Weakly-Supervised Stitching Network for Real-World Panoramic Image Generation

Dae-Young Song[1], Geonsoo Lee[1], HeeKyung Lee[2], Gi-Mun Um[2], and Donghyeon Cho*[1]

[1] Chungnam National University, Daejeon, South Korea
{201501747.o,geonsoo.o,cdh12242}@cnu.ac.kr
[2] Electronics and Telecommunication Research Institute, Daejeon, South Korea
{lhk95,gmum}@etri.re.kr

Abstract. Recently, there has been growing attention on an end-to-end deep learning-based stitching model. However, the most challenging point in deep learning-based stitching is to obtain pairs of input images with a narrow field of view and ground truth images with a wide field of view captured from real-world scenes. To overcome this difficulty, we develop a weakly-supervised learning mechanism to train the stitching model without requiring genuine ground truth images. In addition, we propose a stitching model that takes multiple real-world fisheye images as inputs and creates a 360° output image in an equirectangular projection format. In particular, our model consists of color consistency corrections, warping, and blending, and is trained by perceptual and SSIM losses. The effectiveness of the proposed algorithm is verified on two real-world stitching datasets.

Keywords: image stitching, 360° panoramic image

1 Introduction

Image stitching is a task that combines multiple images obtained from different viewpoints to generate a single panoramic image with a larger field of view (FOV). By exploiting this advantage, image stitching technique can be used in various applications such as street view service, virtual reality [27], video surveillance [16], and Mars exploration [6]. Traditional stitching proceeds in the order of feature point extraction, feature matching, homography estimation, warping, and blending. For instance, Brown and Lowe [3] proposed an automatic stitching method that finds correspondence of feature points using SIFT [35], estimates global homography by RANSAC [11], aligns two images using estimated homography, and combines them by multi-band blending. Since then, a lot of following methods have been developed for creating high-quality panoramic images,

* Corresponding author.
Project page is at https://eadcat.github.io/WSSN

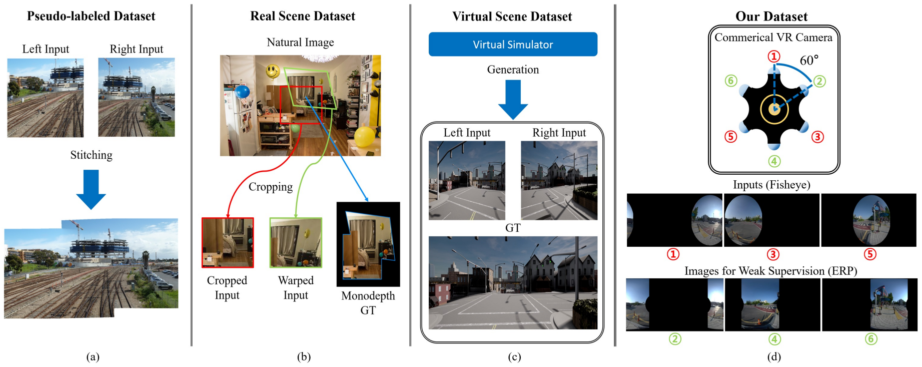

Fig. 1: Comparison of existing stitching dataset for training. (a) The pseudo-labeled dataset is constructed by existing stitching methods. (b) The real scene dataset is generated by cropping with global homography. (c) Virtual scene dataset with a simulator. (d) Our dataset for weak supervisions.

and the main research issue of them is to deal with parallax distortion caused by depth differences. To overcome parallax distortion, spatially varying multi-homography estimation methods [13, 33, 14, 52, 19, 26] and non-uniform mesh-based warping methods [54, 17, 55] have been introduced. Additionally, another challenge to consider in real-world stitching is to overcome visually unpleasant artifacts such as structural distortion and the difference in overall tones between input images. To seamlessly combine input images without suffering these distortions, constraints on line and structure can be explicitly included in the stitching algorithm [51, 30, 21], and color consistency correction can be applied to input images based on a parametric color model [10, 50]. However, the aforementioned approaches depend on the performance of the algorithm that estimates the correspondence of feature points between the input images. Therefore, when the overlapping area between input images is too small or there are many repetitive patterns, feature matching becomes challenging, resulting in parallax distortion and visually unpleasant artifacts, or stitching itself may fail. In other words, the success rate of stitching depends on the performance of the matching algorithm.

Recently, the limitations of these traditional approaches have been solved by the CNN-based feature matching technique [44] and deep homography estimation methods [8, 38, 48, 24, 56]. Furthermore, researches on modeling the entire stitching process as a single pipeline based on neural network are being introduced [45, 28, 23, 39, 40, 47, 7]. Unlike feature matching and homography estimation, it is difficult to construct the inputs-GT pairs for training the end-to-end deep stitching model. To get inputs-GT pairs, Shen *et al.* [45] built a unique hardware system that can capture the real-world scene with fixed viewpoints, but it cannot contain dynamic objects due to its systemic limitation. In addition, there were several efforts [27, 28, 7] to make pseudo GT labels by applying existing stitching methods to real-world images. However, pseudo GT labels may be sensitive to the methods used to create them. In [39], inputs-GT pairs were constructed by cropping sub-images from natural images with random geometric transformations. However, it cannot cover various depths because it is a crop-based method. To handle multiple depth layers and moving objects, there have been studies [23, 47] using a virtual simulator such as CARLA [9] to generate inputs-GT pairs. However, it is difficult to use stitching models trained with synthetic datasets on real-world images without the help of domain adaptation. In summary, constructing real inputs-GT pairs that take the depth of the scene into account for training an end-to-end stitching model is a very challenging problem. Therefore, in this paper, we present a weakly-supervised learning method for training a deep stitching model. To this end, we use a commercial camera to capture six fisheye images uniformly rotated at 60° intervals. We use half of the captured images (0°, 120°, 240°) as inputs and the other remaining images (60°, 180°, 300°) as weak supervisions. Note that all images are captured simultaneously, thus dynamic scenes and objects can be covered in our dataset. Then, we introduce a novel mechanism to train an end-to-end stitching model using our dataset. Meanwhile, we develop a deep stitching model that performs color consistency corrections, warping, and blending. In addition, our model and training mechanism can be applied to the existing pseudo GT-based dataset [27]. Comparisons of training datasets for the stitching model are shown in Fig. 1. Our contributions can be summarized as follows:

- We introduce a novel weak-supervised method for training a stitching network to create real-world 360° panoramic images.
- Our stitching model can effectively deal with parallax distortion due to depth differences as well as inconsistent colors between input images.
- We provide a variety of ablation studies, including the results of training the proposed model using the existing CROSS dataset [27].

