

Active 3D Motion Visualization Based on Spatiotemporal Light-Ray Integration

Fumihiko Sakaue Jun Sato
Nagoya Institute of Technology
Gokiso Showa Nagoya, 466-8555, Japan
{sakaue, junsato}@nitech.ac.jp

Abstract

In this paper, we propose a method of visualizing 3D motion with zero latency. This method achieves motion visualization by projecting special high-frequency light patterns on moving objects without using any feedback mechanisms. For this objective, we focus on the time integration of light rays in the sensing system of observers. It is known that the visual system of human observers integrates light rays in a certain period. Similarly, the image sensor in a camera integrates light rays during the exposure time. Thus, our method embeds multiple images into a time-varying light field, such that the observer of the time-varying light field observes completely different images according to the dynamic motion of the scene. Based on this concept, we propose a method of generating special high-frequency patterns of projector lights. After projection onto target objects with projectors, the image observed on the target changes automatically depending on the motion of the objects and without any scene sensing and data analysis. In other words, we achieve motion visualization without the time delay incurred during sensing and computing.

1. Introduction

In computer vision, 3D motion estimation has a long history, and many efficient methods have been proposed under various conditions [17, 4, 13, 14, 6]. Standard methods first extract point correspondences and optic flows in sequential images [7, 2, 10, 3]. The extracted point correspondences and optic flows are then used to compute 3D motion and the structures of the scene [17, 4, 14]. Active sensors, such as 3D range sensors, are also used to measure the 3D distance, which is then used to compute 3D motion [15]. The direct motion estimation method is also proposed based on the Doppler shift of the reflected light on the moving object [6].

Although 3D motion estimation methods have become very advanced in recent years, all existing methods have an essential and unavoidable problem caused by their system

structure: an inevitable time delay in measurements from real dynamic scenes. Indeed, the estimated 3D motions are not the current motions, but rather the past motions in the scene.

In all existing methods, 3D motion is estimated in two steps: image data or 3D data is first obtained using cameras or 3D sensors; and 3D motion is then computed from the observed data based on changes in the observations during a specific time interval. As a result, these existing methods require a certain amount of time to estimate 3D motion, because they need to obtain at least two observations by sensors at different instants in time before computing the motion based on these multiple observations. Furthermore, estimating 3D motion becomes unstable with a shorter sampling interval of observations. This is because 3D motion is computed from the change in observations, and the signal-to-noise ratio of the change in observation degrades drastically in small intervals of observations. Hence, we cannot shorten this interval for stable motion estimations. Observation and computational costs are consequently never zero, even when expensive sensors and powerful computers are used.

Observation and computation delays are severe problems in real-time computer vision applications, such as driver assistance systems for vehicles [16]. In these real-time systems, delayed motion estimations impose a delay on the driver's decisions and actions, risking serious accidents in traffic environments.

In this paper, we propose a novel method of visualizing the dynamic information in a scene, by projecting images with projectors without using any sensors. With the proposed method, the appearance of target objects changes drastically according to their dynamic motion, as shown in Fig. 1. Furthermore, the proposed method can visualize motion information using complex images—for example, when an angry face appears on a forward-moving surface, whereas a smiling face appears on a backward-moving surface. All the processes can be done merely by projecting lights from projectors. As a result, the proposed method can achieve 3D motion estimation and visualization exclusively



Figure 1. Motion visualization from image projection: the observed image (color in this figure) on a target surface changes spontaneously according to its motions without using any sensor-feedback system such as a camera or 3D sensor. In this example, the projected pattern (the string) changes according to the relative motion of the vehicle.

by projecting lights and without any time delay.

For this objective, we focus on the time integration of light rays in the sensing system of observers. It is known that the visual system of human observers integrates light rays in a certain period. Similarly, the image sensor in a camera integrates light rays during the exposure time. Thus, our method embeds multiple images into a time-varying light field, such that the observer of the time-varying light field observes completely different images according to the dynamic motion of the scene.

The proposed method is a new framework for 3D motion estimation and representation. As such, it can be implemented for various applications. For example, to control vehicle headlights using the hardware described in [16], we can visualize the relative speed of other vehicles and determine the danger of collision without latency, merely by projecting lights from the vehicles headlights.

To our knowledge, this is the first paper to achieve motion visualization with zero latency, and we believe that this paper opens a new research field for 3D motion estimation in computer vision.

