.038	0.113
Our method	6.78	5.57	43.7	75.3	91.7	0.190	0.032	0.100

Note. RMSE = root-mean-square error; SFS = shape-from-shading method.

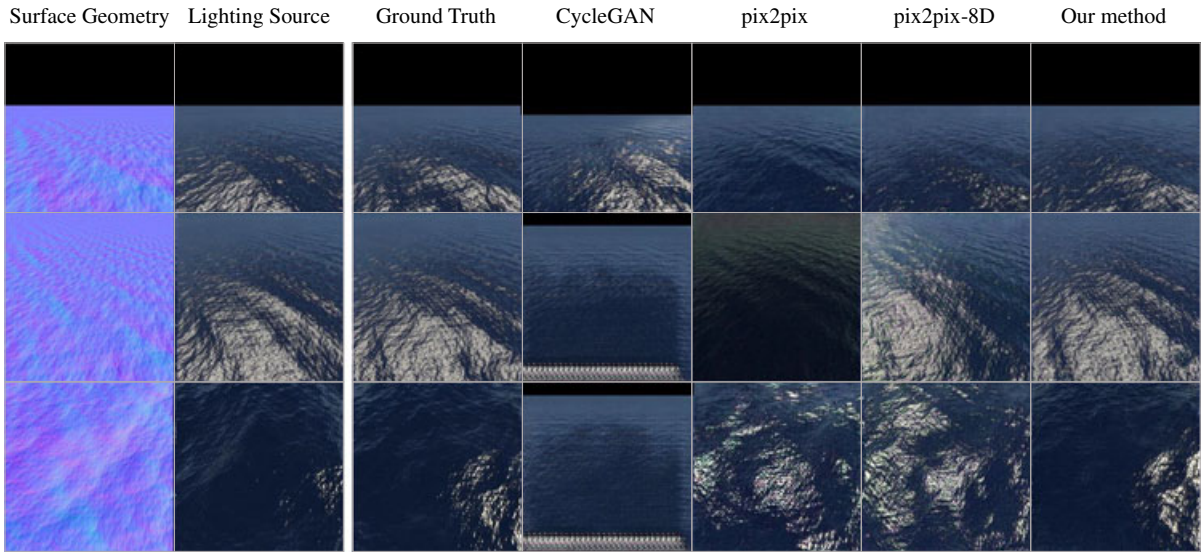

FIGURE 6 A comparison of the image synthesis using different lighting conditions and geometries. Our approach achieves photorealistic water images with correct lighting. In contrast, CycleGAN handles the sky mask and lighting erroneously, whereas pix2pix is capable of accurately recovering the water geometries. However, despite being trained with the imaging parameters, pix2pix-8D still cannot reproduce the lighting correctly

TABLE 3 The quality measurements of rendering image

Method	MSE	PSNR	MAE	SSIM
CycleGAN	3272	13.2	36.7	0.28
pix2pix	3671	13.2	36.0	0.32
pix2pix-8D	3575	13.8	37.2	0.25
Our method	1314	18.2	18.2	0.55

Note. MSE = mean-squared error; PSNR = peak signal-to-noise ratio; MAE = mean absolute error; SSIM = structural similarity index.

but the lighting conditions are beyond its capability to recover. To further demonstrate the effectiveness of reusing the lighting conditions, we append the five-dimension imaging parameters to the three-channel normal map and use the new eight-dimension feature as the input of pix2pix. After retraining, the lighting conditions still cannot be correctly captured by this classic method (tagged as pix2pix-8D). In comparison, our subnetwork approach to extract and reuse lighting conditions has no issue under all the tested combinations of surfaces and imaging parameters. Furthermore, we measure the performance of images synthesized by the above methods, as shown in Table 3, where our method outperforms others in all the metrics.

Lighting controllability. Due to a novel subnetwork designed for reusing imaging parameters, we can synthesize the water image with variable light sources, which cannot be achieved straightforwardly by pix2pix or CycleGAN. The examples of reusing the lighting conditions are provided in Figure 7.

