Deep Upright Adjustment of 360 Panoramas Using Multiple Roll Estimations

Junho Jeon, Jinwoong Jung, and Seungyong Lee

POSTECH, Pohang, South Korea
{zwitterion27, jinwoong.jung, leesy}@postech.ac.kr

Abstract. Misalignment of the orientations between a 360 camera and the scene results in a wavy and distorted spherical panorama image, which may look unstable and have poor perceptual quality. To automatically correct such mis-oriented 360 panoramas, this paper proposes a novel upright adjustment framework based on a convolutional neural network. Instead of directly predicting the 3D rotation of the camera on a given panorama image, our method estimates the rotation by analyzing the projected 2D rotations of multiple images sampled from the panorama. To accurately estimate the rotations of 2D sampled images, we train a 2D roll estimation network using a large-scale labeled image dataset generated by cropping 360 spherical panoramas with various view orientations. Experimental results demonstrate that the proposed method accurately and robustly handles upright adjustment of rotated panoramas while outperforming the previous methods on test datasets that consist of a variety of scenes.

1 Introduction

With the progress of imaging sensors, consumer-level 360 panorama cameras (e.g., Ricoh Theta S, Samsung Gear 360) become cheaper and popular. Typical 360 cameras do not have viewfinder screens while simultaneously capturing the entire view directions (omni-direction) in any poses, so casual users tend not to care about the camera orientations during the capture. However, misalignment of the orientations between the camera and the scene results in wavy and distorted panorama images (Fig. 1a). The wavy horizon and distorted objects look visually unstable, especially when the panorama is used for a VR application with a narrow field-of-view display, such as HMD [14]. Moreover, the objects on the wavy horizon may suffer from severe deformations, which drastically reduce the perceptual quality of panorama images as salient parts of a scene are usually located around the horizon.

Correction of this mis-orientation of the camera is called *upright adjustment* [19], and it requires the estimation of the 3D camera rotation relative to the scene. Few recent works [3, 14] addressed upright adjustment of 360 panoramas by analyzing the structural features of the scene, especially horizontal and vertical lines as well as vanishing points. Unfortunately, these methods work well only if the given scene follows the specific assumption on the orthogonality of

(a) input distorted panorama (b) upright adjustment result

Fig. 1. Our upright adjustment method works accurately and robustly for a severely rotated 360 spherical panorama even when the vanishing structures are not clearly detectable.

scene structures, such as Manhattan or Atlanta world [6, 26], which could be violated in many cases, e.g., natural scenes.

In this paper, we propose a novel automatic upright adjustment framework for 360 panorama images based on a deep convolutional neural network. Instead of using hand-crafted features extracted from a given 360 spherical panorama, our method indirectly estimates the camera rotation by analyzing the projected 2D rotations estimated from multiple narrow field-of-view images sampled from the input panorama. Our proposed framework exploits the semantically trained CNN features rather than straight line segments and vanishing points, and can robustly handle a variety of scenes no matter whether they follow the Manhattan or Atlanta world assumption.

Various experiments on a test dataset show that our method achieves the state-of-the-art performance in both accuracy and robustness, compared to previous methods that are based on straight lines and vanishing points detected in the scene. Furthermore, our method works well on datasets of various scene categories which do not follow the Manhattan and Atlanta world assumptions, demonstrating the benefits of our deep neural network based approach that could learn the semantic upright direction of the scene even when the horizontal and vertical lines are not clearly detectable.

In summary, the key contributions of our work are as follows:

- we propose a 3D camera rotation estimation method for 360 spherical panoramas by exploiting the geometric relationship between a 3D rotation and projected 2D rotations.
- we generate a large-scale labeled narrow field-of-view image dataset that can be used for supervised learning of roll (in-plane image rotation) angle estimation, by cropping 360 spherical panoramas using various view orientations.
- our trained convolutional neural network shows mean absolute error (MAE) about 1° in roll angle estimation for a 2D image.
- our 3D camera rotation estimation for 360 panoramas shows the state-of-the-art performance in both accuracy and robustness, and can handle a variety of scenes where the Manhattan or Atlanta world assumption is not satisfied.

2 Related Work

2.1 Upright adjustment and camera orientation estimation

Automatic correction of a tilted photograph has been researched for a decade, and the existing works are mainly based on vanishing structure analysis [2,8,10, 23]. Recently Lee et al. [18,19] proposed a set of criteria to straighten-up slanted photos while minimizing perceptual distortions.

Camera calibration estimates the relative orientation of a camera in the environment. With the Manhattan and Atlanta world assumptions [6, 26], scene structure analysis for camera calibration has been actively studied. Bazin et al. [4] proposed a method to estimate the globally optimal vanishing points in a Manhattan world. Joo et al. [12] presented a precise Atlanta direction estimation algorithm with inlier set maximization. Recently Zhai et al. [30] proposed a horizon and vanishing point detection algorithm without assuming the Manhattan or Atlanta world.

