Virtual Reality with Motion Parallax by Dense Optical Flow-based Depth Generation from Two Spherical Images

Sarthak Pathak, Alessandro Moro, Hiromitsu Fujii, Atsushi Yamashita, and Hajime Asama

***Abstract*— Virtual reality (VR) systems using head-mounted displays (HMDs) can render immersive views of environments, allowing change of viewpoint position and orientation. When there is a change in the position of the viewpoint, different objects in the scene undergo different displacements, depending on their depth. This is known as 'Motion Parallax' and is important for depth perception. It is easy to implement for computer-generated scenes. Spherical cameras like the Ricoh Theta S can capture an all-round view of the environment in a single image, making VR possible for real-world scenes as well. Spherical images contain information from all directions and allow all possible viewpoint orientations. However, implementing motion parallax for real-world scenes is tedious as accurate depth information is required, which is difficult to obtain. In this research, we propose a novel method to easily implement motion parallax for real world scenes by automatically estimating all-round depth from two arbitrary spherical images. The proposed method estimates dense optical flow between two images and decomposes it to the depth map. The depth map can be used to reproject the scene accurately to any desired position and orientation, allowing motion parallax.**

I. Introduction

Virtual reality (VR) aims to virtually insert the viewer into an environment by rendering accurate images of the environment using a head mounted display (HMD). Usually, the head position and orientation is tracked using inertial sensors of the HMD and the appropriate scene corresponding to the particular head position and orientation is rendered, creating an all-round immersive view. Computer generated VR can render accurate scenes at any head position and orientation. Changing the head orientation can allow the user to look in all directions. Meanwhile, changing the head position can allow the user to look at the scene from various viewpoints. Under a change of position, objects closer to the viewer undergo a larger displacement as compared to objects far away, as shown in Fig. 1. This is known as 'Motion parallax', and it forms an important component of VR systems as it can provide depth perception to viewers, alongside left-right eye stereo.

Recently developed spherical cameras like the Ricoh Theta S (shown in Fig. 2) can capture a full 360 degree spherical image in a single shot. They can extend the capabilities of VR to real-world environments. However, VR systems created from spherical images cannot render accurate information on a change in the head position. In order to so, it is necessary to have accurate 3D depth information of the environment. For computer-generated VR, this is easy to perform using the known 3D model of the artificial environment. But for real-world scenes, it is difficult to obtain 3D information. Hence, motion parallax is not easy to create. In the absence of such motion parallax, the user can become disoriented as the scene does not change as expected by the change of position [1], [2] also validated experimentally that motion parallax is required for perceiving depth, especially for real world scenes.

All authors are with the Department of Precision Engineering, Graduate School of Engineering, The University of Tokyo, Japan {pathak,moro,fujii,yamashita,asama}@robot.t.u-tokyo.ac.jp

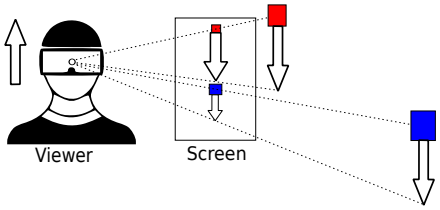


Fig. 1. Under motion parallax, objects away from the camera undergo lesser displacement. This is important for depth perception.

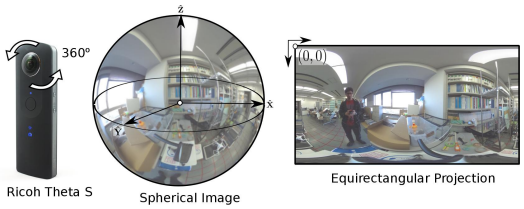


Fig. 2. Spherical cameras like the Ricoh Theta S, that can capture a fully spherical images in a single shot.

In this paper, we propose a new method to create a motion parallax effect by automatic generation of depth from two spherical images clicked at arbitrary, displaced positions in a given environment. The image capture can be done by hand at arbitrary positions. Our system can process the two arbitrary spherical images and refine them to a rectified stereo image pair and simultaneously estimate the depth map based on the dense optical flow between them. Once the depth map is estimated, virtual images at any position and orientation can be created through fast pixelwise interpolation, allowing head-motion parallax at real-time speeds.

