Received January 3, 2019, accepted January 28, 2019, date of publication February 19, 2019, date of current version March 5, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2899221

2D-to-Stereo Panorama Conversion Using GAN and Concentric Mosaics

JIE LU[1,2,3], YANG YANG[4], RUIYANG LIU[2], SING BING KANG[5], (Fellow, IEEE), AND JINGYI YU[2], (Member, IEEE)

[1]Shanghai Institute of Microsystem and Information Technology, Chinese Academy of Sciences, Shanghai 200050, China
[2]School of Information Science and Technology, ShanghaiTech University, Shanghai 201210, China
[3]University of Chinese Academy of Sciences, Beijing 100049, China
[4]DGene Inc., Santa Clara, CA 95050, USA
[5]Microsoft Research, Redmond, WA 98052, USA

Corresponding author: Jie Lu (lujie1@shanghaitech.edu.cn)

This work was supported in part by the Program of Shanghai Subject Chief Scientist under Grant 17XD1402900, in part by the Shanghai Science and Technology Action Plan under Grant 17511108201, and in part by the STCSM under Grant 17JC1403800.

ABSTRACT We describe a learning-based technique to automatically convert a 2-D panorama to its stereoscopic version. In particular, we train a generative adversarial network using perspective stereo pairs as inputs. Given a 2-D panorama, we partition it into overlapping local perspective views. To satisfy the panoramic stereo condition, we generate a sequence of left and right stereo view pairs and stitch them to produce concentric mosaics. We also describe experiments on synthetic and real datasets as well as comparisons with competing state-of-the-art techniques, which validate our technique.

INDEX TERMS 2D-to-stereo panorama, GAN, concentric mosaics, depth peeling loss, selector, stereo panorama synthesis.

I. INTRODUCTION

Capturing panoramic content is made simpler with a wide choice of consumer panoramic imaging systems available on the market. Examples include Ricoh Theta S, Samsung Gear 360, VSN Mobil V.360, Kodak PixPro SP360, 360fly, and Giroptic 360cam, to name just a few. Panoramic content can be used for immersive VR experiences using headsets such as Oculus Rift, HTC Vive, Sony PlayStation VR, Samsung Gear VR, and Google Daydream View. Panoramas can also be visualized and shared on mobile devices, e.g., on Facebook 360 Photos and Google Photo Sphere.

There is a significant amount of legacy 2D panoramic content online; for such content, the ''3Dness'' experience is lacking. In this paper, we show how we can generate a stereoscopic panorama from a single 2D panorama using a supervised learning approach. To conform to the panoramic stereo condition described in [36],[1] we construct concentric mosaics (CMs) [37].

One approach to generate the second view is to directly recover the depth of the original view and then warp. However, this approach has issues associated with disocclusions (which need to be carefully filled to avoid visual discontinuity or implausible appearance) and warping artifacts (especially blur due to resampling). The goal of our work is to generate high-quality views without these artifacts. Generative Adversarial Networks (GANs) [8], [12], [22], [32] are a perfect fit for our work, given their improved performance in appearance prediction compared to regular deep networks. We adapted GAN in two significant ways: (1) we insert a selection layer at the end of the generator (similar to [9] and [42]), and (2) we add a ''pseudo'' depth peeling loss to the objective function.

While GANs are capable of generating sharp-looking views from training stereo pairs, these views tend to lose the original details (exhibiting weird colors). This is especially true for highly textured objects and objects with predicted large disparities. To avoid loss of detail, we add a selection layer (in the spirit of [9] and [42]) at the end of the generator to predict the probability at each disparity level. Furthermore, to better recover an object with significantly predicted disparity, we extend the original GANs design by introducing a ''pseudo'' depth peeling loss (described in Section III-B).

One simple approach would be to partition the panorama into local perspective views (LPVs), infer their stereo

The associate editor coordinating the review of this manuscript and approving it for publication was Zhanyu Ma.

[1] More specifically, Seitz [36] shows that rays in stereo panoramas lie on a family of epipolar hyperboloids.

counterparts using the network, and then stitch them back to produce the second panorama. We explain in Section V-B that this approach creates a stereo pair that violates the panoramic stereo condition [36], [38].

We instead adopt the idea of the concentric mosaic (CM) [26], [27], [37]: each LPV from the input panorama is treated as a central view, with a synthesized left/right stereo pair. All the left images are stitched to form the left panorama, and all the right images are used to generate the right panorama in a similar way. The CM representation satisfies the panoramic stereo condition since it has a circular generator and produces epipolar hyperboloids [36]. We validate our approach on three datasets: (1) synthetic data rendered in 3Ds Max, (2) SUN 360 dataset [41], and (3) real data captured by multiple 360 capture systems. Experiments show that our system can produce high-quality stereo panoramas as shown in Fig. 1.

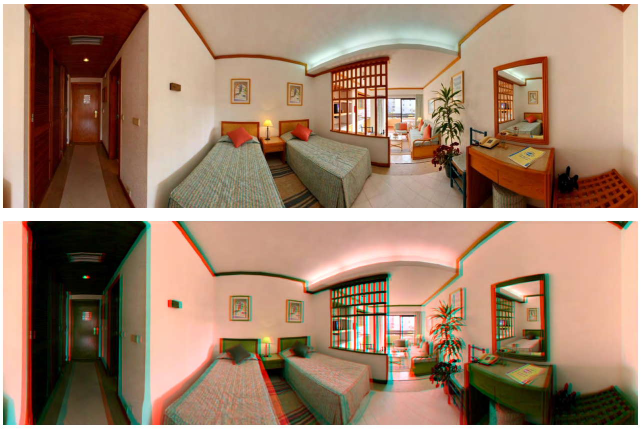

FIGURE 1. A representative result for a real scene. First row: input 2D panorama image. Second row: output stereo panorama shown as a red-cyan anaglyph.

II. RELATED WORK

In this section, we briefly review relevant techniques and system for stereoscopic panorama capture and generation, 2D-to-stereo generation, 2D-to-3D conversion, and GANs.

A. STEREOSCOPIC PANORAMA CAPTURE

Early systems for capturing stereoscopic panoramas are either based on a moving camera or are catadioptric. In the first case (e.g., [26], [29]), a camera is rotated while capturing the scene; a stereoscopic panoramic pair is generated by sampling from different parts of the image sequence. Examples of catadioptric cameras are those of Gluckman *et al.* [11] (with parabolic mirrors), Kawanishi *et al.* [16] (with six cameras and a hexagonal pyramidal mirror), Lin and Bajcsy [20] (with a reflective surface, a beam splitter, and two perspective cameras), and Yi and Ahuja [44] (with a concave lens and a convex mirror).