2.2 Upright adjustment of 360 spherical panoramas

Although omnidirectional vision has been actively researched in computer vision and robotics [5, 15, 17, 25], estimation of 3D camera rotation for 360 panoramas, i.e., upright adjustment of a single 360 panorama image, has not been much studied yet. Bazin et al. [3] presented a top-down approach for rotation estimation of an omnidirectional camera based on vanishing point detection with the Manhattan world assumption. Jung et al. [14] proposed vertical/horizontal line clustering and iterative optimization to robustly estimate 360 camera rotation under the Atlanta world assumption.

These methods [3, 14] assume the environment as a Manhattan or Atlanta world, and may fail for panoramas which do not follow the assumption. In addition, as the methods depend on detecting and clustering straight line segments and vanishing points, lack of such features would incur inaccurate and unstable upright adjustment results. In this paper, we overcome these limitations by proposing a deep learning based approach that does not assume any specific scene structure.

2.3 CNN-based upright adjustment

With the success of deep learning on image analysis and understanding, there have been a few attempts to apply deep learning for image orientation estimation. Fischer et al. [9] trained a CNN with an artificially rotated image dataset to predict in-plane image rotation (roll). Olmschenk et al. [24] proposed a network to estimate pitch and roll at the same time for a given image. Recently Joshi and Guerzhoy [13] trained a CNN classifier to distinguish the orientation of an image in four directions $\{0°, 90°, 180°, 270°\}$.

All these existing methods focus on ordinary images with conventional field-of-views, and a deep learning based approach has not been exploited yet for upright adjustment of 360 spherical panoramas.

3 3D Rotation Estimation of 360 Spherical Panoramas

A 360 spherical panorama captured with a rotated camera shows wavy and distorted appearance when it is represented using equirectangular projection. A narrow field-of-view projection of the panorama shows a tilted image, as the 3D rotated world is projected onto the 2D image plane. In this section, we formulate the relationship between a 3D camera rotation and a projected 2D rotation, and propose a novel framework to compute the 3D rotation using multiple samples of 2D rotation estimates.

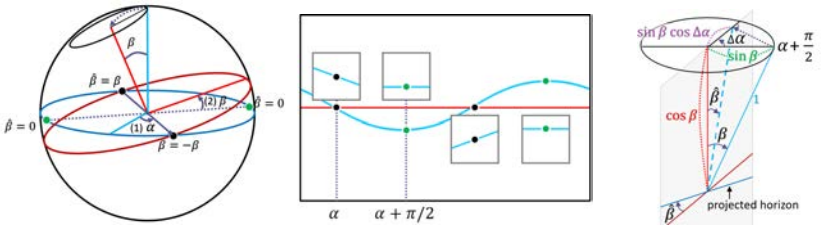
3.1 Rotation of 360 spherical panorama

Upright adjustment of a 360 spherical panorama can be considered as 3D rotation estimation for a 360 camera from the input panorama. Once we have estimated the camera rotation, the given panorama can easily be made upright by applying the inverse rotation. Since a 3D rotation can be represented by various conventions, we first clarify the notation before describing the relationship between 2D and 3D rotations.

A 3D rotation that should be estimated for upright adjustment of 360 panoramas has only two degree-of-freedoms (DOFs) as it can be modeled as a position change of the north pole on the globe [14], as illustrated in Fig. 2a. In the previous work [14], a 3D rotation is represented with two rotation angles, tilt and roll. For better understanding, in this paper, we use a different but equivalent definition of a 3D *intrinsic* rotation of a 360 camera using *yaw* α and *roll* β. Intrinsic rotation means that each element rotation (yaw and roll) occurs with the intrinsic vertical axis attached to the camera itself. As shown in Fig. 2a, an arbitrary camera rotation (α, β) is performed in two steps. The viewing direction of the camera is first rotated by α around the vertical axis, and then the camera with the vertical axis is rotated by β around the rotated viewing direction.

As shown in Fig. 2b, an equirectangular panorama captured by a rotated 360 camera shows a wavy horizon along the horizontal centerline of the image. If we crop the panorama into perspective images with narrow field-of-views, which we call *sampled images*, this wavy horizon would be shown as tilted (slanted horizon) or nodded (higher or lower eye level) in accordance with the viewing directions. Intuitively, the horizon is most slanted when the viewing direction agrees with the yaw α of the 3D rotation, while the horizon becomes flat for the viewing direction orthogonal to the yaw.

With this observation, we can develop an algorithm that estimates a 3D rotation (α, β) for a given distorted (rotated) 360 spherical panorama by analyzing the rotations of multiple sampled images. To this end, we formulate the relationship between the 3D rotation of a 360 camera and the 2D rotations of sampled images in the following section.



