



Spherical Panorama Image Watermarking Using Viewpoint Detection

Jihyeon Kang, Sangkeun Ji and Heungkyu Lee

EasyChair preprints are intended for rapid dissemination of research results and are integrated with the rest of EasyChair.

October 25, 2018

Spherical Panorama Image Watermarking Using Viewpoint Detection

Jihyeon Kang¹, Sang-Keun Ji², and Heung-Kyu Lee²

¹ Graduate school of information security, School of Computing, Korea Advanced Institute of Science and Technology, Korea, kangji@kaist.ac.kr

² School of Computing, Korea Advanced Institute of Science and Technology, Korea, sangkeun.ji@kaist.ac.kr heunglee@kaist.ac.kr

Abstract. Even though interest in spherical panorama content has increased rapidly, few studies have examined watermarking techniques for this content. We present a new watermarking technique to protect spherical panorama images as well as view-images that are rendered with a specific viewpoint. Solving the watermark synchronization problem in the detection process requires finding the viewpoint of a view-image. Scale Invariant Feature Transform (SIFT) and Euclidean transformation matrix are used to find viewpoint information of a detection target view-image. Using the viewpoint information, a view-image can be recovered to a source image and then we can detect watermark from it. The experimental results show robustness against several attacks such as JPEG compression, Gaussian filter, and noise addition attack.

Keywords: Image watermarking · Spherical panorama · Omni-directional image · 360 VR watermarking.

1 Introduction

Recently, interest in spherical panorama content (a.k.a. omni-directional contents, 360×180 degree contents, VR contents) has increased rapidly. Unlike other existing media content, it allows viewers to choose their viewpoint. Thanks to recent improvements in hardware and software for spherical panorama content, one can easily create, distribute, and appreciate their content. Thus, the market for spherical panoramas has grown both qualitatively and quantitatively and the copyright protection problem has become an important issue. Watermarking techniques have been proposed as a solution to the image copyright problem. However, watermarking techniques that can properly protect spherical panorama content are absent.

A view-image comes from spherical panorama's source image with a specific viewpoint. A normal user can only see view-images rather than the spherical panorama's source image, so view-images are easier to leak than the entire source image. Furthermore, a view-image can have sufficiently high content value and a replicated spherical panorama source image can be made using several view-images. Therefore, spherical panorama watermarking techniques should detect

watermarks from view-images. However, existing watermarking techniques for 2D images or 3D stereoscopic images cannot be applied directly to spherical panoramas because its distortion varies depending on viewer’s viewpoint in the rendering process.

This paper proposes a spherical panorama image watermarking technique. For the synchronization between location of embedding and location of detecting, watermark is embedded and detected on spherical panorama’s source image that has a equirectangular form. Therefore, to detect watermark from a view-image, we should render the view-image into the source-image. The rendering needs the view-image’s viewpoint information; we propose using the Scale Invariant Feature Transform (SIFT) feature points matching algorithm [1] and Euclidean matrix transformation to get viewpoint information of a view-image.

This paper is outlined as follows. First, we will explain related work involving image watermarking for various content in Section 2, we will show a spherical panorama image watermarking algorithm in Section 3, and Section 4 shows the experimental results. Finally, future research directions are proposed and the paper is concluded.

2 Related works: Image watermarking for various contents

2D image, Stereoscopic 3D, Depth Image Based Rendering (DIBR), and spherical panoramas are the examples of various types of image content. Watermarking methods have been proposed for various image content and since the various types of content each have their own characteristics, a watermarking technique should be designed that takes account of their characteristics.

To date, numerous watermarking techniques have been proposed for 2D image. Frequency domains such as DCT [2], DFT [3], DWT [4], and contourlet [5] are often used for invisibility and robustness. In various domains, various methods such as spread spectrum [6] and quantization index modulation [7] are used. Furthermore, there are many 2D image watermarking techniques that use template [12] or feature points [8] for the robustness against geometric distortion. Recently, 2D image watermarking techniques have been introduced that uses a deep learning [17][18].

3D image have two kinds of format for 3D content distribution: stereoscopic 3D (S3D) and Depth-image-based rendering (DIBR). S3D uses two images(left image, right image) to get one view and S3D image watermarking schemes emphasize the Human Visual System (HVS) to reduce visual fatigue in the 3D rendering process [11]. S3D and DIBR both use two images, but DIBR uses a center image and a depth image; DIBR makes left and right images using the center and depth images. Therefore, DIBR can control the degree of depth in an image. DIBR watermarking schemes [13][14][15] should consider variable depth depending on user preference. Both DIBR and spherical panorama image watermarking schemes should detect watermark from changing image according to setting, but the degree of change differs. DIBR rendering can only makes local

horizontal translations; therefore, it is not adjustable to spherical panoramas.