2. Related Works

Our method is closely related to coded light projection and light-field displays. In order to measure the 3D shape, coded light projections (i.e., structured light projections) have been studied for decades [1]. More recently, the first version of the Kinect sensor [18] used spatially coded lights to identify the corresponding points between projected lights and observed image points and to recover the 3D structure of a scene. Coded light projections have also been used for other objectives. Nayer et al. proposed a method of separating direct and global components using spatial high-frequency illumination [12]. The temporal coding of projected lights was also used for 3D measurements, among other applications [11]. Although many methods have been developed with coded light projections, coded lights are designed to be observed by sensors and analyzed only subsequently. Therefore, existing methods with coded light projections incur the computational cost to obtain fi-

nal results. By contrast, the proposed method visualizes 3D motion without any computation, and hence without any computational delay. Thus, our framework is completely different from that of existing 3D measurements from coded light projections.

Light-field displays have also been studied extensively in recent years [9]. The light field is the subspace of a 7D plenoptic function. The plenoptic function is 7D space, which consists of 3D position, 2D orientation, 1D wavelength, and 1D time. In most cases, however, the light field is considered 4D space—2D position and 2D orientation—assuming that there no degradation to the light power in light travel and neglecting variations to the wavelength and time [9, 19, 8]. Thus, light-field displays are typically considered 4D devices. Wetzstein et al. [19] proposed a method of organizing a 3D display using multiple layered 2D display panels. Huang et al. [8] used a 4D light-field display to correct the visual aberrations of observers, showing deblurred images to near- or far-sighted observers without corrective lenses. Because existing light-field display techniques consider only the spatial position and orientation of lights, these techniques cannot encode visual information in the time domain. By contrast, we here consider light-ray integration in the time domain. We show that it is possible to encode visual information into the time domain of the light field and that the encoded visual information can be decoded by object motion. Thus, we consider the 3D motion of an object as a decoder of coded light.

3. Observation Model

We first consider an intensity observation model of observers such as humans and cameras. The image observed by them can be considered to be a process of sensing the light rays in the light field of the scene.

Let $E(x, y, t)$ be a light ray from a point (x, y) at time t toward an orientation of an observer. If the observer observes the light ray at a particular moment, the ray—i.e., the light ray $E(x, y, t)$ —is observed directly. However, general observers, e.g., humans and cameras, do not observe a moment of light rays, but rather only integrated light-rays in the time domain. Indeed, the effect of the integration appears as a blurred motion when an observed object is in motion. Thus, to observe dynamic scenes, we need to consider integration with respect to the time domain as follows:

$$I(x, y, T_o) = \int_{T_o-T}^{T_o} E(x, y, t) \quad (1)$$

where T_o is the current time and T is the exposure time. In ordinary observers, such as cameras and humans, time integration of the observed light rays occurs in general. For example, although fluorescent lights blink with high frequency, humans do not perceive this. Similarly, video pro-

jectors with a micro-mirror array, such as digital light processing (DLP) projectors, can represent varying brightness and colors by using the flicker of light.

Here, Eq.(1) has an ambiguity in E . That is, the input light E that satisfies Eq.(1) is not unique. In other words, an observed intensity $I(x, y, T_o)$ is identical even when the input light E changes, provided that Eq. (1) is satisfied. This is an important property of observed intensity in this paper. In the proposed method, we embed multiple images into the ambiguity of the observed intensity, such that these images appear adaptively according to the motion of objects.

4. Motion Visualization Based on Light-Ray Integration

We next consider image embedding based on the observation model described in the previous section. We first consider the geometric relationship between a projector and an observer to derive a method of embedding images. In this paper, we assume that the relative position between the projector and the observer is fixed. Then, we consider the epipolar geometry between the projector and the observer.

4.1. Epipolar Geometry for Projectors and Observers

Let \mathbf{P} and \mathbf{P}' be projection matrices of a projector and an observer, respectively. Suppose a 3D point \mathbf{X} in the scene is projected to a point \mathbf{x} in a projected image and is observed as \mathbf{x}' by an observer such as a camera. Then, the relationship among them can be described as follows:

$$\lambda \tilde{\mathbf{x}} = \mathbf{P} \tilde{\mathbf{X}} \quad (2)$$

$$\lambda' \tilde{\mathbf{x}}' = \mathbf{P}' \tilde{\mathbf{X}} \quad (3)$$

where (\cdot) denotes the homogeneous representation of a vector. Then, a geometric constraint, the so-called epipolar constraint[5], holds between \mathbf{x} and \mathbf{x}' as follows:

$$\tilde{\mathbf{x}}'^T \mathbf{F} \tilde{\mathbf{x}} = 0 \quad (4)$$

where \mathbf{F} denotes a 3×3 fundamental matrix whose rank is 2.