(a) 3D rotation of camera (b) spherical panorama and projected images (c) projection of 3D rotation

Fig. 2. Relation between 3D and 2D rotations. (a) 3D rotation (α, β) of the camera and the captured 360 panorama image. (b) Original ground plane (blue) becomes a wavy horizon when projected along the horizontal centerline of the rotated 360 camera (red). Perspective projections of the panorama with a narrow field-of-view produce tilted or nodded images according to the viewing directions (small rectangles). (c) The angle between the original upright direction and a rotated direction (roll of a 2D image) becomes smaller as it is projected onto a rotated image plane.

3.2 3D rotation estimation from multiple 2D sampled images

The geometric derivation between 2D and 3D rotations in a spherical panorama can be explained with Fig. 2. The blue vertical line in Fig. 2a represents the vertical axis perpendicular to the ground. As shown in Fig. 2c, when the vertical axis is projected to a rotated image plane (gray), the angle $\hat{\beta}$ between the vertical axis of the camera (solid red line) and the projected vertical axis (dashed blue line) can be derived using a trigonometric relationship.

We call $\hat{\beta}$ as the *projected roll* because the angle is identical to the in-plane rotation angle of the projected horizon. When the view direction is rotated with an angle $\Delta\alpha$ from the yaw α, the projection image plane is also rotated away from $\alpha + \pi/2$ by the amount of $\Delta\alpha$. We can then derive the following function f that relates the rotation angle $\Delta\alpha$ with the projected roll $\hat{\beta}$:

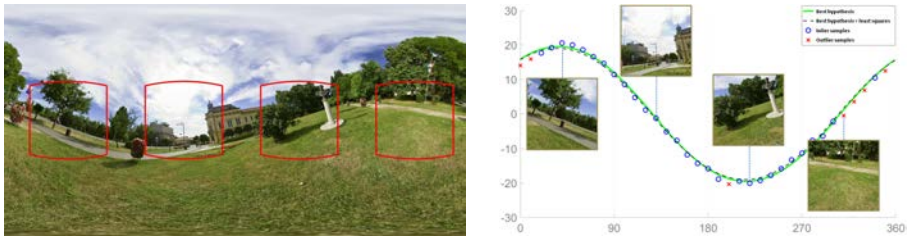
$$\tan\hat{\beta} = \frac{\sin\beta\cos(\Delta\alpha)}{\cos\beta}$$
$$\hat{\beta} = f(\Delta\alpha;\beta) = \tan^{-1}\big(\tan\beta\cos(\Delta\alpha)\big),$$

where the function $f(\Delta\alpha;\beta)$ is a periodic function that has the maximum β and minimum $-\beta$ at $\Delta\alpha = 0$ and $\Delta\alpha = \pi$, respectively.

To measure the yaw angle α, we introduce an auxiliary variable $\hat{\alpha}$, where $\hat{\alpha} = 0$ at the leftmost of the panorama image, and define $\Delta\alpha$ as the difference between $\hat{\alpha}$ and the yaw angle α, i.e., $\Delta\alpha = \hat{\alpha} - \alpha$. Then the function f can be represented as follows:

$$\hat{\beta} = f(\hat{\alpha};\alpha,\beta) = \tan^{-1}\big(\tan\beta\cos(\hat{\alpha}-\alpha)\big).$$

If we know two distinct points $(\hat{\alpha}_1,\hat{\beta}_1)$, $(\hat{\alpha}_2,\hat{\beta}_2)$ on $f(\hat{\alpha};\alpha,\beta)$, we can determine a unique solution for α and β, i.e., 3D rotation of the camera. However,



(a) input spherical panorama (b) estimated samples and fitted curve

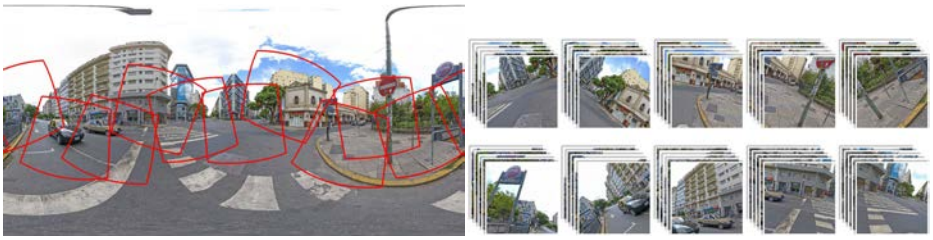
Fig. 3. Estimation of 3D rotation function $f(\hat{\alpha}; \alpha, \beta)$. (a) input panorama image. (b) sample estimates (blue dots) and some corresponding images with the best hypothesis (black dashed curve) and the least squares solution (green curve).

for more robust estimation, we determine a 3D rotation (α, β) using the roll estimation results from multiple sampled images. Fig. 3 illustrates the pipeline of our framework and the detailed steps are in the following.