In the case of spherical panoramas, which have recently entered the spotlight, watermarking techniques are uncommon. Miura et al. proposed a data hiding technique for omnidirectional images [20], but they did not consider robustness because they focused on hiding information. Furthermore, they only used equirectangular images and did not deal with view-images.

Kang et al. proposed spherical panorama image watermarking using feature points [19]; they embedded watermarks into several original images before combining them into an equirectangular image. The several combined watermarked images become one watermarked equirectangular image; then they could detect a watermark in the equirectangular image. Protecting only the equirectangular image can be performed using existing 2D image watermarking techniques. Usually, several original images that are taken from a specific position to make one equirectangular image are taken by one person. Therefore, a different watermark is not needed for each original image. Furthermore, the view-image can easily be stolen, while the source image (mostly equirectangular image) is relatively hard to be stolen due to the characteristics of the spherical panorama. This is why we embed watermarks into equirectangular images and detect from view-images.

3 Proposed spherical panorama watermarking algorithm

This paper presents a watermarking algorithm for spherical panorama images. As mentioned in the introduction, a spherical panorama watermarking algorithm should not only be able to detect watermark from the entire source image but also from view-images. Considering rendering distortion, detecting watermarks in view-images is similar to detecting watermarks in extremely cropped and almost randomly warped 2D images. Therefore, we propose recovering detection-targeted view-image to equirectangular source image for watermark synchronization.

This section briefly explains spherical panorama images and then the watermark embedding and detection processes are described separately.

3.1 Spherical panorama image

Spherical panorama images are also well known as VR images, omni-directional images, or 360° panorama images. They are very effective for expressing virtual space as real space. Making one spherical panorama image requires several images that include all direction views taken from one point. After that, it is necessary to combine several images into one through a process called stitching. Stitching multiple images containing all directions into one image is like expressing a three-dimensional sphere in a 2D plane. There are many ways to project a 3D sphere into one 2D image and no method can avoid distortion, but the equirectangular form is most commonly used in spherical panorama images.

In equirectangular formed image, width refers to the 360° horizontal view and height refers to the 180° vertical view. Therefore, the length of the width

is always twice that of the height in an equirectangular formed image. Common world maps are the most well-known equirectangular formed image that express the spherical Earth's surface into a single 2D image.

When someone wants to view a spherical panorama image, the equirectangular image is projected onto a sphere and then the viewer observes the partial surface of the sphere from its center. Therefore, spherical panoramas have 360° horizontal and 180° vertical fields of view. The partial image depends on the user viewpoint, which we will call the 'view-image'. Figure 1, 2 will clarify this.

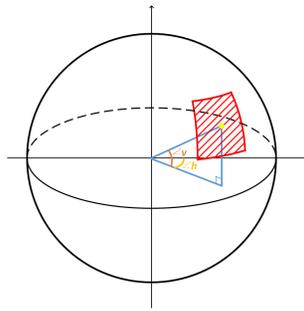


Fig. 1. Viewing a spherical panorama is like viewing a partial of surface from the center. The red area refers to 'view-image'.

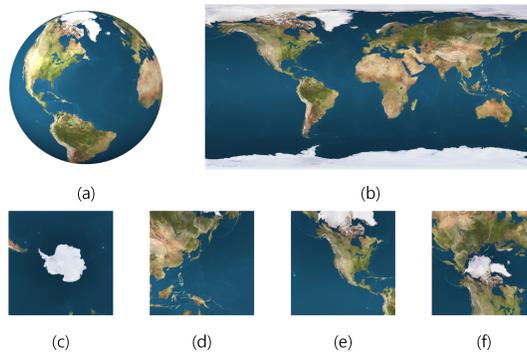


Fig. 2. (a): sphere, (b): equirectangular (source image), (c)-(f): view-images

3.2 Watermark Embedding

Watermark patches We embed watermark into the original source image (the equirectangular image) and we detect watermarks from view-image. This is similar to defending against a random extreme cropping attack. We want at least one complete watermark pattern in the view-image regardless of the viewpoint; we do this by dividing the source image into $n * 2n$ number of blocks, each of blocks has one watermark patch and all the blocks have same watermark patch. Since adjacent pixels are affected by each other in the spherical panorama rendering process, the edge areas of the blocks are not used and the center area of the blocks is only used as watermark patch. Figure 3 shows an example of watermark patches' area without considering watermarks' invisibility.