II. Previous Research and Our Approach

Many methods attempt to resolve this problem by capturing depth information using stereo panoramic images [3]. The two camera positions can be calibrated using a checkerboard. In spherical images, corresponding pixels lie along epipolar circles, as explained in [4]. Once the camera positions are known, the depth map can be easily generated by

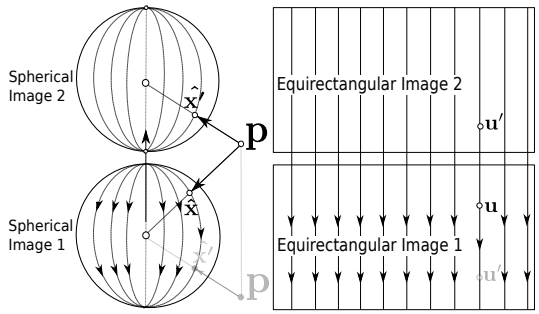


Fig. 3. If the cameras are displaced vertically, all pixel movements are in the vertical direction in the equirectangular projection.

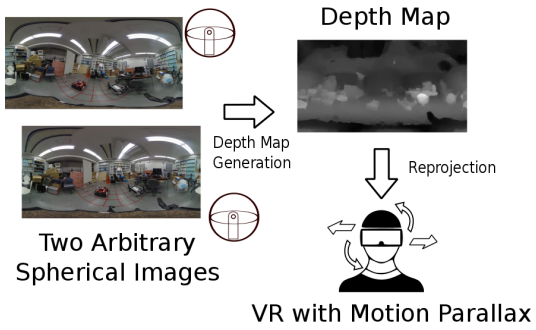


Fig. 4. Overview of the proposed approach

searching for corresponding pixels along the epipolar circles. However, this requires the use of more than one camera.

Instead, [5] attempted to displace a single spherical camera in the vertical direction. On expanding the spherical images to the equirectangular projection, all the epipolar circles on the sphere collapse to vertical lines in the equirectangular projection, as shown in Fig. 3. They were able to estimate the depth map easily by searching for corresponding pixels in the same vertical line. Using a similar concept, [6] attempt to use a single spherical camera displaced in the vertical direction. They perform depth map estimation using the known camera positions in a manner similar to [5]. The depth map is then used to generate virtual images at any head position, inducing head-motion parallax. In all these methods, the camera positions are restricted to allow easy capture of 3D information. Hence, these methods are not generalizable and depend on precise mechanical alignment of the camera, which is difficult to achieve.

[7] also used a vertically displaced orientation and used feature detection and matching to correct for the errors of this vertical displacement. However, feature detection and matching is not easy in distorted spherical images and there can be many mistakes. The presence of even a few mistakes can lead to a large loss of accuracy in sparse feature-based methods. Instead, [8] used dense optical flow to estimate the epipolar geometry of two perspective images. They obtained very accurate results based on the fact that dense optical flow algorithms try to estimate a smooth flow field, removing individual outliers. They can not only provide accurate camera geometry, but also provide the depth map in terms of the disparity of each pixel.

Thus, in this research, we use the vertical rectification scheme of [6] along with the optical flow based depth map estimation proposed in [8] in order to render virtual images for VR with head-motion parallax. The benefit and novelty of our method, unlike [6], is that it can process any two images clicked at arbitrary positions, without any user intervention.

III. Overview

As input, our approach uses two spherical images clicked in the desired environment with translation between them. A translational distance of atleast more than 1/20th of the average scene depth is required for accurate parallax. The images can be clicked by hand as well. The translation can be in any direction, but best results are obtained by translating approximately in the vertical direction, as the accuracy is the highest at all points located perpendicularly from the direction of translation.

Our system consists of two steps. One is the preprocessing step in which the two spherical images are rectified using an iterative procedure based on dense optical flow. We take advantage of the complete spherical field of view, i.e. the property that they can be rotated to any orientation without loss of information. The two arbitrary spherical images are rectified to an equirectangular stereo pair through multiple image rotations, via an iterative minimization based on the equirectangular optical flow field. This step simultaneously estimates the 5 DoF epipolar geometry and the dense depth map.

In the second step, this dense depth map is utilized to reproject one of the images, pixel-to-pixel, to a virtual image at any desired position and orientation. It can then be unwarped at any direction, providing the perspective image from that viewpoint. An overview of our approach is shown in Fig. 4.

