Hallucinating Scenes

Ting-I Hsieh*, Hwann-Tzong Chen*, Chia-Ming Cheng†, and Yan-Hao Huang‡
* National Tsing Hua University, Taiwan
† MediaTek Inc., Taiwan
‡ Industrial Technology Research Institute, Taiwan

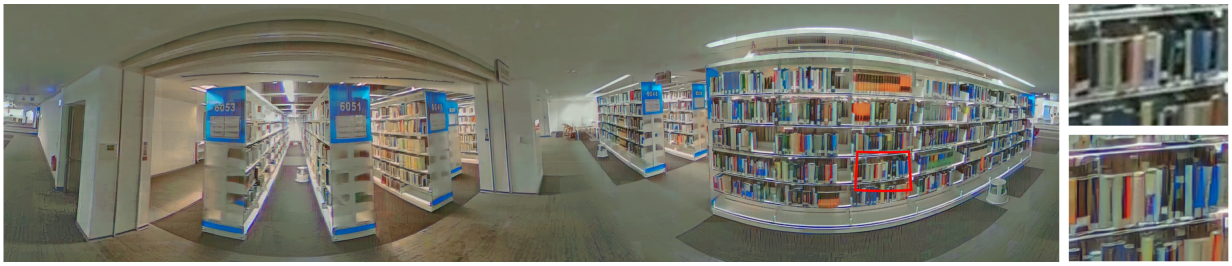


Fig. 1. A hallucinated high-resolution panoramic scene. On the right we show the corresponding low-resolution (top) and high-resolution (bottom) patches of the red-framed region in the panorama.

***Abstract*—The goal of this work is to synthesize high-resolution panoramic images of a scene. We propose a system that takes low-resolution panoramic video frames as the input, extracts geometric information as the reference, and produces high-resolution panoramas of the scene as the output. The deep learning module of our system benefits from the structure-from-motion geometry and learns to decouple the content and the style of an input low-resolution patch for hallucinating its high-resolution version. We show that the high-resolution panoramas generated by our system achieve a better quality of detail and color enhancement than those produced by existing super-resolution and style-transfer methods.**

I. Introduction

The motivation of this work is to investigate and propose an alternative way of acquiring high-quality panoramic views of a scene. We consider the process of capturing a scene as a constrained sampling problem. Suppose that we are allowed to use a panoramic camera to take high-resolution panoramas at only a few spots in the scene. (In our experiment we only take four high-resolution panoramas for a scene.) To obtain the high-resolution views at other spots in the scene, one may apply some view-synthesis techniques to generate the novel views. However, the underlying issue of multi-view geometry is itself a challenging problem, in particular if we would like to generate high-resolution novel views. Now, consider an augmented scenario in which we are also allowed to take densely sampled low-resolution panoramas. By "densely sampled" we mean capturing views at more spots that are close to each other in spatial domain. Such a setting avoids the issues of novel-view synthesis and instead reformulates the problem as how to hallucinate high-resolution images from low-resolution ones.

The proposed problem setting and formulation are different from those of conventional image super-resolution. For image super-resolution, a main issue is the lack of real high-resolution and low-resolution pairs of image patches for evaluation and for training. Existing approaches often create the high- and low-resolution pairs by downsampling the high-resolution ground truth to get the low-resolution counterpart. However, the downsampling operation is unlikely to be coherent with the real process of resolution degradation. Our approach, on the other hand, only relies on the structure-from-motion (SfM) information to find correlated high- and low-resolution pairs, but the corresponding image patches do not have to be accurately aligned. Nevertheless, the pairs of corresponding image patches provide a more realistic transformation in appearance than simply downsampling. The proposed learning method can decouple the corresponding image patches to extract details for better enhancement in resolution.

Our approach also differs from existing style-transfer and image-to-image translation methods. We integrate the geometry information of the scene obtained by SfM into the deep learning module, and therefore the low-resolution and high-resolution pairs are not totally uncorrelated. Existing style-transfer methods do not take into account the geometric cues and correspondences derived from SfM.