2 Related Works

In this section, we review both traditional stitching methods and recent deep learning-based stitching methods.

2.1 Traditional Stitching Methods

After Brown and Lowe [3] introduced an automatic stitching method using SIFT feature [35], RANSAC [11], and multi-band blending, lots of follow-up studies addressing various issues have been introduced.

Parallax distortion. To handle multiple depth layers in the scene, Gao *et al.* [13] proposed a method that estimates dual homography for two separate regions: ground plane and distant plane. Lin *et al.* [33] introduced a spatially varying affine field to adaptively align pixels. Zaragoza *et al.* [54] proposed as-projective-as-possible (APAP) image stitching based on moving direct linear transformation (Moving DLT) for allowing local non-projective deviations. In [55], input images are aligned by estimated homography, then content-preserving warping is applied to solve local parallax distortion. However, the existing methods have problems such as perspective distortion when stitching multiple images. Perazzi *et al.* [43] proposed a video stitching technique using multiple scenes from unstructured camera arrays. It deals well with parallax and perspective distortions, but takes a long time due to its large computational complexity. In addition, seam-driven image stitching that finds the best homography based on the quality of seam-cut was introduced in [14], while seam-guided local alignment methods were proposed in [31]. Herrmann *et al.* [19] proposed a robust stitching method that generates multiple registrations and combines them using Markov random field (MRF) with energy terms discouraging duplication and tearing effects. Recently, Lee and Sim [26] introduced a novel concept of warping residual to deal with large parallax using locally optimal warping. For a similar goal, our network includes warping operations that take global and local information into account.

Visually unpleasant distortion. In human visual perception, distortion tends to be particularly noticeable on thin objects such as lines or curves. Xiang *et al.* [51] proposed a line-guided local warping method with a global similarity constraint to overcome projective distortions. Liao *et al.* [30] presented two single-perspective warpings consisting of parametric warping and mesh-based warping for enhancing the naturalness of stitched images. Jia *et al.* [21] presented a structure-preserving method based on line-guided warping and line-point constraint. Also, methods exploiting semantic information about pedestrians [12], faces [41], human perception [29] and objects [18] were introduced for natural stitching. In addition, color and tone differences between input images are noticeable distortions. Especially in the near of the seam lines, the distortion becomes more prominent. Doutre and Nasiopoulos [10] proposed a method that corrects differences in color between images using simple linear regression. A more advanced color consistency correction method using convex quadratic programming for the stitching problem is proposed in [50]. To satisfy human visual perception, we utilize perceptual loss [25] in the training step and include color correction operation in our stitching model.

2.2 Deep Learning-based Stitching Methods

To train a deep stitching model, it is necessary to construct pairs of input images with a narrow FOV and a GT image with a wide FOV. Shen *et al.* [45] built the hardware system with a flat mirror to create the dataset and trained a stitching model using the constructed dataset. Since it is not practical to use a specialized camera, there have been several studies that make inputs-GT pair as follows.

Dataset with pseudo GT. Li *et al.* [27] captured 4 fisheye images taken by lenses rotated at 90° intervals. Then, two images facing opposite directions are used as inputs, and the stitched image using the other two images is used as a pseudo GT image. To create the stitched images, a method with the highest mean opinion score (MOS) among existing stitching methods is used for each image. Using this dataset, Li *et al.* [28] introduced an attentive deep stitching approach consisting of two modules for deformation and resolution. Similarly, Dai *et al.* [7] generated pseudo GT images using existing stitching methods and used them to train an edge-guided composition network. An example of pseudo GT is illustrated in Fig. 1-(a). However, pseudo GT labels are sensitive to the methods used to generate them.

Dataset with only global homography. Nie *et al.* [39] presented a deep learning-based view-free stitching model consisting of global homography estimation, structure stitching, and content revision. For the training, they constructed inputs-GT pairs using natural images such as the COCO dataset [32] as shown in Fig. 1-(b). Specifically, given an image, two sub-images having overlapping regions are extracted, then geometric transformation is applied to one of them. Thus, these two sub-images have different perspectives, and their geometric relationship can be modeled by a global homography. These two sub-images are used as inputs to the stitching model while an image containing both sub-images is used as GT. However, the problem with their dataset is that depth is not considered when generating two input images. It means that a single depth layer is assumed, which is unrealistic in the real-world scenario. As a result, there is a limitation to stitching images containing scenes with multiple depth layers. Furthermore, parallax distortion caused by depth differences that may occur in the real-world environment cannot be dealt with. Recently, Nie *et al.* [40] proposed an unsupervised learning method for a view-free stitching model composed of coarse alignment and image reconstruction. However, the unsupervised coarse alignment module is performed by a global homography. Thus, parallax distortion induced by depth difference still causes visual artifacts, even though the image reconstruction module enhances the quality of the output image.

Dataset using virtual simulator. There have been several studies that train a stitching model by using inputs-GT pairs generated from a virtual simulator such as CARLA [9]. Since it is possible to control camera configuration and the scene in the virtual space, depth information can be included in the relationship between inputs as shown in Fig. 1-(c). Thus, with these virtual datasets, parallax distortion due to depth differences can be covered. Using the virtual dataset, Lai *et al.* [23] proposed a pushbroom stitching network that estimates flow maps in fixed view, and Song *et al.* [47] developed an end-to-end virtual image stitching network via multi-homography estimation. However, these stitching models trained with virtual dataset has limitations in applying them to real-world images, and additional techniques such as domain adaptation may be required. In summary, it is hard to obtain real-world datasets that take the depth information of the scenes into account. In this paper, we use real-world images captured at different viewpoints themselves as inputs to the stitching model. In this case, there are no GT images with a wide FOV, thus we propose a new mechanism for training the stitching model.