IV. Optical Flow-based Depth Map Generation

The first step in our approach is to preprocess the two images and generate a depth map. We use an optical flow-based, iterative technique based on the vertical displacement approach used in [6], as shown in Fig. 3. Spherical images are not easy to store and handle in memory as they do not exist on a regularized grid. Instead of complicated spherical image processing, our method can naturally handle the distortion of spherical images on a planar equirectangular grid. We make use of dense pixel motions in an iterative minimization to rectify the two arbitrary images to this rectified arrangement, simultaneously estimating the epipolar geometry and the disparity.

Since spherical images have information from all directions, they can be rotated to any orientation without loss of information. It can be seen that an arbitrary spherical camera motion in space differs from this arrangement only by rotational transformations. On knowing the precise 5 DoF epipolar geometry, first the rotation between the two images can be corrected by aligning them to the same orientation.

Following this, they can both be rotated in order to align the translation vector in a vertical direction, as shown in Fig. 3. In this arrangement, the equirectangular images of both should have all pixel displacements aligned in a vertical direction - which can be checked using dense optical flow. Thus, we try to estimate the epipolar geometry that best fits the desired vertically oriented dense optical flow field. The pipeline is given below, as shown in Fig. 5.

Given two spherical images in the equirectangular format, I_1 and I_2, arbitrarily displaced by a translation vector $\mathbf{t}$ and rotation matrix $\mathbf{R}$, we first project each pixel (u, v) of each equirectangular image to its spherical unit vector $\hat{\mathbf{x}} = [x, y, z]^T$ to form their respective spherical images S_1 and S_2. Since the translation vector $\mathbf{t}$ only represents a direction, it can be represented in 2 DoF as (θ, ϕ). Meanwhile, the rotation matrix $\mathbf{R}$ is represented as three euler angles (α, β, γ). Thus, $G = (\alpha, \beta, \gamma, \theta, \phi)$ are given as the parameters to be optimized in the minimization (in order to implicitly enforce 5 DoF on the epipolar geometry). The images are rectified as follows: First, S_2 is derotated to the same orientation as S_1:

$$S_{2,1} = \mathbf{R}^{-1} \times S_2 = \Big(\mathbf{R}_x(\alpha)\mathbf{R}_y(\beta)\mathbf{R}_z(\gamma)\Big)^{-1} \times S_2, \quad (1)$$

where $S_{2,1}$ indicates S_2 in the same orientation as S_1, and $\mathbf{R}_x(\alpha)$, $\mathbf{R}_y(\beta)$, and $\mathbf{R}_z(\gamma)$ denote the individual rotation matrices along the x, y, and z axes. Following this, we compute $\mathbf{R}_v$, the rotation matrix that rectifies both images to a vertically displaced orientation. The translation vector $\mathbf{t}$ is expressed in cartesian coordinates from (θ, ϕ). The angle Ω between the translation vector $\mathbf{t}$ and the vector $\mathbf{n} = [0, 0, 1]^T$, which is the vertical direction, is calculated as:

$$\Omega = \arccos\left(\frac{\mathbf{t} \cdot \mathbf{n}}{|\mathbf{t}||\mathbf{n}|}\right). \quad (2)$$

The axis of rotation $\mathbf{a}$ is the cross product of $\mathbf{t}$ and $\mathbf{n}$:

$$\mathbf{a} = \mathbf{t} \times \mathbf{n}. \quad (3)$$

Thus, $\mathbf{R}_v$ can be written as a rotation matrix of angle Ω around the axis $\mathbf{a}$:

$$\mathbf{R}_v = \mathbf{R}_{\mathbf{a}}(\Omega). \quad (4)$$

Finally, both images are rotated by $\mathbf{R}_v$ to the rectified orientations, $S_{1,r}$ and $S_{2,r}$ and thereafter expanded to the equirectangular projection for refinement.

$$S_{1,r} = \mathbf{R}_v \times S_1, \quad (5)$$

$$S_{2,r} = \mathbf{R}_v \times S_{2,1}. \quad (6)$$

All rotations are done pixel-wise between equirectangular images after projecting each pixel (u, v) to its spherical unit vector $\hat{\mathbf{x}} = [x, y, z]^T$, with bilinear interpolation to fill the gaps due to discretization of pixels. Thus, I_1 and I_2 are converted to two rectified equirectangular images $I_{1,r}$ and $I_{2,r}$ via multiple rotations. We denote the equirectangular pixel coordinate system in the rectified state as (u_r, v_r). The dense optical flow field $\mathbf{f}(u_r, v_r)$ is computed between $I_{1,r}$ and $I_{2,r}$. In order to rectify the two images, the following minimization is defined on the horizontal component of the optical flow field: $f_u(u_r, v_r)$.