In sum, our system integrates SfM, deep networks, and image stitching to form a pipeline for generating high-resolution panoramas. SfM provides the geometric information of the scene for extracting image patches from low-resolution panoramas as the training data. Our deep learning module learns with SfM geometry to decouple the content and the style of low-resolution patches for producing detail-enhanced patches. The hallucinated patches are stitched to form a high-resolution panorama as the final output. The experiments show that our system can synthesize visually appealing high-resolution panoramas with enhanced color and detail.

A. Related Work

The conventional super-resolution problem has been studied for a long time [15]. Successful commercial applications are available for digital cameras, TV production, and medical imaging, *etc*. Recent super-resolution techniques often use deep learning for further improvements. For example, SR-CNN [3] adopts a deep architecture for end-to-end training. Dong *et al*. show that using a larger filter size is better than using a deeper CNN architecture. DRCN [8] uses deep CNN layers with shared weights to reduce the number of parameters, and achieves significant improvements.

Ledig *et al*. [11] propose a GAN-based approach that is capable of inferring photo-realistic natural images. They use a perceptual loss function to model the natural image manifold via a discriminator for distinguishing super-resolved images from original realistic images. Shrivastava *et al*. [17] propose Simulated+Unsupervised (S+U) learning with a GAN-based simulator, where the task is to learn a model for improving the simulator's output using unlabeled real data. Gharbi *et al*. present deep bilateral learning for real-time image enhancement [5]. They introduce a new neural architecture for image upsampling via predicting the coefficients of a locally-affine model in bilateral space.

Another closely related topic is image-based rendering [4], [18] in computer graphics and computer vision. Its goal is to render novel views based on a set of images of a scene. Previous techniques can be categorized with respect to how much the geometric information is used: rendering without geometry, rendering with implicit geometry, and rendering with explicit geometry. Kopf *et al*. [9] introduce a model-based viewing system for browsing, enhancing, and manipulating casual outdoor photographs by combining them with already existing georeferenced digital terrain and urban models.

MUNIT [7] and DRIT [12] share similar ideas that image representation can be decomposed into a content code that is domain-invariant, and a style code that captures domain-specific properties. Our work differs from MUNIT and DRIT in that it includes geoference information derived from SfM. We are able to provide low-resolution images as the reference for inferring the corresponding high resolution images.

II. Our Approach

The objective of our approach is to learn a generator $\mathbb{G}$ that can synthesize a high-resolution patch $\hat{x}_{L\rightarrow H}$ from the low-resolution patch x_L and its georeference code g_L:

$$\hat{x}_{L\rightarrow H} = \mathbb{G}_H(x_L, g_L)\,. \quad (1)$$

First, in the pre-processing stage, we establish the geometry model from the collected datasets, including densely sampled low-resolution panoramic video frames of size 1920 × 453 and sparsely sampled high-resolution panoramic images of size 5376 × 1269. The georeference information includes rotation matrix R, translation vector t, and the associated 3D reconstruction of feature points. Second, during the training phase, we generate sample patches with embedded geometric information. We use x_L and x_H to denote the image patches in low-resolution $\mathcal{L}$ and high-resolution $\mathcal{H}$ domains. We denote the georeference codes by g_L and g_H, queried from the geometry model and image location (u, v) in the panorama. The generator $\mathbb{G}$ and the cross-domain translation model $\mathbb{T}$ are trained based on the pairs of image patches (x_L, x_H), which do not have to be well aligned. Last, in the inference phase, we use the generator to convert input low-resolution video frames to high-resolution ones.