Loss function. We compare our full loss against its several variants (see Figure 8). Training without L_{cGAN} makes the results substantially blurry. On the other hand, if $L_{\cos}$ is removed, some degradations in reconstructing highlight areas can be noticed. We also try removing universality loss L'_I, as displayed in Figure 9. In this situation, the lighting is not completely separated from the geometries by the subnetwork and cannot be safely reused in synthesizing new images. We therefore conclude that all the losses are critical to our network.

Reconstruction from real photos. We further test our trained surface reconstruction network on real water photos and achieve better results than the well-known SFS method, as shown in Figure 10, although some errors still occur in the highlight regions. Being data driven, the performance of our method depends largely on how well the training set represents the actual application scenario. Tailoring our method to a particular kind of water or specific waveforms is thus relatively easy, provided that enough training data are available.

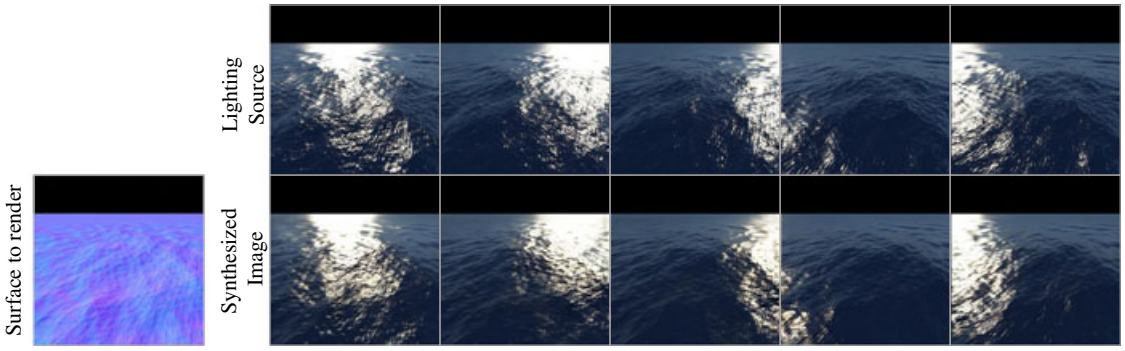

FIGURE 7 Examples of lighting controllability. The surface normal (bottom left) is rendered using lighting extracted from different images (top row). Equipped with this function, our method offers a viable editing tool for water images and videos

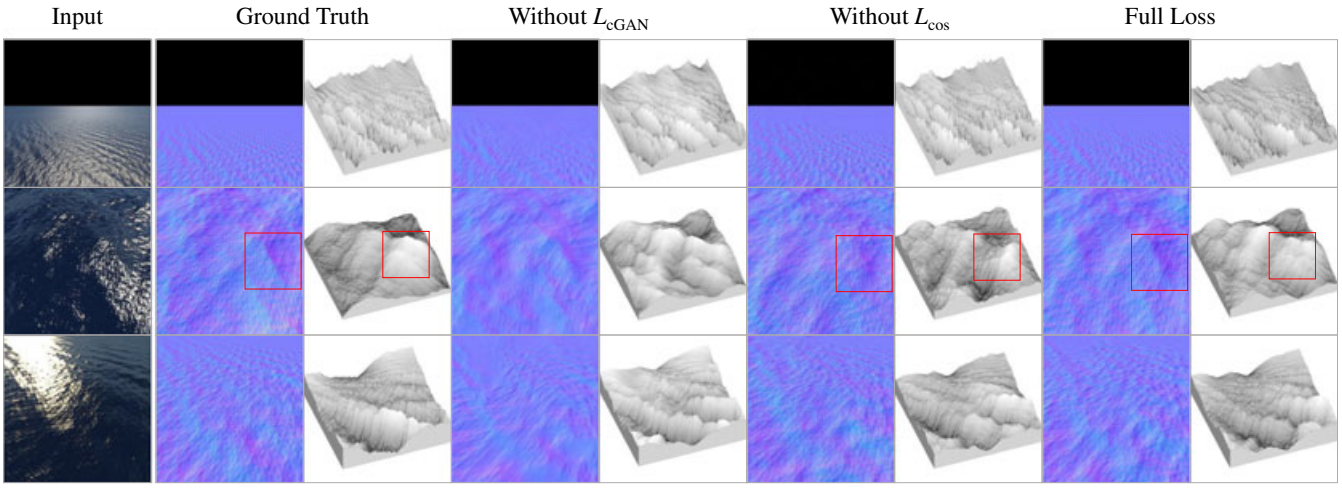

FIGURE 8 Comparison among various variants of the loss function. The result trained without L_{cGAN} is substantially blurry. Without L_{cos}, some deviations in the highlight areas can be noticed, marked with red boxes