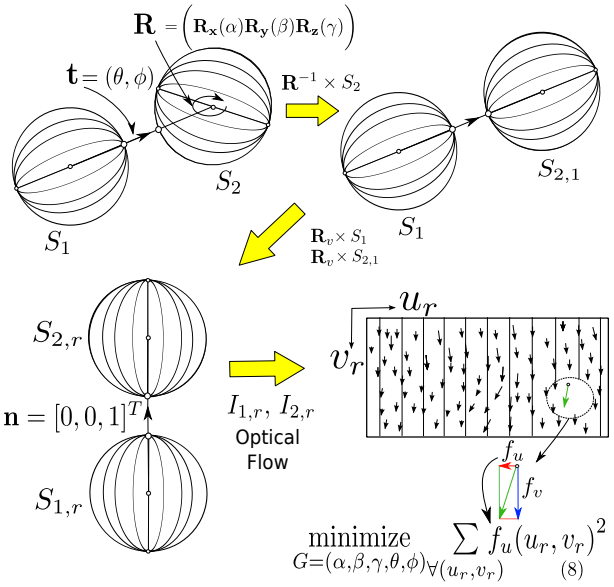


Fig. 5. Rectifying a pair of spherical images with arbitrary motion

$$\underset{G=(\alpha,\beta,\gamma,\theta,\phi)}{\text{minimize}} \sum_{\forall(u_r,v_r)} f_u(u_r, v_r)^2. \quad (7)$$

This minimization can be solved using the Levenberg-Marquardt approach [9] as a non-linear least squares problems, as is commonly used for most dense image-based approaches. Since an initial value is required, we use A-KAZE [10] feature points to initialize the epipolar geometry estimate, as described in [11]. In this research, we used the Deepflow [12] algorithm to compute the dense optical flow between $I_{1,r}$ and $I_{2,r}$. Since the dense optical flow needs to be computed in each iteration, it could become very time-consuming. In order to avoid this, we simply reproject the dense optical flow state in every iteration based on the first computation. Since the images in successive iterations are only related by spherical rotational transformations, it is easy to track pixel movements.

At the end of the minimization, all dense pixel motions are oriented vertically. We denote this vertically oriented optical flow field as $\mathbf{f}_v(u_r, v_r)$. Its magnitude component $|\mathbf{f}_v(u_r, v_r)|$ is nothing but the disparity of pixels and directly forms the depth map of the environment. Thus, in this manner, the dense optical flow pattern between the two spherical images can be rectified to satisfy the vertical alignment of Fig. 3 and hence be decomposed to the depth map of the environment. Using this depth map, 3D positions of all pixels can easily be triangulated and then reprojected to any desired position and orientation as described in the next section.

V. Virtual Image Generation by Reprojection

Generation of the depth map takes around 100s for equirectangular images of resolution 1000 x 500 pixels as

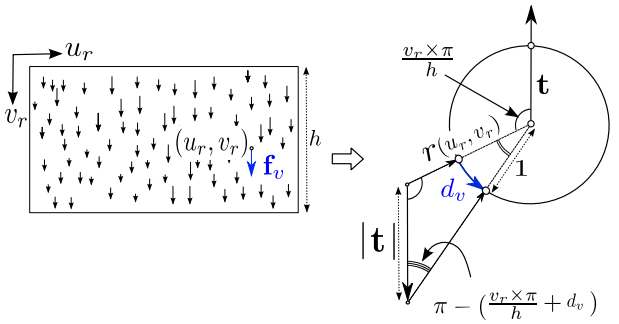


Fig. 6. Estimating the 3D position of each pixel from the depth map

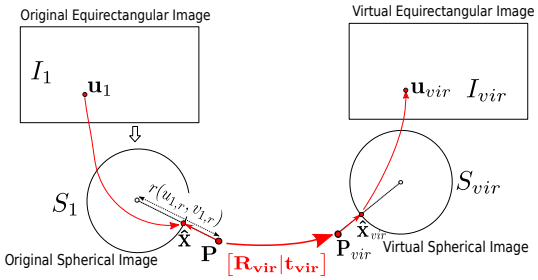


Fig. 7. Pixelwise Reprojection Pipeline from I_1 to I_{vir}

it is a time-consuming, iterative process. However, it is intended as a preprocessing step. Once the depth map is estimated, virtual images can quickly be generated through a fast pixelwise transform. We reproject the depth map to render virtual spherical images at any position and orientation $\mathbf{t}_{vir}$ and $\mathbf{R}_{vir}$.

