# Recognizing Artwork in Panoramic Images Using Optimized ASIFT and Cubic Projection

Dayou Jiang and Jongweon Kim

*Abstract*—**Few studies have been published on recognizing objects in panoramic images. To prevent copyright infringement related to artwork in 360° images, this paper proposes an efficient method for recognizing such artwork. First, we employ an improved cubic projection approach to transform distorted panoramas. Next, we use an optimized affine invariant feature transform (ASIFT) algorithm to extract local features of the transformed images. Finally, we employ point feature matching based on a one-to-one mapping constraint. We investigate the method's overall performance on a panorama dataset and compare the results with those for other popular local feature extraction methods as well as the original panorama image. The experimental results show that the proposed method is both faster and can improve accuracy by around 30% for highly-distorted panoramas.**

*Index Terms*—**Panoramic image, artwork, recognition, ASIFT, cubic projection.**

## I. Introduction

Recently, owing to the continuing improvement of panoramic photography and the advent of easy-to-use 360° cameras, such as the Samsung Gear 360 and LG G5 360 CAM, 360° videos and images have become increasingly popular. Users can easily capture the view in all directions simultaneously in a single 360° image, and feel a strong sense of immersion when viewing the image [1]. Given these developments, the potential risk of copyrighted artworks being photographed without permission is greater than ever before, hence the infringement of artwork copyrights using 360° images is a hot topic. We therefore need to find a way to detect and recognize unauthorized artworks in panoramic images.

Although many image recognition studies have been published, relatively few studies have been conducted on 360° images, never mind detecting artworks in such images. Nonetheless, methods that have been found useful for image recognition can sometimes be helpful for recognizing objects in panoramas as well.

Since the 1990s, image content-based methods have been a popular way to solve image recognition problems. These methods describe the image's content by extracting low-level visual features, and can perform well in terms of both accuracy and speed [2]. Various local feature extraction algorithms have been proposed in recent years. Of these, the most typical and widely-used method is scale-invariant feature transform (SIFT) [3]. Other algorithms have also been found to perform quite well, such as the speeded up robust features (SURF) [4], affine SIFT (ASIFT) [5], oriented FAST and rotated BRIEF (ORB) [6], binary robust invariant scalable keypoints (BRISK) [7], and fast retina keypoint (FREAK) [8] algorithms. Currently, deep learning methods, such as AlexNet [9], ZFNet [10], GoogLeNet [11], or ResNet [12], are employed to learn suitable local feature vectors and obtain classification models for large dataset tasks such as the ImageNet Large Scale Visual Recognition Challenge.

With respect to object recognition in panoramic images, Xiao [13] introduced the problem of scene viewpoint recognition and also studied the canonical view biases exhibited by people photographing particular locations. Yang [14] addressed the problem of recognizing the room structure from a 360° cylindrical panorama by transforming the original panorama into sub-images projected from four different perspectives. An algorithm has also been proposed that can detect and recognize road lane markings from panoramic images [15].

Zhang [16] advocated the use of 360° full-view panoramas for understanding scenes and proposed a three-dimensional (3D) whole-room context model. A region-based convolutional neutral network (R-CNN) has been trained and then tested on a set of indoor panoramas [17], and a novel panorama-to-panorama matching process has been developed [18] that involves either aggregating the features of a group of individual images or explicitly constructing a larger panorama. In addition, an improved ASIFT algorithm for matching indoor panoramas has been analyzed and compared with algorithms such as SIFT, SURF, and ASIFT [19].

Few studies have been conducted on recognizing artwork in panoramic images. However, an artwork identification approach has been hypothesized [20] that transforms the 360° image into a 3D sphere and surrounds it with a polyhedron [20]. The results of [20] show that this method can significantly increase identification precision for artwork, displayed on a monitor, that would otherwise be severely distorted. In addition, the use of different local features for feature matching was analyzed. Although employing more polyhedra can significantly improve performance, more time would be required for feature detection and matching, and the panorama's visual appearance would also worsen.

The aim of this study is to develop an efficient method for recognizing artworks in 360° images. In attempting to solve this problem, we focus on two main issues: (1) achieving better performance by generating faster results, and (2) using a simple, feasible projection to ensure the transformed panoramas offer a good visual experience.

The remainder of the paper is organized as follows. Section

Manuscript received October 10, 2017; revised January 19, 2018.

Dayou Jiang is with Dept. of Copyright Protection, Sangmyung University, Seoul, Korea (e-mail: dyjiang@cclabs.kr).