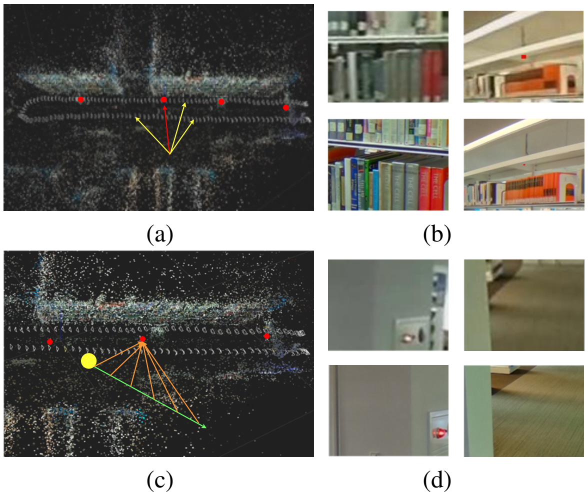
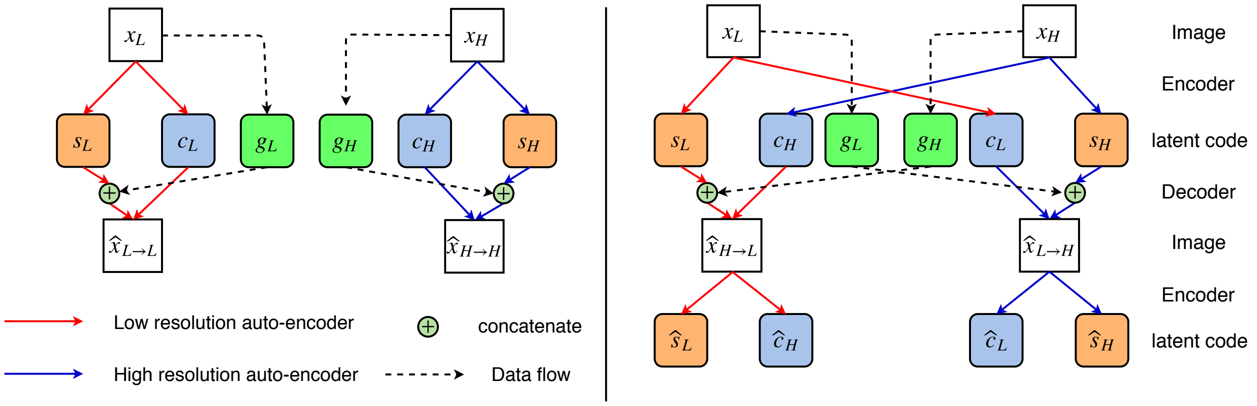


Fig. 3. (a) The red line represents a 3D point re-projecting onto a high-resolution image, which is represented as a red point. The yellow lines represent the re-projections of the same 3D point onto three different low-resolution images. Given a 3D point, we sample the patch pairs according to the angles and positions. (b) Examples of sampled patch pairs that correspond to the same 3D points. (c) Patches of texture-less regions without corresponding 3D points: Starting from the center of a patch in a low-resolution image, represented as the yellow dot, the green line represents the ray from the camera center through the patch center into the 3D space. By equispacedly sampling along the ray, we can obtain candidate high-resolution patches in the corresponding high-resolution image, represented as a red point. The patch among the candidates with highest similarity is selected as the training pair. (d) Examples of texture-less low- and high-res patch pairs.

A. Collecting the Data

We record low-resolution panorama videos to cover a target scene using a 360 camera 'RICOH THETA S', and take sparsely sampled high-resolution panoramic images at only four spots in the scene. We then perform OpenSfM [1], [2] to establish the geometry relations from the input video frames and estimate the 3D structures of the corresponding feature points. Both of the georeference information and the distribution of 3D points provide valuable cues in the training phase, which will be further discussed in the next subsection.

1) Extracting Training Samples: We obtain training patches in two ways, depending on the number of feature points. For a patch containing more than five feature points in low-resolution space, we take account of the 3D geometry information, as illustrated in Fig. 3(a), where the 3D points are reprojected to low-resolution images and high-resolution image. For a patch containing less than five feature points (less-textured regions), we generate samples by checking image



(a) within-domain (b) cross-domain

Fig. 4. The proposed deep model.