First, we estimate the 3D location $\mathbf{P}$ of each pixel $\mathbf{u}_1$ on I_1. In the rectified image $I_{1,r}$, the radius $r(u_r, v_r)$ of each pixel can be calculated from its disparity $|\mathbf{f}_v|$ as follows.

The angular distance of (u_r, v_r) from the topmost point of the sphere (i.e. the epipolar point in the rectified alignment) is $= \frac{v_r \times \pi}{h}$. The magnitude of $\mathbf{f}_v$ on the equirectangular image is a difference of latitudes on the sphere, and can be converted to the angular disparity on the sphere as follows:

$$d_v = \frac{|\mathbf{f}_v| \times \pi}{h}, \tag{8}$$

where h is the height of the equirectangular image.

As seen from Fig. 6, the law of sines gives the radius r of 3D point $\mathbf{P}$:

$$r(u_r, v_r) = |\mathbf{t}| \times \frac{\sin(\frac{v_r \times \pi}{h} + d_v)}{\sin(d_v)}. \tag{9}$$

The translation magnitude $|\mathbf{t}|$ can be set as 1 without loss of generality. Following this, each pixel on the rectified orientation, $r(u_r, v_r)$ is multiplied with its spherical unit vector $\hat{\mathbf{x}}$ to give its 3D location $\mathbf{P}$ and rotated back by ${\mathbf{R}_v}^T$ to its original orientation, to that of $I_1(u, v)$.

Now, we have $\mathbf{P}(u, v)$, the 3D locations of every pixel in I_1. Our objective is to reproject these 3D coordinates to a virtual image I_{vir} at position and orientation $\mathbf{t}_{vir}$ and $\mathbf{R}_{vir}$. We transform $\mathbf{P}(u, v)$ by the transformation matrix $[\mathbf{R_{vir}}|\mathbf{t_{vir}}]$.

$$\mathbf{P}_{vir} = [\mathbf{R}_{vir}|\mathbf{t}_{vir}] \times \mathbf{P}. \tag{10}$$

Then, the point $\mathbf{P}_{vir}$ is reduced to its spherical unit vector $\hat{\mathbf{x}}_{vir}$ as seen from the virtual image as follows:

$$\hat{\mathbf{x}}_{vir} = \frac{\mathbf{P}_{vir}}{|\mathbf{P}_{vir}|}. \tag{11}$$

Finally, $S_{vir}(\hat{\mathbf{x}}_{vir})$ is converted to the desired virtual equirectangular image $I_{vir}(\mathbf{u}_{vir})$. Thus, a pixelwise mapping from I_1 to I_{vir} is developed:

$$I_1(\mathbf{u}_1) \rightarrow I_{vir}(\mathbf{u}_{vir}). \tag{12}$$

If this pixel-wise transform is performed directly, it can create gaps in the virtual image as some pixels of I_1 will be stretched and some will be contracted. A perfect mapping of all pixels from I_1 to I_{vir} cannot exist due to the distortions present in them. Hence, this transformation is implemented as an image warping that incorporates interpolation in order to fill the gaps. The entire pipeline of this pixelwise reprojection is shown in Fig. 7.

Thus, using our method, virtual equirectangular images at any position and orientation can be created. This allows not only a change of orientation, but also the position of the viewed image. The use of the estimated depth information allows accurate rendering of the scene from any position, allowing motion parallax, as demonstrated in the next section.