It is hard for catadioptric systems with curved mirrors to capture high-resolution stereo panoramas due to mirror curvature that generates blur [24], and moving camera systems are unable to capture dynamic scenes. Multi-camera setups [2], [4], [5], [16] are capable of capturing high-resolution dynamic scenes, but they have issues with size, expense, and camera self-occlusion (resulting in wasted pixels). Furthermore, they require careful post-processing to avoid visible seams.

Another complication is that of centricity, i.e., whether rays associated with a panorama pass through a common center of projection. This raises the question of how to maintain stereoscopy when viewing a panorama pair. Seitz [36] has classified all possible stereo pairs regarding their epipolar geometry. Svoboda *et al.* [38] independently obtained a similar result. They have shown that the epipolar geometry has to be a double ruled surface.

There are only a few specific types of stereo panoramas that satisfy stereoscopy. One example is concentric mosaics (CMs) [26], [27], [37], which can be generated from a sequence of perspective images captured with a circularly rotated camera. A specific pixel column from each image is collected to form a panorama; here, all rays are tangent to a 3D circle. The Google Jump system follows a similar design by using a circular array of video cameras. Each image is partitioned into left and right components and is individually stitched to generate CMs. Richardt *et al.* [29] further resolved issues caused by perspective distortion and described how to upsample the set of captured and corrected rays using optical-flow-based interpolation techniques.

B. 2D-TO-STEREO CONVERSION

Techniques to directly generate the second view given the input view tend to be learning-based. In principle, these are a cleaner approach to creating stereoscopic image pairs because they avoid computing depth and then warping, which has resampling and disocclusion issues. Learning-based approaches [1], [18], [42] typically generate the right view using convolutional neural nets (CNNs). For example, Flynn *et al.* [9] employ a deep network and applied regression to directly predict the color of the pixels without inferring their depths. Xie *et al.* [42] train a CNN model with ground truth stereo pairs extracted from a broad set of 3D movie collections. The availability of abundant training data further improves the robustness and quality. Nonetheless, most techniques handle perspective images only, and results produced by these techniques tend to exhibit loss of detail.

C. 2D-TO-3D CONVERSION

An alternative to directly generating stereo images is to first add depth to the input 2D image. Techniques for perspective image 2D-to-3D conversion are either based on depth-from-X or are learning-based. Depth-from-X methods [6], [13], [35], [48] use geometric and environmental cues to overcome the ill-posedness issue; such cues include defocus, scattering (in fog or haze), and indoor scene geometry (planes and lines constrained by Manhattan world assumption). However, these methods work under specific conditions and may be sensitive to image noise or lack of visual features.

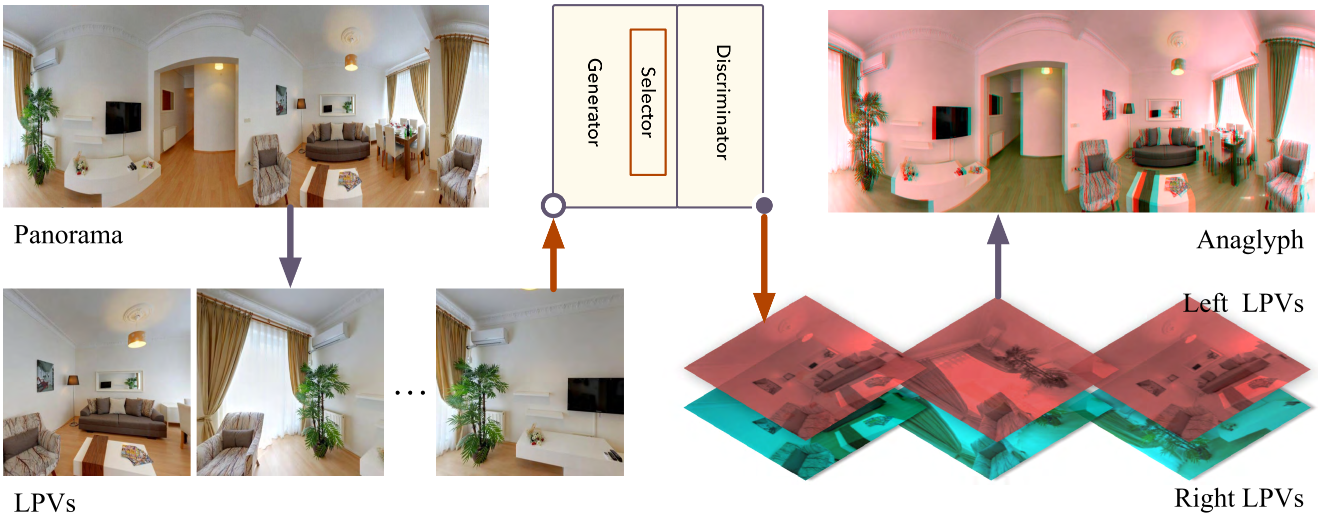

FIGURE 2. Overview of our fully automatic 2D-to-stereo conversion system. It takes as input a single 2D panorama, samples locally perspective views (LPVs), generates left/right pairs for each LPV using a GAN architecture, then constructs a concentric mosaic (CM) before finally mapping to a stereoscopic panorama output (shown as anaglyph here).

Learning-based techniques (e.g., [7], [15], [33], [34]) generate a depth map by training on extensive 3D data. For example, the system of Karsch *et al.* [15] automatically generates depth maps using a non-parametric depth sampling technique. More recent approaches are now CNN-based. Liu *et al.* [21] use deep CNN and continuous CRF, while Wang *et al.* [40] use a trained CNN to jointly predict a global layout composed of pixel-wise depth values and semantic labels. Roy and Todorovic [31] propose a neural regression forest model combined with CNNs.

D. GENERATIVE ADVERSARIAL NETWORKS (GANs)

Since its introduction, the use of GAN [12], [46] has seen impressive successes. For example, Denton *et al.* [8] propose a generative parametric model to produce high-quality samples of natural images. To perform image editing operations, Zhu *et al.* [47] use GAN to learn the manifold of natural images as a constraint. Mathieu *et al.* [23] use a GAN model that learns to separate the factor of variation associated with the labels from the other sources of variability. More recently, the conditional GAN is introduced for conditional image generation applications. To translate visual concepts from characters to pixels, Reed *et al.* [28] use a GAN-based deep architecture. Pathak *et al.* [25] present an unsupervised visual feature learning algorithm with GAN to perform context-based pixel prediction. Isola *et al.* [14] use conditional GAN for image-to-image translation. In this paper, we show how to combine the selection layer and ‘‘pseudo’’ depth peeling loss with adversarial networks to further improve visual results, in addition to using CMs to produce conceptually-valid stereo panoramas.

