Visualizing Motion of Natural Objects by Deep Learning Optical Flow Estimation in an Omnidirectional Image for Virtual Sightseeing

Motoki Kakuho[1,*], Norihiko Kawai[1]

[1]Graduate School of Information Science, Osaka Institute of Technology

Abstract

Services using omnidirectional images have become increasingly popular. For example, Google Street View enables users to view the scenery of a location online without physically visiting it. However, the use of still images limits the sense of presence. This study proposes a method that focuses on natural elements such as water, sky, and trees within a single omnidirectional image and utilizes deep learning to reproduce their motion in 3D space, generating omnidirectional videos. Experiments demonstrate the effectiveness of the proposed method by comparing results with conventional methods.

Keywords

Omnidirectional Image, Video Generation, Motion Reproduction, Virtual Sightseeing

1. Introduction

Virtual sightseeing services using omnidirectional images have been increasing. For example, TOWNWARP [1] and AirPano [2] enable users to enjoy the scenery of famous tourist spots and cities as videos online without physically visiting them. Additionally, there are studies that combine virtual tourism with education by synthesizing virtual objects into omnidirectional images. For example, CoSpaces [3] provides functions to place virtual objects such as information boards, explanatory text, and human avatars in virtual environments created with omnidirectional images, which can support various types of learning. For ruins tourism, an application has been developed that allows users to learn and enjoy the scenery of the past of a historical site not only as VR at arbitrary locations but also as Indirect AR at the site by synthesizing virtual buildings that existed in the past into omnidirectional images and presenting them to the user [4]. While such services allow users to virtually experience and learn various places around the world, users cannot view locations other than famous tourist spots chosen by the content creators.

In contrast, Google Street View [5] is an example of services that allow users to explore any location worldwide. However, since this service presents still images, it lacks the sense of presence. One solution to this issue is to record videos from fixed points while traveling around the world. However, this method would require a significant amount of time for collecting video data.

To solve this problem, we propose a method that focuses on natural objects such as water, sky, and trees within a single omnidirectional image and reproduces their motion to generate omnidirectional videos for highly realistic virtual sightseeing at arbitrary locations. In the proposed method, for water surface and sky regions, a part of the target omnidirectional image is converted into a perspective projection image, and the optical flow of the water surface and sky is estimated by a deep learning-based approach. The optical flow is then transformed into the motion in 3D space and projected back onto the omnidirectional image, reproducing the motion of the water and sky in the omnidirectional image. For trees, the optical flow is obtained from a reference video in the perspective projection, converted into the motion on vertical 3D planes, and then applied to the omnidirectional image. Semantic segmentation [6] is also used to clearly separate the sky, water surface, and tree regions. This process generates an omnidirectional video where motion is reproduced only in the regions of water, sky, and trees.

2. Related Work

Various studies have been conducted on converting still images into videos by moving objects in them. Among these, there are studies [7, 8, 9, 10] that focuses on the movement of natural objects. For instance, Creating Fluid Animation from a Single Image using Video Database [9] generates high-quality animations by efficiently assigning target images using a Markov Random Field (MRF) and leveraging a fluid video database. Another example is Animating Landscape [10], which generates videos that reproduce the motion of the sky and water surfaces

APMAR'24: The 16th Asia-Pacific Workshop on Mixed and Augmented Reality, Nov. 29-30, 2024, Kyoto, Japan

*Corresponding author.

m1m23a07@oit.ac.jp (M. Kakuho); norihiko.kawai@oit.ac.jp (N. Kawai)

using machine learning with neural networks. However, these methods deal with images in perspective projection. Therefore, for example, if the method [10] is applied to omnidirectional images in equirectangular projection, the generated motion appears unnatural because the model is trained on perspective projection images. It also suffers from parameter dependency, causing motion in regions where no motion should occur. In addition, even though the left and right edges of the omnidirectional image are connected, the conventinal methods do not take this into account. Therefore, when looking around the omnidirectional image as a perspective projection image, we can observe a misaligned border in the texture at the edges.