1. Sample N narrow field-of-view images $\{I_i\}_{i=1}^N$ around the horizontal centerline of a given spherical panorama using uniformly distributed viewing angles $\{\hat{\alpha}_i\}_{i=1}^N$.
2. Estimate the projected roll $\{\hat{\beta}_i\}_{i=1}^N$ of the sample images using our *roll estimation network*, which will be described in the following section. Now we have multiple yaw/roll pairs $\{(\hat{\alpha}_i, \hat{\beta}_i)\}_{i=1}^N$.
3. Use exhaustive search to find the best hypothesis (α', β'), which maximizes the number of inlier samples among the multiple pairs.
4. Determine the function parameters (α, β) using the least-squares fitting of inlier samples.

Through all experiments in this paper, we used 36 uniformly sampled images (i.e., $N = 36$) with 60° for the field-of-view. As the number of samples are only 36, we use an exhaustive search to find the best hypothesis for outlier rejection. If a larger number of samples are used for better accuracy, RANSAC could be an option, instead of the exhaustive search, for avoiding too much increase of the computation time.

Even though we use outlier rejection and least-squares fitting with inlier samples for robust estimation, the performance of our algorithm would highly depend on the accuracy and reliability of 2D roll estimation of sampled images. As described in Section 2, previous rotation estimation methods for 2D images mainly utilize hand-crafted features, such as lines and vanishing points, with specific assumptions on the scene structure, i.e., the Manhattan or Atlanta world. To overcome the limitations, we present a CNN-based approach that can precisely estimate the rotation of a 2D image.



(a) input panorama (b) projected image dataset

Fig. 4. Large-scale dataset generation with various perspective projections.

4 CNN-based 2D Roll Estimation

In this section, we first present a massive 2D image dataset, where individual images are labeled with how much they have been rotated. The dataset is generated by cropping 360 spherical panoramas into perspective narrow field-of-view images. Our CNN is trained with the dataset to estimate the projected roll angle $\hat{\beta}$ of a given sampled image.

4.1 Rotated image dataset generation using 360 panoramas

Recent tremendous success of image understanding using deep learning is built upon large-scale labeled datasets [7, 21]. Similarly we need a large-scale dataset for training our roll estimation network. However, it would be time-consuming to label the rotation angles for many captured images, and applying artificial rotations to upright images [9] may suffer from information losses due to the cropping after image rotation. To annotate the ground truth rotation angles, Olmsschenk et al. [24] used the camera poses of a RGB-D stream estimated by 3D reconstruction, and additionally utilized the onboard accelerometer of a RGB-D sensor to extend the dataset. However, as the RGB-D dataset only consists of images captured from indoor environments, the dataset is not suitable for 2D images sampled from 360 spherical panoramas that may come from a variety of scenes.

In this paper, we generate a novel large-scale dataset for 2D image roll estimation using an existing 360 spherical panorama dataset. A 360 spherical panorama has 360° horizontal and 180° vertical field-of-views. So even when we sample narrow field-of-view images multiple times, resulting images are seldom much overlapped with each other. In addition, as a 360 panorama contains information from all view directions, no cropping is needed for a rotated image.

We use SUN360 panorama dataset [29]. The dataset consists of 30K high-resolution 360 spherical panoramas from a large variety of scene categories. Since the dataset contains non-upright panoramas, we manually correct the 3D rotations for randomly chosen 2000 panoramas, where the selection covers various scene categories, both indoor and outdoor environments. Then we generate a

labeled image dataset by projecting the panoramas onto intentionally rotated viewing angles and image frames as follows.

For the purpose of uniform sampling, we first divided the 360° yaw angle into 10 uniform intervals. Then we randomly sampled a yaw value from each interval, and for each yaw value, we sampled 30 images with random rotations, where the angle ranges are -20°$\leq$ pitch $\leq$20° and -40° $\leq$ roll $\leq$ 40°. We used 45° and 60° for the field-of-views to obtain perspective diversity of feature learning. Consequently, 600 narrow field-of-view images were sampled from each 360 spherical panorama with random 3D rotations. The ground truth label for a sampled image becomes the roll angle used for the sampling. As a result, we generated 1.2M randomly sampled images labeled with rotation angles in total.

We randomly chose 100 out of 2000 panoramas for a validation set and sampled training and validation images from the disjoint sets of panoramas. We used 1140K training images and 60K validation images for training our roll estimation network.

4.2 Network structure and training details

We choose Deep Residual Network (ResNet) [11] as our base network architecture, and modify the last layer to produce a single real value that represents the rotation angle of a given image, instead of the original 1000 class probabilities. To reduce the burden for exploring semantic features from scratch, we use a ResNet34 model pretrained for ImageNet-1K classification task. The network parameters are trained using stochastic gradient descent algorithm [16] for 142K iterations (16 epochs) with mini-batch size of 128. We set the initial learning rate to 5e-4, and decrease the learning rate by 0.1 times when the validation error does not drop for two epochs. The final learning rate was 5e-7 after training. We use L1 loss to minimize the absolute error between the prediction and ground truth angles.