III. STEREO IMAGE SYNTHESIS

The only input to our system is a single panoramic image; the pipeline for generating the panoramic stereo pair is shown in Fig. 2. We first apply perspective projection to obtain locally perspective views (LPVs). A special GAN architecture is used to recover stereo views for each LPV. The left/right pairs are then used to construct a concentric mosaic (CM), after which, the final panoramic stereo pair is created. We now describe how we sample LPVs.

A. LOCAL PERSPECTIVE VIEW EXTRACTION

The typical way of generating 2D panoramas is to capture multiple perspective images while rotating the camera, and then stitch these images [39]. For our work, we do the reverse: we project the input panorama to overlapping perspective images (which we call local perspective views or LPVs). The input is a 2D full-view panorama, with horizontal FoV of 360° and vertical FoV of 180°. As shown in Fig. 3 (left), to extract the LPVs, we first map the original panorama into spherical coordinates.

We synthesize an LPV with a virtual camera placed at the sphere center pointing out within the $x - y$ plane; the LPVs are then generated by rotating the virtual camera about the z axis. We extract 32 overlapping LPVs with horizontal and vertical FoV of 90°. Fig. 3 (right) shows representative extracted LPVs.

Once the LPVs have been extracted, the next step is to generate stereo views for each LPV. To do this, we use a modified GAN architecture. We now describe how we formulate our objective for this architecture.

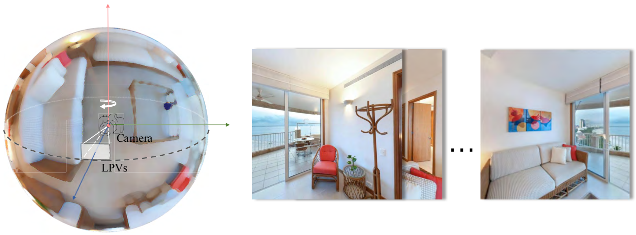

FIGURE 3. Extraction of local perspective views (LPVs). Left: Panorama mapped on a sphere, with a virtual camera. Right: Representative LPVs.

B. OBJECTIVE FORMULATION

GAN [12] estimates the generative models that learn a mapping from random noise vector z to output y by minimizing an adversarial loss. During the training process, we train two models simultaneously: a generative model G that represents the data distribution and a discriminative model D that estimates the probability to differentiate a sample whether is a ground truth data or generated by G. We extend the usual GAN architecture by introducing a new ‘‘pseudo’’ depth peeling loss that applies bigger weights to larger disparity regions. We train the network to predict both left and right views.

1) ADVERSARIAL LOSS

By using GAN, the generator network is encouraged by the discriminator network to produce non-blurry results. For our left-to-right view conversion task, the objective is expressed as

$$L_{GAN}(G, D) = E[\log D(RV)] + E[\log(1 - D(G(LV)))], \quad (1)$$

where RV and LV are the right and left views, respectively, G is the generator that attempts to minimize the objective function, and D is the discriminator that tries to maximize it. Note that we also perform right-to-left view conversion, using the same formulation as above, except the views are swapped.

2) ‘‘Pseudo’’ DEPTH PEELING LOSS

In using conventional GAN, we noticed that the quality of reconstruction is disparity dependent, with larger disparities creating more loss in detail. To address this problem, we modulate the weights based on the amount of disparity, with higher weights at larger disparity regions and lower weights at small disparity regions. More specifically, we evenly partition the image into three parts (foreground, middle, and background) based on their depth. This specific loss function, cast as an $L1$-based objective, is

$$L_{P_{L1}}(G) = E\left[\sum_{i=1}^{3} \omega_i ||l_i(RV) - l_i(G(LV))||_1\right], \quad (2)$$

where l_i is the i^{th} partition and ω_i is its weight.

3) FULL OBJECTIVE FORMULATION

To extract the stereo views from a single image, we optimize

$$E = \arg\min_G \max_D [L_{GAN}(G, D) + L_{P_{L1}}(G)] \quad (3)$$

by training a network. However, instead of learning the distribution of color information to estimate the intensity of a pixel, our network learns to predict a probability of disparity map representing the shifting of a pixel at different depth levels. This is done because the probability of disparity map is differentiable, unlike the disparity map itself. Then we feed the probability map into the selector and we show in Section V that using the selector generates higher quality results compared to directly inferring color [14], especially on the highly textured objects.

C. NETWORK ARCHITECTURE

In this section, we describe our network architecture for optimizing the objective described in the previous section. In addition to the usual generator and discriminator, we also add a selector at the end of the generator; this is done to improve the quality of detail in the output.

1) GENERATOR

The core parts of our generator G are U-net and selector. Similar to [14] and [30], we also use the U-net skip block in our network. Specifically, we use 9 convolution layers in the downsampling path, as the input size is $512 \times 512 \times 3$. All convolutions are 4×4 spatial filters applied with stride 2. The downsampling factor is 2 in the downsamping path, but the number of filters doubles until 512. Meanwhile, the upsampling factor is the same as the downsampling factor and the number of filters mirrors the downsampling path as well. As Fig. 4 shows, there are concatenate skip connections between downsampling path and upsampling path. The size of the output in the final convolution layer is $512 \times 512 \times 26$, with 26 being the disparity range (0 to 25) in our training dataset. The selector interprets this representation as the probability disparity map across the different disparities.

2) SELECTOR

We observe that the original GAN [12] or the image-to-image translation approach [14] can produce non-blurry outputs by training stereo views. However, these methods are less successful in recovering texture. Inspired by Flynn *et al.* [9] and Xie *et al.* [42], we add a selector at the end of the generator network. Instead of estimating the color of each pixel, the selector generates a set of probability disparity maps (shifting on each disparity level) indicating the likelihood of each pixel on the given depth level. This is similar to the idea of the plane sweep [3], except that the evidence is learned rather than derived from multiple displaced images of the same scene. We modify the disparity range of selector based on our training data. Fig. 5 shows our selector structure, where the brighter regions of probability map represent a higher likelihood of having that disparity. The output of the selector is the predicted view.

3) DISCRIMINATOR

As shown in Fig. 4, we have 6 convolution layers in the downsampling path. We also use the LeakyReLU activation with a slope of 0.2. The first four convolutions are 4×4 spatial filters applied with a stride 2. The last two convolutions are 4×4 spatial filters applied with a stride 1. At the end of the discriminator, there is a sigmoid activation function to output the probability of the generated image being real, namely, the Binary Cross Entropy (BCE) associated with the target label (1 = real, 0 = fake).