Jongweon Kim is with Dept. of Electronics Engineering, Sangmyung University, Seoul, Korea (corresponding author; e-mail: jwkim@smu.ac.kr).

doi: 10.18178/ijmlc.2018.8.1.663

II discusses panorama transformation, feature extraction, and feature matching methods. Section III demonstrates the performance of the proposed artwork recognition method via experiments on a dataset of panoramas involving artworks. Finally, Section IV presents our conclusions.

## II. Panorama Transformation and Optimized ASIFT

### A. *Transforming Panoramic Images*

Panoramas captured using 360 ° cameras are mainly in a equi-rectangular format with a 2:1 aspect ratio. Such projections are widely used to map 3D scenes onto two-dimensional (2D) planes; however they cause serious distortions, particularly near the two poles, and are thus a poor approach to use for artwork recognition.

Recently, several alternative projections have been investigated for 360 °images and videos. Kim [21] proposed a framework to automatically generate content-aware 2D videos using a normal view perspective from 360 ° videos, based on a Panini projection [22] model. A recent study has also considered Oculus 360 °video streaming using an offset cube map [23]. Following these investigations, Brown (a Google engineer) presented an equi-angular cube map method [24] that offers better results and uses resources more efficiently, aiming to resolve virtual reality video quality issues.

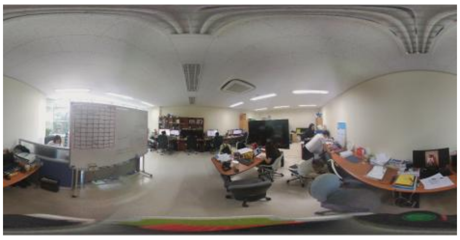
Fig. 1. A panoramic image.

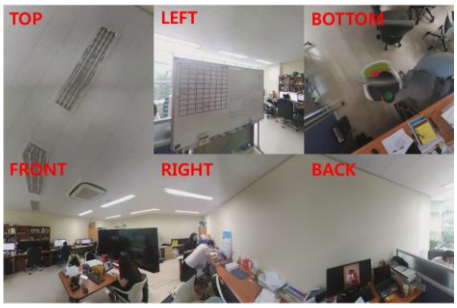

Fig. 2. Cube map corresponding to Fig. 1.

When considering possible panorama projections, a natural starting point are map projections such as the Mercator, Goode homolosine, Natural Earth, or cube map. However, most of these projections cannot be used for artwork recognition, owing to two reasons: transforming a panorama into a 3D sphere and then mapping it onto these projections is computationally costly, and despite considerably longer processing times, the result will still have unavoidable distortions.

One exception to these limitations is the cube map, which is easily calculated and has relatively less distortion. Therefore, in this study, we chose to begin with a cube map and attempt to improve it. A standard cube map converts a panorama into the six faces of a cube, where each face is a rectilinear image. Figure 1 shows an example of a low-resolution and low-distortion panorama, and Fig. 2 shows a standard cube map created from this panorama.

For our improved method, we required the transformed images to show the object of interest clearly, which can be achieved by adjusting the pitch and yaw angles and the field of view (FOV). Figure 3 (taken from [23]) illustrates the concepts of yaw, pitch, and camera roll, and Fig. 4 illustrates a cube map, highlighting the front and back faces. The *x*, *y*, and *z* directions represent the pitch, yaw, and roll of the view, respectively. The pitch and roll angles are varied depending on the image and screen to suit the situation, whereas the FOV is fixed at 90 °. The transformation matrix is given by

$$M_T = roty(yaw) * rotx(pitch) * rotz(roll)\,, \tag{1}$$

where *rotx*, *roty*, and *rotz* perform clockwise rotations around the *x*, *y*, and *z* axes, respectively, and *pitch*, *yaw*, and *roll* are the desired rotations around these axes. We constrain the pitch values to be 20 °–50 ° south for large monitors, and 10 °–30 °south for small monitors.

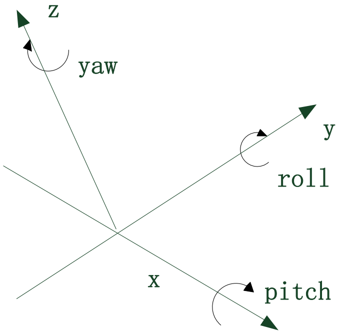

Fig. 3. Illustration of yaw, pitch, and camera roll.

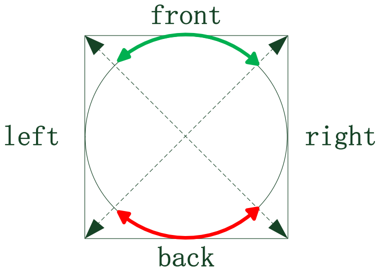

Fig. 4. Illustration of a standard cube map.