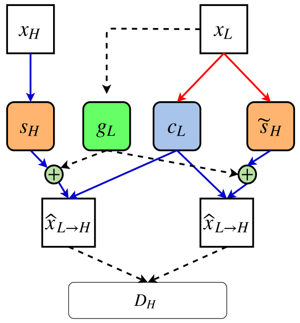


Fig. 5. At the training phase, we have two paths to get a high-resolution patch, depending on how we obtain the style code: *i)* using x_L to generate style code $\tilde{s}_H$, and *ii)* deriving s_H from x_H. Further, we integrate $\tilde{s}_H$ with c_L and g_L to generate $x_{L\to H}$. Note that, at the testing phase, only the first path can be used to get $x_{L\to H}$ since x_H is unknown.

similarity between the low-resolution patch and corresponding high-resolution candidate patches along the epipolar line, as shown in Fig. 3(c).

B. Modeling Different Domains

Let $\mathbb{E}_*$ denote an encoder to decompose a patch x_* into its latent code. The latent code will further be decomposed into a content code c_* and a style code s_*. We use a generator $\mathbb{G}$ to reproduce an image patch by decoding from the latent codes, formulated as $\hat{x}_* = \mathbb{G}_*(\mathbb{E}^c_*(x_*), \mathbb{E}^s_*(x_*), g_*)$, where $*$ can be either L or H. Similar to previous approaches [7], [12], the content code in our method is assumed to be domain-invariant, while the style code captures domain-specific properties. To translate a low-resolution patch x_L to a high-resolution patch, we recombine the content code with a style code $\tilde{s}_H$ learned from the high-resolution domain by a translation model.

Fig. 4 shows an overview of our model. It consists of a within-domain constructor and a cross-domain transfer. Each patch is factorized into content code c_* and style code s_*, where $(c_*, s_*) = (\mathbb{E}^c(x_*), \mathbb{E}^s(x_*))$. To transfer low resolution to high resolution or vice versa, we swap encoder-decoder pairs, see in Fig. 4(b). For example, to translate x_L to $\hat{x}_{L\to H}$, we extract the content code $c_L = \mathbb{E}^c_L(x_L)$, and find the style code from paired patch $s_H = \mathbb{E}^s_H(x_H)$, and use the decoder to generate patch images $\hat{x}_{L\to H} = \mathbb{G}_H(\mathbb{E}^c_L(x_L), \mathbb{E}^s_L(x_H), g_L)$.

Our model contains two discriminators. The discriminator $\mathbb{D}_H$ aims to distinguish $\hat{x}_{L\to H}$ from real patch x_H. On the other hand, $\mathbb{D}_L$ distinguishes $\hat{x}_{H\to L}$ from real patch x_L. The model is trained with a set of loss terms detailed as follows.

Bidirectional reconstruction losses compute the reconstruction error of image patch and latent code, which are defined by

$$\mathcal{L}^{x_H}_{recon} = E\left[||\mathbb{G}_H(\mathbb{E}^c_H(x_H), \mathbb{E}^s_H(x_H), g_H) - x_H||_1\right] \quad (2)$$

$$\mathcal{L}^{c_L}_{recon} = E\left[||\mathbb{E}^c_H(\mathbb{G}_H(c_L, s_H, g_L)) - c_L||_1\right], \quad (3)$$

$$\mathcal{L}^{s_H}_{recon} = E\left[||\mathbb{E}^s_H(\mathbb{G}_H(c_L, s_H, g_L)) - s_H||_1\right]. \quad (4)$$

Adversarial loss ensures that generated patches should be indistinguishable from the real ones. Fig. 5 illustrates the two paths to generate high-resolution patches depending on how we obtain the style code.