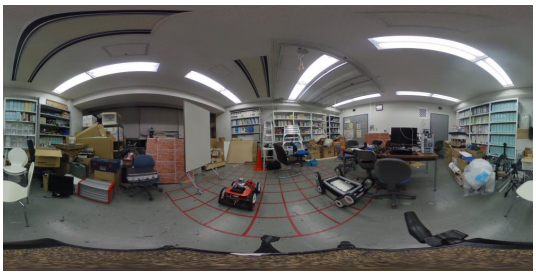
VI. Experiment

In order to demonstrate the effect of the proposed motion parallax technique, an experiment was conducted in a cluttered room. The Ricoh Theta S spherical camera was used to capture two spherical images, displaced along an arbitrary, approximately vertical direction. The two images are shown below in Fig. 8.

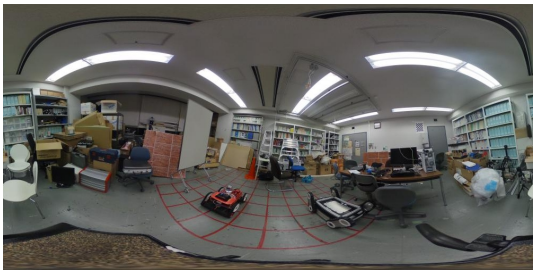
The dense optical flow between the images was estimated using DeepFlow [12] and rectified to generate the depth map. The initial and rectified states of the dense optical flow are shown in Fig. 9 and the resultant depth map is shown in Fig. 10.

Virtual images were generated under a number of translations and rotations and a small area was unwarped to form a perspective video, to be rendered on a VR display, as shown in Fig. 11. Some frames from the resultant video[1] are shown below in Fig. 12. Specifically, three sequences of frames showing leftward, upward, and forward camera translation are shown in top-to-bottom sequences. In addition, a split-channel overlay combining all three frames is shown at the bottom of each sequence to clearly visualize the motion parallax. It can be noticed that the bookshelf away from the camera undergoes a much lesser shift as compared to objects closer to the camera, located near the edges. The results show an accurate, natural looking motion parallax.

[1]Please refer to the video attachment for a clearer demonstration of the motion parallax effect. Also available at: https://youtu.be/6Vqe0mBg0BA



(a) Image 1



(b) Image 2

Fig. 8. Two spherical images were captured in a cluttered room, displaced along an arbitrary, approximately vertical direction

(a) Initial State

(b) Rectified State

Fig. 9. (a) Initial state of the dense optical flow between the two images and (b), the final, rectified state showing vertically aligned optical flow vectors

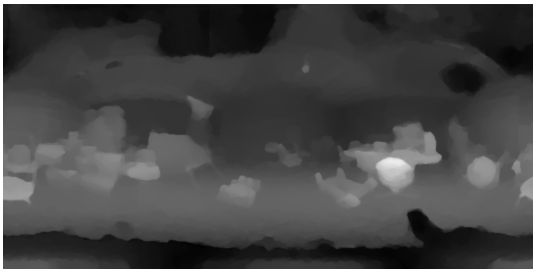


Fig. 10. Resultant depth map, generated from rectifying the dense optical flow

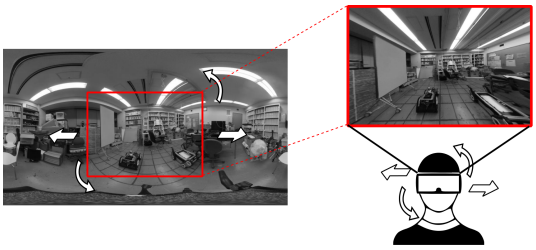


Fig. 11. Virtual images were generated under a number of translations and rotations and a small area was unwarped to a perspective view to form a perspective video.

The reprojection was possible at 5 frames per second for equirectangular images of resolution 1000 x 500 pixels on a 2.8 GHz CPU, without any parallel processing.

VII. Discussion and Conclusion

In this research, the aim was to augment VR systems based on spherical images with a motion parallax feature. This was done by estimating depth from two spherical images captured in the desired environment. Unlike previous methods, our method can generate a depth map under any arbitrary camera motion by decomposing and rectifying the optical flow field between them. The depth map of the environment was extracted from the optical flow and used to reproject one of the images to any desired position and orientation. This provided the necessary depth information to achieve a motion parallax effect. Experimental results in a cluttered environment showed accurate and natural-looking motion parallax using only two arbitrary spherical images and no additional information.