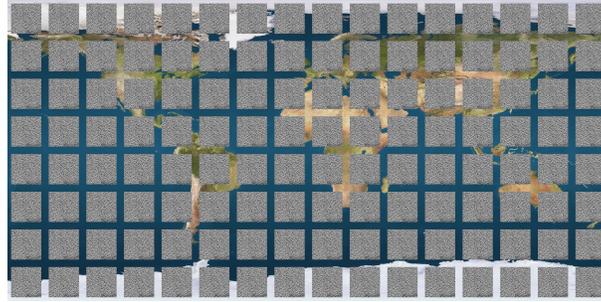


Fig. 3. Example of 9×18 watermark patches (80% of blocks)

Embedding watermark pattern This proposed method can adapt to various existing 2D watermarking methods. For convenience of explanation, we will explain this using a basic DCT domain-based spread spectrum watermarking method.

The secret key (K_s) is used to make a watermark pattern (W_p) that is a random sequence that follows a Gaussian distribution with an average of 0 and a variance of 1. The length of the watermark pattern is proportional to the size of the watermark patch. W_p and the original DCT coefficients (C_o) are used to create watermarked coefficients (C_w). Making the watermarked coefficients (C_o) equation is as follows.

$$C_{w(i,k)} = C_{o(i,k)} + \alpha_w |C_{o(i,k)}| W_{p(k)} \quad (1)$$

In Equation 1, i denotes the index of the watermark patches and k denotes index of the watermarking-targeted coefficients. Watermarking-targeted coefficients are coefficients of middle frequency in DCT domain. α_w denotes watermark

strength value and this value controls the trade-off between the watermark’s robustness and visibility. Original coefficients (C_o) are replaced by watermarked coefficients (C_w) and then a watermarked RGB-channel source-image is obtained through inverse-DCT.

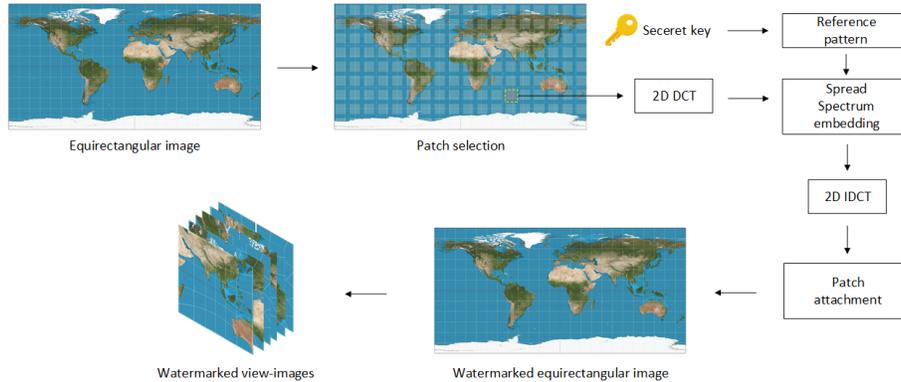


Fig. 4. Watermark embedding process

3.3 Watermark Detection

Spherical panorama watermarking schemes should be able to detect watermarks from source images and view-images. Detecting a watermark from the source image is the same as detecting a watermark from 2D image in this scheme. We propose a way to detect watermarks in a view-image by recovering a view-image to the source image. However, the recovery process needs viewpoint information from the view-image, which is why we need a viewpoint detection process. We divide the viewpoint detection process into two steps: a near-viewpoint detection step and a precise-viewpoint detection step; these are explained in detail in following subsections.

Near-viewpoint Detection Spherical panorama view-image is the part of a sphere’s surface. Therefore, each view-image’s center can be expressed as two spherical angle variables; one is horizontal ($-180^\circ \sim 180^\circ$) and the other is vertical ($-90^\circ \sim 90^\circ$). Obtaining a viewpoint of a view-image by comparing all the each of view-images is almost impossible. We propose a method to determine near-viewpoint using the SIFT matching technique.

We need reference view-images to obtain the near-viewpoint of a detection-targeted view-image. Reference view-images are generated from the original (or watermarked but undamaged) source image. The reference view-images represent near view-images that have a similar viewpoint. The sum of reference view-

images should cover the whole spherical panorama image and there should be an overlapping region between the very next neighbor reference view-images. Too large an overlapping region increases the number of required reference view-images and means lower efficiency. Based on empirical results, we used 26 reference view-images in a source image. Figure 5 shows the example of reference view-images.