3 Approach

In this section, we first describe the procedure of generating training data using real-world fisheye images. Then, we define the problem setup for creating a 360°

panoramic image and introduce an architecture of the proposed stitching model to solve the defined problems. Finally, loss functions for training the proposed stitching model are explained.

3.1 Dataset Preparation

To construct the training data for learning the real-world stitching model, we use a commercial VR camera called Kandao Obsidian R [1] to acquire fisheye images. It can capture six fisheye images simultaneously using six lenses rotated at 60° intervals. We use three fisheye images rotated by 0°, 120°, 240° as inputs to our stitching model while the remaining three images rotated by 60°, 180°, 300° are utilized as weak supervisions. As shown in Fig. 1-(d), overlapping areas between the input images correspond to the central regions of the images for weak supervisions. Therefore, when training a stitching model using three input images, the remaining three images can be used as weak supervisions. Input images are used as themselves whereas pre-processing is applied on images for weak supervisions. Two types of pre-processing are performed on images for weak supervisions as follows.

Geometric calibration. We represent the GT 360° panoramic image in equirectangular projection (ERP) format. Since there are no genuine GT images in our setup, it is required to register images for weak supervisions as much as possible in the GT format in advance. Therefore, we transform images for weak supervisions into ERP coordinates. To this end, we perform geometric calibration for fisheye cameras to compute intrinsic and extrinsic parameters by utilizing NVIDIA VRWorks 360 Video SDK [42]. As shown in Fig. 1-(d), three fisheye images for weak supervisions are well projected on ERP coordinates.

Color consistency correction. Multi-view images captured in a real-world environment may have different color tones. To utilize three images for weak supervisions as GT, the color tones of them should be matched consistently. Therefore, we correct the color consistency of three images for weak supervisions in advance. We use the polynomial curve mapping function to convert the color values of two images (called query image) to the those of remaining one image (called reference image) as

$$\bar{x} = ax^2 + bx + c, \tag{1}$$

where x is the original pixel value in query images, $\bar{x}$ is the corrected pixel value, and a, b, and c are learnable parameters of the polynomial model, respectively. We estimate a, b, and c as follows. First, we find correspondences between query and reference images using SuperGlue [44], then extract patches centered at matched points from both images. Then, we minimize mean squared error (MSE) between corrected patches from query images by (1) and extracted patches from the reference images. Then, we obtain three images with consistent color tones in ERP format by using (1) with the learned a, b, and c. These three images are used for weak supervisions to train our stitching model.

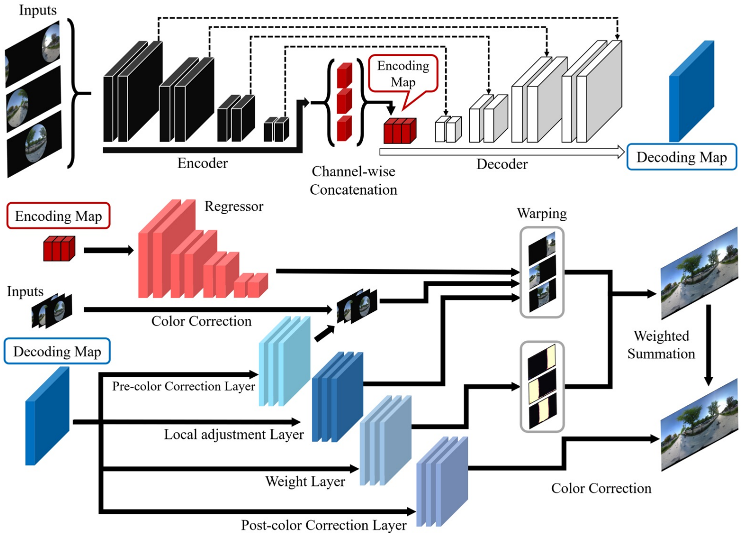

Fig. 2: The entire pipeline of our stitching model. Our model takes N inputs and produces global warping maps, pre- and post-color correction maps, local adjustment maps, and weight maps. After extracting an encoding map (red) and a decoding map (blue), a final warping map U_n is created by adding a global warping map and a local adjustment map, and then the input is warped with U_n. All warped images are weighted by weight maps and merged into a panorama. Color correction is applied once before warping and once after weighted summation using color correction maps.

3.2 Problem Definition

In this paper, we aim to create a 360° panoramic image by stitching N adjacent images taken with fisheye lenses rotated at different angles. Our stitching model $\mathbf{S}(\cdot)$ takes N fisheye images I_n as inputs and generates a pre-color correction map C_n^{pre}, a global warping map G_n, a local warping adjustment map L_n, and a weight map W_n for each input image. It also produces a post-color correction map C^{post} for an input pair of N images. It is defined as

$$\mathbf{S}(I_1, ..., I_N; \theta) \rightarrow (C_n^{pre}, G_n, L_n, W_n, C^{post}), \tag{2}$$

where θ and n are learnable parameters of $\mathbf{S}(\cdot)$ and the index of input images, respectively. In our experiment, N is 3, the vertical FOV of each fisheye image is 185°, and the lens for each input is rotated 60° from each other. The panoramic image is created by applying all estimates from the stitching model in (2) to the input fisheye images.

3.3 Architecture

Our stitching model $\mathbf{S}(\cdot)$ is composed of an encoder $\mathbf{E}(\cdot)$, a regressor $\mathbf{R}(\cdot)$, and a decoder $\mathbf{D}(\cdot)$ as illustrated in Fig. 2. The role and details of each component are described as follows.

Encoder. In our stitching model, there are N encoders to extract visual features f_n of each input fisheye image as

$$f_n = \mathbf{E}(I_n; \theta_e), \tag{3}$$

where I_n is one of the input fisheye images and θ_e is the learnable parameters of the encoder. Our encoder consists of a series of convolutional layers, batch normalization layers, and ELU activations [5]. Learnable parameters of each encoder are shared. Visual features extracted from each input image are concatenated along the channel axis and used as input for a regressor and a decoder.