The epipolar constraint stipulates that a pair of corresponding points, \mathbf{x} and \mathbf{x}' , lie on epipolar lines \mathbf{l} and \mathbf{l}' , respectively, as shown in Fig. 2. These epipolar lines are computed as follows:

$$\mathbf{l}' = \mathbf{F} \mathbf{x} \quad (5)$$

$$\mathbf{l} = \mathbf{F}^T \mathbf{x}' \quad (6)$$

This epipolar constraint shows that the observer observes a 3D point, which is illuminated by a pixel on the epipolar line \mathbf{l} corresponding to \mathbf{x}' on the epipolar line \mathbf{l}' . This means that we only need to consider corresponding epipolar lines to analyze the relationship between the projector and the observer. Therefore, we consider the following derivation exclusively on the epipolar lines.

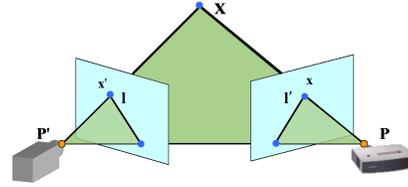


Figure 2. Epipolar plane and epipolar lines: the epipolar plane is defined by a 3D point \mathbf{X} and the optical centers of an observer and a projector. The epipolar lines are the intersections of image planes and the epipolar plane.

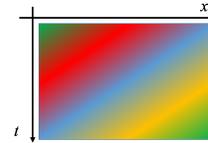
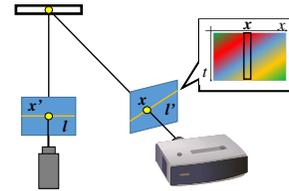
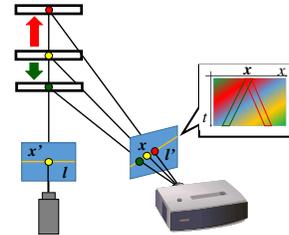


Figure 3. Spatiotemporal image projected by a projector. The horizontal axis shows the epipolar line and the vertical axis shows time.



(a) No motion



(b) Forward motion and backward motion

Figure 4. Change in integration in the spatiotemporal image: the black box in (a), blue box and red box in (b) show regions integrated in the observed intensity under each object motion.

4.2. Motion Visualization by Light-Ray Integration

We next consider the changes of observation when a target object moves. As described in the previous section, at point \mathbf{x}' , an observer observes pixels on a corresponding epipolar line \mathbf{l} . In addition, epipolar lines in an image do not intersect with each other except at the epipole. Therefore, we consider only an epipolar line in this section.

Let us now consider the case where the projector projects dynamic images. If we focus on an epipolar line, these dynamic images can be regarded as a spatiotemporal image, as shown in Fig.3. In this figure, the horizontal axis shows the epipolar line and the vertical axis shows changes over

time. In the spatiotemporal image, we consider the change in observation caused by the motion of the target object.

Let us consider the case where a projector illuminates a planar screen and observed by a camera, as shown in Fig.4. The exposure time of the camera is equivalent to the time length of the spatiotemporal image. We first consider an observation where the planar target screen does not move, as shown in Fig.4(a). In this case, a corresponding pixel on an epipolar line does not change, and thus the observer observes the same point. Consequently, the observed image is the integration of the black box in the figure.

We next consider the case where the planar screen moves forward, as shown in Fig.4(b). In this case, the corresponding point on the epipolar line moves to the left. Thus, the integration of the spatiotemporal image also changes to the blue box in this figure. Therefore, the observed image changes drastically from that in the case without motion. Similarly, if the planar screen moves backward, the corresponding point moves to the right, and thus the integration of spatiotemporal image changes to the red box. As a result, the observed image also changes.

These three examples show that the observation changes drastically according to the target motion. That is, object motion can be visualized using the proposed method. Note that although a planar surface is used for this discussion, there is no constraint on the object shape, insofar as the method is based on the change in depth of an observed target.

4.3. Spatiotemporal Image Observation

We next consider observational details in order to control the observed image under object motion.

Let us consider the case where a 3D point \mathbf{X} is illuminated by a point \mathbf{x} on the projected image, and the 3D point \mathbf{X} is observed at a point \mathbf{x}' on the camera image. When the 3D point \mathbf{X} moves toward the optical center of the camera with speed v , the observed point \mathbf{x}' does not change, whereas the illumination point \mathbf{x}_t on the projected image at time t changes as follows:

$$\mathbf{x}_t = \mathbf{x} + \alpha(vt)\mathbf{d} \quad (7)$$

where \mathbf{d} is a unit vector in the direction of the epipolar line, and α is a map from the change in depth to the change in point \mathbf{x} . On the epipolar line, this can be rewritten in 1D notation as follows:

$$x_t = x + \alpha(vt) \quad (8)$$

Assuming that the function $\alpha(x)$ is linear, the equation can be rewritten as follows:

$$x_t = x + t\alpha(v) \quad (9)$$

Let $E(x, t)$ be the illumination value at point x and time t . If the exposure time of the observer is T , the observed

intensity $I(x')$ at x' on the camera image can be computed as follows:

$$I(x') = \int_0^T E(x + t\alpha(v), t) dt \quad (10)$$

This continuous notation can be approximated by discrete notation as follows:

$$I(x') = \sum_{t=0}^{T-1} E(x + t\alpha(v), t) \quad (11)$$

Here, Eq.(11) indicates that the observed intensity changes according to the motion v of the object. That is, the observed image for each motion can be controlled by the spatiotemporal image.

Note that the function α describes changes of disparity by the change of depth. Although disparity is inversely proportional to the depth and it is nonlinear, it can be approximated by a linear function, in a small range.

4.4. Projection Pattern Estimation

We next consider a method of estimating projection patterns for motion visualization.

Let us consider the case where we want to show an objective image $\hat{\mathbf{I}}_1$ to the observer when the target object moves with speed v_1 . Let $\hat{I}_1(x)$ be the intensity of $\hat{\mathbf{I}}_1$ at point x . Then, the projected image for presenting this objective image can be obtained by minimizing the following evaluation value ϵ_1 :

$$\epsilon_1 = \sum_{x=1}^N \left(\hat{I}_1(x) - \sum_{t=0}^{T-1} E(x + t\alpha(v_1), t) \right)^2 \quad (12)$$

where N is the number of pixels on the epipolar line.

This evaluation can be minimized with an ordinary least-means square (LMS) method. However, projectors cannot project negative intensities in general, and they have limited intensity:

$$0 \leq I(x) \leq I_{\max} \quad (13)$$

where I_{\max} is the maximum intensity value. Conditional LMS can solve this equation. In addition, we degrade the contrast of the objective image. By this degradation, the range of the projection intensity virtually enhanced relative to the objective image.

We next consider the case where the target object moves with M different speeds $v_i (i = 1, \dots, M)$, and the observer observes M different images $\hat{\mathbf{I}}_i (i = 1, \dots, M)$ according to the motion. The projected images for such an observation can be derived by minimizing the following ϵ :

$$\epsilon = \sum_{i=1}^M \sum_{x=1}^N \left(\hat{I}_i(x) - \sum_{t=0}^{T-1} E(x + t\alpha(v_i), t) \right)^2 \quad (14)$$



(a) in a vehicle (b) Environment

Figure 5. Experimental environment in outdoor scene. (a) Camera and projector in a vehicle and (b) experimental scene.

Note that the number of samples T of exposure should be more than M , because the equations do not have an explicit solution if $T < M$. The number of samples can be increased using a high-frequency projector. By projecting the derived spatiotemporal patterns, different motion can be visualized with varying patterns of image.

5. Experimental Results

5.1. Results in Outdoor Scene

In this section, we show the efficiency of the proposed method by providing results from real image experiments. We first show the experimental results from an outdoor scene. In this experiment, a camera and a projector were equipped on a vehicle, as shown in Fig.5. The relative position between the camera and the projector were fixed in the vehicle. Unfortunately, the projector could project only 60 images each second. Therefore, we decreased the frames per second (fps) of the camera, thus virtually increasing the fps of the projector. The camera observed five images each second. Therefore, 12 images were integrated into each observation.

The projector pattern was generated, such that the observer could see the three different images shown in Fig.6 according to the motion of the vehicle: Image (a) for a static scene, Image (b) for forwarding motion, and Image (c) for backward motion. We degraded the contrast of each image to enhance the image representation ability since the projectable range of intensity was limited by Eq.(13). We derived projected images using the proposed method. These images were transformed by projective transformations, such that their horizontal axes were parallel to the epipolar line with the camera. After the transformation, the projected images were observed by the camera. The vehicle moved forward and backward in front of a wall, and the camera observed images projected onto the wall. The wall had slight 3D structure as shown in Fig.5(b) and Fig.8

Figure8 shows the results observed by the camera. In these results, three different images were observed corresponding to the vehicles motion. The results indicate that our proposed method can visualize 3D motion without any sensing. In addition, the results indicate that our method is



(a) Static (b) Forward (c) Backward

Figure 6. Objective images for (a) a static scene, (b) forward motion, and (c) backward motion.