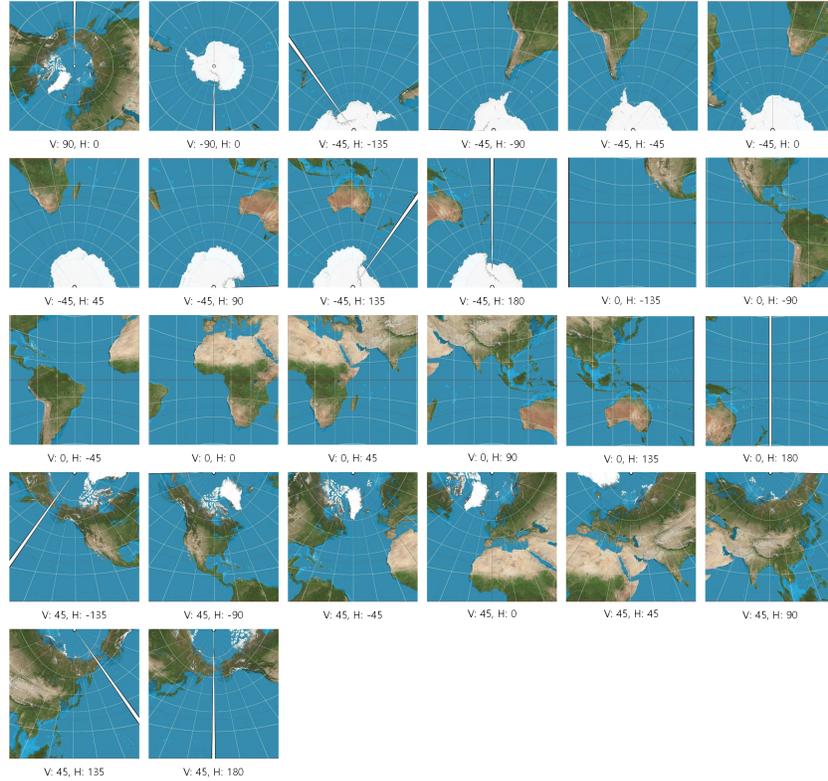


Fig. 5. Example of 26 reference view-images. V: vertical, H: horizontal

If any two view-images have a similar viewpoint, then the two view-images will likely contain several of the same objects. If there is no severe distortion, SIFT, which is robust to Rotation, Scaling, and Translation (RST) can match same objects between them. In other words, if there are many matched SIFT feature points between two view-images, then the two view-images have a similar viewpoint. A near-viewpoint can be obtained using this characteristic.

Initially, SIFT matching is performed between a detection-targeted view-image and each reference view-images. Then, the number of matching points becomes a measure of viewpoint similarity. If the maximum SIFT feature points

matching number is not over a specific threshold, it is determined that the targeted view-image is not derived from the corresponding spherical panorama image. Conversely, if the maximum SIFT matching number exceeds a threshold value, the viewpoint of the reference view-image is determined as the near-viewpoint.

Precise-viewpoint detection Precise-viewpoint can be obtained by searching surround of near-viewpoint. SIFT feature point matching method and Euclidean transformation are used to search.

Initially, we set the candidate-viewpoint as the near-viewpoint that was obtained in the previous step. The candidate-viewpoint is the last guessed viewpoint that could be a precise-viewpoint. After that, the Peak Signal Noise Ratio (PSNR) value between the view-image comes from candidate-viewpoint and a detection-targeted view-image is obtained. We can use the PSNR value to determine whether the viewpoints of the two view-images are the same. In other words, if the PSNR value exceeds a threshold, it means that the viewpoint of the detection-targeted view-image has been found and the candidate-viewpoint becomes the precise-viewpoint. Then, we can move on to the next step. However, if the candidate-viewpoint does not equal the viewpoint of the targeted view-image, the candidate-viewpoint should be changed to a reasonable guess.

SIFT matching information is obtained between view-image from the candidate-viewpoint and targeted view-image. This information is used to guess the vertical, horizontal, and rotation differences between the two images using the Euclidean transformation matrix. The general Euclidean transformation matrix that considers vertical translation, horizontal translation, and rotation can be expressed as Equation 2.