### B. *Optimized ASIFT Algorithm*

The original ASIFT algorithm involves the following steps.

1) Transforming the image by rotating it to change the directions of the latitude $\theta$ and longitude $\varphi$, according to the tilt *t*. Here, *t* is a T-subsampling in the height direction, as follows.

$$t = \left|1/\cos\theta\right| \tag{2}$$

$$\text{height}' = \left|\text{height}/t\right| \tag{3}$$

2) Rotating the image by changing the values of $\varphi$ and $\theta$ so that maximum objects of interest can be seen clearly in the resulting images.
3) Using the SIFT algorithm to detect features in all the simulated images.

Figure 5 (taken from [5]) shows the irregular sampling of

$\theta$ and $\varphi$ parameters used by ASIFT to detect features.

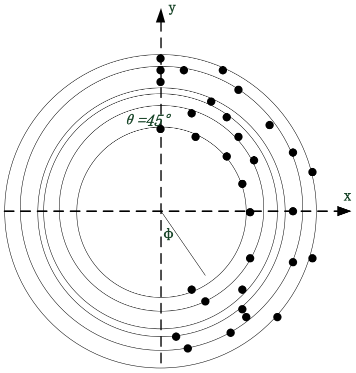

Fig. 5. Sampling of $\theta$ and $\varphi$ parameters.

In designing our optimized ASIFT algorithm, we first evaluated possible sampling steps using SIFT with different simulated tilts, eventually fixing $\Delta t$ experimentally as $\sqrt{2}$ . In contrast, we change the value of $\Delta \varphi$ based on the variation in $t$. For example, if the tilt is 2, then $t = \sqrt{2}$ and hence $\Delta \varphi$ = 45°, whereas if the tilt is 3, then $t = 2$ and $\Delta \varphi$ = 30°. Thus, setting the tilt to 2 means we generate four simulated images at 45° intervals, whereas setting the tilt to 3 produces six simulated images at 30°intervals.

Figure 6 shows some example images simulated using ASIFT. Navigating from left to right and top to bottom, these images represent transforms 1 ($t$ = 1, $\varphi$ = 0°), 2 ($t = \sqrt{2}$ , $\varphi$ = 0°), 3 ($t = \sqrt{2}$ , $\varphi$ = 45°), 4 ($t = \sqrt{2}$ , $\varphi$ = 90°), 5 ($t = \sqrt{2}$ , $\varphi$ = 135°), and 6 ($t$ = 2, $\varphi$ = 0°). As can be seen in Fig. 6, transforms 3 and 5 are larger than the others, implying that feature detection and extraction will take longer, and hence, finding a way to process these transforms more simply might be helpful. In addition, the rotation transformation affects the height direction comparatively less; thus, using fewer rotation transforms could reduce the running time and improve the relative performance as compared with using all the transforms.

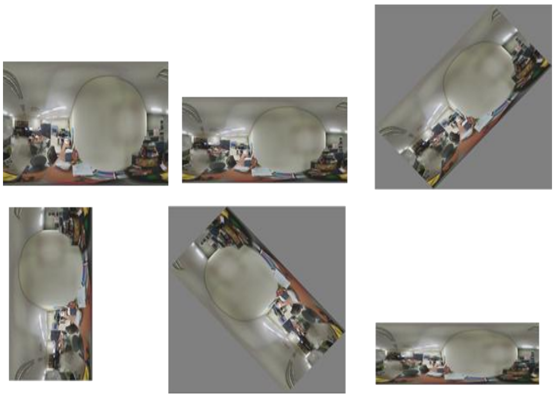

Fig. 6. Images simulated using ASIFT.

### C. One-to-One Mapping

In our experiments, we used the UBCMATCH feature matching algorithm [25], which finds the closest matching feature in one image for each corresponding feature in the other image. Although the resulting matches can be filtered for uniqueness using a threshold (the ratio of the distances between the best and second-best matching keypoints), errors can still exist. After several tests, we found that the erroneous matches were usually crowded around a point, and sometimes one point in the original artwork was matched to many points in the training image when using UBCMATCH with a high threshold (3.0, rather than the default 2.0). Therefore, one-to-one mapping was adopted to solve the issue of such erroneous one-to-many matches. In Fig. 7, one keypoint in the panorama has been matched to two different keypoints from the original artwork image, and the corresponding one-to-one mapping results are shown in Fig. 8. As compared with using UBCMATCH exclusively, one-to-one mapping has reduced the number of mismatched keypoints, clearly illustrating its benefits.