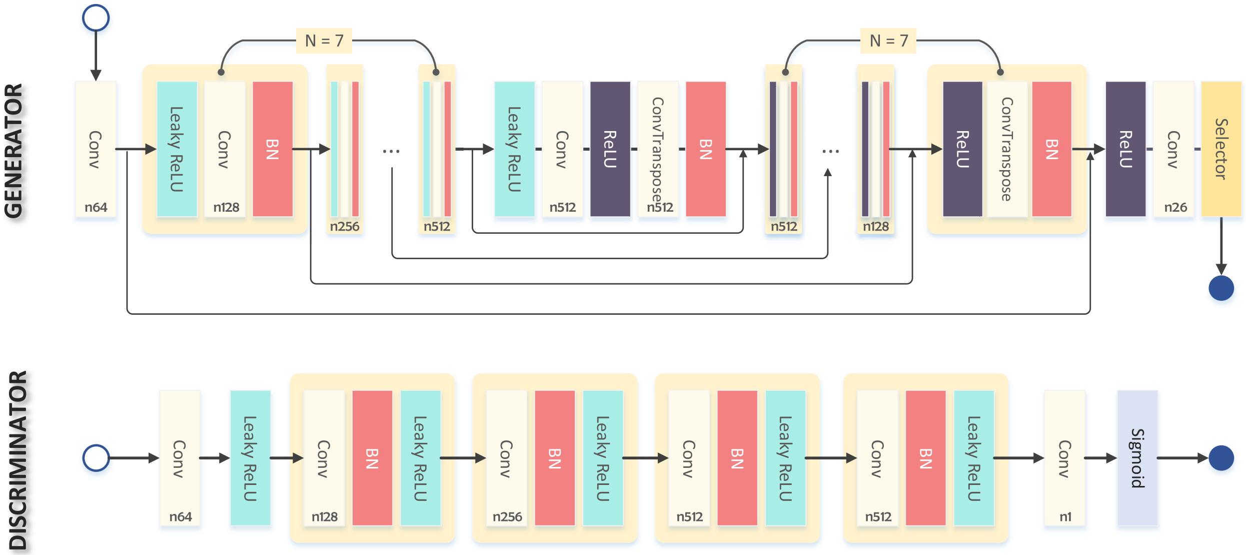

FIGURE 4. Our network architecture. The generator and discriminator are shown in detail, with corresponding kernel size information. The input to the generator is an LPV. while the output of the selector is the estimated novel view. The input to the discriminator is the output generated from the generator or the ground truth view, while the output of the sigmoid module is used to measure the Binary Cross Entropy (BCE) with the target label. $N = 7$ represents the number of the repeated blocks.

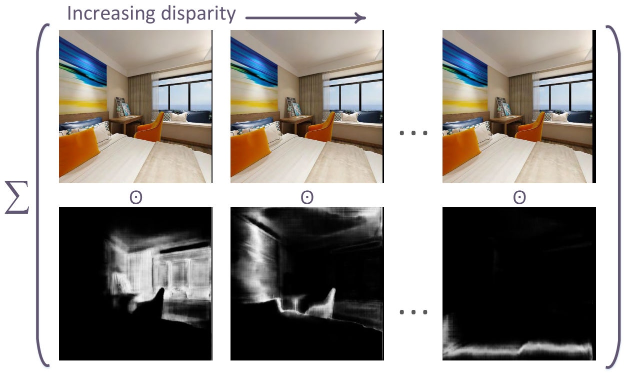

FIGURE 5. Selector structure. Each column represents a different uniform disparity map and its corresponding color image. The operation between disparity map and color image represents dot product. Top row: each image is a shifted version of the original based on the uniform disparity. Bottom row: each image depicts the probability distribution for image pixels having that particular disparity.

IV. STEREO PANORAMA SYNTHESIS

Our stereo panorama output is generated from concentric mosaics (CMs) [37]. A CM can be constructed from a series of images captured along a circular path. In our approach, we train our network to produce this series of stereo images from a single 2D panorama. As the left of Fig. 6 shows, for a given LPV (‘‘Center View’’), we generate its left and right LPVs. More specifically, we train two models using our network for the left and right LPV synthesis: we use the left LPVs as inputs to train the right LPV synthesis network and vice versa. This helps to generate two LPVs with horizontal parallax and with a predefined interocular distance.

The series of stereo LPVs generated for all the input LPVs can be thought of as being arranged along a circular path whose radius is the shift in perspective with respect

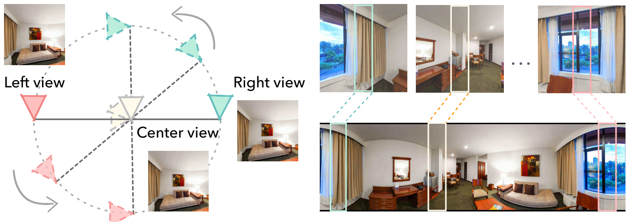

FIGURE 6. Stereo panorama synthesis. Left: Each originally sampled LPV is the ''Center view'', and the output stereo LPVs are denoted as ''Left view'' and ''Right view''. Note that the output views are along a circle. Right: The same columns from different right LPVs are resampled to constitute the right panorama, in the exact same manner as CM. The left panorama is created the same way with left LPVs.

to the original LPV. This is exactly the setup for concentric mosaics (CMs). The right of Fig. 6 shows how the right panorama is constructed from the same columns of the right LPVs; the left panorama is constructed in the same way using the left LPVs.

Please note that generating either the left or right panorama only is not sufficient. Suppose we use the original panorama and the left panorama as the stereo pair. For any given stereo view, the mid-interocular position is between the panorama center and a point on the circle corresponding to the locus of left LPVs. As the user rotates, this mid-interocular position moves along another circle. This is in comparison with our left-right synthesis, where the mid-interocular position remains at the center. This reasoning is consistent with the panoramic stereo condition described in [36].

V. EXPERIMENTS

In this section, we describe how we acquire synthetic and real data for testing, and show both qualitative and quantitative results. We also show results of a user study involving the use of an HTC VIVE headset to experience two versions of the stereoscopic panorama, namely ours (left-right panorama pair) and a reference method (input-right panorama pair). This user study is done as validation of the panoramic stereo condition.

a: Data Acquisition

For synthetic data, we downloaded 263 room models online; their virtual sizes range from 2×2 m^2 to 4×4 m^2. To render the stereo pairs, we set one camera roughly in the middle of the room and the second camera next to it with the baseline 35 mm (this is about half the average adult human interocular distance). To synthesize the right-view CM, we rotate the virtual camera about its center by 360° to render input LPVs, while rotating the corresponding virtual right camera along a circle (with the baseline as the radius) to render the right LPVs (and depth images of right LPVs simultaneously). The left-view CM is synthesized similarly. All LPVs are of resolution 542 × 542 with 90° field of view (FoV).

Since we train using the left and right views, we are training for the baseline of $2 * 35 = 70$ mm, which is close to the average human interocular distance. We do not use publicly available stereo datasets (e.g., KITTI [10]) due to their typically wide baseline. We render 6016 pairs of stereo views and randomly separate them into 5516 groups for training and 500 groups for testing. For real data, we collect 40 panoramas from SUN360 database [41]. We also captured 60 panoramas using our panoramic capture system, which consists of a Canon 5D Mark II camera (with a fish-eye lens) mounted on a tripod.