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} s \cdot \cos \theta & s \cdot -\sin \theta \\ s \cdot \sin \theta & s \cdot \cos \theta \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} c \\ d \end{bmatrix} \quad (2)$$

In Equation 2, x and y refer to the positions before conversion and x' and y' refer to the positions after conversion; c represents the degree of translation in the x-axis and d represents the degree of translation in the y-axis. s refers to the scale and θ represents the degree of rotation based on the origin considering the scale (s) change. After replacing $s \cdot \cos \theta$ with a and $s \cdot \sin \theta$ with b , we can expand it and then rewrite it with determinants a, b, c , and d . After that, n SIFT feature point matching pairs between the two view-images can be substituted into it and then it can be expressed as Equation 3.

$$\begin{bmatrix} x_1 & -y_1 & 1 & 0 \\ y_1 & x_1 & 0 & 1 \\ x_2 & -y_2 & 1 & 0 \\ y_2 & x_2 & 0 & 1 \\ \vdots & \vdots & \vdots & \vdots \\ x_n & -y_n & 1 & 0 \\ y_n & x_n & 0 & 1 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \\ d \end{bmatrix} = \begin{bmatrix} x'_1 \\ y'_1 \\ x'_2 \\ y'_2 \\ \vdots \\ x'_n \\ y'_n \end{bmatrix} \quad (3)$$

In Equation 3, the pseudo inverse can be used to get the best approximate values for a, b, c , and d . Using c and d , we can estimate the degree of parallel translation of both x-axis and y-axis. In other words, we can estimate the degree of vertical and horizontal translation. Furthermore, we can estimate the degree of rotation (θ) using Equation 4.

$$s = \sqrt{a^2 + b^2}, \quad \cos \theta = \frac{a}{s} \quad (4)$$

After obtaining the guessed vertical, horizontal, and rotational transformation information, the information is used to estimate the viewpoint of the detection-targeted view-image. In other words, new candidate-viewpoint can be obtained by adjusting the transformation information.

Newly obtained candidate-viewpoints are checked for whether one is the precise-viewpoint. The method compares the PSNR value between the detection-targeted view-image and the view-image comes from the candidate-viewpoint to the threshold, and its method is the same as before. If the PSNR value exceeds a threshold, the candidate-viewpoint becomes the precise-viewpoint and we move on to the next step. Otherwise, we should repeat this step until we find the precise-viewpoint.

Recover to source image The obtained precise-viewpoint is used to recover the detection-targeted view-image to the source-image. It is impossible to reconstruct the entire source image because only information used in the rendering process is recoverable. After reconstruction, an interpolation process is needed to reduce the hole-effect that interferes with watermark detection. Figure 6 presents examples of view-images and corresponding reconstructed source-images and interpolated source-images.

Watermark detection Watermark patches in fully recovered areas are used for detection. This paper only uses patches that have over 95% of recovered pixels. The DCT coefficients are obtained from each patch, then the correlation values between the watermark pattern and the coefficients of each patch can be obtained. As a result, one correlation value comes from one recovered patch. Optionally, some of the highest and lowest correlation values can be excluded to remove outliers.

After that, the average value of the remaining correlation is obtained, and the average value is compared to a threshold value. If the average value exceeds that threshold value, it is determined that the watermark has been detected; otherwise, it is determined that a watermark does not exist or the embedded watermark pattern and the watermark pattern that the process tried to detect are different.

4 Experimental results

The experiments used various types of a hundred equirectangular formed source images [16]. The resolutions of equirectangular images ranged from 1024×2048

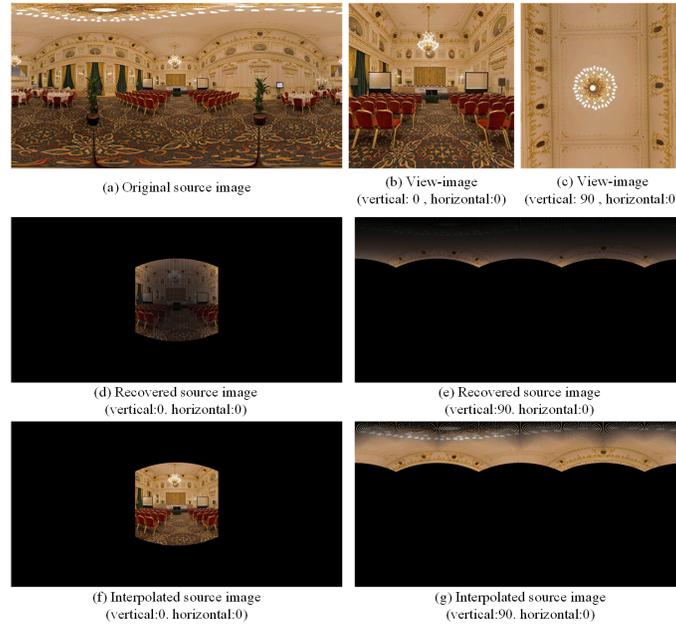


Fig. 6. Examples of recovering from view-image to source-image and its interpolation

to 3000×6000 , the resolution of view-image was 400×400 , and view-image's horizontal and vertical field of view values were 90° . The number of blocks of a source-image was 9×18 and each block had one watermark patch in the center area; the patch size was equal to 75% of the block size. The length of the reference pattern equaled 40% of the number of pixels in a patch and the watermark insertion strength was 0.2.