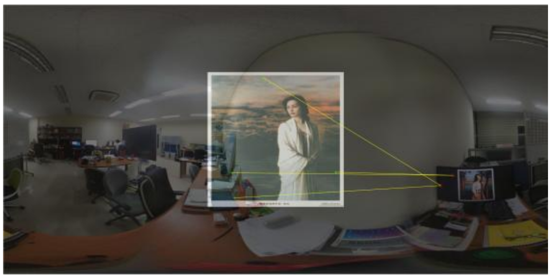

Fig. 7. Feature matches produced using UBCMATCH.

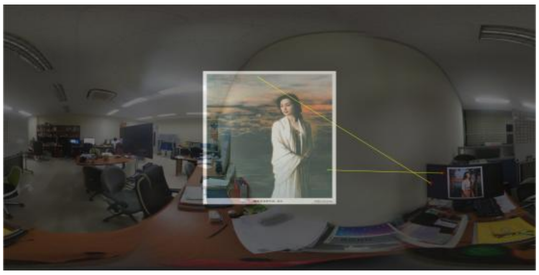

Fig. 8. Feature matches produced using one-to-one mapping.

## III. Simulation Results

For our experiments, we used a dataset of panoramic images comprising artworks, created using images of 50 famous artworks taken from Google, as shown in Fig. 9. The images were downloaded in JPG format and were of significantly different file sizes: the largest was 7.22 MB, while the smallest was 89.1 KB, and at least half were less than 1 MB. The panoramas were captured using an LG G5 360 CAM that has dual wide-angle cameras. The default panorama size was 5660 × 2830 at 72 dots per inch.

Fig. 9. Popular artworks used in the experiment.

To make the simulation more realistic, we captured three different distorted panoramas from three different positions for each artwork, which was displayed on two different screens, 32 and 79 inches in size. This resulted in a panorama artwork dataset comprising 300 images, evenly distributed among the six categories thus created. In panoramas with larger distortions, the artworks were significantly further from the equator.

The experiments were conducted using a computer with a Xeon 3.50 GHz CPU and 8 GB RAM, running Windows 7 Professional K (64-bit).

As using large images means large numbers of feature points can be detected and most of them cannot be matched correctly, reducing the size of the experimental images could not only reduce the computing time but also improve the percentage of correct matches. Therefore, the panorama images were resized to 800 ×1600, the artwork images to 400 ×400, and the generated transformed images were resized to 800 ×1200.

When extracting the features, the SIFT method uses default parameters; however, we employed a Parallel function with 4 local CPU pools for the ASIFT method to accelerate processing. Where the original ASIFT algorithm applies SIFT five times (i.e., uses transforms 1–5), our optimized ASIFT algorithm only applies it four times (i.e., uses transforms 1, 2, 4, and 6).

We used a UBCMATCH threshold value of 3.0 for feature matching as the default value of 2.0 generated too many erroneous matches for the more distorted panoramas. For example, as shown in Figs. 10 and 11, the number of matches was sometimes higher when comparing the panorama with an artwork other than the correct one, leading to false positives. To ensure the experiments were reliable, we conducted several test evaluations before selecting the parameters discussed above.

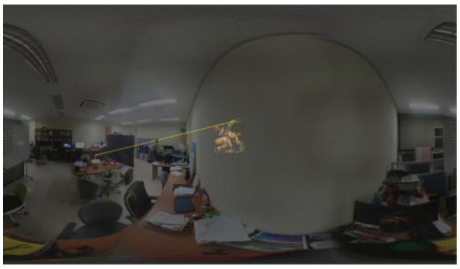

Fig. 10. Feature matching results when comparing the panorama with the correct artwork, with a UBCMATCH threshold of 2.0.

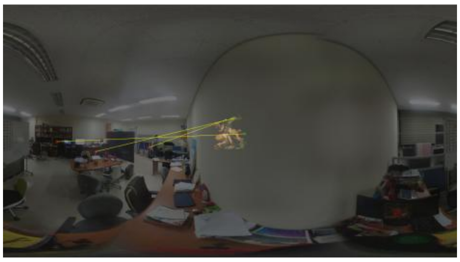

Fig. 11. Feature matching results when comparing the panorama with a different artwork, with a UBCMATCH threshold of 2.0.

We measured the performance of the proposed method using three types of comparison experiments. First, we constructed the following example to compare SIFT's recognition performance for images generated via cubic projection and original panoramas. Figs. 12–15 show feature-matching results for low-distortion images displayed on a 32-inch monitor.

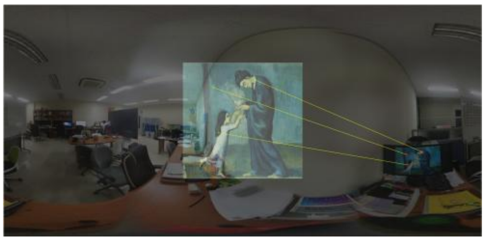

Fig. 12. Feature matching results when comparing the original panorama with the correct artwork.