Regressor. The purpose of the regressor is to find affine transformation matrices that can warp the pixel values of each input image to the pixel coordinates of the output image globally. The regressor takes the visual features f_n as input and generates affine matrices A_n as follows.

$$A_n = \mathbf{R}(f_n; \theta_r), \tag{4}$$

where θ_r is the learnable parameters of the regressor. Using the estimated affine matrices, a global warping map G_n for each input image is created. The global warping map contains x- and y-direction information on where the pixels of the input image are moved to the coordinates of the output pixels. Global warping can be viewed as global registration by a single homography.

Decoder. Except for the global warping map G_n, the remaining components in (2) needed to make the final output are generated by the decoder as

$$\mathbf{D}(f_n; \theta_d) \rightarrow (C_n^{pre}, L_n, W_n, C^{post}), \tag{5}$$

where θ_d is the learnable parameters of the decoder. Specifically, the decoder consists of a shared decoder and four private decoders for each output component. The shared decoder takes the visual features obtained from the encoder as inputs and generates shared features. Shared features are passed as input to each private decoder to create each output component. A shared decoder consists of a series of convolutional blocks and upsampling layers, and each private decoder consists of several convolutional blocks.

Output generation processes. First, the color of the N input images with different color tones is corrected by using the estimated pre-color correction map C_n^{pre}. Inspired by Zero-DCE [15], we convert the color intensity values of input images by a monotonic quadratic curve as follows:

$$\hat{I}_n = I_n + C_n^{pre} I_n (1 - I_n). \tag{6}$$

The color-corrected images $\hat{I}_n$ will have color tones harmonized with each other. Then, each color-corrected input fisheye image is warped to the output pixel

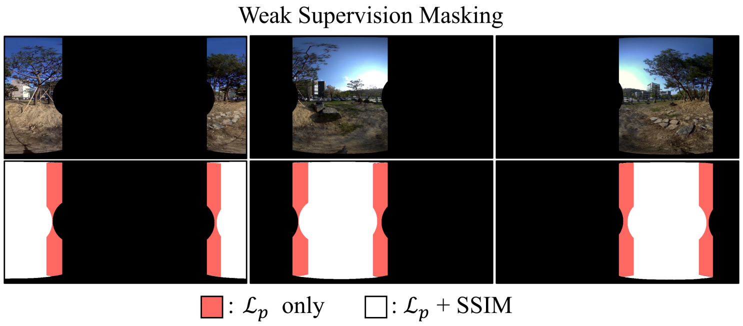

Fig. 3: The specific application area of loss functions. Note that SSIM loss is valid only in the white area.

grid. Our output image is in 360° ERP format. Warping is performed by a global warping map G_n and a local warping adjustment map L_n for each input image. Since the entire depth layer of the scene cannot be covered with only a single global warping map, the local warping adjustment map is used to supplement it as follows.

$$U_n = G_n + \alpha L_n, \tag{7}$$

where α is a balancing factor and set to 0.3 in our experiment. Using the final warping map U_n, color-corrected fisheye images $\hat{I}_n$ are warped as

$$\bar{I}_n = \mathbf{warp}(\hat{I}_n, U_n), \tag{8}$$

where $\mathbf{warp}(\cdot)$ is a pixel mapping function. After that, all warped images are weighted and merged to create a panoramic image P as follow:

$$P = \sum_{n=1}^{N} \bar{I}_n W_n, \tag{9}$$

where W_n is a per-pixel weight map for fusing warped images. Finally, a post-color correction map C^{post} is applied to generate the final panoramic image O as

$$O = P + C^{post} P(1 - P). \tag{10}$$

Detailed formulations of the architecture are in the supplementary material.

3.4 Training

Learnable parameters of our stitching model $\mathbf{S}(\cdot)$ are trained using the images for weak supervisions generated by the method described in Section 3.1. Since genuine GT images do not exist in our settings, we use perceptual loss [25] instead of pixel-wise loss as follows:

$$\mathcal{L}_p(\theta) = \sum_{n=1}^{N} \sum_{i=3}^{5} \mathcal{L}_1(\phi_i(\bar{O}_n), \phi_i(M_n O)), \tag{11}$$

where $\mathcal{L}_1(\cdot)$ and $\phi_i(\cdot)$ are functions of L_1 distance and feature extractor at i-th maxpooling layer of VGG16 [46], respectively. $\bar{O}_n$ represents the image for weak supervisions and M_n is the mask representing the valid pixels of $\bar{O}_n$. Note that M_n is the union of the red and white areas in the bottom row of Fig. 3. Also, for the consistency in color tone and contrast of the input images, we use SSIM loss as follows:

$$\mathcal{L}_{SSIM}(\theta) = \sum_{n=1}^{N}[(1 - SSIM(\hat{M}\bar{O}_n, \hat{M}O))], \tag{12}$$

where $SSIM(\cdot)$ is a function of the structural similarity [49], and $\hat{M}$ is a mask representing non-overlapping regions between the images for weak supervisions. By using this loss function, our model can harmonize the color tone in the overlapping regions between inputs. Note that $\hat{M}$ is only white areas in the bottom row of Fig. 3. Overall loss for training our stitching model is defined as

$$\mathcal{L}(\theta) = (1 - \lambda)\mathcal{L}_p(\theta) + \lambda\mathcal{L}_{SSIM}(\theta), \tag{13}$$

where λ represents the balancing factor between two losses. We set λ to 0.4 in our experiments.

4 Experiments

4.1 Implementation Details

For the experiments, we use our dataset as well as the CROSS dataset [27]. For our dataset, we use 47,063 sets of images for the training and 1,400 for the test. Each training set includes three input fisheye images, three ERP images for weak supervisions, and three masks. For the CROSS, we divide the dataset into 1,146 for the training and 128 for the test. Each set of the CROSS includes two fisheye inputs, and a GT that is pseudo-labeled by SamsungGear. SamsungGear's MOS obtain the highest in most data, thus we choose it as our pseudo-labeling method. For both datasets, all images have a resolution of 1024 × 512, and data augmentations such as brightness and tone adjustments are randomly applied during the training. Our model was trained by Adam Optimizer [22] with a learning rate of 0.0004. The number of epochs for our dataset and the CROSS is set to 20 and 1200, respectively. Our method is implemented using Pytorch 1.8.1 with CUDA 11.1 on Ubuntu 18.04.