Initially, we compared four domains to find one that was robust against spherical panorama rendering. We embedded watermark patterns into each spatial, DCT, DWT, and DFT domain patch of the source-images. Except for the spatial domain, we embedded watermark patterns into the middle frequency area. We adjusted the other variables to make the PSNR values similar (The PSNR values that were obtained were between the original view-images and the watermarked view-images).

We used 20 original source-images for this experiment. We made 80 watermarked source-images through four domains. We used four representative view-points to make 320 watermarked view-images in total; then, we obtained average correlation values using the proposed spherical panorama watermarking scheme. Table 1 shows the results.

As a result, DCT domains show the best performance. Therefore, watermark pattern values are inserted into DCT coefficients in the following experiments.

Table 2 shows the results of invisibility experiments using the average PSNR

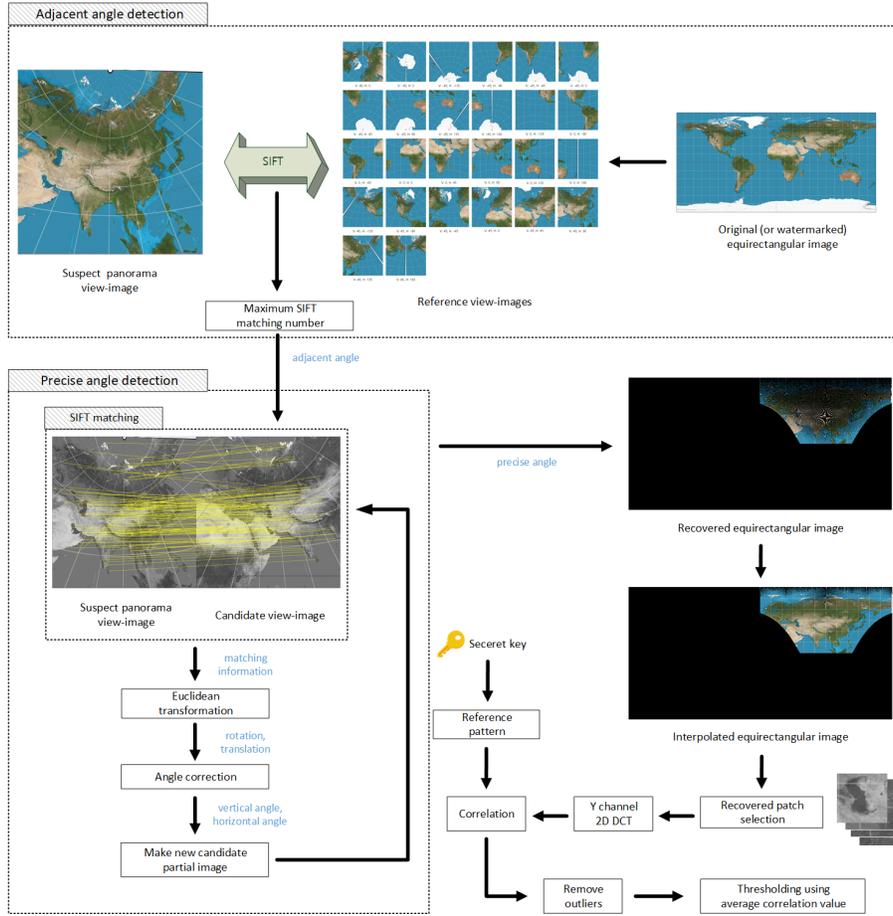


Fig. 7. Watermark detection process

value between the original image and watermarked images. The source-image experiments use a hundred equirectangular formed images and the view-image experiments use 10 random-viewpoint view-images for each source-image.

Figure 8 shows examples of the original and watermarked images. It is difficult to find the difference between them by the human eye.

We experimented with the detection ratio. The detection ratio means how correctly the viewpoint of a view-image is detected. This experiment used about 1600 view-images from a hundred source images. The viewpoint of the view-images was randomly chosen and the detection ratio was 83.15%.

