Data hiding technique for omnidirectional JPEG images displayed on VR spaces

Yasutoshi MIURA, Xuefei LI, Seok KANG, Yuji SAKAMOTO
Graduate School of Information Science and Technology, Hokkaido University
N14, W9, Kita-ku, Sapporo, 060-0814, Japan
Email: ymiura@ist.hokudai.ac.jp

***Abstract*—We propose a data hading technique for images in this paper. Our technique hides additional data of images into image data area for not being separated each other easily. The proposed technique uses omnidirectional JPEG images as cover data because of the huge amount of the embeddable area. Moreover, the embedded omnidirectional image can be viewed on VR space and the embedded data are extracted and playing in VR spaces using HMDs. We verified that our technique embedded the additional image into images efficiently in terms of PSNR, payload and overhead.**

Keywords*—*Data hiding; Omni-directional image; Virtual reality; HMD; JPEG;

I. INTRODUCTION

Nowadays, to view virtual Reality (VR) content using head mounted displays (HMDs) is very popular[1][2]. VR applications make us feel like we are in a real world without actually going to that specific place. In addition, omnidirectional cameras are popular, and people take omnidirectional images everywhere.

Omni directional images have a high affinity with VR spaces because the images have an all-around view from camera's coordinates. We can get a realistic view by simply mapping images to VR spaces and looking at them through HMDs. Such VR applications would be improved if additional data, like surrounding sound and object information, came from where objects in the images are located. This sort of application is popular, but coming up with a way to manage and save this additional data as related files has not yet been done.

Typically, the additional data is saved in the same folder or related folder of images. That is, additional data like mp3s or JPEG, are tagged with images in no particular way. However, in most cases, this connection is as such that if the images are taken from the original folder, there is no retrieving the additional data using only the image. Saving the additional data in the same environment in which images are taken is preferable. For example, it is hard to say what a simple omnidirectional image is displaying because of its characteristic distortion. However, if the omnidirectional image has a thumbnail that is taken in a VR space or a real space, we can easily grasp the point of the image without decoding or displaying a lot of image data. Furthermore, if the additional data is combined with image data, we could easily share that additional data using a conventional image management system.

Generally, the header of image data has been used as a place to save additional data. The header is a data area that includes metadata for image data and a free area anyone can see and edit. However, this space can be unintentionally deleted for privacy protection, and the size of the header is imposed on the image size as-is. Thus, we propose saving additional data into the image data using a data hiding technique that hides one type of data in another type of data without increasing file size.

We developed the above-mentioned VR system and also confirmed that the proposed data hiding technique can extract additional data in response to user's sightline.

II. PROPOSED TECHNIQUE

A. Overview

An overview of the proposed VR image-viewing system is shown in Fig. 1. The images are JPEGs input from omnidirectional cameras. In the data hiding process (Fig. 2), first, JPEG images are decoded to the level of 8×8 DCT blocks. Then, the least significant bit (LSB) replacement data hiding for DCT blocks is implemented to hide the bitstream of additional data. The reason for embedding in DCT blocks is the ease of the operation. Since JPEG file sizes depend on these blocks, it has been reported that their file size can be controlled if their coefficient values are properly changed[3].

At the end of the embedding process, embedded DCT blocks are encoded as-is. Through these processes, we can get the embedded images. For the extracting process, the above procedure is reversed. After that, a 3D sphere object is set in a VR space and its inside surface is wrapped with omnidirectional images. By standing at the center of the sphere and seeing specified regions through HMDs, we can access the additional data corresponding with an omnidirectional view.

For comparison, practical examples of the application are shown in Fig. 3. In this figure, a human voice, portrait data, and metadata are set as additional data. In a conventional working space, the additional data and omnidirectional images are managed separately by metadata, so the additional