$$\begin{aligned}\mathcal{L}^{x_H}_{GAN} &= E\left[\log(1 - \mathbb{D}_H(\mathbb{G}_H(c_L, s_H, g_L)))\right] \\ &+ E\left[\log(1 - \mathbb{D}_H(\mathbb{G}_H(c_L, \tilde{s}_H, g_L)))\right] \\ &+ E\left[\log \mathbb{D}_H(x_H)\right].\end{aligned} \quad (5)$$

Transfer loss ensures the style code $\tilde{s}_H$ estimated by $\mathbb{T}_L(x_L)$ is similar to the real style code s_H extracted from x_H.

$$\mathcal{L}^{x_L}_{Trans} = E\left[\mathbb{T}_L(x_L) - s_H\right]. \quad (6)$$

The final objective combines all of the loss terms:

$$\begin{aligned}&\min_{\{\mathbb{E}_H, \mathbb{E}_L, \mathbb{G}_H, \mathbb{G}_L, \mathbb{T}_H, \mathbb{T}_L\}} \max_{\{\mathbb{D}_H, \mathbb{D}_L\}} \mathcal{L}^{x_H}_{GAN} + \mathcal{L}^{x_L}_{GAN} \\ &+ \lambda_x(\mathcal{L}^{x_H}_{recon} + \mathcal{L}^{x_L}_{recon}) + \lambda_c(\mathcal{L}^{c_H}_{recon} + \mathcal{L}^{c_L}_{recon}) \\ &+ \lambda_s(\mathcal{L}^{s_H}_{recon} + \mathcal{L}^{s_L}_{recon}) + \lambda_s(\mathcal{L}^{x_H}_{Trans} + \mathcal{L}^{x_L}_{Trans}).\end{aligned} \quad (7)$$

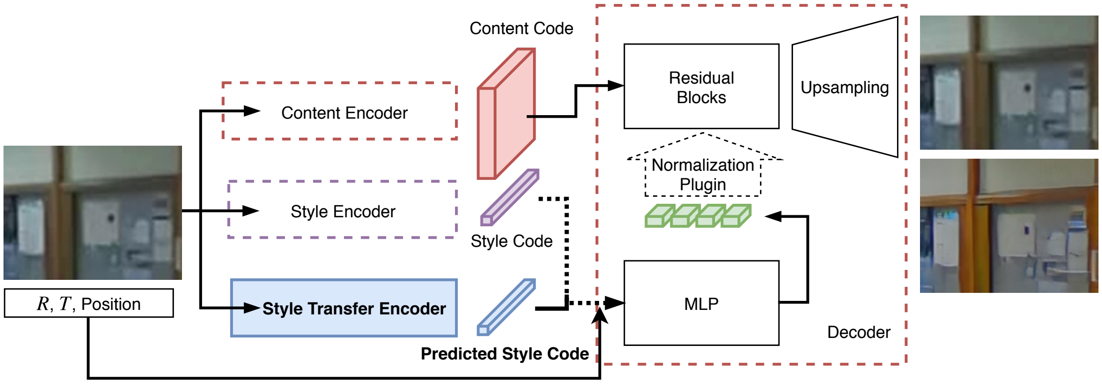


Fig. 6. The pipeline of generating high-resolution patches.

C. Testing

During testing we can only use the low-resolution patches to generate high-resolution patches. As shown in Fig. 5, we get $c_L = \mathbb{E}_L^c(x_L)$ and the georeference g_L. We also estimate $\tilde{s}_H$ from x_L using $\mathbb{T}$, *i.e.*, $\tilde{s}_H = \mathbb{T}_L(x_L)$. A high-resolution patch can therefore be generated by $x_H = \mathbb{G}(c_L, \tilde{s}_H, g_L)$. After getting all high resolution patches, we use Poisson blending [14] to stitch the patches and obtain the high-resolution panorama.