5 Experimental Results

In this section, we present various experimental results that show our method achieves the state-of-the-art performance on upright adjustment of 360 spherical panoramas. We first show some 2D image upright adjustment results to demonstrate the performance of our trained network. We then compare our 360 upright adjustment results qualitatively and quantitatively on a test dataset [14]. In addition, with a newly constructed test dataset, we show that our method produces robust results for input panoramas violating the Manhattan and Atlanta world assumptions.

Implementation details For outlier rejection, we set the inlier threshold to 2°, which is the distance between a sample and the hypothesis function. We used the Levenberg–Marquardt algorithm [20, 22] for the non-linear least squares fitting of function $f(\hat{\alpha}; \alpha, \beta)$ to obtain the final 3D rotation. We tested our algorithm on

(a) (b) (c) (d)

Fig. 5. Correction of slanted 2D images (top: input images, bottom: results). Our network precisely predicts the rotation angles of images sampled from spherical panoramas (a, b) and works even for natural images that were captured by ordinary perspective cameras (c, d). We selected the test natural images in (c, d) from DIV2K dataset [1].

an Intel i7-6700K 4.00GHz CPU, 32GB RAM and NVIDIA GeForce GTX 1080. For an input image of size 9104×4552, our algorithm takes about 0.6 seconds on average. Jung et al. [14] consumes less than one second, and Bazin et al. [3] takes about five seconds with a MATLAB implementation.

5.1 Rotation estimation for 2D images

After network training, the average absolute error on the validation set is 1.08°. We also define the in-k accuracy as the percentage of samples whose error is less then k°, which is similar to top-k accuracy in classification tasks. Our network shows 67.5%, 94.9%, and 97.9% on the validation set for in-1, 3, and 5 accuracies, respectively.

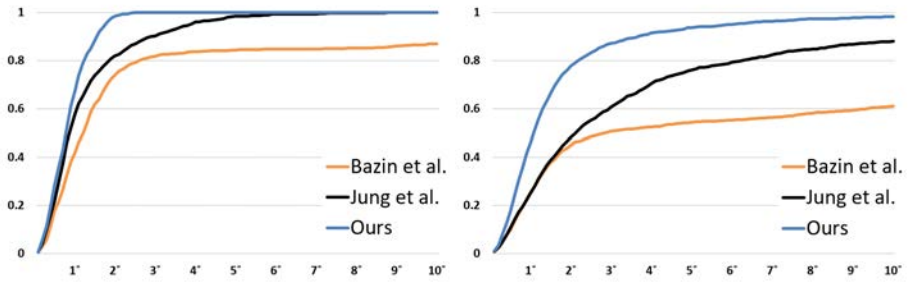
Although our CNN is trained on images sampled from 360 spherical panoramas, it shows satisfying prediction results on ordinary perspective images. Fig. 5 shows 2D roll correction results on different images, where the correction works well even when the image is cluttered and does not have enough straight lines.

5.2 New test datasets

To evaluate the performance of our framework, we use two different kinds of test datasets. Jung et al. [14] constructed a test dataset by applying various rotations to 360 panoramas that have been carefully taken with no camera rotations. It consists of 840 test images generated from 14 panoramas. As Jung et al.'s work [14] assumes the Atlanta world for the scene, most of test images in the dataset follow the assumption well. In contrast, our method does not assume any specific scene structures, and the dataset would not be enough to thoroughly evaluate the performance in various situations.

Table 1. Numerical comparison with existing 360 panorama upright adjustment methods on Jung et al.'s test dataset [14] and our newly generated test datasets with four categories. We measured the average absolute angular error between the predicted rotation and the ground truth label.

	Jung et al.'s dataset [14]	our test datasets				
		street	indoor	park	mountain	total
Bazin et al. [3]	3.66°	3.76°	4.13°	13.29°	17.67°	9.71°
Jung et al. [14]	1.12°	1.78°	2.21°	3.75°	12.98°	5.18°
ours (best hypothesis)	1.02°	1.14°	1.09°	1.58°	3.85°	1.92°
ours (best hypothesis+lsq.)	**0.84°**	0.96°	0.81°	1.50°	3.80°	**1.77°**



(a) Jung et al.'s test dataset [14] (b) Our test dataset

Fig. 6. Cumulative histograms of the errors of different methods on Jung et al.'s test dataset [14] and our dataset. Our method outperforms previous methods on both datasets.