Once we have collected the data, we extract the LPVs from each panorama using the technique described in Section III-A. Each LPV has a resolution of 512 × 512 and the same FoV (90°) as the rendered image. Note that the synthetic dataset has a slightly larger resolution to enable data augmentation during training.

b: Training details and parameters

We set $\omega_1 = 50$, $\omega_2 = 30$, and $\omega_3 = 20$ in Equ. 2. For training, we first collect the input images with the resolution of 542 × 542. We randomly crop them to the resolution of 512 × 512 for data augmentation. Before training, we normalize the images between -1 and 1. We initialize the learning rate to 0.0002. After 100 epochs, we reduce it to 0.0001. For optimization, we have used Adam [17] with $\beta_1 = 0.5$. Meanwhile, we set the weights based on a Gaussian distribution with the mean 0 and the standard deviation 0.02. Our training process takes about 14 hours on an NVIDIA Tesla M4000 GPU with the batch size of 8 for 150 epochs. All networks are implemented with PyTorch.

A. RESULTS

To prove the selection layer is effective, we conduct an experiment with and without selector structure. To analyze the contribution of the "pseudo" depth peeling loss, we conduct an ablation study of comparison with GAN and GAN+L1. To further evaluate our method, we use Deep3D [42], MonoDepth [43] and Pix2Pix [14] as baselines. Please note that we use the same training data to retrain their networks without changing any settings.

Our system takes about 4 seconds for LPV extraction. Our trained model takes about 25 seconds for predicting and saving 64 LPVs (with the batch size of 1 in inference process), after which the output panoramic stereo pair is synthesized in 5 seconds.

1) QUANTITATIVE EVALUATION

We compute the mean absolute error (MAE, also used in [42]) and SSIM on the validation set to evaluate the quality of our results. MAE is defined as $\text{MAE} = |N - G|/n$, where n is the number of pixels, N is the inferred novel view, and G is the ground truth.

Fig. 7 shows L_1 error maps for two LPVs. Using GAN alone produces the highest errors. It generates images that are nearly copies of the original because it does not factor in disparity shifts. Our proposed "pseudo" depth peeling loss produces better results than GAN+L1, especially on foreground regions with large disparity.

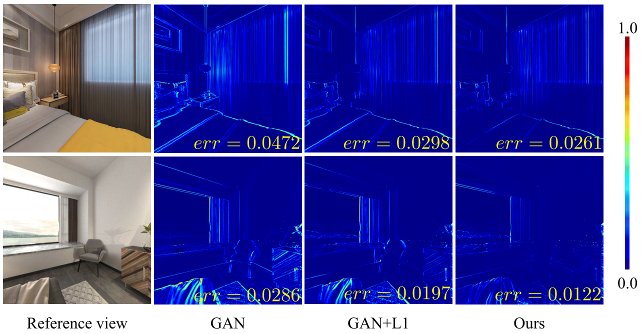

FIGURE 7. Comparisons of error maps. The error for our "pseudo" depth peeling loss is typically less that that for GAN and GAN+L1.

TABLE 1. Quantitative comparisons using MAE (mean absolute error) and SSIM metrics. Our approach produced the best numbers.

Method	MAE	SSIM
Ours	**4.329**	**0.911**
Ours w/o selection layer	4.617	0.899
GAN	7.847	0.778
GAN+L1	4.710	0.893
Deep3D [42]	5.750	0.877
MonoDepth [43]	5.523	0.879
Pix2Pix [14]	6.248	0.868

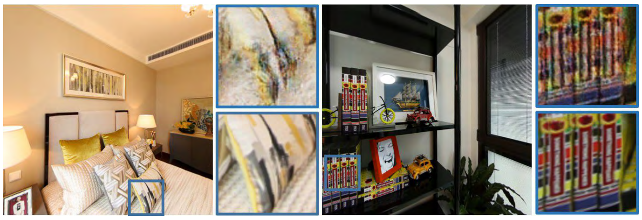

FIGURE 8. Visual comparisons of ours without selection layer and ours. The closeup views are, from top to bottom: ours without selection layer, ours. Our results recover more detail of texture.