Most of the viewpoint detection failures correspond to one of two types. The first type was when view-images had almost no features, such as images of the

	Viewpoint (vertical, horizontal)			
	(0°, 0°)	(30°, 0°)	(60°, 0°)	(90°, 0°)
Spatial [8]	0.039	0.062	0.057	0.037
	PSNR: 42.47 dB			
DCT [2]	0.109	0.104	0.092	0.076
	PSNR: 42.35 dB			
DWT [4]	0.055	0.055	0.047	0.043
	PSNR: 40.56 dB			
DFT [3]	0.048	0.063	0.052	0.052
	PSNR: 42.61 dB			

Table 1. Average correlation and PSNR values depending on the domain and viewpoint

	Source-image	View-image
PSNR	45.12 dB	46.84 dB

Table 2. Invisibility experimental result

sky or ocean. The other type was when almost similar objects were repeatedly present in the image. Because the algorithm uses SIFT feature points matching technique for viewpoint detection, so the two types above can be a limitation. However, the first limitation is not a problem because the view-images that have almost no feature points are unlikely to be worth protecting. Figure 9 shows examples of the two viewpoint detection failure types; the upper lined view-images are included in the first type and the lower lined view-images are included in the second type.

We experimented on the watermarking performance; we used about 500 view-images for each test in this experiment. Initially, we obtained average correlation coefficients from correct and incorrect watermarks. A correct watermark means that the same watermark pattern is used in embedding and detecting and incorrect watermark means that different watermark patterns are used. The detection ratio result was obtained by comparing the threshold with the average correlation coefficient. Table 3 and Figure 10 show the results; the results show the meaningful difference between the correct watermark and the incorrect watermark; we set the threshold value to 0.02 to get the detection ratio.

	Correct WM	Wrong WM
Average correlation	0.0796	-0.0002
Detection rate	88.8 %	0 %

Table 3. Correlation results without attack



Fig. 8. Examples of original and watermarked images

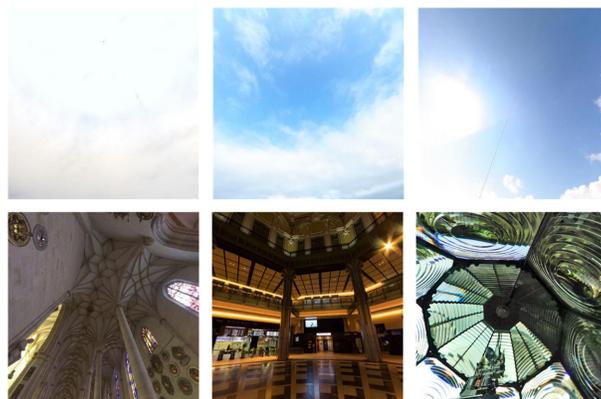


Fig. 9. Examples of viewpoint detection failed view-images

Further experiments were conducted to determine the robustness of the watermarking against JPEG compression, noise addition attack, and blurring attack; the results are as shown in Table 4.

Each attack experiment used about 300 random viewpoint view-images. The experiment used four types (90, 80, 70, and 60) of JPEG compression attack with different compression qualities. The noise addition attack experiments used Gaussian noise with an average of 0 and a variable of 0.0005, while the blurring attack experiments used a Gaussian filter with 0.6 sigma. The PSNR values between the watermarked view-images and watermarked and attacked view-images show the degree of attacks.

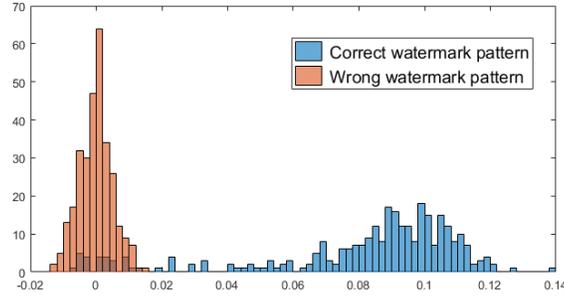


Fig. 10. Correlation histogram using correct and wrong watermark pattern

	JPEG 90	JPEG 80	JPEG 70	JPEG 60	Noise	Gaussian
Average correlation	0.0536	0.0516	0.0466	0.0387	0.0725	0.0665
Detection rate	83.8 %	85.1 %	87.2 %	79.7 %	88.8 %	90.2 %
PSNR	36.34 dB	33.57 dB	32.23 dB	31.69 dB	33.15 dB	34.22 dB

Table 4. Correlation result with attacks

5 Conclusion

In this paper, we proposed spherical panorama image watermarking technique. We used equirectangular formed source image to synchronize embedding and detecting location. Viewpoint detection process was used to recover from view-image to source image. The experiments shows robustness against JPEG compression, Gaussian filter, and noise addition attacks. Unfortunately, this proposed scheme have limitations for some kinds of images and this is non-blind watermarking technique. As we known, this is first watermarking technique for spherical panorama contents in the right scenario and we showed the possibility of watermarking for spherical panorama contents.