4.2 Comparisons

Results on our dataset. Since there are no genuine GT images in our datasets, we utilize a perceptual distance P_d using VGG16 as an evaluation metric. The perceptual distance is computed by using making in the same way as in the

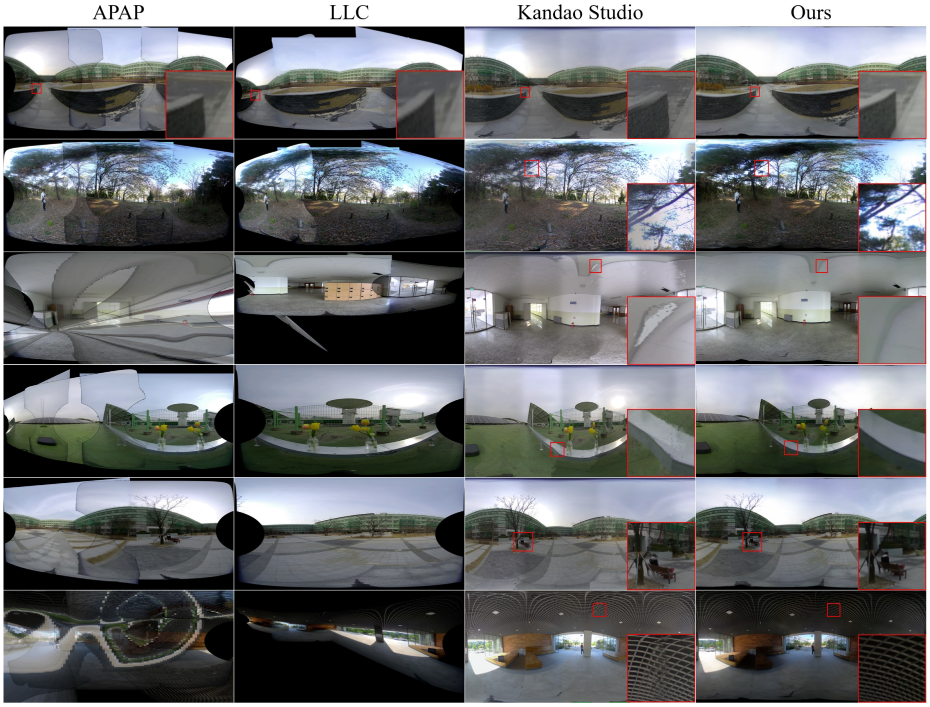

Fig. 4: Qualitative comparisons on our dataset. Please refer to the supplementary material for the rest of the test examples.

training step, but there is a difference that unlike in training, the distance is calculated using all five feature maps from five max pooling layers as follows:

$$P_d = \sum_{n=1}^{N} \sum_{i=1}^{5} \mathcal{L}_1(\phi_i(\bar{O}_n), \phi_i(M_n O)), \tag{14}$$

As a result, P_d can evaluate low-level features such as edges. Since our model is trained in the same way, this evaluation can be unfair. Therefore, to compensate for this, we also utilize SIQE [36], LPIPS [57], and FID [20] as quantitative evaluation metrics. As for the competition methods, APAP [54], LLC [21], and Kandao Studio [1] are selected for which the softwares are publicly available. We use ERP format input images for the APAP and the LLC because they were not developed for fisheye inputs. Qualitative comparisons are shown in Fig. 4. Our method produces the most natural, high-quality 360° panoramic images without structural distortions and color inconsistency. In Table 1, there are quantitative comparisons with the existing methods. As ablation studies, we also compare our model without a post-color correction map. In addition, we measure the average running time per image for all methods. Note that the running time of the kandao studio includes time for saving a 1920 × 960 image because there are

Table 1: Quantitative result of our 1,400 test dataset. **bold**: best. Note that Ours† is our model without the post-color correction map.

Metric	APAP [54]	LLC [21]	Kandao [1]	Ours†(CPU/GPU)	Ours (CPU/GPU)
Time Spent (s)	8.5887	16.5126	0.8275	1.3010/**0.0347**	1.3870/0.0363
P_d (↓)	6.498	6.625	5.308	2.773	**2.731**
LPIPS (Alex) (↓)	0.647	0.722	0.266	0.122	**0.118**
LPIPS (VGG16) (↓)	0.652	0.690	0.408	0.178	**0.175**
SIQE [36] (↑)	22.644	20.602	29.399	**39.528**	37.714
FID [20] (↓)	585.6	608.1	224.0	**132.4**	140.8

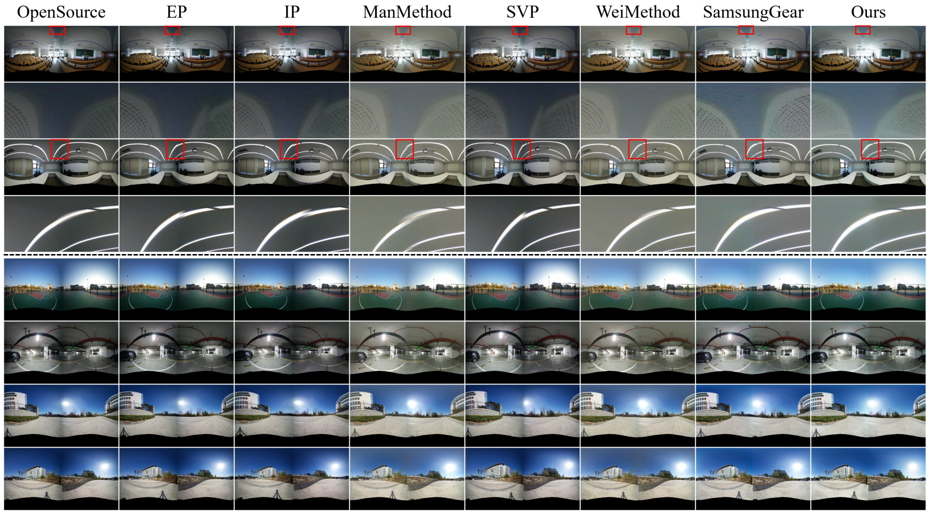

Fig. 5: Qualitative results on CROSS dataset. Top: our method well preserves structural patterns compared to existing stitching models. Bottom: our method produces more color-consistent results than other existing methods.

no open-source codes. As reported in Table 1, our method performs better and much faster than the existing methods. However, the results were not significantly different according to a post-color correction map. Also, as expected, the proposed method using GPU acceleration is much faster than other algorithms, including the commercial kandao studio.