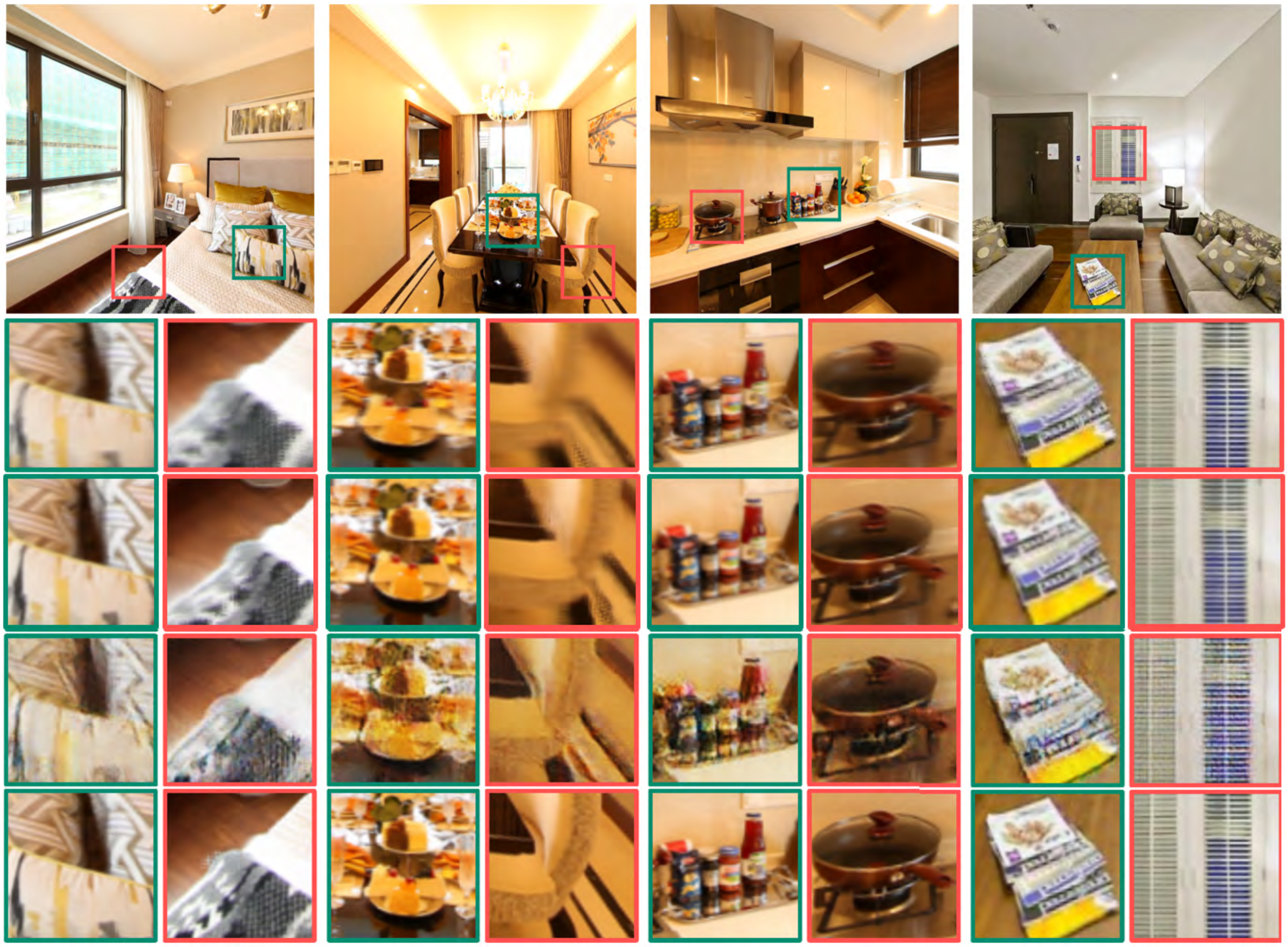

FIGURE 9. Visual comparisons. The closeup views are, from top to bottom: Deep3D [42], MonoDepth [43], Pix2Pix [14], ours. Our results appear sharp with fewer artifacts.

Table 1 compares quantitative performance (in terms of MAE and SSIM) on the validation set. Our system produces the best values among ours without selection layer, GAN, GAN+L1, and three state-of-the-art approaches [14], [42], [43]. These results validate our design decisions that include using the selector and "pseudo" depth peeling loss.

2) QUALITATIVE EVALUATION

Fig. 8 shows the visual comparisons between ours without selection layer and ours. The results of ours with selection layer can recover more detail of texture, which prove the selector is effective. Fig. 9 shows results for representative *real* scenes (from left to right: bedroom, dining room, kitchen, and living room). We also show results from three state-of-the-art approaches, namely those of Deep3D [42], MonoDepth [43] and Pix2Pix [14]. Specifically, Pix2Pix is a GAN based image translation method, and network structures of other two methods are normally convolutional neural networks. The results of our system appear to have the best fidelity. By comparison, results from Deep3D [42] and MonoDepth [43] tend to be blurry, and this is likely caused by Euclidean distance minimization. Results from Pix2Pix [14] exhibit noisy colors, and this is likely due to the image-to-image transformation not preserving stereo geometry.

Fig. 11 shows four additional stereo panorama results of real indoor scenes, converted to red-cyan anaglyphs.[2]

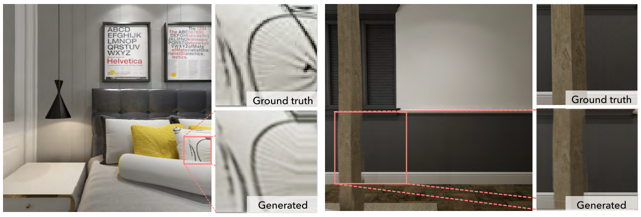

FIGURE 10. Two failure cases with distortion artifacts.

Fig. 10 shows two failure cases. In the first case, our method fails to recover the right-most part of the object. This is because the selector can only estimate pixel shifting, but not

[2] Note that these anaglyphs are "optimized" versions without red, to reduce retinal rivalry. See http://3dtv.at/Knowhow/AnaglyphComparison_en.aspx.

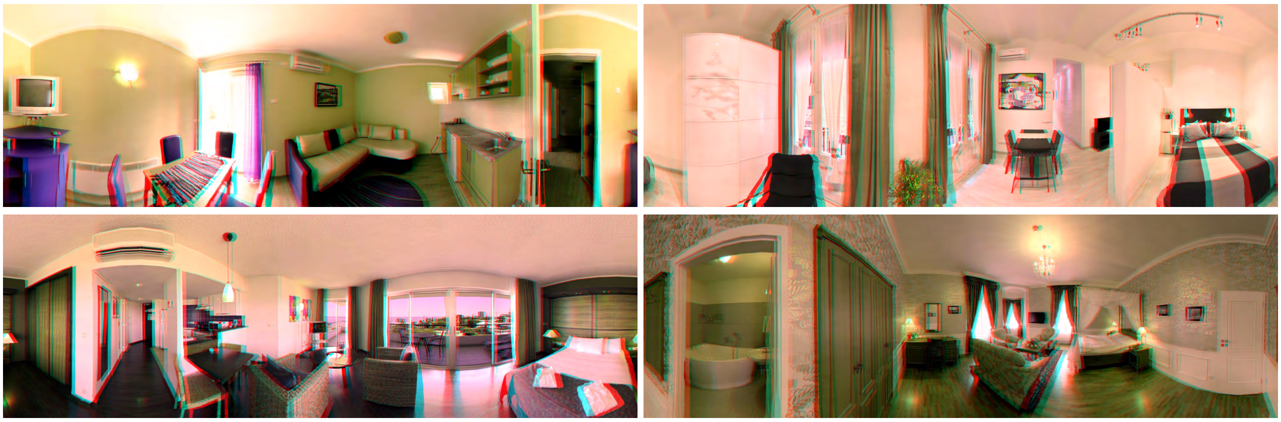

FIGURE 11. Four stereo panorama results of real indoor scenes in red-cyan anaglyph form.

recover missing pixels. In the second case, our approach fails to recover the scene of larger disparity range, and this is because of the fixed disparity range used during learning. Pixels with disparities beyond this range tend to be clamped to similar values, causing the observed distortion artifacts.

B. USER STUDY

As with [45], we ran a user study to evaluate user experience of stereo panoramas. In particular, we wish to validate our design decision to generate left-right panoramic pairs, as opposed to the simpler (reference) approach of using the original panorama plus the right-view panorama. Please note that technically, our output conforms to the stereo panorama condition while the reference output does not.

To generate the reference stereo pair, we use the original (center view) panorama as the "left" view, while the "right" panorama is created right LPVs. The baseline for the reference stereo is the same as that for ours, i.e., 70 mm.

Each subject is showed 10 different scenes, with each generated in two ways (ours and the reference). The order of the panorama stereo shown is randomized. The device used for visualization (in the form of stereoscope, not anaglyph) is the HTC VIVE with GoPro VR Player. Fifteen subjects (7 females and 8 males) took part in our user study; they are university students who all have normal stereopsis perception. They are also not told how each stereo panorama was generated. Each subject takes on average 1-2 minutes to visualize a panorama stereo pair.