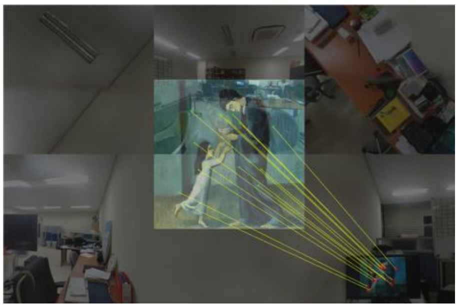

Fig. 13. Feature matching results when comparing the cubic projection with the correct artwork.

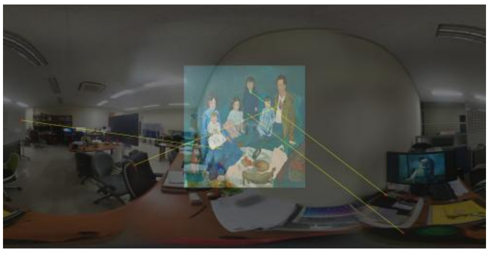

Fig. 14. Feature matching results when comparing the original panorama with a different artwork.

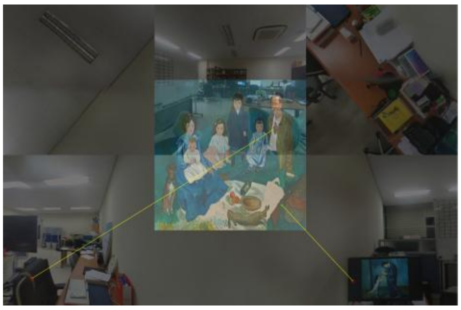

Fig. 15. Feature matching results when comparing the cubic projection with a different artwork.

Figures 12 and 14 show that using the original panorama yielded some true matches for the correct artwork together with some erroneous matches for different artworks. In contrast, Figs. 12 and 13 show that cubic projection yielded more true matches than the panorama for correct artwork, and Figs. 13 and 15 show that it yielded fewer erroneous matches than the panorama for a different artwork. Therefore, these results indicate that the proposed cube map transform is more suitable than using the panorama directly for recognizing artwork in panoramas.

Second, we compared the performance of SIFT and ASIFT. Figures 16–19 show feature matching results generated using ASIFT. As our previous experiments used SIFT, we only show the matching results for ASIFT with cubic projection images here. If we compare Figs. 16 and 17 (ASIFT) with Figs. 13 and 15 (SIFT), ASIFT may have yielded fewer matched points, but it outperformed SIFT in feature extraction and matching for more distorted images. In addition, if we compare ASIFT's performance with transforms two and five, we find that using more transforms actually makes ASIFT less effective; although Fig. 18 shows more correctly-matched points than Fig. 16, Fig. 19 also shows more erroneously-matched points than Fig. 17. However, further experiments using different distorted images will be needed to verify the ASIFT's performance with different numbers of affine transforms.

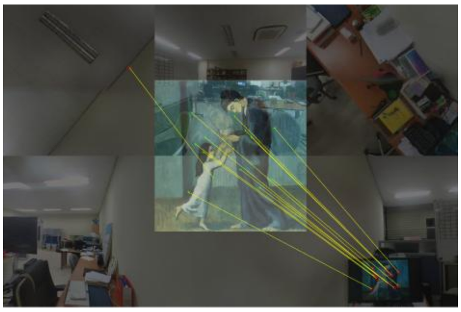

Fig. 16. Feature matching results when using ASIFT with two transforms (1, 2) for correct artwork.

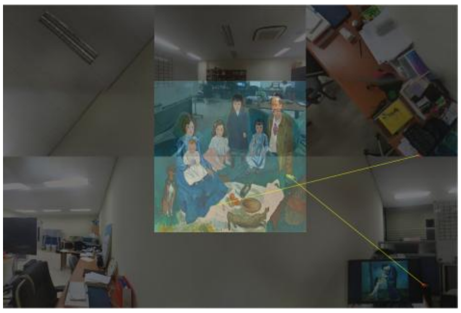

Fig. 17. Feature matching results when using ASIFT with two transforms (1, 2) for a different artwork.

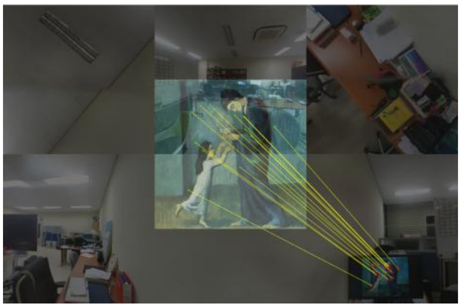

Fig. 18. Feature matching results when using ASIFT with five transforms (1–5) for correct artwork.