III. Implementation and Experiment

Fig. 6 summarizes the pipeline of our system. The architecture is inspired by [6], except that we include the georeference information and predict the style code using a *style transfer encoder*. The content encoder is composed of residual blocks The style transfer encoder predicts the high-resolution style code from a low-resolution patch. The decoder reconstructs the image patch from the content code and style code. We concatenate the style code and georeference information, and use a multilayer perceptron (MLP) to obtain the normalization weights and biases for the residual blocks in the decoder.

The discriminator $\mathbb{D}_*$ consists of two components $\mathbb{D}_*^{Image}$ and $\mathbb{D}_*^{Lap}$, where $\mathbb{D}_*^{Image}$ distinguishes real patches from generated ones using the RGB cues and $\mathbb{D}_*^{Lap}$ using the Laplacian features, inspired by [16], [10]. We use LSGAN [13] as our discriminator backbone.

A. Evaluation Metrics

We conduct user study to evaluate the quality of different methods. At each round of evaluation, we show the user seven randomly picked patches generated by the following methods: MUNIT [7], CycleGAN [20], SRGAN [11], ours, ours w/o Laplacian, ours w/o georeference g, ours w/o georeference and Laplacian. With the input low-resolution patch as reference, the user has to select two among seven patches generated by different methods in terms of visual quality.

In addition to qualitative evaluation, we adopt LPIPS (Learned Perceptual Image Patch Similarity) [19] as a metric to measure the perceptual similarity between real and generated patches, x_H and $\hat{x}_{L \to H}$.

B. Results

Table III-B shows the evaluation results of different methods with respect to LPIPS score and user study. We split the LPIPS scores into Quartile (Q1,Q2,Q3) and mean, and the last column of the table shows the results of user study. Many less-textured patches have high similarity with their real high-resolution versions so the scores will be within Q1. On the other hand, complex patches often have lower similarity with the real patches, so the scores will distribute within (Q2,Q3). SRGAN is not good at adapting to color change, especially for complex patches. Therefore, SRGAN only has a good score in (Q1), but achieves much worse performance in (Q2,Q3). MUNIT performs well in user study, in comparison with CycleGAN and SRGAN. Our method achieves better LPIPS scores than all other methods.

Fig. 7 shows some examples of high-resolution panoramas generated by different methods. We also include the original low-resolution panorama at the bottom of the figure for reference. MUNIT does not have a style-transfer encoder as our method does, and therefore it can only arbitrarily choose a style code for generating a high-quality patch. Some patches might not be suitable for the chosen style code and thus the generated patch would exhibit an irrelevant style. MUNIT, CycleGAN, and SRGAN do not use the geometric information, and therefore the visual quality of generated panorama might degrade at some locations.

IV. Conclusion

This work suggests a new way of acquiring high-resolution panoramas of a scene. We present a learning-based system that learns to hallucinate high-resolution panoramas from low-resolution patches. The proposed system differs from previous style-transfer and super-resolution methods in that it uses the geometric information derived from SfM reconstruction. On the other hand, novel view synthesis from captured high-resolution panoramas is not applicable here because only three or five high-resolution panoramas are captured at far-apart spots and from different perspectives. The style transfer encoder of our system can directly predict the style code from

	LPIPS				User
Model	Q1	Q2	Q3	Mean	Studey
Low resolution	0.335	0.616	0.73	0.553	N/A
MUNIT [7]	0.447	0.578	0.682	0.562	42.18%
CycleGAN [20]	0.405	0.527	0.623	0.503	13.34%
SRGAN [11]	0.298	0.618	0.739	0.543	5.62%
w/o g and Lap	0.446	0.543	0.628	0.526	7.9%
w/o g	0.326	0.492	0.599	0.500	28.58%
w/o Lap	0.369	0.552	0.644	0.482	5.34%
Ours	**0.289**	**0.487**	**0.593**	**0.435**	**48.58%**

TABLE I

Evaluation by LPIPS score and user study. Lower LPIPS means better visual quality. For user study, the percentage means how likely the user considers that the patch quality is good.

low-resolution patches, which allows the system to produce high-resolution results without paired correspondences. The high-resolution panoramas generated by our approach are visually appealing and exhibit better quality than the results of state-of-the-art style-transfer and super-resolution methods, in terms of LPISP scores and user study.