After each experience, the participants are asked to respond to two questions: "Do you perceive 3D?", and "Do you feel comfortable of viewing the stereo panoramas?" Each response is in the form of a rating between 1 to 10, with 10 representing complete agreement. Results are shown in Table 2. This user study, while limited in scope, appears to support our design decision to generate left-right panoramic stereo views through LPV and CM synthesis.

TABLE 2. User study results (mean scores μ and their standard deviations σ).

	Perception of 3D	Comfort
	μ (σ)	μ (σ)
Ours	**7.4** (0.80)	**6.9** (0.92)
Reference	6.4 (0.95)	4.6 (0.61)

VI. DISCUSSION AND FUTURE WORK

Our work is currently constrained to indoor scenes. Conceptually, since our technique is data-driven, it should be applicable to outdoor scenes as well, as long as sufficient data are available. It would be interesting to see how well our system works beyond indoor scenes.

An immediate future direction is to fuse our approach with geometry-based methods. More specifically, we can impose additional priors on the structure of the scene to both improve both depth estimation and accelerate the view synthesis process. We can first estimate the coarse scene geometry using geometric cues such as vanishing points and Manhattan World structures and integrate such geometry into our deep learning solution process.

VII. CONCLUSIONS

We propose a technique to automatically convert a 2D panorama to a stereoscopic pair. We adapted the GAN architecture to add the selection layer on the generator and introduced the "pseudo" depth peeling loss term. Another contribution is we adhere to the panoramic stereo condition by generating concentric mosaics (CMs) instead of merely creating another panorama with shifted views. Experiments, both quantitative and qualitative, show that our technique can generate high-quality 3D stereoscopic panoramas, thus justifying our design decisions.

REFERENCES

[1] V. Appia and U. Batur, "Fully automatic 2D to 3D conversion with aid of high-level image features," *Proc. SPIE, Stereoscopic Displays Appl. XXV*, vol. 9011, Mar. 2014, Art. no. 90110W.

[2] V. Chapdelaine-Couture and S. Roy, "The omnipolar camera: A new approach to stereo immersive capture," in *Proc. ICCP*, Apr. 2013, pp. 1–9.

[3] R. T. Collins, "A space-sweep approach to true multi-image matching," in *Proc. CVPR*, Jun. 1996, pp. 358–363.

[4] V. Couture, M. S. Langer, and S. Roy, "Panoramic stereo video textures," in *Proc. ICCV*, Nov. 2011, pp. 1251–1258.

[5] V. C. Couture, M. S. Langer, and S. Roy, "Omnistereo video textures without ghosting," in *Proc. Int. Conf. 3D Vis.-3DV*, Jun./Jul. 2013, pp. 64–70.

[6] F. Cozman and E. Krotkov, "Depth from scattering," in *Proc. CVPR*, Jun. 1997, pp. 801–806.

[7] E. Delage, H. Lee, and A. Y. Ng, "A dynamic Bayesian network model for autonomous 3D reconstruction from a single indoor image," in *Proc. CVPR*, vol. 2, Jun. 2006, pp. 2418–2428.

[8] E. L. Denton, S. Chintala, A. Szlam, and R. Fergus, "Deep generative image models using a Laplacian pyramid of adversarial networks," in *Proc. NIPS*, 2015, pp. 1486–1494.

[9] J. Flynn, I. Neulander, J. Philbin, and N. Snavely, "Deep stereo: Learning to predict new views from the world's imagery," in *Proc. CVPR*, Jun. 2016, pp. 5515–5524.

[10] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The KITTI dataset," *Int. J. Robot. Res.*, vol. 32, no. 11, pp. 1231–1237, Sep. 2013.

[11] J. Gluckman, S. K. Nayar, and K. J. Thoresz, "Real-time omnidirectional and panoramic stereo," in *Proc. Image Understand. Workshop*, Nov. 1998, pp. 299–303.

[12] I. Goodfellow *et al.*, "Generative adversarial nets," in *Proc. NIPS*, 2014, pp. 2672–2680.

[13] V. Hedau, D. Hoiem, and D. A. Forsyth, "Recovering the spatial layout of cluttered rooms," in *Proc. ICCV*, Sep./Oct. 2009, pp. 1849–1856.

[14] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros. (2016). "Image-to-image translation with conditional adversarial networks." [Online]. Available: https://arxiv.org/abs/1611.07004

[15] K. Karsch, C. Liu, and S. B. Kang, "Depth transfer: Depth extraction from video using non-parametric sampling," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 11, pp. 2144–2158, Nov. 2014.

[16] T. Kawanishi, K. Yamazawa, H. Iwasa, H. Takemura, and N. Yokoya, "Generation of high-resolution stereo panoramic images by omnidirectional imaging sensor using hexagonal pyramidal mirrors," in *Proc. ICPR*, Aug. 1998, pp. 485–489.

[17] D. P. Kingma and J. Ba. (2014). "Adam: A method for stochastic optimization." [Online]. Available: https://arxiv.org/abs/1412.6980

[18] J. Konrad, M. Wang, P. Ishwar, C. Wu, and D. Mukherjee, "Learning-based, automatic 2D-to-3D image and video conversion," *IEEE Trans. Image Process.*, vol. 22, no. 9, pp. 3485–3496, Sep. 2013.

[19] C. Li and M. Wand, "Precomputed real-time texture synthesis with Markovian generative adversarial networks," in *Proc. ECCV*, Sep. 2016, pp. 702–716.

[20] S.-S. Lin and R. Bajcsy, "High resolution catadioptric omni-directional stereo sensor for robot vision," in *Proc. ICRA*, Sep. 2003, pp. 1694–1699.

[21] F. Liu, C. Shen, and G. Lin, "Deep convolutional neural fields for depth estimation from a single image," in *Proc. ICCV*, Jun. 2015, pp. 5162–5170.

[22] M.-Y. Liu and O. Tuzel, "Coupled generative adversarial networks," in *Proc. NIPS*, 2016, pp. 469–477.

[23] M. Mathieu, J. J. Zhao, P. Sprechmann, A. Ramesh, and Y. LeCun, "Disentangling factors of variation in deep representation using adversarial training," in *Proc. NIPS*, 2016, pp. 5040–5048.

[24] S. K. Nayar, "Catadioptric omnidirectional camera," in *Proc. CVPR*, Jun. 1997, pp. 482–488.

[25] D. Pathak, P. Krahenbuhl, J. Donahue, T. Darrell, and A. A. Efros, "Context encoders: Feature learning by inpainting," in *Proc. CVPR*, Jun. 2016, pp. 2536–2544.

[26] S. Peleg, M. Ben-Ezra, and Y. Pritch, "Omnistereo: Panoramic stereo imaging," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 23, no. 3, pp. 279–290, Mar. 2001.