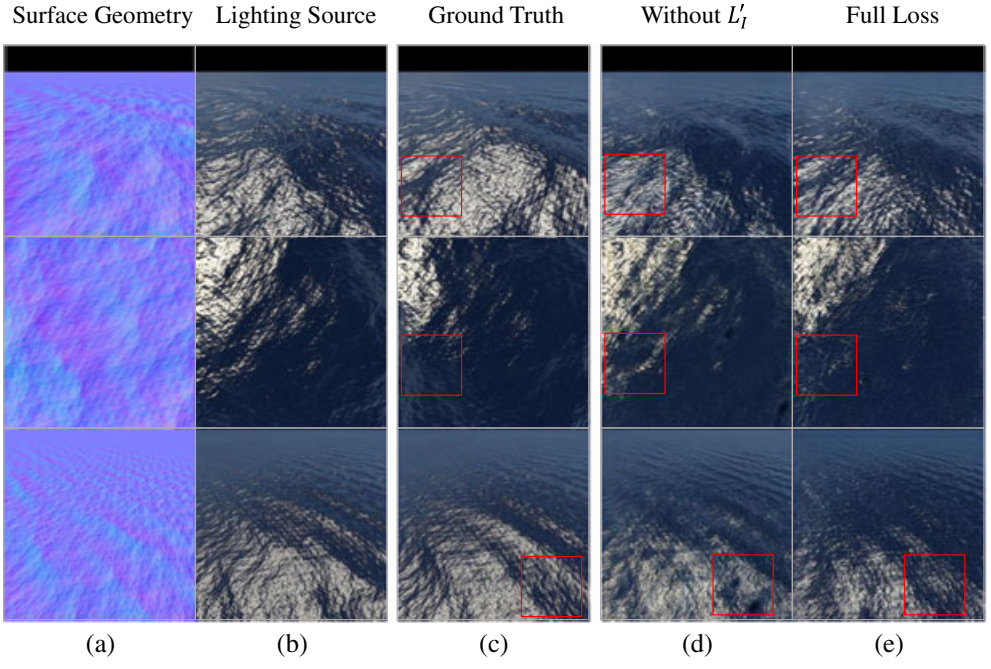

FIGURE 9 Rendering with or without the universality loss L'_I. Without L'_I, the lighting is not completely separated from the geometries and cannot be safely reused to synthesize new images