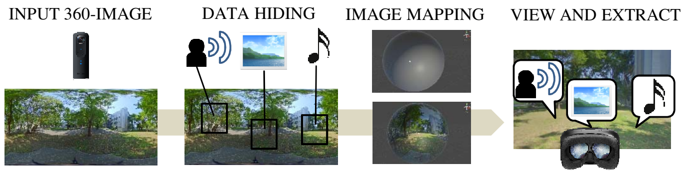
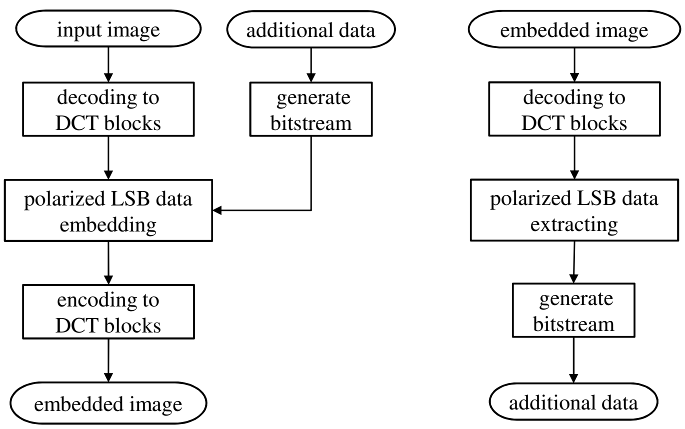


Fig. 1. Proposed VR image viewing system



(a) embedding process

(b) extracting process

Fig. 2. Data hiding process

data cannot be retrieved if the image is taken away. This is because the working space of conventional applications is folder based. In contrast, the proposed working space is image based. Thus, the additional data could be retrieved as long as users have images. In addition, the total file size of the working space would be less than a conventional one by using the polarized LSB replacement method.

B. Polarized LSB replacement data hiding for image

We applied polarized LSB replacement data hiding[4] to embed the additional data. This technique changes the least bit of DCT coefficients with a bit of additional data. Such a technique is also called the LSB substitution method, and it is used in steganography[5] and watermarking. An example of the process of polarized LSB replacement data hiding is shown in Fig. 4. The target DCT blocks are processed by raster scanning from the upper left block of images. A DCT block is composed of one direct current (DC) coefficient at the upper left and the 63 alternating current (AC) coefficients. We use the nonzero AC coefficients as a target because a slight change at the DC coefficient and the zero AC coefficient causes a lot of image deterioration. The nonzero AC coefficients are ordered in zig-zag scanning as a parameter O_i. After that, the bitstreamof additional data A_i and O_i are compared in accordance with the polarized LSB replacement data hiding algorithm as shown in Fig. 4. Then the processed data P_i is obtained. For example, when $O_i = 2, -4, -3, -1, 1, 2, -2, 1$ and $A_i = 10111010$, O_i is changed to $P_i = 1, -3, -2, 1, 1, 2, -2, -1$. When extracting, we can get A_i by confirming the polarity and parity of the O_i value while referring to the algorithm.

In this algorithm, an integer O_i can be reduced or increased by 1 at most. That means one nonzero AC coefficient has one additional bit. That is to say, the number of nonzero AC coefficients is the same as the payload the image has. Although polarized LSB replacement data hiding is irreversible and causes image-quality deterioration, this embedding technique has a large embedding capacity. In case of using the reversible data hiding technique[6][7], it is hard to get a large embedding capacity because of the redundancy when embedding. Furthermore, the polarized LSB replacement technique absolutely reduces the file size of the original cover data after the processing. It is because polarized LSB replacement data hiding intentionally changes the coefficient's absolute value in order to not go higher than the original's. Since the compression rate of JPEG is determined by the length of the zero coefficient sequence and how close the absolute value of coefficients is to zero in DCT blocks, the more we embed, the more the file size of images decreases.

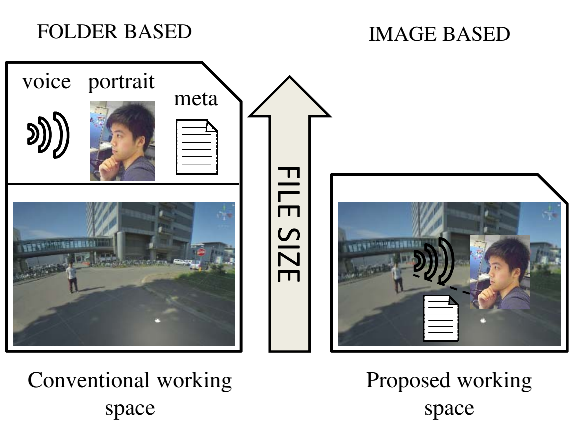


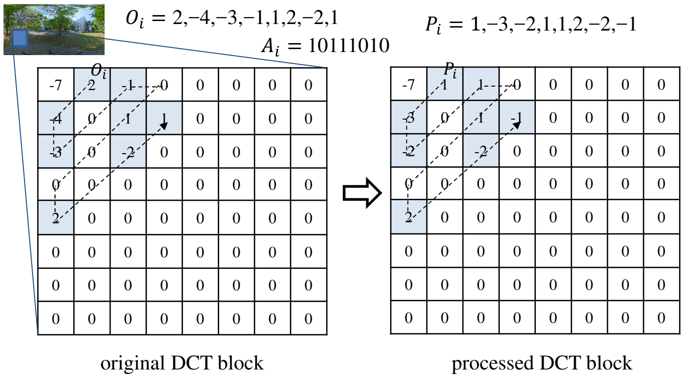
Fig. 3. Practical usage of the application