Results on the CROSS. To validate the versatility of the proposed method, we evaluate the proposed method on the CROSS dataset, which contains pseudo GT 360° panoramic images as supervisions. As shown in Fig. 5, our method produces more visually pleasing results compared to the existing stitching methods. In particular, our results demonstrate robustness to structural distortion and vignetting artifacts. To measure PSNR, SSIM, and P_d, we use the pseudo-labeled GT images, because the SamsungGear method obtains the highest MOS in [27]. Note that $M_n = 1$ in all pixels because masking is not required. As reported in Table 2, the proposed model outperforms the existing methods.

Table 2: Quantitative comparisons on CROSS dataset [27]. **bold**: best.

Metric	OpenSource [2]	EP [34]	IP [4]	ManMethod	SVP [37]	WeiMethod [53]	Ours
PSNR (↑)	16.417	15.908	15.177	18.943	16.110	18.730	**22.440**
SSIM (↑)	0.589	0.565	0.546	0.611	0.562	0.595	**0.736**
P_d (↓)	3.31	3.55	3.79	3.12	3.69	3.23	**2.53**

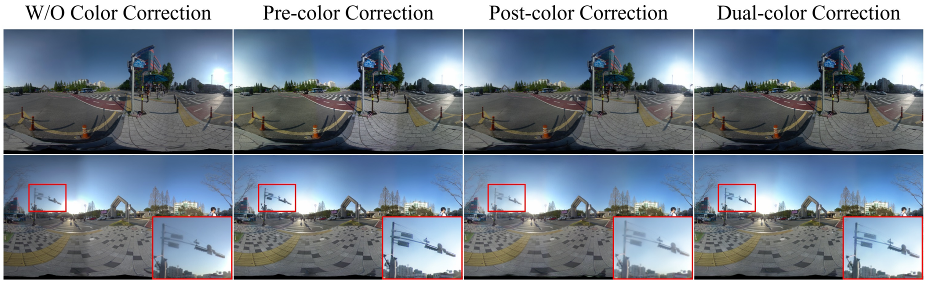

Fig. 6: Ablation studies of our model. The w/o color correction and the pre-color correction model have the unpleasant boundaries. The post-color correction model is suffered from fading.

4.3 Ablation Studies

Effects of color correction. We conduct experiments depending on whether the pre-color correction map C_n^{pre} and the post-color correction map C^{post} are used. As shown in Fig. 6, the color tone around boundary lines is inconsistent when only pre-color correction is applied. In addition, results of only using post-color correction suffer from fading effects as shown in the second row of Fig. 6. Overall, the dual-color correction model using both pre-color correction and post-color correction produces the most comfortable results.

Effects of loss. Since the images for weak supervisions have parallax between themselves, it may not be appropriate to use a common pixel-wise regression loss. Considering this point, we adopt the perceptual loss as in (11). Therefore, as ablation studies, we train our model with a pixel-wise L_1 loss instead of (11). As shown in Fig. 7, models trained using L_1 loss are vulnerable to parallax distortion, which causes noticeable distortion.

Table 3: Self-comparisons according to the number of warpings K on our full test dataset (14 sets). The number of epochs is set to 10. **bold**: best.

Metrics	K=1	K=2	K=3	K=4	K=5
P_d (↓)	4.001	3.936	3.944	**3.904**	3.962
SIQE (↑)	**22.89**	21.16	15.28	17.89	18.29

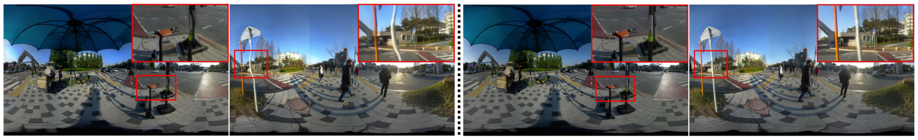

Fig. 7: Effects of loss. Utilizing L_1 loss instead of $\mathcal{L}_p$ (left). Ours (right).

Effects of the number of warpings. Inspired by [47], we modified our model to perform multiple K warpings. However, as shown in Table 3, multiple warpings do not have a significant effect on the quantitative results. We guess that it is because our model uses different input and output coordinates from the stitching model in [47]. Note that cylindrical coordinates are used in [47] while our model is operated on fisheye input and ERP output. Based on the above results, we use the simplest model with $K = 1$ for all experiments.

5 Limitations and Future Works

Even though our model can be trained without genuine GTs, our research does not take view-free inputs into account. We believe that subsequent studies based on this paper can be extended to studies on view-free stitching. Another promising future work is video stitching to cover dynamic scenes. Although the proposed method is developed for a static scene, it can be extended to video, and we believe that temporal artifacts such as waving effects can be solved by temporal consistency loss as in [23].

6 Conclusion

In this paper, we present a weakly supervised method for training the real-world stitching model. Our model takes multiple fisheye images as inputs and generates a 360° panorama image. For training, we generate images of weak supervisions and utilize them for perceptual and SSIM losses. We verify the proposed method on our stitching dataset as well as the CROSS dataset. Through the various experiments, we demonstrate superior stitching performance over existing methods. In particular, it is more robust to structural artifacts and color inconsistency problems compared to existing methods.