For these problems, in our previous study [11], we reproduced the motion of the sky and water surface in omnidirectional images by assuming that the sky and water surface could be expressed by straight-line motion on a plane. The proposed method in this study is the extended version, and reproduces more natural motion by using optical flow estimated by deep learning, and also reproduces the motion of trees.

3. Proposed Method

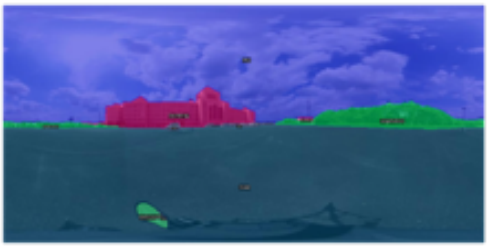
3.1. Overview

The flow of the proposed method is as follows. First, (1) we input an omnidirectional landscape image containing either sky, water or trees, as shown in figure 1(a). This study assumes that omnidirectional images are generated by equirectangular projection so that the bottom pixel is in the direction of gravity obtained from the accelerometer in the camera. Next, (2) we apply the semantic segmentation [6] to the image to divide it into regions such as water surface, sky, trees, and others as shown in Figure 1(b). From the segmented image, we generate a mask image that mask all objects above the horizon except the sky area, as shown in Figure 1(c). Here, due to inaccuracies near the boundaries of the semantic segmentation, the mask regions are expanded to fully include objects except the sky area. Next, (3) using the generated mask image, we generate an image in which all areas above the horizon have sky textures by inpainting [12], as shown in Figure 1(d).

Next, (4) we generate the motion of the water surface, sky and trees by copying pixel values using calculated optical flows. The motion of the water surface and sky are calculated by estimating the motion in 3D space based on the deep learning-based optical flow estimation [10]. The tree motion is calculated by acquiring the motion from a perspective projection video of the trees and reproducing the motion in 3D space.

Finally (5) We combine the video of each region generated in (4) with the input image using the segmented image as shown in Figure 1(b) to generate a video in which only the water surface, sky, and trees move. Here, alpha blending is performed at the boundary of the mask to reduce the unnaturalness at the boundary between the moving and static regions. The following sections describe the details of motion generation in Step (4).

(a) Example of input image



(b) Semantic segmentation

(c) Mask image

(d) Inpainting result

Figure 1: Example of input image and intermediate results.

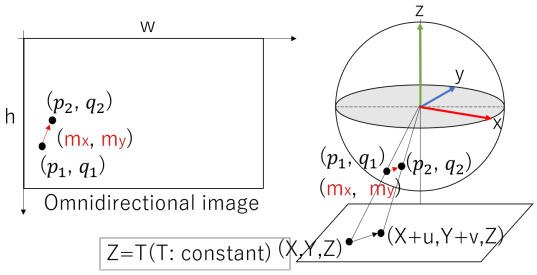


Figure 2: Relationship between sphere and 3D position on water surface.

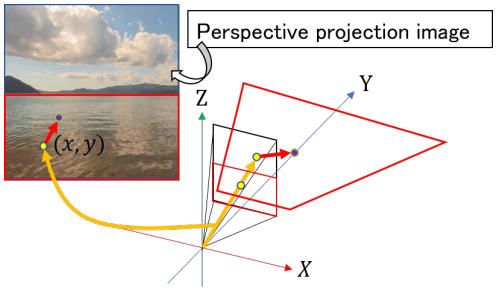


Figure 3: Relationship between 2D and 3D coordinates of flows.

3.2. Generation of Water Motion

For the motion of water, we assume that the water is moving along a planar surface in 3D space. This 3D motion on a plane is represented as a 2D optical flow on the omnidirectional image.