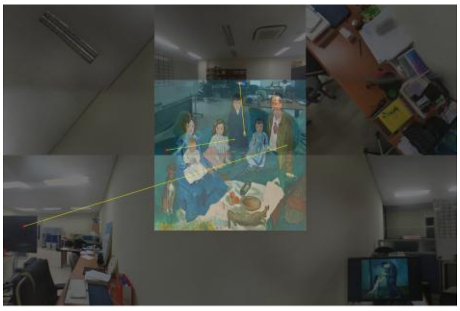

Fig. 19. Feature matching results when using ASIFT with five transforms (1–5) for a different artwork.

Even though we applied the Parallel function with four local pools to accelerate ASIFT's processing, its slowness remains a problem for real-time recognition. Therefore, we need to find a way to achieve satisfactory performance in less time. The following example is taken from an experiment where we compared the standard (transforms 1–5) and optimized (transforms 1, 2, 4, and 6) ASIFT algorithms. Here, highly-distorted images were displayed on a 79-inch monitor, and the results are shown in Figs. 20–23.

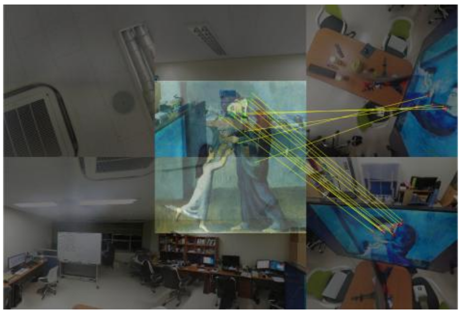

Fig. 20. Feature matching results when using standard ASIFT for correct artwork.

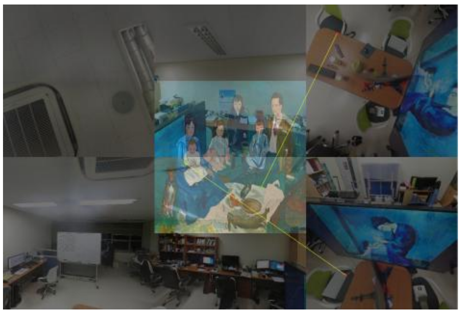

Fig. 21. Feature matching results when using standard ASIFT for a different artwork.

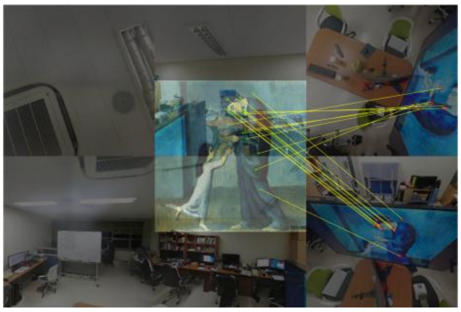

Fig. 22. Feature matching results when using optimized ASIFT for correct artwork.

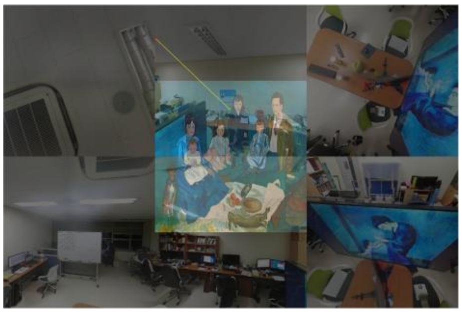

Fig. 23. Feature matching results when using optimized ASIFT for a different artwork.

Finally, we measured the proposed method's performance using the entire artwork panorama dataset. Here, we computed the numbers of matched points, then sorted them in descending order. As the parameters used were carefully selected after testing, the number of erroneous matches could be ignored. Thus, to simplify the calculations, all matches were assumed to be true matches. Whether or not each artwork was considered as having been recognized was determined by the number of matches for each image. If the correct artwork had the highest number of matches, it was recognized; otherwise, it was not recognized. Next, we calculated the accuracy of each method in each panorama category as a fraction of correctly-recognized images in that category. Figure 24 compares the results for the proposed method with those for other methods.