To this end, we built a new test dataset by collecting four scene categories from SUN360 dataset [29]: street, indoor, park, and mountain. Intuitively, street and indoor scenes usually follow the Manhattan or Atlanta world assumption, and contain lots of structural features useful for rotation estimation. However, park scenes consist of mainly natural structures, such as trees, which may yield many false vanishing structures. Lastly, mountain scenes contain images captured at mountain tops, which rarely include straight lines and vanishing structures and hardly follow the Atlanta world assumption. We manually corrected the randomly chosen 50 panoramas from each category in SUN360 dataset, 200 panoramas in total. Then we randomly rotated each panorama image with 10 different rotations within a range of $\pm 30°$, resulting in 2000 labeled test cases in total.

5.3 Comparisons and quantitative evaluation

We quantitatively evaluated our method on two test datasets: Jung et al.'s and ours. We measured the angular error between the predicted vertical direction

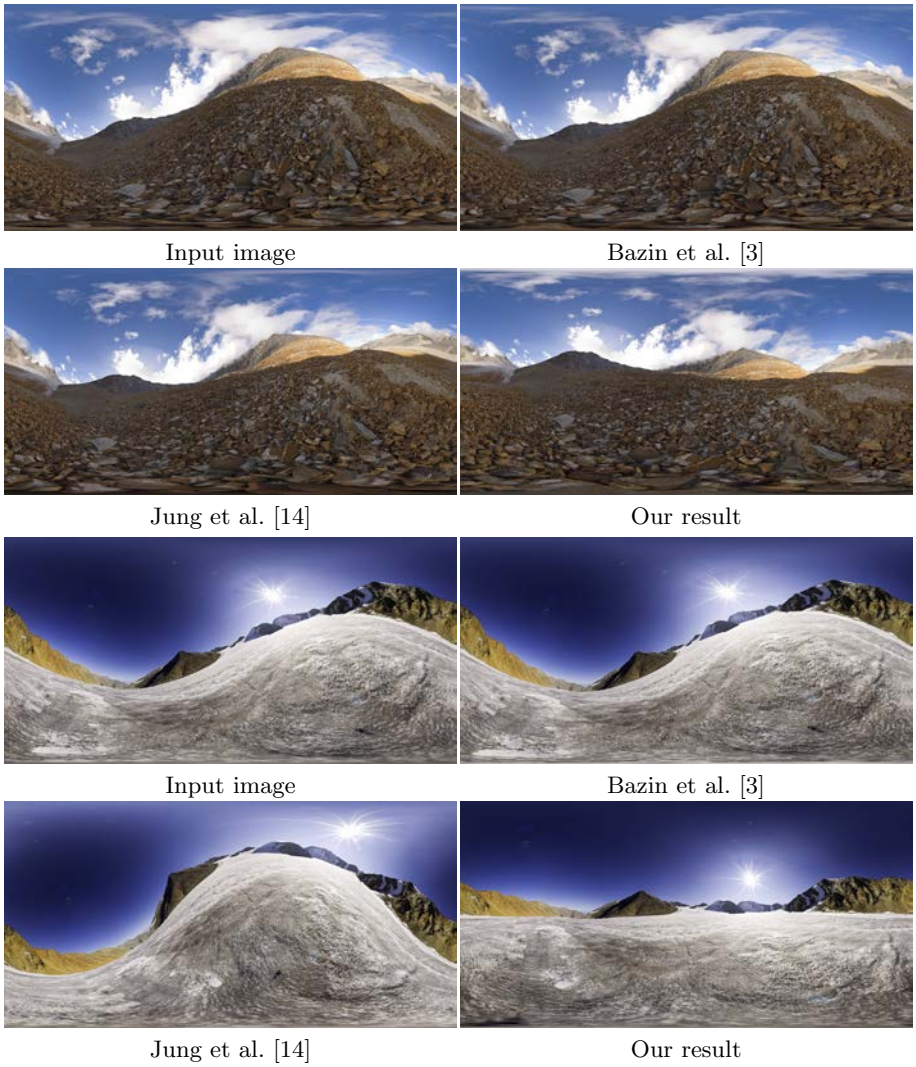


Fig. 7. Visual comparisons of the upright adjustment results on 360 panorama images from mountain scenes that do not contain strong structural features.

of the rotated camera and the ground truth label. Table 1 shows average absolute angular errors on both datasets. Fig. 6 shows the cumulative histograms of prediction errors comparing with existing methods [3, 14]. The table and figure show the superior performance of our method in both accuracy and robustness. For Jung et al.'s test dataset [14] that follows the Atlanta world assumption, the average error of our method reaches the state-of-the-art performance and more than 99% of test panoramas are corrected with the errors less than 2.2°. Interestingly, our errors of upright adjustment of 360 panoramas are lower than