Acknowledgement

This work was supported by Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korea government(MSIT) (No. 2018-0-00207, Immersive Media Research Laboratory) and the National Research Foundation of Korea(NRF) grant funded by the Korea government(MSIT) (No.2021R1A4A1032580, No.2022R1C1C1009334).

References

1. Kandao. https://www.kandaovr.com/, accessed: 2022-03-05
2. Dualfisheye. https://github.com/ooterness/DualFisheye (2016)
3. Brown, M., Lowe, D.G.: Automatic panoramic image stitching using invariant features. Int'l Journal of Computer Vision (IJCV) **74**(1), 59–73 (2007)
4. Cai, D., He, X., Han, J.: Isometric projection. In: Association for the Advancement of Artificial Intelligence (AAAI). pp. 528–533. AAAI Press (2007)
5. Clevert, D.A., Unterthiner, T., Hochreiter, S.: Fast and accurate deep network learning by exponential linear units (elus). International Conference on Learning Representation (ICLR) (2016)
6. Coates, A., Jaumann, R., Griffiths, A., Leff, C., Schmitz, N., Josset, J.L., Paar, G., Gunn, M., Hauber, E., Cousins, C.R., et al.: The pancam instrument for the exomars rover. Astrobiology **17**(6-7), 511–541 (2017)
7. Dai, Q., Fang, F., Li, J., Zhang, G., Zhou, A.: Edge-guided composition network for image stitching. Pattern Recognition (PR) **118**, 108019 (2021)
8. DeTone, D., Malisiewicz, T., Rabinovich, A.: Deep image homography estimation. CoRR **abs/1606.03798** (2016)
9. Dosovitskiy, A., Ros, G., Codevilla, F., Lopez, A., Koltun, V.: CARLA: An open urban driving simulator. In: Proc. of Conference on Robot Learning (CoRL). pp. 1–16 (2017)
10. Doutre, C., Nasiopoulos, P.: Fast vignetting correction and color matching for panoramic image stitching. In: IEEE Int'l Conf. on Image Processing (ICIP). pp. 709–712 (2009)
11. Fischler, M.A., Bolles, R.C.: Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. Communications of the ACM **24**(6), 381–395 (1981)
12. Flores, A., Belongie, S.: Removing pedestrians from google street view images. In: Proc. of Computer Vision and Pattern Recognition Workshops (CVPRW). pp. 53–58. IEEE (2010)
13. Gao, J., Kim, S.J., Brown, M.S.: Constructing image panoramas using dual-homography warping. In: Proc. of Computer Vision and Pattern Recognition (CVPR). pp. 49–56. IEEE (2011)
14. Gao, J., Li, Y., Chin, T.J., Brown, M.S.: Seam-driven image stitching. In: Eurographics. pp. 45–48 (2013)
15. Guo, C., Li, C., Guo, J., Loy, C.C., Hou, J., Kwong, S., Cong, R.: Zero-reference deep curve estimation for low-light image enhancement. In: Proc. of Computer Vision and Pattern Recognition (CVPR). pp. 1780–1789 (2020)
16. He, B., Yu, S.: Parallax-robust surveillance video stitching. Sensors **16**(1), 7 (2016)
17. He, K., Chang, H., Sun, J.: Rectangling panoramic images via warping. ACM Trans. on Graph. (ToG) **32**(4), 1–10 (2013)
18. Herrmann, C., Wang, C., Bowen, R.S., Keyder, E., Zabih, R.: Object-centered image stitching. In: Proc. of European Conf. on Computer Vision (ECCV). pp. 821–835 (2018)
19. Herrmann, C., Wang, C., Strong Bowen, R., Keyder, E., Krainin, M., Liu, C., Zabih, R.: Robust image stitching with multiple registrations. In: Proc. of European Conf. on Computer Vision (ECCV). pp. 53–67 (2018)
20. Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., Hochreiter, S.: GANs trained by a two time-scale update rule converge to a local nash equilibrium. In: Proc. of Neural Information Processing Systems (NeurIPS). pp. 6626–6637 (2017)

21. Jia, Q., Li, Z., Fan, X., Zhao, H., Teng, S., Ye, X., Latecki, L.J.: Leveraging line-point consistence to preserve structures for wide parallax image stitching. In: Proc. of Computer Vision and Pattern Recognition (CVPR). pp. 12186–12195 (2021)
22. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. International Conference on Learning Representation (ICLR) (2014)
23. Lai, W.S., Gallo, O., Gu, J., Sun, D., Yang, M.H., Kautz, J.: Video stitching for linear camera arrays. In: British Machine Vision Conf. (BMVC). pp. 1–12 (2019)
24. Le, H., Liu, F., Zhang, S., Agarwala, A.: Deep homography estimation for dynamic scenes. In: Proc. of Computer Vision and Pattern Recognition (CVPR). pp. 7652–7661 (2020)
25. Ledig, C., Theis, L., Huszár, F., Caballero, J., Cunningham, A., Acosta, A., Aitken, A., Tejani, A., Totz, J., Wang, Z., et al.: Photo-realistic single image super-resolution using a generative adversarial network. In: Proc. of Computer Vision and Pattern Recognition (CVPR). pp. 4681–4690 (2017)
26. Lee, K.Y., Sim, J.Y.: Warping residual based image stitching for large parallax. In: Proc. of Computer Vision and Pattern Recognition (CVPR). pp. 8198–8206 (2020)
27. Li, J., Yu, K., Zhao, Y., Zhang, Y., Xu, L.: Cross-reference stitching quality assessment for 360 omnidirectional images. In: Proceedings of the 27th ACM International Conference on Multimedia. pp. 2360–2368 (2019)
28. Li, J., Zhao, Y., Ye, W., Yu, K., Ge, S.: Attentive deep stitching and quality assessment for 360° omnidirectional images. IEEE Journal of Selected Topics in Signal Processing **14**(1), 209–221 (2019)
29. Li, N., Liao, T., Wang, C.: Perception-based seam cutting for image stitching. Signal, Image and Video Processing **12**(5), 967–974 (2018)
30. Liao, T., Li, N.: Single-perspective warps in natural image stitching. IEEE Trans. on Image Processing (TIP) **29**, 724–735 (2019)
31. Lin, K., Jiang, N., Cheong, L.F., Do, M., Lu, J.: Seagull: Seam-guided local alignment for parallax-tolerant image stitching. In: Proc. of European Conf. on Computer Vision (ECCV). pp. 370–385. Springer (2016)
32. Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft coco: Common objects in context. In: Proc. of European Conf. on Computer Vision (ECCV). pp. 740–755. Springer (2014)
33. Lin, W.Y., Liu, S., Matsushita, Y., Ng, T.T., Cheong, L.F.: Smoothly varying affine stitching. In: Proc. of Computer Vision and Pattern Recognition (CVPR). pp. 345–352. IEEE (2011)
34. Ling, S., Cheung, G., Le Callet, P.: No-reference quality assessment for stitched panoramic images using convolutional sparse coding and compound feature selection. In: IEEE International Conference on Multimedia and Expo (ICME). pp. 1–6 (2018). https://doi.org/10.1109/ICME.2018.8486545
35. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. Int'l Journal of Computer Vision (IJCV) **60**(2), 91–110 (2004)
36. Madhusudana, P.C., Soundararajan, R.: Subjective and objective quality assessment of stitched images for virtual reality. IEEE Trans. on Image Processing (TIP) **28**(11), 5620–5635 (2019)
37. Maneshgar, B., Sujir, L., Mudur, S., Poullis, C.: A long-range vision system for projection mapping of stereoscopic content in outdoor areas. In: VISIGRAPP (1: GRAPP). pp. 290–297 (01 2017). https://doi.org/10.5220/0006258902900297
38. Nguyen, T., Chen, S.W., Shivakumar, S.S., Taylor, C.J., Kumar, V.: Unsupervised deep homography: A fast and robust homography estimation model. IEEE Robotics and Automation Letters (RAL) **3**(3), 2346–2353 (2018)