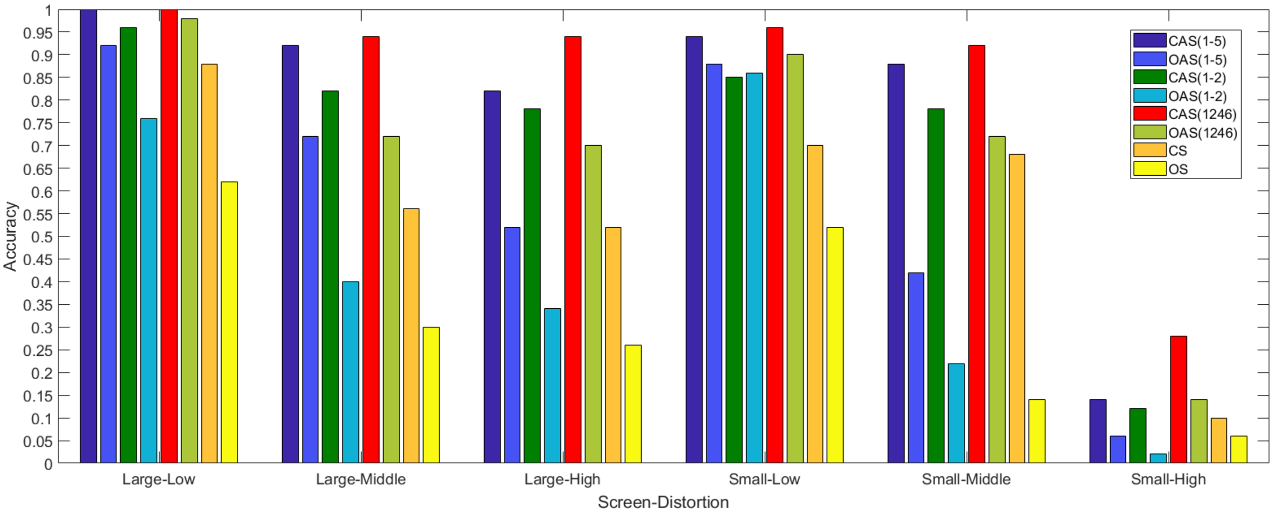

Fig. 24. Accuracy of both the proposed and baseline methods on the artwork dataset, under several different conditions.

Here, “C” and “O” represent a cube map and the original panorama, respectively, while “S” and “AS” denote SIFT and ASIFT, respectively; the following numbers indicate the transforms used: “1–5” denotes transforms 1 to 5, whereas “1246” denotes transforms 1, 2, 4, and 6.

These results indicate that, for a small screen with high distortion, even though the accuracy of the proposed method (CAS(1246)) was relatively low, it was still the best-performing method in the test. Furthermore, the proposed method also exhibited better performance than other methods in all the other cases tested, with recognition accuracies that were consistently more than 85%.

Therefore, we can conclude that the proposed panorama transformation method can offer significant improvements compared with using the original panorama directly. We can also see that ASIFT has a notable advantage over SIFT for image recognition, and that the optimized ASIFT algorithm can achieve possibly even better performance than ASIFT in comparatively less time. In summary, the proposed method for recognizing artwork in 360° images based on cubic projection and the optimized ASIFT algorithm is highly efficient.

## IV. Conclusion

The advantage of the proposed method over previous approaches to recognizing artwork in panoramic images is that this method is able to recognize artwork efficiently while providing a good visual experience and better performance.

In this study, we have used cubic projection to transform distorted panorama images and then employed an optimized ASIFT algorithm to accelerate computation and improve recognition accuracy. We have also adopted a one-to-one mapping constraint to eliminate the majority of erroneous feature matches.

Our experimental results show that our proposed approach offers clear improvements in both accuracy and computing time. Satisfactory results can be obtained despite the panoramas being seriously distorted.

However, the algorithm might face issues in real-world situations, where artworks can be displayed anywhere. In addition, the efficiency could be improved using a GPU to accelerate the feature extraction and matching tasks. Therefore, in future work, we will consider adding a powerful GPU to accelerate the algorithm. In addition, we will apply the algorithm to large artwork datasets that cover as wide a variety of situations as possible.

## Acknowledgments

This work was supported by the Ministry of Culture, Sports and Tourism (MCST) and the Korea Creative Content Agency (KOCCA) in the Culture Technology (CT) Research & Development Program 2017.

## References

[1] T. Ho and M. Budagavi, “Dual-fisheye lens stitching for 360-degree imaging,” *IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 2172-2176, 2017.

[2] G. Zhen, L. Zhuo, J. Zhang, and X. G. Li, “A comparative study of local feature extraction algorithms for web pornographic image recognition,” *Informatics and Computing*, pp. 87-92, 2015.

[3] D. G. Lowe, “Distinctive image features from scale-invariant keypoints,” *International Journal of Computer Vision*, vo1. 60, no. 2, pp. 91-110, 2004.