References

[1] P. F. Alcantarilla, A. Bartoli, and A. J. Davison. KAZE features. In *ECCV*, 2012.

[2] P. F. Alcantarilla, J. Nuevo, and A. Bartoli. Fast explicit diffusion for accelerated features in nonlinear scale spaces. In *BMVC*, 2013.

[3] H. Chen, X. He, C. Ren, L. Qing, and Q. Teng. Super-resolution of compressed images using deep convolutional neural networks. *Neurocomputing*, 2018.

[4] P. E. Debevec, C. J. Taylor, and J. Malik. Modeling and rendering architecture from photographs: A hybrid geometry- and image-based approach. In *SIGGRAPH*.

[5] M. Gharbi, J. Chen, J. T. Barron, S. W. Hasinoff, and F. Durand. Deep bilateral learning for real-time image enhancement. *ACM*, 2017.

[6] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

[7] X. Huang, M. Liu, S. J. Belongie, and J. Kautz. Multimodal unsupervised image-to-image translation. In *ECCV*, 2018.

[8] J. Kim, J. K. Lee, and K. M. Lee. Deeply-recursive convolutional network for image super-resolution. In *IEEE*, 2016.

[9] J. Kopf, B. Neubert, B. Chen, M. F. Cohen, D. Cohen-Or, O. Deussen, M. Uyttendaele, and D. Lischinski. Deep photo: model-based photograph enhancement and viewing. *ACM Trans. Graph.*, 2008.

[10] W. Lai, J. Huang, N. Ahuja, and M. Yang. Deep laplacian pyramid networks for fast and accurate super-resolution. In *CVPR*, pages 5835–5843, 2017.

[11] C. Ledig, L. Theis, F. Huszar, J. Caballero, A. Cunningham, A. Acosta, A. P. Aitken, A. Tejani, J. Totz, Z. Wang, and W. Shi. Photo-realistic single image super-resolution using a generative adversarial network. In *CVPR*, 2017.

[12] H. Lee, H. Tseng, J. Huang, M. Singh, and M. Yang. Diverse image-to-image translation via disentangled representations. In *ECCV*, 2018.

[13] Q. Lu, Q. Tao, Y. Zhao, and M. Liu. Sketch simplification based on conditional random field and least squares generative adversarial networks. *Neurocomputing*, 2018.

[14] J. M. D. Martino, G. Facciolo, and E. Meinhardt-Llopis. Poisson image editing. *IPOL Journal*, pages 300–325, 2016.

[15] S. C. Park, M. K. Park, and M. G. Kang. Super-resolution image reconstruction: a technical overview. *IEEE Signal Processing Magazine*, 2003.

[16] Y. Shih, S. Paris, C. Barnes, W. T. Freeman, and F. Durand. Style transfer for headshot portraits. *ACM Trans. Graph.*, pages 148:1–148:14, 2014.

[17] A. Shrivastava, T. Pfister, O. Tuzel, J. Susskind, W. Wang, and R. Webb. Learning from simulated and unsupervised images through adversarial training. In *IEEE*, 2017.

[18] H. Shum and S. B. Kang. Review of image-based rendering techniques. In *VCIP*, 2000.

[19] T. Wang, M. Liu, J. Zhu, A. Tao, J. Kautz, and B. Catanzaro. High-resolution image synthesis and semantic manipulation with conditional gans. *CoRR*, 2017.

[20] J. Zhu, T. Park, P. Isola, and A. A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *ICCV*, pages 2242–2251, 2017.