Specifically, we first define the coordinate system for the omnidirectional image and the plane. As shown in Figure 2, position (X, Y, Z) on the water surface corresponding to pixel (p_1, q_1) of the omnidirectional image is determined in a coordinate system where the center of the sphere corresponding to the omnidirectional image is at the origin, as follows:

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} T \tan \frac{\pi q_1}{h} \cos \frac{2\pi p_1}{w} \\ T \tan \frac{\pi q_1}{h} \sin \frac{2\pi p_1}{w} \\ T \end{bmatrix}, \tag{1}$$

where w and h are the width and height of the omnidirectional image, and T is a negative constant representing the height of the water surface.

In this coordinate system, we compute the flow (u, v) at the water surface. First, as shown in Figure 3, a part of the omnidirectional image is extracted as a perspective projection image so that its horizon is at the center of the image height. The optical flow is then estimated by the deep learning-based method in [10]. As illustrated in Figure 3, both the original pixel and the pixel after moving based on the flow are projected onto the plane at height T using the focal lengths f_x, f_y and the image center c_x, c_y of the perspective projection image. The 3D coordinates after projecting the pixel (x, y) onto the plane are calculated as follows:

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} \frac{Tf_y(x-c_x)}{f_x(y-c_y)} \\ \frac{Tf_x}{(y-c_y)} \\ T \end{bmatrix}. \tag{2}$$

Next, the flow map on the plane is determined from the differences between the respective projected 3D coordinates. This process is performed on the pixels in the lower half of the perspective projection image. However,

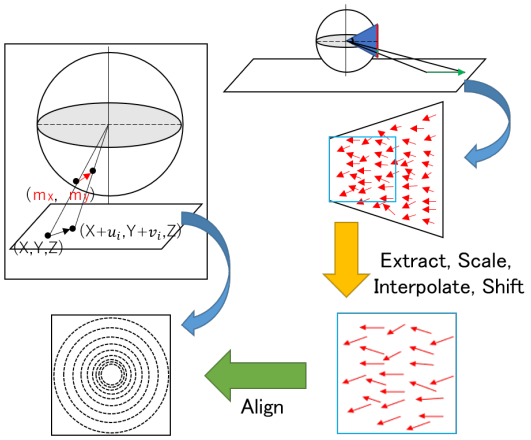


Figure 4: Alignment of flow maps.

the motion calculated on the plane here only corresponds to a part of the lower part of the omnidirectional image. To handle the entire water area in the omnidirectional image, this study assumes that the motion of the water at any given location is similar. Here, since the projected region from the perspective projection image is a trapezoidal shape, as shown by the red outline in Figure 3, the flow map of the region is extracted, scaled, interpolated, and shifted to align with the square region projected from the lower half of the omnidirectional image, as shown in Figure 4.

Next, as shown in Figure 2, the X, Y coordinates on the plane obtained by equation (1) are shifted by flow (u, v), and projected onto the surface of the sphere. Pixel (p_2, q_2) in the omnidirectional image corresponding to the shifted coordinate $(X+u, Y+v, Z)$ on the horizontal plane is determined as follows:

$$\begin{bmatrix} p_2 \\ q_2 \end{bmatrix} = \begin{bmatrix} \frac{w}{2\pi} \tan^{-1} \frac{\frac{Y+v}{z}}{\frac{X+u}{z}} \\ \frac{h}{\pi} \cos^{-1} \frac{}{\sqrt{(X+u)^2+(Y+v)^2+Z^2}} \end{bmatrix}. \tag{3}$$

Finally, the difference between the transformed pixel (p_2, q_2) and the original pixel (p_1, q_1) is calculated as the

optical flow (m_x, m_y) on the omnidirectional image. By performing this process for all pixels, the optical flows for the entire water surface on the omnidirectional image are obtained. Based on the optical flows, pixel values are copied to generate an image where the water surface has moved. This process is repeated for each frame, and a video is generated by combining all the frames.

3.3. Generation of sky Motion

For the motion of the sky, assuming that clouds in the sky move on a plane in 3D space above the scene, the motion is estimated in the same manner as for the water. The optical flows in the upper part of the perspective projection image in Figure 3 is estimated by the deep learning-based method [10], and the motion is projected on the plane, and the motion on of the upper part of the omnidirectional image is finally determined by re-projecting the motion on the plane onto the sphere representing the omnidirectional image.