[4] H. Bay, T. Tuytelaars, and L. V. Gool, “SURF: Speeded up robust features,” in *Proc. European Conference on Computer Vision (ECCV)*, 2006, vol. 3951, no. 3, pp. 404-417.

[5] J. M. Morel and G. Yu, “ASIFT: A new framework for fully affine invariant image comparison,” *SIAM Journal on Imaging Sciences*, vol. 2, no. 2, pp. 438-469, 2009.

[6] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, “ORB: An efficient alterative to SIFT or SURF,” in *Proc. International Conference on Computer Vision (ICCV)*, 2011, pp. 2564-2571.

[7] L. Stefan, M. Chli, and R. Y. Siegwart, “BRISK: Binary robust invariant scalable keypoints,” in *Proc. International Conference on Computer Vision (ICCV)*, 2011, pp. 2548-2555.

[8] A. Alahi, R. Ortiz, and P. Vandergheynst, “FREAK: Fast retina keypoint,” in *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012, pp. 510-517.

[9] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” *Advances in Neural Information Processing Systems*, pp. 1097-1105, 2012.

[10] M. D. Zeiler and R. Fergus, “Visualizing and understanding convolutional networks,” in *Proc. European Conference on Computer Vision*, 2014, pp. 818-833.

[11] C. Szegedy *et al*., “Going deeper with convolutions,” in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 1-9.

[12] K. M. He, X. Y. Zhang, S. Q. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 770-778.

[13] J. X. Xiao, K. A. Ehinger, A. Oliva, and A. Torralba, "Recognizing scene viewpoint using panorama place representation," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, 2012, pp. 2695-2702.

[14] H. Yang and H. Zhang, "Indoor structure understanding from single 360 cylindrical panorama image," *Computer-Aided Design and Computer Graphics*, pp. 421-422, 2013.

[15] C. Li, L. Creusen, L. Hazelhoff, and P. H. N. D. With, "Detection and recognition of road markings in panorama images," in *Proc. Asian Conference on Computer Vision*, 2016, pp. 448-458.

[16] Y. D. Zhang, S. R. Song, P. Tan, and J. X. Xiao, "Pano-context: A whole-room 3d context model for panorama scene understanding," in *Proc. European Conference on Computer Vision*, 2014, pp. 668-686.

[17] F. Deng, X. Zhu, and J. Ren, "Object detection on panorama images based on deep learning," in *Proc. Control, Automation and Robotics* (*ICCAR)*, 2017, pp. 375-380.

[18] A. Iscen, G. Tolias, Y. Avrithis, T. Furon, and O. Churn, "Panorama to panorama matching for location recognition," in *Proc. ACM on International Conference on Multimedia Retrieval*, 2017, pp. 392-396.

[19] H. Fu, D. H. Xie, R. F. Zhong, Y. Wu, and Q. Wu, "An improved ASIFT algorithm for indoor panorama image matching," in *Proc. Ninth International Conference on Digital Image Processing*, 2017, vol. 10420, p. 104201C.

[20] X. Jin and J. W. Kim, "Artwork identification for 360-degree panorama images using polyhedron-based rectilinear projection and keypoint shapes," *Applied Sciences*, vol. 7, no. 5, p. 528, 2017.

[21] Y. W. Kim *et al*., "Automatic content-aware projection for 360° videos," *Computing Research Repository (CoRR)*, pp. 4753-4761, 2017.

[22] T. K. Sharpless, B. Postle, and D. M. German, "Pannini: A new projection for rendering wide angle perspective images," in *Proc. the Sixth International Conference on Computational Aesthetics in Graphics, Visualization and Imaging*, 2010, pp. 9-16.

[23] C. Zhou, Z. Li, and Y. Liu, "A measurement study of oculus 360," *Degree Video Streaming*, pp. 27-37, 2017.

[24] C. Brown. Bring pixels front and center in VR video. [Online]. Available: https://www.blog.google/products/google-vr/bringing-pixels-front-and-center-vr-video , 2017.

[25] VLFeat.org. [Online]. Available: http://www.vlfeat.org/index.html

**Dayou Jiang** received his M.S. degree in Computer Application Technology from YanBian University, China, in 2016. He is currently pursuing a Ph.D. degree in Copyright Protection at Sangmyung University, Korea. His research interests include image retrieval, image identification, music identification, and digital forensics.

**Jongweon Kim** received a Ph.D. degree from the University of Seoul, Korea, majoring in signal processing, in 1995. He is currently a professor in the Dept. of Electronics Engineering and director of Creative Content Labs at Sangmyung University, Korea. He has a lot of practical experience in digital signal processing and copyright protection technology in institutional, industrial, and academic environments. His research interests are in the fields of copyright protection technology, digital rights management, digital watermarking, and digital forensic marking.