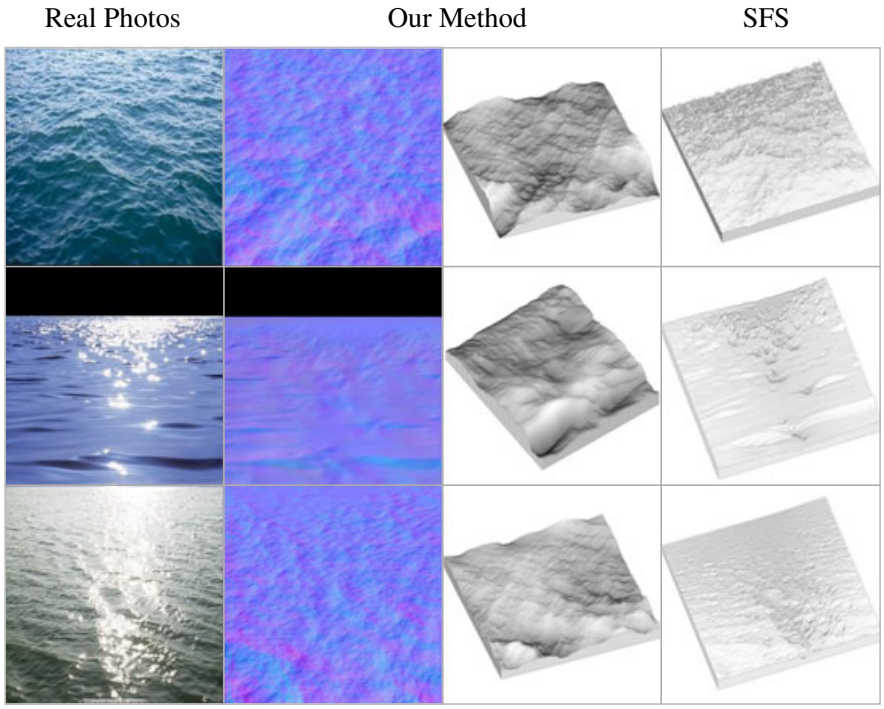

FIGURE 10 Reconstruction of surface normals and geometries from real photos with our method and the shape-from-shading (SFS) method

6 | CONCLUSION

We have presented a novel approach for two tasks, monocular water surface reconstruction and realistic synthesis of water images, by building a multitask deep learning network based on the cyclic bilateral translation between water images and surface geometries with imaging parameters. Abundant experimental results validate that our method is both visually plausible and computationally efficient. The key and novel ingredient of our method is the subnetwork for extracting and applying imaging parameters, which makes the image synthesis flexible. We further improve the performance using other techniques, for example, adding universality loss to make the estimated parameters independent of geometries. Moreover, our method could be regarded as a viable solution for content editing on fluid images and videos. The source code is available at https://github.com/XueguangXie-BUAA/Water-Surface-Reconstruction-and-Image-Synthesis.

ACKNOWLEDGEMENTS

This work was partially supported by National Natural Science Foundation of China under Grants 61872347, 61532002, 61672149, 61672077, and 61872020; National Science Foundation of USA under Grants IIS-1715985, IIS-1812606 IIS-0949467, IIS-1047715 and IIS-1049448; Special Plan for the Development of Distinguished Young Scientists of the Institute of Software, Chinese Academy of Sciences, under Grant Y8RC535018; Chinese Academy of Sciences Key Research Program of Frontier Sciences under Grant QYZDY-SSW-JSC041; National Key R&D Program of China under Grant 2017YFF0106407; and Applied Basic Research Program of Qingdao under Grant 161013xx.

ORCID

Xiao Zhai https://orcid.org/0000-0001-8964-3704
Fei Hou https://orcid.org/0000-0001-8226-6635

REFERENCES

1. Ding Y, Li F, Ji Y, Yu J. Dynamic fluid surface acquisition using a camera array. In: Metaxas DN, Quan L, Sanfeliu A, Gool LJV, editors. IEEE International Conference on Computer Vision, ICCV 2011, Barcelona, Spain, November 6–13, 2011. Washington, DC: IEEE Computer Society; 2011. p. 2478–2485.