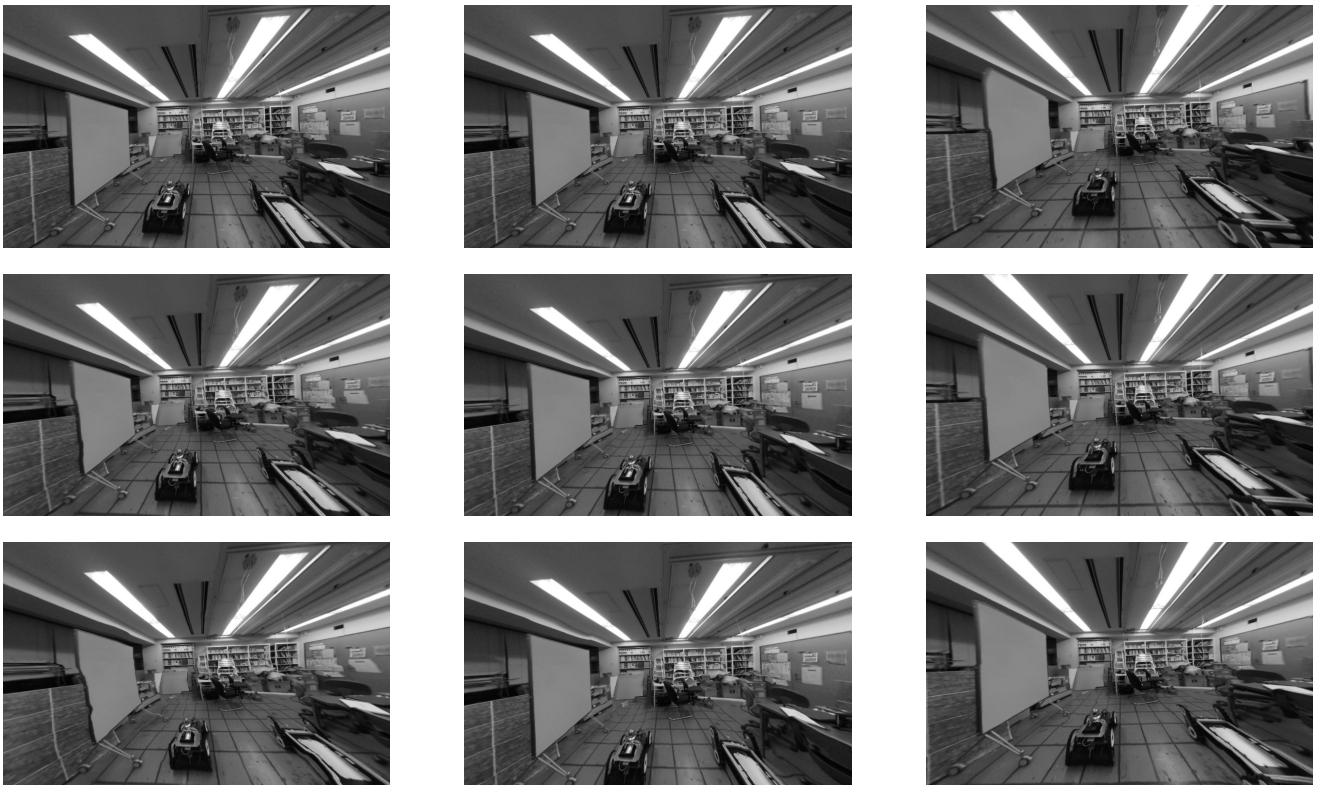
However, one drawback of our method is that the real-world scale cannot be estimated and must be manually adjusted to give natural-looking parallax. Moreover, it must be noted that while the range of possible orientations is unlimited, the set of possible positions for an accurate, natural motion parallax is limited to be close to the original positions of the input images. This is because if the camera moves too far away, regions that were occluded in the two images can come into view. Such regions cannot be rendered properly as the information from those regions is unknown. In order to solve this, multi-view depth map estimation is necessary, which will be our future work.