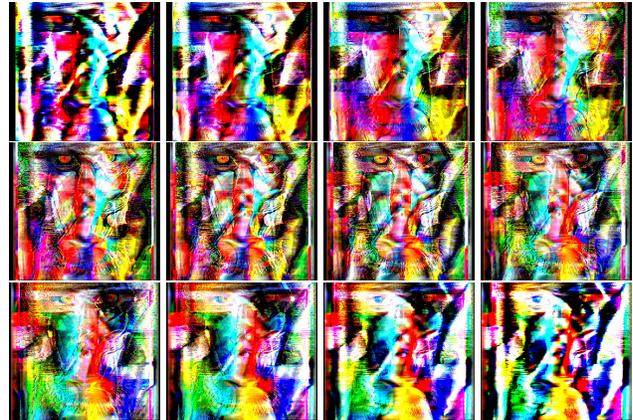


Figure 7. Projected images generated by our proposed method.



(a) Static scene (b) Forward motion (c) Backward motion

Figure 8. Images observed by a camera: (a) an observed image when the vehicle stopped, (b) an observed image when the vehicle moved forward, and (c) an image observed with backward motion.

robust to changes in speed, insofar as the speed of the vehicle could not be controlled accurately. Furthermore, the scene included various types of noisy light rays from street lights, buildings, and so on. Despite this noise, our method worked well. This further indicates that our method is robust to noise. All the results indicate that our method can visualize relative 3D motion in the real world without any sensing or computing.

5.2. Results in Indoor Scene

We next show the experimental results for an indoor scene. In this experiment, a camera and a projector were placed, as shown in Fig.9. The camera was used exclusively as an observer. The projector projected dynamic patterns generated by the proposed method. The planar was placed

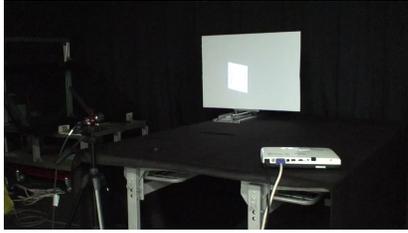


Figure 9. Experimental environment



(a) Static (b) Forward (c) Backward

Figure 10. Objective images for (a) static object, (b) forward motion, and (c) backward motion.



(a) Static scene (b) Forward motion (c) Backward motion

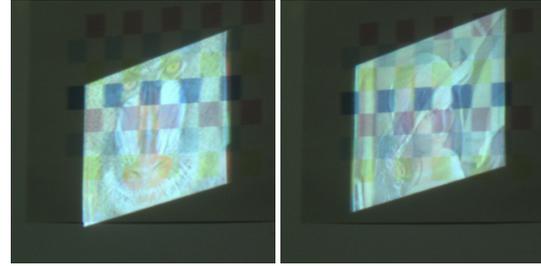
Figure 11. Observed images by a camera: (a) an observed image when the screen is static, (b) an observed image when the screen moved forward, and (c) an image observed with backward motion.

on a moving stage and moved forward and backward iteratively. The speed of the motion was approximately 1 cm/s. The camera and the projector were fixed in the scene and only the screen was moved.

The projector pattern was generated, such that the camera could see the three different images shown in Fig.6 according to the motion of the object: Image (a) for static object, Image (b) for forward motion, and Image (c) for backward motion.

The observed results under three different types of object motion are shown in Fig.11. Although the observed images differ slightly from the objective images, we find that the proposed method provides us with entirely different images for forwarding and backward motion and that the difference in these motions is visible from the images. In addition, the camera could observe almost the same image even when the camera position slightly moved. The fact indicates that our proposed method is robust against the movement of the camera and the projector. The detail of this result is in the supplemental material.

We next show the result when a target screen is more



(a) Forward motion (b) Backward motion

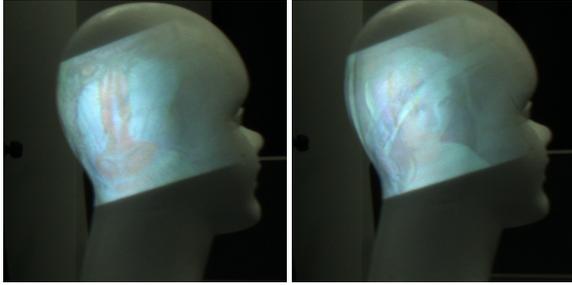
Figure 12. Observed images on the colored screen: (a) and (b) are the images observed when the screen moved forward and backward, respectively.

complicated. First, we projected the proposed patterns onto a colored screen, as shown in Fig.12. The color patterns were printed on the screen, and our proposed patterns were projected onto the screen. In this case, reflected light rays were attenuated by the albedo on the screen, such that the observed patterns could be changed accordingly. Figure 12 shows the observed result when the screen moved (a) forward and (b) backward. In these results, the color pattern on the screen and the projected images were simultaneously observed. This is because the number of integrated pixels for each observed pixels was not so large. Consequently, the observed pixels did not change drastically, even when the albedo of the screen was not white. Note that the observed results differ from the target image because the observed images include not only the projected patterns but also the images on the screen. However, human eyes can perceive these projected images, even when the screen is not white, owing to high adaptation abilities. Thus, our proposal can appropriately project images even when target objects are colored.