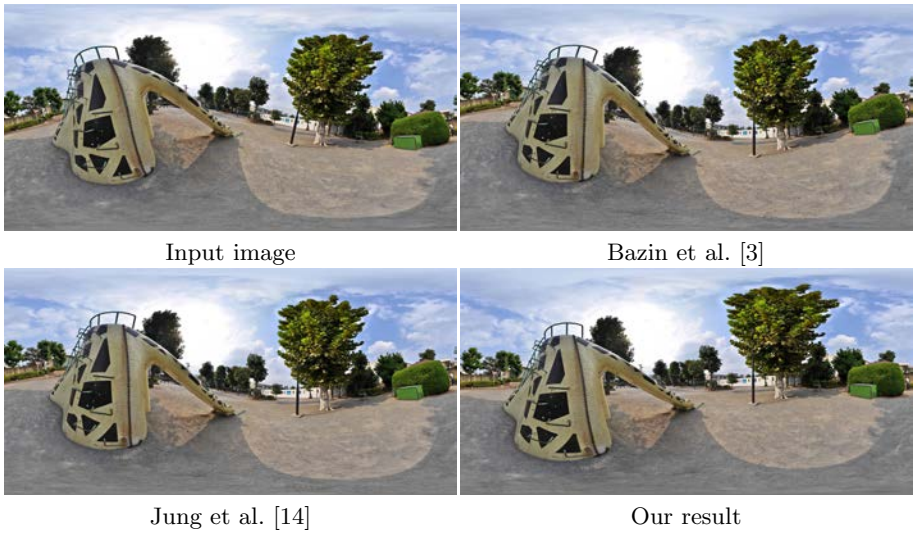


Fig. 8. Comparison on a scene not following the Atlanta world assumption (e.g., the slide in the playground).

those of 2D roll estimation network. The reason would be our framework uses outlier rejection and least-squares fitting to robustly find the best 3D rotation despite possible mistakes in roll predictions.

For our newly generated test dataset, our method outperforms previous methods by a large margin. For easy scenes, i.e., street and indoor, all methods show similar performances compared to the previous test set of Jung et al. [14]. However, as can be expected, for the test images from the hard case of mountain scenes, previous methods easily fail as the scenes rarely contain strong structural features such as straight lines. In contrast, our method uses a roll estimation network trained with a large-scale dataset, and robustly estimates semantically correct upright directions regardless of the existences of structural features. Fig. 7 shows visual comparisons of the upright adjustment results for some mountain scenes.

In addition, as shown in Fig. 8, our method works well on a scene that does not follow the Atlanta world assumption. Various examples of our algorithm in Fig. 9 and the supplementary material show that our method can robustly and accurately handle spherical panoramas from a variety of scenes.

6 Conclusions

Upright adjustment of a mis-oriented 360 spherical panorama improves the perceptual quality by correcting the wavy horizon and distorted objects. In this paper, we presented a novel upright adjustment framework for 360 spherical panoramas which indirectly uses a convolutional neural network. By exploiting

Input image Our upright adjustment result

Fig. 9. Additional results of our method for a variety of scenes.

the relationship between a 3D camera rotation and projected 2D rotations of multiple sampled images, our method accurately estimates the 3D camera rotation relative to the scene from a given single 360 panorama. Differently from existing methods based on straight lines and vanishing points, our method uses a CNN for 2D roll estimation by training it using a massively generated 2D

image dataset, enabling the network to learn semantic upright directions (e.g., standing people). Extensive experiments on test datasets demonstrate that our method provides highly accurate and robust results for upright adjustment of 360 panoramas even when existing methods could fail.

There remains a limitation of our method. A uniform sampling along the horizontal centerline of the image could crop only the ground or sky when the camera has been severely rotated, introducing many outliers in roll estimation. Adaptive sampling based on saliency or objectness would improve our method.

Directly applying CNN to a 360 spherical panorama has not been straightforward due to severe distortions around the north and south poles of the panorama and the lack of a large-scale labeled dataset. In this paper, to avoid the difficulties, we took an indirect approach that uses CNN to estimate rotations of 2D sampled images and computes the 3D rotation from the estimated 2D rotations. Recently Su and Grauman [28] proposed a novel approach to extract CNN features directly from a spherical panorama image by adjusting the shapes of pre-trained convolution kernels according to the spherical distortions. Directly predicting the 3D camera rotation using CNN from a given panorama would be interesting future work.

Acknowledgements

This work was supported by the Ministry of Science and ICT, Korea, through IITP grant (IITP-2015-0-00174), NRF grant (NRF-2017M3C4A7066317), and Giga Korea grant (GK18P0300).