[27] Y. Pritch, M. Ben-Ezra, and S. Peleg, "Optics for omnistereo imaging," in *Foundations of Image Understanding*. Springer, 2001, pp. 447–467.

[28] S. Reed, Z. Akata, X. Yan, L. Logeswaran, B. Schiele, and H. Lee. (2016). "Generative adversarial text to image synthesis." [Online]. Available:https://arxiv.org/abs/1605.05396

[29] C. Richardt, Y. Pritch, H. Zimmer, and A. Sorkine-Hornung, "Megastereo: Constructing high-resolution stereo panoramas," in *Proc. CVPR*, Jun. 2013, pp. 1256–1263.

[30] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *Proc. Int. Conf. Med. Image Comput. Comput.-Assist. Intervent.*, Nov. 2015, pp. 234–241.

[31] A. Roy and S. Todorovic, "Monocular depth estimation using neural regression forest," in *Proc. CVPR*, Jun. 2016, pp. 5506–5514.

[32] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen, "Improved techniques for training gans," in *Proc. NIPS*, 2016, pp. 2234–2242.

[33] A. Saxena, S. H. Chung, and A. Y. Ng, "Learning depth from single monocular images," in *Proc. NIPS*, 2005, pp. 1–8.

[34] A. Saxena, M. Sun, and A. Y. Ng, "Make3D: Learning 3D scene structure from a single still image," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 5, pp. 824–840, May 2009.

[35] A. G. Schwing, T. Hazan, M. Pollefeys, and R. Urtasun, "Efficient structured prediction for 3D indoor scene understanding," in *Proc. CVPR*, Jun. 2012, pp. 2815–2822.

[36] S. M. Seitz, "The space of all stereo images," in *Proc. ICCV*, Jul. 2001, pp. 26–33.

[37] H. Y. Shum and W. L. He, "Rendering with concentric mosaics," in *Proc. 26th Annu. Conf. Comput. Graph. Interact. Techn.*, Jun. 1999, pp. 299–306.

[38] T. Svoboda, T. Pajdla, and V. Hlaváč, "Epipolar geometry for panoramic cameras," in *Proc. ECCV*, Jun. 1998, pp. 218–231.

[39] R. Szeliski, "Image alignment and stitching: A tutorial," *Found. Trends Comput. Graph. Vis.*, vol. 2, no. 1, pp. 1–104, 2007.

[40] P. Wang, X. Shen, Z. Lin, S. Cohen, B. Price, and A. L. Yuille, "Towards unified depth and semantic prediction from a single image," in *Proc. CVPR*, Jun. 2015, pp. 2800–2809.

[41] J. Xiao, J. Hays, K. A. Ehinger, A. Oliva, and A. Torralba, "Sun database: Large-scale scene recognition from abbey to zoo," in *Proc. CVPR*, Jun. 2010, pp. 3485–3492.

[42] J. Xie, R. Girshick, and A. Farhadi, "Deep3D: Fully automatic 2D-to-3D video conversion with deep convolutional neural networks," in *Proc. ECCV*, Sep. 2016, pp. 842–857.

[43] C. Godard, O. M. Aodha, and G. J. Brostow, "Unsupervised monocular depth estimation with left-right consistency," in *Proc. CVPR*, Jul. 2017, pp. 6602–6611.

[44] S. Yi and N. Ahuja, "An omnidirectional stereo vision system using a single camera," in *Proc. ICPR*, vol. 4, Aug. 2006, pp. 861–865.

[45] F. Zhang and F. Liu, "Casual stereoscopic panorama stitching," in *Proc. CVPR*, Jun. 2015, pp. 2002–2010.

[46] J. Zhao, M. Mathieu, and Y. LeCun. (2016). "Energy-based generative adversarial network." [Online]. Available: https://arxiv.org/abs/1609.03126

[47] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros. (2017). "Unpaired image-to-image translation using cycle-consistent adversarial networks." [Online]. Available: https://arxiv.org/abs/1703.10593

[48] S. Zhuo and T. Sim, "On the recovery of depth from a single defocused image," in *Proc. CVPR*, 2009, pp. 889–897.

JIE LU received the B.S. degree from Xidian University, Xi'an, China, in 2016. He is currently pursuing the master's degree with the Shanghai Institute of Microsystem and Information Technology, Chinese Academy of Sciences, and ShanghaiTech University, Shanghai, China. His research interests include computer vision, computational photography, and computer graphics.

YANG YANG received the master's degree from the Stevens Institute of Technology, in 2012, and the Ph.D. degree in computer science from the University of Delaware, in 2017. In 2017, he joined DGene Inc., as a Research Engineer. His research interests include computer vision and computer graphics.

RUIYANG LIU received the B.S. degree from the Dalian University of Technology, Dalian, China, in 2016. She is currently pursuing the Ph.D. degree in computer science with ShanghaiTech University, Shanghai, China. Her current research interests include computer vision, computational photography, and computer graphics.

SING BING KANG (M'90–F'12) received the Ph.D. degree in robotics from Carnegie Mellon University, Pittsburgh, in 1994. He is currently a Principal Researcher with Microsoft Corporation. He has co-edited two books: *Panoramic Vision* and *Emerging Topics in Computer Vision*) and co-authored two books: *Image-Based Rendering* and *Image-Based Modeling of Plants and Trees*. His research interests include image and video enhancement, and image-based modeling. On the community service front, he has served as the Area Chair for the major computer vision conferences and as a Papers Committee Member for SIGGRAPH and SIGGRAPH Asia. He was the Program Chair of ACCV 2007 and CVPR 2009, and an Associate Editor-in-Chief of the IEEE Transactions on Pattern Analysis and Machine Intelligence, from 2010 to 2014.

JINGYI YU received the B.S. degree from the California Institute of Technology, Pasadena, CA, USA, in 2000, and the Ph.D. degree from the Massachusetts Institute of Technology, Cambridge, MA, USA, in 2005. He is currently a Professor with the School of Information Science and Technology, ShanghaiTech University, Shanghai, China, and an Associate Professor with the Department of Computer and Information Sciences and the Department of Electrical and Computer Engineering, University of Delaware, Newark, DE, USA. His current research interests include computer vision and computer graphics, in particular, computational cameras and displays. He is an Editorial Board Member of the IEEE Transactions on Pattern Analysis and Machine Intelligence, *The Visual Computer Journal*, and *Machine Vision and Application*. He was a recipient of the NSF CAREER Award and the AFOSR YIP Award. He has served as the Program Chair for the 2011 Workshop on Omnidirectional Vision and Camera Networks, the General Chair for the 2008 International Workshop on Projector-Camera Systems, and the Area and Session Chair for the 2011 International Conference on Computer Vision.