In addition, colored patterns on the screen were slightly blurred because the target screen moved slowly. However, projected patterns could be clearly observed because the patterns were computed for clear observations when the screen moved. Therefore, our proposed method clearly observes images even when the target screen moves.

We next show the case when a target object is not planar, as shown in Fig. 13. In this case, reflected light rays are attenuated by changes in the angle between the light ray and the surface. Figure 13 shows observed results with forwarding motion and backward motion. In these results, we can observe the target images on the screen object for the same reasons as those mentioned above. Although the non-planar screen distorted the observed images, the target images could be recognized appropriately. This indicates that our method can achieve motion visualization, even if the target object is not planar.

Because these results can be obtained merely by projecting images toward moving objects, the proposed method is



(a) Forward motion (b) Backward motion

Figure 13. Observed images on the non-planar surface: (a) and (b) are the images observed when the screen moved forward and backward, respectively.

efficient at visualizing object motion without latency.

6. Evaluation

6.1. Speed Change

We evaluated the accuracy of our proposed method. For a quantitative evaluation, we used a synthetic environment. In the synthetic environment, we simulated a projector, a camera, and a planar object, as shown in Fig.15. The planar object was moved toward the camera at various speeds. The epipolar lines in the projected image and the observed image were parallel to the horizontal axis of the images, and the image point in the observed image moved by 1 pixel when the target object moved approximately 1 mm toward the camera.

We first examined the change of view along with the target object. As shown in Eq. (14), our proposed method essentially visualizes discrete 3D motion. However, general objects move at continuous speeds in the real world. Therefore, we examined how the projected image was observed when the target object moved at speeds different from the target speed. In this experiment, three images were used as objective images, as shown in Fig.11: backward (-5 mm/s), static, and forward (5 mm/s) motion. These were observed at several speeds differing from the target speed. Figure 14 shows the observed images at each speed.

The results show that clear images could be observed when the target object moved at the same speed as the target speed. In addition, images similar to the objective images could be observed even when the speed of the target differed slightly from the target speed. This fact indicates that our proposed method is robust to changes in the target speed. Furthermore, the observation results gradually changed when the speed of the target changed. For example, a morphed image of parrots and Lena was observed when the target moved by 0.6 cm/s. This indicates that our proposed method can represent not only discrete 3D motion but also continuous motion. That is, users will perceive morphed motion from a morphed observed image.

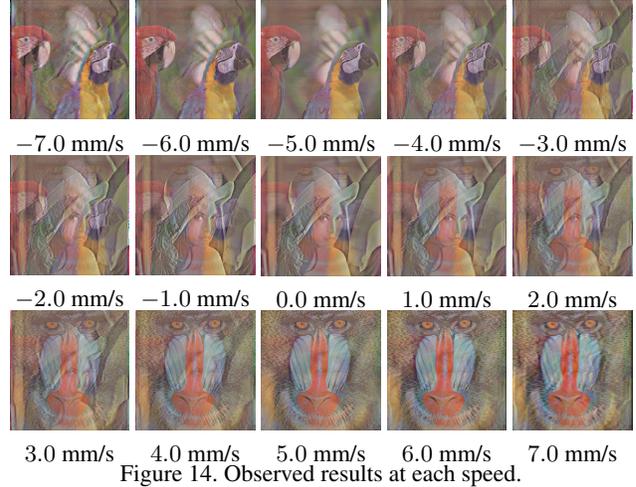


Figure 14. Observed results at each speed.

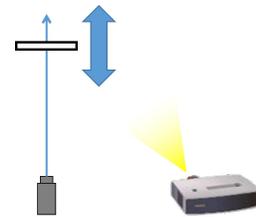


Figure 15. Synthetic environment.

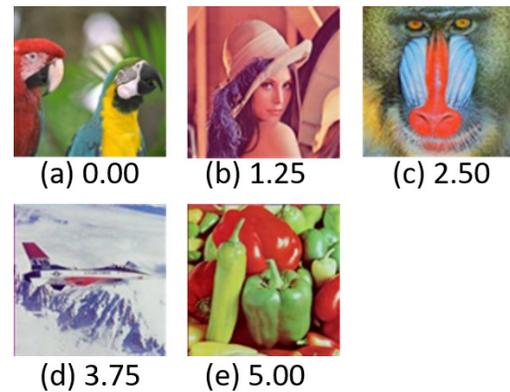


Figure 16. Objective images of five different motion speeds.