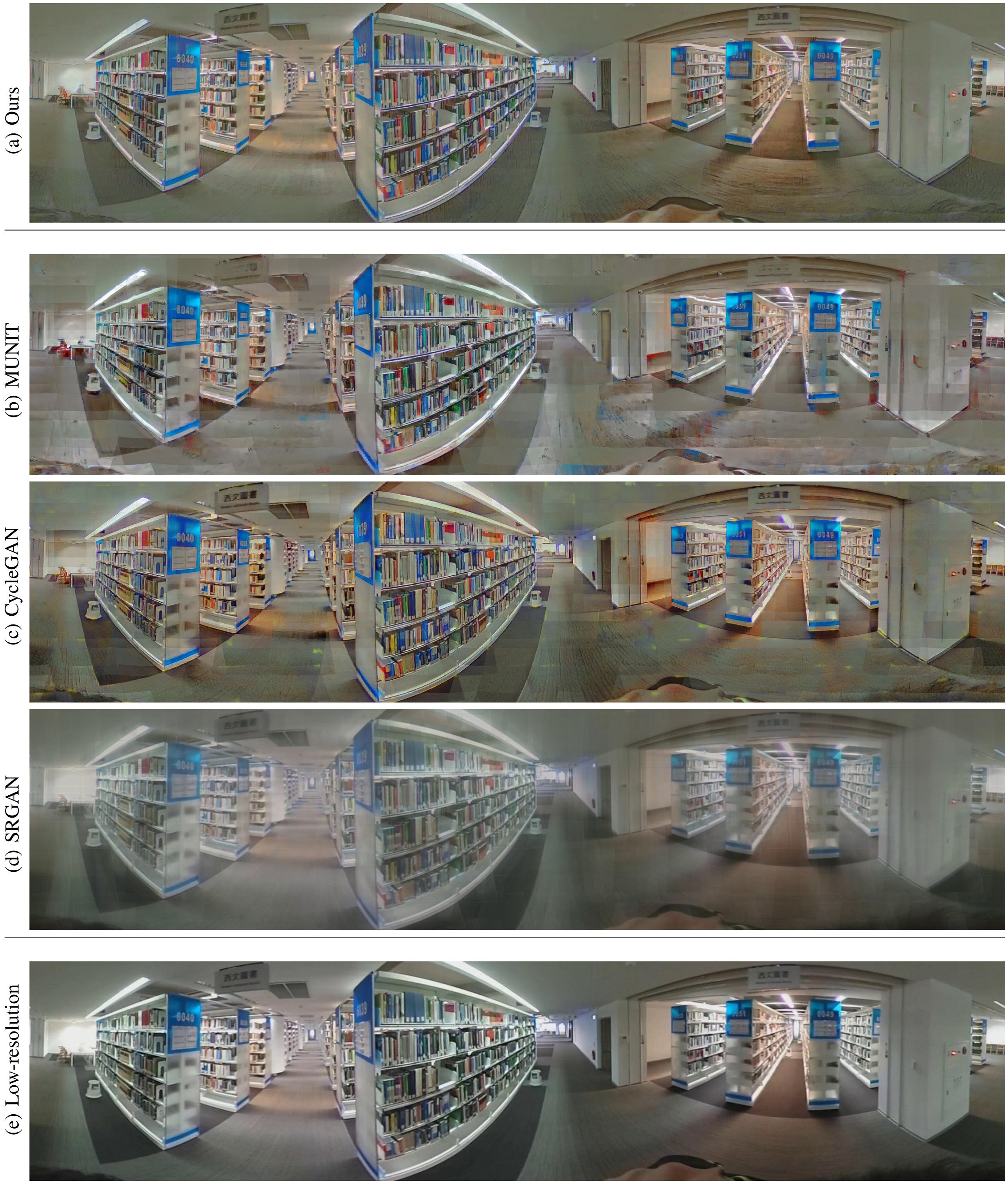


Fig. 7. (a) Our result. (b)-(d) Different methods: MUNIT, CycleGAN, SRGAN. (e) Low-resolution input.