III. Experiments

A. Embedding

We evaluated the quality deterioration rate of embedded images in terms of peak signal noise ratio (PSNR) and the payload (embedded data size) and overhead (file size increase). Input images are eight JPEG images of 5376×2688 pixels taken by an omnidirectional camera. Images for the experiment are shown in Fig. 5 and each image is numbered $(x_1, x_2, \ldots, x_7, x_8)$. Additional data input is one mp3 that has a long enough bitstream. The first target DCT block is the upper left block of images where $(y, x) = (0, 0)$. If the additional data bitstream is embedded into the last coefficient of the DCT blocks, the next block $(0, 1)$ is processed by raster scan and so on.

When all nonzero coefficients are embedded onto the lower right DCT block of images, the processed image data size and payload are shown in Fig. 6. An overhead ratio and payload ratio to original image data size are




```
if (O_i > 0)
   if (A_i = 1)
      reduce O_i to odd integer
   if (A_i = 0) & (O_i ≠ 1)
      reduce O_i to even integer
   if (A_i = 0) & (O_i = 1)
        O_i = −1
if (O_i < 0)
   if (A_i = 0)
      increase O_i to odd integer
   if (A_i = 1) & (O_i ≠ −1)
      increase O_i to even integer
   if (A_i = 1) & (O_i = −1)
        O_i = 1
```

A_i : *Additional data*
O_i : *Original data*
P_i : *Processed data*
i : *number of nonzero AC coefficients* ($0 \le i \le 63$)

polarized LSB replacement data hiding algorithm

Fig. 4. Polarized LSB replacement data hiding for image

Fig. 5. Omnidirectional JPEG images for experiment (5376×2688)

shown in Fig. 7, and these mean results are shown in Table I. As shown in Fig. 7 and Table I, every image can contain payload for about 10% of an original image's data size. Moreover all image sizes decrease about 3%. A PSNR recorded above 50 dB and this line is high enough quality that we can hardly tell the difference between the original images and the embedded images. A comparison of processed omnidirectional images is shown in Fig. 9. As we can see, there is no conspicuous deterioration in the processed omnidirectional image compared to the original omnidirectional image.

In terms of payload, we could embed an average 617,789 bytes of an input mp3 file. Considering that the mp3 file is 128bps, the payload is equal to 37 seconds of sound data. It is conceivable that this amount of additional data is enough for one image.

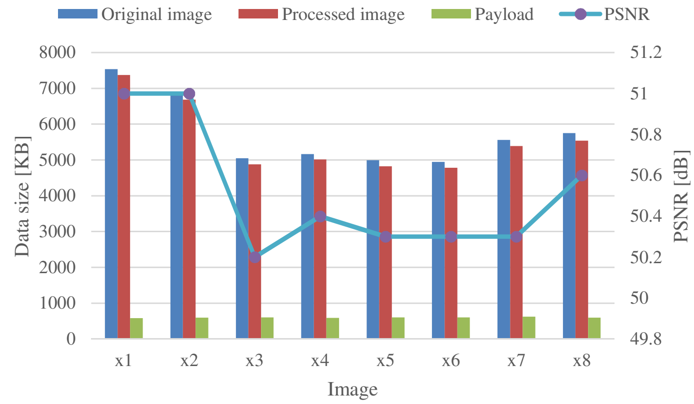


Fig. 6. File size of processed image and PSNR after embedding

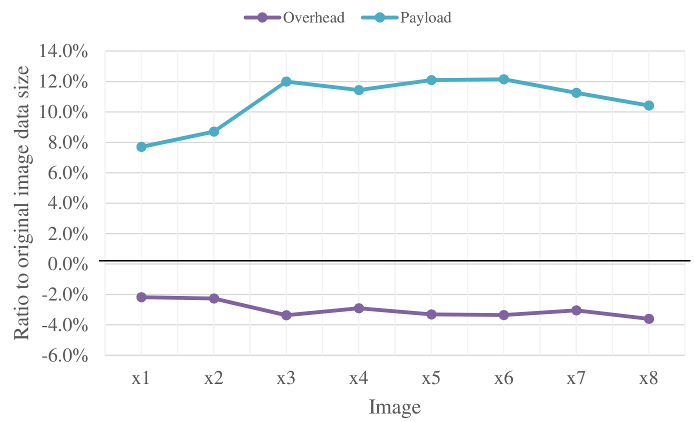


Fig. 7. Overhead ratio and payload ratio to original images

B. Application working

A working environment of the proposed application is shown in Fig. 8. Putting on a headset and earphones, the application's users can immerse themselves in a VR world. Inside the HMD, users can look at every angle they want to see.