Note that, as described in section 3.1, by removing all areas other than the sky using inpainting, the plausible sky texture is generated in the areas. Even when the flow is from behind buildings, the generated texture is copied, and the motion of the sky can be reproduced.

3.4. Generation of tree Motion

For the motion of trees, rather than assuming a single plane like water and sky, we assume that, as shown in Figure 5, they move on a vertical plane perpendicular to the radial line from the center of the sphere to the sphere surface at height 0, for each column. Similar to the sky and water, this 3D motion is expressed as a 2D optical flow on the omnidirectional image.

Specifically, a reference video is first input and the optical flows are estimated by Farneback method [13]. The flow map is resized to match the tree region. Next, the 2D coordinates of the input image in the mask region for trees are converted into 3D coordinates as follows:

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} \cos \frac{2\pi p_1}{w} \\ \sin \frac{2\pi p_1}{w} \\ \frac{1}{\tan \frac{\pi q_1}{h}} \end{bmatrix}. \quad (4)$$

The 3D coordinate shifted on the vertical plane based on the flow map, and the shifted 3D coordinate is re-projected onto the shpere. The flow on the omnidirectinal image is determined by the original and the re-projected pixels. By repeating this process for the number of frames in the reference video, a video with the tree motion is generated.

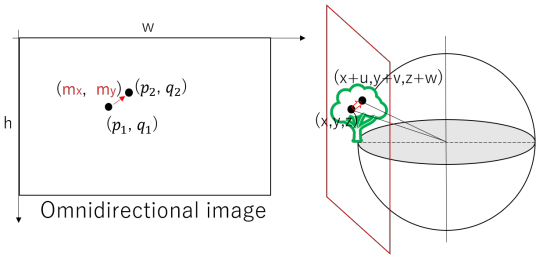


Figure 5: Relationship between sphere and vertical plane for tree motion.

Figure 6: Reference video for tree flow extraction.

4. Experiments and Discussions

4.1. Experimental Settings

We conducted experiments to generate a video from a single omnidirectional image. As input, we used an image captured with the 360° camera RICOH THETA Z1 and an image obtained from Google Street View, which were resized to a resolution of 1600 × 800. We used the image captured with the 360° camera as Case 1, and the image obtained from Google Street View as Case 2. In the experiments, we set the height of the planes representing the sky and water along the Z-axis to 2 and -2, respectively. We set the focal lengths f_x, f_y and image center c_x, c_y of the perspective projection image with a resolution of 384 × 384 to 192. We obtained the motion of the trees from the reference video as shown in Figure 6. The generated video consisted of 199 frames. Additionally, in Case 2, we compared the results with those obtained by directly applying the conventional method [10] to the equirectangular omnidirectional image. The following sections describe the experiments for Cases 1 and 2 in turn.

4.2. Experimental Results

4.2.1. Result of Case 1

Figure 7 shows the images of the 1st, 60th, 120th, and 180th frames of the video in equirectangular projection generated from the input omnidirectional image (1st

Frame 1

Frame 60

Frame 120

Frame 180

Figure 7: Results in equirectangular projection in Case 1.

Frame 1

Frame 60

Frame 120

Frame 180

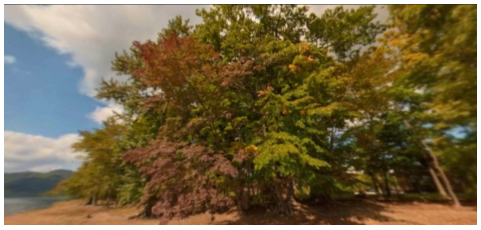
Figure 8: Results of sky and water in perspective projection in Case 1.

frame) by the proposed method in Case 1. Figures 8 and 9 show the result of converting these frames into perspective projection images in a specific direction. In this experiment, we convereted the omnidirectional image into the perspective projection image as shown in Figure 10(a). Figure 10(b) shows the the calculated optical flow at the 30th frame. From this flow map, we generated the flow maps of the water and sky planes as shown in Figures 10(c) and (d). In these figures, the angle of motion is represented by hue, the relative magnitude of the motion is represented by brightness, and the saturation is fixed at 1.