Acknowledgement

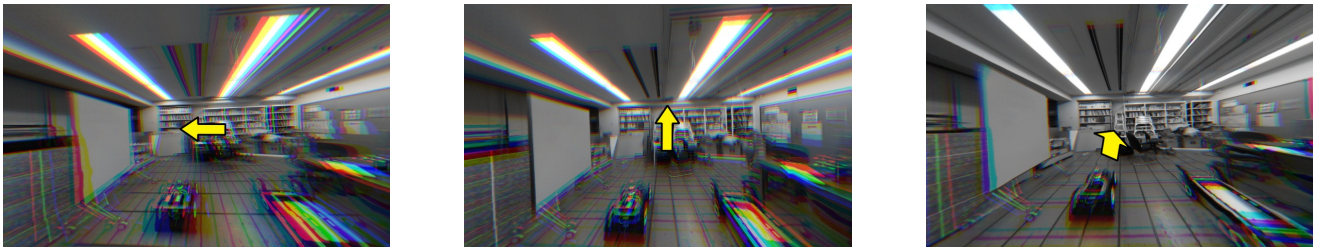
This work was in part supported by the Council for Science, Technology and Innovation, "Cross-ministerial Strategic Innovation Promotion Program (SIP), Infrastructure Maintenance, Renovation, and Management" (funding agency: NEDO).

References

[1] J. A. Jones, J. E. Swan, II, G. Singh, E. Kolstad, and S. R. Ellis, "The effects of virtual reality, augmented reality, and motion parallax on egocentric depth perception," in *Proceedings of the 5th Symposium on Applied Perception in Graphics and Visualization*, August 2008, pp. 9–14.



Leftward camera translation | Upward camera translation | Forward camera translation



Split-channel overlay for leftward translation | Split-channel overlay for upward translation | Split-channel overlay for forward translation

Fig. 12. Selected frames showing examples of different types of camera motion in three top-to-bottom sequences for leftward, upward, and forward camera translation. Towards the bottom, the three images have been shown in an overlay of different color channels to show the motion parallax effect. It can be noticed that the objects near the edges which are closer to the camera are displaced more in the appropriate direction for each kind of translation.

[2] B. Rogers and M. Graham, "Motion parallax as an independent cue for depth perception," *Perception*, vol. 8, no. 2, pp. 125–134, April 1979.

[3] C. Richardt, Y. Pritch, H. Zimmer, and A. Sorkine-Hornung, "Megastereo: Constructing high-resolution stereo panoramas," in *Proceedings of the 2013 IEEE Conference on Computer Vision and Pattern Recognition*, June 2013, pp. 1256–1263.

[4] A. Torii, A. Imiya, and N. Ohnishi, "Two-and three-view geometry for spherical cameras," in *Proceedings of the sixth workshop on omnidirectional vision, camera networks and non-classical cameras*, October 2005, pp. 81–88.

[5] H. Kim and A. Hilton, "3d scene reconstruction from multiple spherical stereo pairs," *International Journal of Computer Vision*, vol. 104, no. 1, pp. 94–116, August 2013.

[6] J. Thatte, J. B. Boin, H. Lakshman, and B. Girod, "Depth augmented stereo panorama for cinematic virtual reality with head-motion parallax," in *2016 IEEE International Conference on Multimedia and Expo (ICME)*, July 2016, pp. 1–6.

[7] A. Banno and K. Ikeuchi, "Omnidirectional texturing based on robust 3d registration through euclidean reconstruction from two spherical images," *Computer Vision and Image Understanding*, vol. 114, no. 4, pp. 491 – 499, April 2010.

[8] L. Valgaerts, A. Bruhn, M. Mainberger, and J. Weickert, "Dense versus sparse approaches for estimating the fundamental matrix," *International Journal of Computer Vision*, vol. 96, pp. 212–234, January 2012.

[9] M. Lourakis, "levmar: Levenberg-marquardt nonlinear least squares algorithms in C/C++," [web page] `http://www.ics.forth.gr/~lourakis/levmar/`, July 2004.

[10] P. Alcantarilla, J. Nuevo, and A. Bartoli, "Fast explicit diffusion for accelerated features in nonlinear scale spaces," in *Proceedings of the British Machine Vision Conference*, September 2013, pp. 13.1–13.11.

[11] A. Pagani and D. Stricker, "Structure from motion using full spherical panoramic cameras," in *Proceedings of the IEEE International Conference on Computer Vision (Workshops)*, November 2011, pp. 375–382.

[12] P. Weinzaepfel, J. Revaud, Z. Harchaoui, and C. Schmid, "DeepFlow: Large displacement optical flow with deep matching," in *Proceedings of the IEEE International Conference on Computer Vision*, December 2013, pp. 1385 – 1392.