6.2. Resolution

We next evaluated the resolution of visualized speed. In this experiment, the speed of the object motion was changed from 0 mm/s to 5 mm/s at an interval of 1.25 mm/s. Thus, five different motion speeds were estimated by the proposed method. The objective images for these five motion speeds are shown in Fig. 16. The projected images for visualizing this motion were generated and projected by the proposed method. Figure 17 shows the observed images under these five different motion speeds. In the environment, variations to observed images depended on the target speed evaluated.

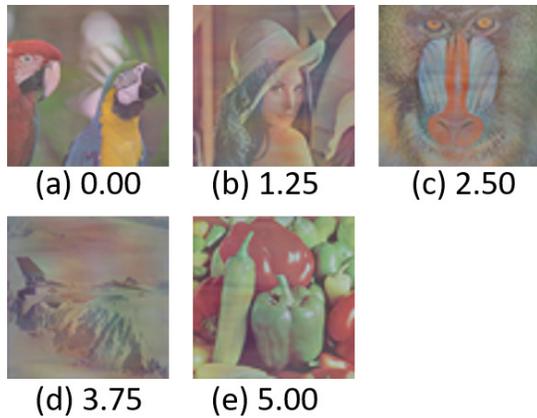


Figure 17. Observed images with five different motion speeds.

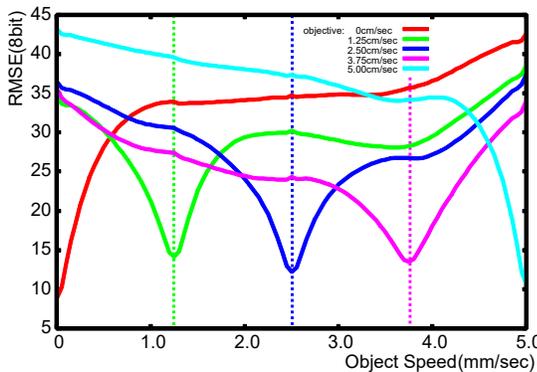


Figure 18. Errors between the observed image and the objective image at five different speeds. The horizontal axis shows the speed of the object, and each line shows the root-mean-square error between the observed image and each objective image under different object speeds.

Figure 18 shows the error of the observed image with respect to each objective image. For example, the green line shows the root-mean-square error (RMSE) between the observed image and the objective image at 1.25 mm/s. It required minimum speed of 1.25 mm/s, as we expected. From this graph, we find that the RMSE of true motion is very small compared to those of other motions. Thus, the proposed method can visualize several types of motion.

6.3. Frame Rate

Finally, we evaluated the relationship between the frame rate of the projector and the accuracy of motion representation. The frame rate of the projector was varied from 5 fps to 50 fps, while the frame rate of the observer was fixed at 1 fps. The number of motions distinguished by the proposed method was also changed, from two to five. Under these conditions, the RMSE of the observed image with respect to the objective image was evaluated. Figure 19 shows the RMSE of the observed images. From this figure, we find that the accuracy of motion representation improves as the

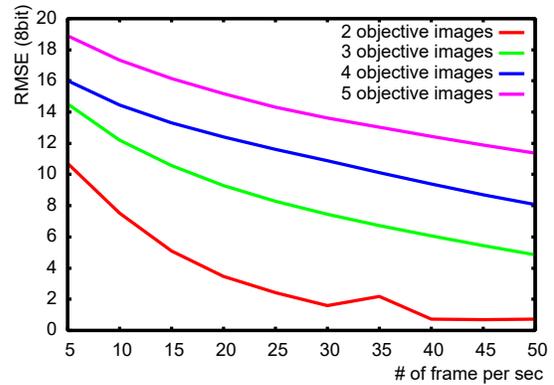


Figure 19. Relationship between the frame rate of the projector and the accuracy of motion representation. The horizontal axis shows the frame rate of the projector, and the vertical axis shows the root-mean-square error of the observed images. The number of motions distinguished by the proposed method was varied from two to five.

frame rate of the projector increases. We also find that the accuracy of motion representation degrades as the number of motions increases. However, the accuracy of motion representation can be recovered by increasing the frame rate of the projector, even with a large number of motions. Thus, it is important to use high-frequency projectors to represent several types of object motion with the proposed method.