From these experimental results, we can observe that the sky moves naturally in the sky region, and we can also feel perspective because the clouds just above us moves faster than those in the distance. As for the water, we can see that the water surface moves in various directions, resulting in successfully representing waves. In the flow map of the water plane in Figure 10(c), we can observe various hues between green and yellow, and the brightness also varies, indicating that the complex motion of the water is well-represented. In contrast, the flow of the sky shows less variation in hue compared to the water, confirming that it moves in a mostly consistent direction. Regarding the trees, although the motion of the trees in the reference video is reflected in the omni-

Frame 1

Frame 60



Frame 120

Frame 180

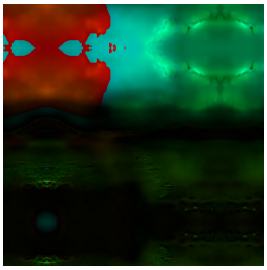
Figure 9: Results of trees in perspective Projection in Case 1.

directional video, if we look at the details, the branches and leaves move in blocks regardless of their positions, leading to slight unnaturalness. This is likely due to the mismatch between the positions of branches and leaves in the reference video for obtaining the optical flow and those in the target image.

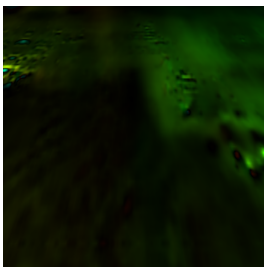
4.2.2. Result of Case2

Figure 11 shows the 1st, 60th, 120th, and 180th frames of the omnidirectional video in equirectangular projection generated from the input omnidirectional image (1st frame) by the proposed method in Case 2. Figure 12 shows the perspective projection image used for generating optical flow by the deep learning-based method [10].

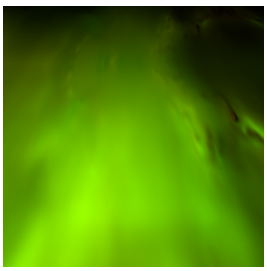
(a) Perspective projection image



(b) Flow of perspective projection image



(c) Flow of water plane



(d) Flow of sky plane

Figure 10: Flow visualisation results in Case 1.

In this scene, only the sky moves. Figure 13 shows the result of converting these frames into perspective projection images in a specific direction. From these results, we confirmed that all regions other than the sky remain static, and the clouds in the sky move uniformly in the same direction, indicating that a video could be successfully generated from a single omnidirectional image obtained from Google Street View.

Next, we compare the results with those obtained by directly applying the conventional method to the omnidirectional image. Figure 14 shows the 120th frame of the omnidirectional image in equirectangular projection generated by the conventional method and the result of converting it into a perspective projection image in a specific direction. As seen in Figure 14, the conventional method result has unnatural distortions of the ground and clouds, whereas the proposed method shows that both the ground and the clouds move without any unnatural distortion. These results demonstrate that the problems of moving static regions unnecessarily and forming unnatural motion have been resolved.

4.2.3. Discussion

In this section, in addition to the issues previously discussed, we examine other remaining issues in the system and their potential improvements.

First, there is a limitation in the accuracy of semantic segmentation, making it difficult to achieve perfect

Frame 1

Frame 60

Frame 120

Frame 180

Figure 11: Results in equirectangular projection in Case 2.

Figure 12: Perspective projection image in Case 2.