2. Ye J, Ji Y, Li F, Yu J. Angular domain reconstruction of dynamic 3D fluid surfaces. In: 2012 IEEE Conference on Computer Vision and Pattern Recognition, Providence, RI, USA, June 16–21, 2012. Washington, DC: IEEE Computer Society; 2012. p. 310–317.
3. Yu M, Quan H. Fluid surface reconstruction based on specular reflection model. J Vis Comput Animat. 2013;24(5):497–510.
4. Li C, Pickup D, Saunders T, et al. Water surface modeling from a single viewpoint video. IEEE Trans Vis Comput Graph. 2013;19(7):1242–1251.
5. Girdhar R, Fouhey DF, Rodriguez M, Gupta A. Learning a predictable and generative vector representation for objects. In: Leibe B, Matas J, Sebe N, Welling M, editors. Computer vision: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part VI. Berlin, Germany: Springer; 2016. p. 484–499. Lecture notes in computer science. Vol. 9910.
6. Hane C, Tulsiani S, Malik J. Hierarchical surface prediction for 3D object reconstruction. In: 2017 International Conference on 3D Vision, 3DV 2017, Qingdao, China, October 10–12, 2017. Washington, DC: IEEE Computer Society; 2017. p. 412–420.
7. Fan H, Su H, Guibas LJ. A point set generation network for 3D object reconstruction from a single image. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21–26, 2017. Washington, DC: IEEE Computer Society; 2017. p. 2463–2471.
8. Wang N, Zhang Y, Li Z, Fu Y, Liu W, Jiang Y-G. Pixel2mesh: Generating 3D mesh models from single RGB images. In: Ferrari V, Hebert M, Sminchisescu C, Weiss Y, editors. Computer vision: 15th European Conference, Munich, Germany, September 8–14, 2018, Proceedings, Part XI. Berlin, Germany: Springer; 2018. p. 55–71. Lecture notes in computer science. Vol. 11215.
9. Xie Y, Franz E, Chu M, Thuerey N. tempoGAN: a temporally coherent, volumetric GAN for super-resolution fluid flow. ACM Trans Graph. 2018;37(4). Article No. 95.
10. Sato S, Dobashi Y, Kim T, Nishita T. Example-based turbulence style transfer. ACM Trans Graph. 2018;37(4). Article No. 84.
11. Kallweit S, Müller T, McWilliams B, Gross MH, Novák J. Deep scattering: rendering atmospheric clouds with radiance-predicting neural networks. ACM Trans Graph. 2017;36(6). Article No. 231.
12. Zhu J-Yan, Park T, Isola P, Efros AA. Unpaired image-to-image translation using cycle-consistent adversarial networks. In: IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22–29, 2017. Washington, DC: IEEE Computer Society; 2017. p. 2242–2251.
13. Qian Y, Gong M, Yang Y-H. Stereo-based 3D reconstruction of dynamic fluid surfaces by global optimization. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21–26, 2017. Washington, DC: IEEE Computer Society; 2017. p. 6650–6659.
14. Wang H, Liao M, Zhang Q, Yang R, Turk G. Physically guided liquid surface modeling from videos. ACM Trans Graph. 2009;28(3). Article No. 90.
15. Eckert M-Lena, Heidrich W, Thürey N. Coupled fluid density and motion from single views. Comput Graph Forum. 2018;37(8):47–58.
16. Fournier A, Reeves WT. A simple model of ocean waves. In: Evans DC, Athay RJ, editors. Proceedings of the 13th Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH 1986, Dallas, Texas, USA, August 18–22, 1986. New York, NY: ACM; 1986. p. 75–84.
17. Tessendorf J. Simulating ocean water. SIGGRAPH'99 Course Note; 2001.
18. Ashikhmin M, Premoze S, Shirley P, Smits B. A variance analysis of the metropolis light transport algorithm. Comput Graph. 2001;25(2):287–294.
19. Premoze S, Ashikhmin M. Rendering natural waters. Comput Graph Forum. 2001;20(4):189–200.
20. Hu Y, Velho L, Tong X, Guo B, Shum H. Realistic, real-time rendering of ocean waves. J Vis Comput Animat. 2006;17(1):59–67.
21. Schneider J, Westermann R. Towards real-time visual simulation of water surfaces. In: Ertl T, Girod B, Niemann H, Seidel H-P, editors. Proceedings of the Vision Modeling and Visualization Conference 2001 (VMV-01), Stuttgart, Germany, November 21–23, 2001. Augsburg, Germany: AKA GmbH; 2001. p. 211–218.
22. Bruneton E, Neyret F, Holzschuch N. Real-time realistic ocean lighting using seamless transitions from geometry to BRDF. Comput Graph Forum. 2010;29(2):487–496.
23. Nalbach O, Arabadzhiyska E, Mehta D, Seidel H-P, Ritschel T. Deep shading: convolutional neural networks for screen space shading. Comput Graph Forum. 2017;36(4):65–78.
24. Kato H, Ushiku Y, Harada T. Neural 3D mesh renderer. In: 2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18–22, 2018. Washington, DC: IEEE Computer Society; 2018. p. 3907–3916.
25. Goodfellow IJ, Pouget-Abadie J, Mirza M, et al. Generative adversarial nets. In: Ghahramani Z, Welling M, Cortes C, Lawrence ND, Weinberger KQ, editors. Advances in neural information processing systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8–13, 2014, Montreal, Quebec, Canada. San Diego, CA: Neural Information Processing Systems Foundation; 2014. p. 2672–2680.
26. Isola P, Zhu JY, Zhou T, Efros AA. Image-to-image translation with conditional adversarial networks. Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR); 2017 Jul 21–26; Honolulu, HI. Washington, DC: IEEE Computer Society; 2017.
27. Mirza M, Osindero S. Conditional generative adversarial nets. arXiv preprint arXiv:1411.1784. 2014.
28. Kim T, Cha M, Kim H, Lee JK, Kim J. Learning to discover cross-domain relations with generative adversarial networks. In: Precup D, Teh YW, editors. Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6–11 August 2017. JMLR.org; 2017. p. 1857–1865. Proceedings of Machine Learning Research. Vol 70.