39. Nie, L., Lin, C., Liao, K., Liu, M., Zhao, Y.: A view-free image stitching network based on global homography. Journal of Visual Communication and Image Representation **73**, 102950 (2020)
40. Nie, L., Lin, C., Liao, K., Liu, S., Zhao, Y.: Unsupervised deep image stitching: Reconstructing stitched features to images. IEEE Trans. on Image Processing (TIP) (2021)
41. Ozawa, T., Kitani, K.M., Koike, H.: Human-centric panoramic imaging stitching. In: Proceedings of the 3rd Augmented Human International Conference. pp. 1–6 (2012)
42. Patil, T., Turkowski, K.: Calibrating stitched videos with VRWorks 360 video SDK. https://developer.nvidia.com/blog/calibrating-videos-vrworks-360-video/ (2018)
43. Perazzi, F., Sorkine-Hornung, A., Zimmer, H., Kaufmann, P., Wang, O., Watson, S., Gross, M.: Panoramic video from unstructured camera arrays. In: Computer Graphics Forum. vol. 34, pp. 57–68. Wiley Online Library (2015)
44. Sarlin, P.E., DeTone, D., Malisiewicz, T., Rabinovich, A.: Superglue: Learning feature matching with graph neural networks. In: Proc. of Computer Vision and Pattern Recognition (CVPR). pp. 4938–4947 (2020)
45. Shen, C., Ji, X., Miao, C.: Real-time image stitching with convolutional neural networks. In: 2019 IEEE International Conference on Real-time Computing and Robotics (RCAR). pp. 192–197. IEEE (2019)
46. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. International Conference on Learning Representation (ICLR) (2015)
47. Song, D.Y., Um, G.M., Lee, H.K., Cho, D.: End-to-end image stitching network via multi-homography estimation. IEEE Signal Process. Lett. (SPL) **28**, 763–767 (2021)
48. Wang, C., Wang, X., Bai, X., Liu, Y., Zhou, J.: Self-supervised deep homography estimation with invertibility constraints. Pattern Recognition Letters (PRL) **128**, 355–360 (2019)
49. Wang, Z., Bovik, A.C., Sheikh, H.R., Simoncelli, E.P.: Image quality assessment: from error visibility to structural similarity. IEEE Trans. on Image Processing (TIP) **13**(4), 600–612 (2004)
50. Xia, M., Yao, J., Xie, R., Zhang, M., Xiao, J.: Color consistency correction based on remapping optimization for image stitching. In: Proc. of Int'l Conf. on Computer Vision Workshops (ICCVW). pp. 2977–2984 (2017)
51. Xiang, T.Z., Xia, G.S., Bai, X., Zhang, L.: Image stitching by line-guided local warping with global similarity constraint. Pattern Recognition (PR) **83**, 481–497 (2018)
52. Xu, B., Jia, Y.: Wide-angle image stitching using multi-homography warping. In: IEEE Int'l Conf. on Image Processing (ICIP). pp. 1467–1471 (2017)
53. Ye, W., Yu, K., Yu, Y., Li, J.: Logical stitching: A panoramic image stitching method based on color calibration box. In: 2018 14th IEEE International Conference on Signal Processing (ICSP). pp. 1139–1143 (2018). https://doi.org/10.1109/ICSP.2018.8652363
54. Zaragoza, J., Chin, T.J., Brown, M.S., Suter, D.: As-projective-as-possible image stitching with moving dlt. In: Proc. of Computer Vision and Pattern Recognition (CVPR). pp. 2339–2346 (2013)
55. Zhang, F., Liu, F.: Parallax-tolerant image stitching. In: Proc. of Computer Vision and Pattern Recognition (CVPR). pp. 3262–3269 (2014)

56. Zhang, J., Wang, C., Liu, S., Jia, L., Ye, N., Wang, J., Zhou, J., Sun, J.: Content-aware unsupervised deep homography estimation. In: Proc. of European Conf. on Computer Vision (ECCV). pp. 653–669. Springer (2020)
57. Zhang, R., Isola, P., Efros, A.A., Shechtman, E., Wang, O.: The unreasonable effectiveness of deep features as a perceptual metric. In: Proc. of Computer Vision and Pattern Recognition (CVPR). pp. 586–595 (2018)