7. Conclusions

In this paper, we proposed a novel method of visualizing object motion using image projection. The proposed method does not require any sensing devices, such as cameras, and does not require any computation. With the proposed method, the appearance of objects changes according to their motion, and feedback from sensors is unnecessary. Consequently, there is no delay when visualizing real object motion. The proposed method is also robust: it does not require sensing information, it is free from observation errors, and it is free from the problem of incorrect correspondences.

These features do not exist in conventional motion estimation methods, and we believe that this paper opens a new research field for 3D motion estimation in computer vision. Especially, our proposed concept that changes observed images based on light integration without any sensing and computation is much useful. This concept can visualize observer motion as well as the movement of the targets. This concept has various applications, for example, the spontaneously changing signboards according to the observer's motion.

References

- [1] K.L. Boyer and A.C. KAK. Color-encoded structured light for rapid active ranging. *IEEE transactions on Pattern Analysis and Machine Intelligence*, 9(1):14–28, 1987.
- [2] T. Brox and J. Malik. Large displacement optical flow: descriptor matching in variational motion estimation. *IEEE transactions on Pattern Analysis and Machine Intelligence*, 33(3):500–513, 2011.
- [3] A. Dosovitskiy, P. Fischer, E. Ilg, P. Hausser, C. Hazirbas, V. Golkov, P. Smagt, D. Cremers, and T. Brox. FlowNet: Learning optical flow with convolutional networks. In *Proc. IEEE International Conference on Computer Vision*, pages 2758–2766, 2015.
- [4] O.D. Faugeras. What can be seen in three dimensions with an uncalibrated stereo rig? In *Proc. European Conference on Computer Vision*, pages 563–578, 1992.
- [5] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2000.
- [6] Felix Heide, Wolfgang Heidrich, Matthias Hullin, and Gordon Wetzstein. Doppler time-of-flight imaging. *ACM Trans. Graph.*, 34(4):36:1–36:11, 2015.
- [7] B.K.P. Horn and B.G. Schunck. Determining optical flow. *Artificial Intelligence*, 17:185–203, 1981.
- [8] F. Huang, G. Wetzstein, B.A. Barsky, and R. Raskar. Eyeglasses-free display: Towards correcting visual aberrations with computational light field displays. *ACM Transaction on Graphics*, 33(4), 2014.
- [9] M. Levoy and P. Hanrahan. Light field rendering. In *Proc. SIGGRAPH*, pages 31–42, 1996.
- [10] C. Liu, J. Yuen, A. Torralba, J. Sivic, and W.T. Freeman. Sift flow: Dense correspondence across different scenes. In *Proc. European Conference on Computer Vision*, volume 3, pages 28–42, 2008.
- [11] S.G. Narasimhan, S.J. Koppal, and S. Yamazaki. Temporal dithering of illumination for fast active vision. In *Proc. European Conference on Computer Vision*, pages 830–844, 2008.
- [12] S.K. Nayar, G. Krishnan, M.D. Grossberg, and R. Raskar. Fast separation of direct and global components of a scene using high frequency illumination. In *Proc. SIGGRAPH*, pages 935–944, 2006.
- [13] M. Pollefeys, M. Vergauwen, K. Cornelis, J. Tops, F. Verbiest, and L. Van Gool. Structure and motion from image sequences. In *Proc. Conference on Optical 3-D Measurement Techniques*, pages 251–258, 2001.
- [14] A.J. Davison R.A. Newcombe, S.J. Lovegrove. Dtam: Dense tracking and mapping in real-time. In *IEEE International Conference on Computer Vision*, 2011.
- [15] J. Shotton, A. Fitzgibbon, M. Cook, T. Sharp, M. Finocchio, R. Moore, A. Kipman, and A. Blake. Real-time human pose recognition in parts from single depth images. In *IEEE Conference on Computer Vision and Pattern Analysis*, 2011.
- [16] Robert Tamburo, Eriko Nurvitadhi, Abhishek Chugh, Mei Chen, Anthony Rowe, Takeo Kanade, and Srinivasa G. Narasimhan. Programmable automotive headlights. In *Proc. ECCV2014*, pages 750–765, 2014.
- [17] C. Tomasi and T. Kanade. Shape and motion from image streams under orthography: a factorization method. *International Journal of Computer Vision*, 9, 1992.
- [18] J. Webb and J. Ashley. *Beginning Kinect Programming with the Microsoft Kinect SDK*. Apress, 2012.
- [19] G. Wetzstein, D. Lanman, W. Heidrich, and R. Raskar. Layered 3d: Tomographic image synthesis for attenuation-based light field and high dynamic range displays. In *Proc. SIGGRAPH*, 2011.