References

1. Agustsson, E., Timofte, R.: Ntire 2017 challenge on single image super-resolution: Dataset and study. In: Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops. vol. 3, p. 2 (2017)
2. Antone, M.E., Teller, S.: Automatic recovery of relative camera rotations for urban scenes. In: Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR). vol. 2, pp. 282–289. IEEE (2000)
3. Bazin, J.C., Demonceaux, C., Vasseur, P., Kweon, I.: Rotation estimation and vanishing point extraction by omnidirectional vision in urban environment. The International Journal of Robotics Research **31**(1), 63–81 (2012)
4. Bazin, J.C., Seo, Y., Demonceaux, C., Vasseur, P., Ikeuchi, K., Kweon, I., Pollefeys, M.: Globally optimal line clustering and vanishing point estimation in manhattan world. In: Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 638–645. IEEE (2012)
5. Bosse, M., Rikoski, R.J., Leonard, J.J., Teller, S.J.: Vanishing points and 3D lines from omnidirectional video. In: Proc. IEEE International Conference on Image Processing (2002)
6. Coughlan, J.M., Yuille, A.L.: The manhattan world assumption: Regularities in scene statistics which enable bayesian inference. In: Advances in Neural Information Processing Systems. pp. 845–851 (2001)

7. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: ImageNet: A Large-Scale Hierarchical Image Database. In: Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 248–255 (2009)
8. Denis, P., Elder, J.H., Estrada, F.J.: Efficient edge-based methods for estimating manhattan frames in urban imagery. In: Proc. European Conference on Computer Vision (ECCV). pp. 197–210. Springer (2008)
9. Fischer, P., Dosovitskiy, A., Brox, T.: Image orientation estimation with convolutional networks. In: Proc. German Conference on Pattern Recognition. pp. 368–378. Springer (2015)
10. Gallagher, A.C.: Using vanishing points to correct camera rotation in images. In: Proc. Canadian Conference on Computer and Robot Vision. pp. 460–467. IEEE (2005)
11. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 770–778 (2016)
12. Joo., K., Oh., T.H., Kwon., I.S., Bazin., J.: Globally optimal inlier set maximization for atlanta frame estimation. In: Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 2408–2415. IEEE (2018)
13. Joshi, U., Guerzhoy, M.: Automatic photo orientation detection with convolutional neural networks. In: Conference on Computer and Robot Vision (CRV). pp. 103–108. IEEE (2017)
14. Jung, J., Kim, B., Lee, J.Y., Kim, B., Lee, S.: Robust upright adjustment of 360 spherical panoramas. The Visual Computer **33**(6-8), 737–747 (2017)
15. Kamali, M., Banno, A., Bazin, J.C., Kweon, I.S., Ikeuchi, K.: Stabilizing omnidirectional videos using 3d structure and spherical image warping. In: Proc. IAPR Conference on Machine Vision Applications. pp. 177–180 (2011)
16. Kiefer, J., Wolfowitz, J., et al.: Stochastic estimation of the maximum of a regression function. The Annals of Mathematical Statistics **23**(3), 462–466 (1952)
17. Kopf, J.: 360° video stabilization. ACM Transactions on Graphics **35**(6), 195 (2016)
18. Lee, H., Shechtman, E., Wang, J., Lee, S.: Automatic upright adjustment of photographs. In: Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 877–884. IEEE (2012)
19. Lee, H., Shechtman, E., Wang, J., Lee, S.: Automatic upright adjustment of photographs with robust camera calibration. IEEE Transactions on Pattern Analysis and Machine Intelligence **36**(5), 833–844 (2014)
20. Levenberg, K.: A method for the solution of certain non-linear problems in least squares. Quarterly of applied mathematics **2**(2), 164–168 (1944)
21. 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. European Conference on Computer Vision (ECCV). pp. 740–755. Springer (2014)
22. Marquardt, D.W.: An algorithm for least-squares estimation of nonlinear parameters. Journal of the society for Industrial and Applied Mathematics **11**(2), 431–441 (1963)
23. Martins, A.T., Aguiar, P.M.Q., Figueiredo, M.A.T.: Orientation in manhattan: equiprojective classes and sequential estimation. IEEE Transactions on Pattern Analysis and Machine Intelligence **27**(5), 822–827 (May 2005)
24. Olmschenk, G., Tang, H., Zhu, Z.: Pitch and roll camera orientation from a single 2D image using convolutional neural networks. In: Conference on Computer and Robot Vision (CRV). pp. 261–268. IEEE (2017)

25. Scaramuzza, D.: Omnidirectional vision: from calibration to robot motion estimation. Ph.D. thesis, ETH Zurich (2008)
26. Schindler, G., Dellaert, F.: Atlanta world: An expectation maximization framework for simultaneous low-level edge grouping and camera calibration in complex man-made environments. In: Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 203–209 (2004)
27. Song, S., Lichtenberg, S.P., Xiao, J.: Sun rgb-d: A rgb-d scene understanding benchmark suite. In: Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 567–576 (2015)
28. Su, Y.C., Grauman, K.: Learning spherical convolution for fast features from 360 imagery. In: Advances in Neural Information Processing Systems. pp. 529–539 (2017)
29. Xiao, J., Ehinger, K.A., Oliva, A., Torralba, A.: Recognizing scene viewpoint using panoramic place representation. In: Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 2695–2702. IEEE (2012)
30. Zhai, M., Workman, S., Jacobs, N.: Detecting vanishing points using global image context in a non-manhattan world. In: Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 5657–5665 (2016)