Acknowledgements

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (NRF-2016R1A2B2009595).

References

1. Lowe, David G. "Distinctive image features from scale-invariant keypoints." International journal of computer vision 60.2 (2004): 91-110.
2. Barni, Mauro, et al. "A DCT-domain system for robust image watermarking." Signal processing 66.3 (1998): 357-372.

3. Pereira, Shelby, and Thierry Pun. "Robust template matching for affine resistant image watermarks." *IEEE transactions on image Processing* 9.6 (2000): 1123-1129.
4. Xia, Xiang-Gen, Charles G. Boncelet, and Gonzalo R. Arce. "A multiresolution watermark for digital images." *Image Processing, 1997. Proceedings., International Conference on*. Vol. 1. IEEE, 1997.
5. Akhaee, Mohammad Ali, S. Mohammad Ebrahim Sahraeian, and Farokh Marvasti. "Contourlet-based image watermarking using optimum detector in a noisy environment." *IEEE Transactions on Image Processing* 19.4 (2010): 967-980.
6. Cox, Ingemar J., et al. "Secure spread spectrum watermarking for multimedia." *IEEE transactions on image processing* 6.12 (1997): 1673-1687.
7. Chen, Brian, and Gregory W. Wornell. "Quantization index modulation: A class of provably good methods for digital watermarking and information embedding." *IEEE Transactions on Information Theory* 47.4 (2001): 1423-1443.
8. Lee, Hae-yeoun, Hyungshin Kim, and Heung-Kyu Lee. "Robust image watermarking using local invariant features." *Optical Engineering* 45.3 (2006): 037002.
9. Chang, Chin-Chen, Piyu Tsai, and Chia-Chen Lin. "SVD-based digital image watermarking scheme." *Pattern Recognition Letters* 26.10 (2005): 1577-1586.
10. Chandra, DV Satish. "Digital image watermarking using singular value decomposition." *Circuits and Systems, 2002. MWSCAS-2002. The 2002 45th Midwest Symposium on*. Vol. 3. IEEE, 2002.
11. Ji, Sang-Keun, Ji-Hyeon Kang, and Heung-Kyu Lee. "Perceptual Watermarking for Stereoscopic 3D Image Based on Visual Discomfort." *International Conference on Information Science and Applications*. Springer, Singapore, 2017.
12. Lu, Wei, Hongtao Lu, and Fu-Lai Chung. "Feature based watermarking using watermark template match." *Applied Mathematics and Computation* 177.1 (2006): 377-386.
13. Lin, Yu-Hsun, and Ja-Ling Wu. "A digital blind watermarking for depth-image-based rendering 3D images." *IEEE transactions on Broadcasting* 57.2 (2011): 602-611.
14. Wang, Shen, Chen Cui, and Xiamu Niu. "Watermarking for DIBR 3D images based on SIFT feature points." *Measurement* 48 (2014): 54-62.
15. Kim, Hee-Dong, et al. "Robust DT-CWT watermarking for DIBR 3D images." *IEEE Transactions on Broadcasting* 58.4 (2012): 533-543.
16. Equirectangular flickr group, <https://www.flickr.com/groups/equirectangular>. Last accessed 6 Juli 2018
17. Mun, Seung-Min, et al. "A Robust Blind Watermarking Using Convolutional Neural Network." *arXiv preprint arXiv:1704.03248* (2017).
18. Kim, Wook-Hyung, et al. "Convolutional Neural Network Architecture for Recovering Watermark Synchronization." *arXiv preprint arXiv:1805.06199* (2018).
19. Kang, I-Seul; Seo, Young-Ho; Kim, Dong-Wook. (2017). Blind Digital Watermarking Methods for Omni-directional Panorama Images using Feature Points. *The Korean Institute of Broadcast and Media Engineers*, 22(6), 2017, 785-799.
20. Miura, Yasutoshi, et al. "Data hiding technique for omnidirectional JPEG images displayed on VR spaces." *Advanced Image Technology (IWAIT), 2018 International Workshop on*. IEEE, 2018.