Frame 1

Frame 60

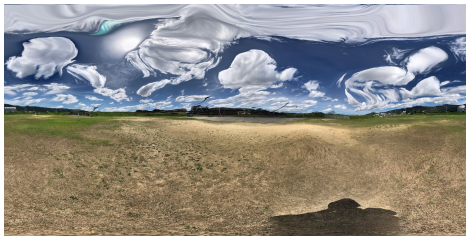
Frame 120

Frame 180

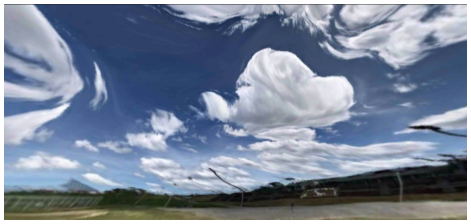
Figure 13: results in perspective projection in Case 2.

results. A specific example of this issue is illustrated in Figure 11, where the sun is also considered as part of the sky, making it move in the same way as the clouds. One solution is to extract the sun from the sky by developing a new semantic segmentation method and then to keep the sun in its original position.

Furthermore, while this study focuses on animating natural objects, many tourist spots also have moving man-made objects such as cars and flags. If these objects are not properly animated, the realism of the video is reduced. We should develop a method for animating man-made objects, further enhancing the realism of the video.



(a) Omnidirectional image



(b) Perspective projection

Figure 14: Results by directly applying the conventional method to omnidirectional image (Flame 120).

5. Conclusion

In this study, we proposed a method for generating videos with motion of natural objects from a single omnidirectional image by the combination of estimating optical flows using deep learning and considering the motion in 3D space for virtual sightseeing. Through experiments, we confirmed that the proposed method is effective. However, while the water and sky regions moved naturally, the tree regions still show some unnatural motion. In future work, we introduce deep learning for the motion of trees as well.

Acknowledgment

This research was partially supported by JSPS KAKENHI JP23K21689.

References

[1] V. T. Consortium, Townwrap, 2024. URL: https://townwarp.net/, last accessed: September 25, 2024.

[2] AirPano, Airpano, 2024. URL: https://www.airpano.com/, last accessed: September 25, 2024.

[3] C. Valero-Franco, A. Berns, A virtual reality app created with cospaces: Student perceptions and attitudes, in: Ethical Considerations of Virtual Reality in the College Classroom, 1st ed., Routledge, 2023, p. 16.

[4] Y. Suganuma, M. Oda, K. Nakayama, S. Nishikawa, S. Hata, K. Paul, S. Wada, N. Kawai, Integrated system of augmented and virtual reality for ruins tourism, in: Proceedings of NICOGRAPH International 2023, 2023, p. 85.

[5] Google, Google street view, 2024. URL: https://www.google.co.jp/maps, last accessed: July 17, 2024.

[6] J. Lambert, Z. Lie, O. Sener, J. Hays, V. Koltun, MSeg: A composite dataset for multi-domain semantic segmentation, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, 2020.

[7] Y.-Y. Chuang, D. B. Goldman, K. C. Zheng, B. Curless, D. Salesin, R. Szeliski, Animating pictures with stochastic motion textures, ACM Transactions on Graphics 24 (2005) 853–860.

[8] M. Okabe, K. Anjyor, T. Igarashi, H.-P., Animating pictures of fluid using video examples, Computer Graphics Forum 28 (2009) 677–686.

[9] M. Okabe, K. Anjyor, R. Onai, Creating fluid animation from a single image using video database, Computer Graphics Forum 30 (2011) 1973–1982.

[10] Y. Endo, Y. Kanamori, S. Kuriyama, Animating landscape: Self-supervised learning of decoupled motion and appearance for single-image video video synthesis, ACM Transactions on Graphics 38 (2019).

[11] M. Kakuho, H. Ikebayashi, N. Kawai, Motion reproduction of sky and water surface from an omnidirectional still image, in: Proceedings of IEEE Global Conference on Consumer Electronics, 2023, pp. 150–151.

[12] J. Y. ans Z. Lin, J. Yang, X. Shen, X. Lu, T. S. Huang, Free-form image inpainting with gated convolution, in: Proceedings of IEEE International Conference on Computer Vision, 2019.

[13] G. Farnebäck, Two-frame motion estimation based on polynomial expansion, in: Proceedings of Scandinavian Conference on Image Analysis (SCIA 2003), volume 2749, 2003.