29. Yi Z, Zhang H(R), Tan P, Gong M. DualGAN: Unsupervised dual learning for image-to-image translation. In: IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22–29, 2017. Washington, DC: IEEE Computer Society; 2017. p. 2868–2876.
30. Sangkloy P, Lu J, Fang C, Yu F, Hays J. Scribbler: controlling deep image synthesis with sketch and color. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21–26, 2017. Washington, DC: IEEE Computer Society; 2017. p. 6836–6845.
31. Eigen D, Puhrsch C, Fergus R. Depth map prediction from a single image using a multi-scale deep network. In: Ghahramani Z, Welling M, Cortes C, Lawrence ND, Weinberger KQ, editors. Advances in neural information processing systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8–13, 2014, Montreal, Quebec, Canada. San Diego, CA: Neural Information Processing Systems Foundation; 2014. p. 2366–2374.
32. Agrawal AK, Raskar R, Chellappa R. What is the range of surface reconstructions from a gradient field?. In: Leonardis A, Bischof H, Pinz A, editors. Computer vision: 9th European Conference on Computer Vision, Graz, Austria, May 7–13, 2006. Proceedings, Part I. Berlin, Germany: Springer, 2006. p. 578–591. Lecture notes in computer science. Vol. 3951.
33. Eigen D, Fergus R. Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. In: 2015 IEEE International Conference on Computer Vision, ICCV 2015, Santiago, Chile, December 7–13, 2015. Washington, DC: IEEE Computer Society; 2015. p. 2650–2658.

AUTHOR BIOGRAPHIES

Xueguang Xie received her BS degree in information and computing science from Inner Mongolia University in 2014 and her MS degree in software engineering from Beihang University in 2017. She is currently a PhD candidate at State Key Laboratory of Virtual Reality Technology and Systems, Beihang University. Her research interests include physics-based fluid simulation, deep learning-based fluid animation in computer graphics, etc.

Xiao Zhai received his BS degree in computer science and engineering from Beihang University in 2013. He is currently a PhD candidate at State Key Laboratory of Virtual Reality Technology and Systems, Beihang University. His research interests include physics-based fluid simulation, data-driven fluid animation, and all the relevant topics in computer graphics.

Fei Hou received his PhD degree in computer science from Beihang University in 2012. He is currently a research associate professor of Institute of Software, Chinese Academy of Sciences. He was a Postdoctoral researcher at Beihang University from 2012 to 2014 and a research fellow in School of Computer Science and Engineering, Nanyang Technological University from 2014 to 2017. His research interests include geometry processing, image-based modeling, data vectorization, medical image processing, etc.

Aimin Hao received his BS, MS, and PhD degrees in computer science at Beihang University. He is a professor in School of Computer Science and Engineering, Beihang University and the associate director of State Key Laboratory of Virtual Reality Technology and Systems. His research interests include virtual reality, computer simulation, computer graphics, geometric modeling, image processing, and computer vision.

Hong Qin received his BS and MS degrees in computer science from Peking University, and his PhD degree in computer science from the University of Toronto. He is a professor of computer science in Department of Computer Science at Stony Brook University. His research interests include geometric and solid modeling, graphics, physics-based modeling and simulation, computer-aided geometric design, human-computer interaction, visualization, and scientific computing. Currently, he serves as an associate editor for The Visual Computer, Graphical Models, and Journal of Computer Science and Technology. He is a senior member of the IEEE and the IEEE Computer Society.

SUPPORTING INFORMATION

Additional supporting information may be found online in the Supporting Information section at the end of the article.

How to cite this article: Xie X, Zhai X, Hou F, Hao A, Qin H. Multitask learning on monocular water images: Surface reconstruction and image synthesis. *Comput Anim Virtual Worlds*. 2019;30:e1896. https://doi.org/10.1002/cav.1896