In the embedding process, users can simultaneously set up specified regions in a VR space simultaneously while choosing an additional data file. The data for regions is also saved and embedded as additional data. When seeing an embedded image, if a user's sightline enters a specified region, it is confirmed that the corresponding additional data is playing correctly (Fig. 10).

IV. Conclusion

We proposed a new omnidirectional image viewing system using HMDs and data hiding techniques. Many VR applications for omnidirectional JPEG images have already been developed, but managing and saving the additional data conveniently when images are taken from

TABLE I. PSNR, OVERHEAD AND PAYLOAD

	psnr [dB]	overhead [%]	payload [%]	payload [Byte]
mean	50.5	-2.9	10.6	614,471

(a) landscape

(b) inside HMD

Fig. 8. Working environment

(a) original omnidirectional image

(b) processed omnidirectional image

Fig. 9. Comparison of omnidirectional images

their original folder. Therefore we tried to embed additional data into the image data so they are not separated. Through our experiments, we verified that our technique efficiently embedded the additional image into images in terms of PSNR, payload, and overhead.

An quantitative evaluation has been left for future work when the embedded bit number per DCT coefficient variably increases.

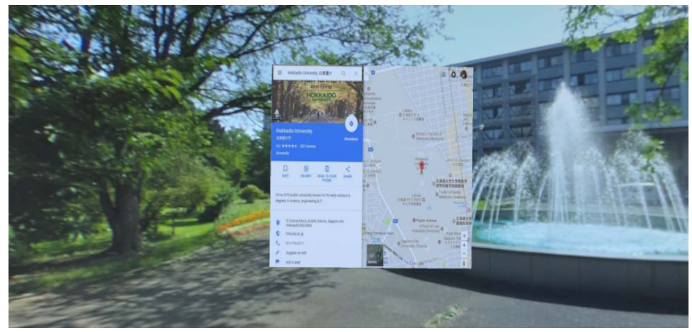
References

[1] A. Fineschi, A. Pozzebon, "A 3D virtual tour of the Santa Maria della Scala Museum Complex in Siena, Italy, based on the use of Oculus Rift HMD, " 2015 International Conference on 3D Imaging(IC3D), pp. 15. IEEE (2015).

[2] Ralf Schfer, Peter Kauff, Robert Skupin, Yago Snchez, Christian weiig, "Interactive Steaming of Panoramas and VR Worlds, " SMPTE Motion Imaging Journal, Vol.126, No.1, (2017).

[3] S. Shimaura, S. Harada, S. Kang, and Y. Sakamoto, "An Evaluation on Controlling File Size Increase in the Image Partial Scrambling Technique Enabled Quantitative Degradation Degree Control by Exchanging DCT Coefficients," IEICE Technical Report, **113**(135), pp. 167–172, (2013).

(a) some hiding regions are set



(b) popup of map information

(c) popup of text message

Fig. 10. Application working

[4] X. Li, Y. Miura, S. Kang, Y. Sakamoto, "A data hiding technique for videos using polarized LSB replacement, " 3rd Symposium of Dong-Eui University and Hokkaido University, (2017).

[5] Wenxu Jiang; Zhenbo Guo; Kaixi Wang; Yongfeng Huang, "A self-contained steganography combining LSB substitution with MSB matching, " 5th International Conference on Computer Science and Network Technology (ICCSNT), pp. 635–640, (2016).

[6] G. Xuan, Y.Q. Shi, Z. Ni, P. Chai, X. Cui, and X. Tong, "Reversible data hiding for JPEG images based on histogram pairs, " 4th Inter.Conf., ICIAR'07, pp.715–727, (2007).

[7] S. Shimaura, S. Kang, and Y. Sakamoto, "Reversible Privacy Protection Technique Controlling Quantitative Scramble Degree and File Size Increase," IEICE Technical Report, **114**(220), pp. 41–46, (2014).