IPSJ Transactions on Computer Vision and Applications Vol.6 25–33 (June 2014)

[DOI: 10.2197/ipsjtcva.6.25]

Research Paper

Motion-invariant Coding Using a Programmable Aperture Camera

Toshiki Sonoda[1,a)] Hajime Nagahara[1,b)] Rin-ichiro Taniguchi[1,c)]

Received: October 22, 2013, Accepted: April 3, 2014, Released: June 24, 2014

Abstract: A fundamental problem in conventional photography is that movement of the camera or captured object causes motion blur in the image. In this research, we propose coding motion-invariant blur using a programmable aperture camera. The camera realizes virtual camera motion by translating the opening, and as a result, we obtain a coded image in which motion blur is invariant with respect to object velocity. Therefore, we can reduce motion blur without having to estimate motion blur kernels or requiring knowledge of the object speed. We model a projection of the programmable aperture camera and also demonstrate that our proposed coding works using a prototype camera.

Keywords: computational photography, coded aperture, image reconstruction, motion-invariant imaging

1. Introduction

Motion blur in an image is the result of either camera shake or object motion in a scene. When an object moves or the camera shakes during an exposure, the captured image contains blur caused by these motions, since the obtained image is a superimposition of the different positions of the objects at the different times. As a result, we obtain an unclear image, which has lost the high frequency component of object texture. Since motion blur is undesirable in any regular photograph, this paper aims to address the problem.

Various methods have been proposed for dealing with motion blur. The simplest solution is short exposure imaging. A camera has a shutter in front of an imager, and in short exposure imaging, the imager is exposed for only a short time when the shutter is opened to capture an image. If the exposure time is short, we can ignore any object motion in the scene and avoid motion blur. However, there is an unavoidable trade-off between motion blur and the signal-to-noise ratio (SNR) of the image since a shorter exposure darkens the captured image.

Another solution is lens or sensor shifting, which has been implemented in some modern cameras to stabilize the image. Here a mechanical actuator is controlled to shift a lens or sensor in real time during the exposure to compensate for motion of the camera [1]. This system is applicable to motion blur caused by camera motion only and not to that resulting from object motion.

An approach that restores a clear scene through deconvolution has been proposed in image processing [2]. However, the blur kernels for deconvolution vary according to the object motion and it is difficult to estimate the kernels or motions. To deal with this problem, various methods have attempted to estimate the point spread functions (PSFs) and restore a sharp image from a single input image [3], [4], [5] or multiple input images [6], [7], [8]. However, since a typical motion blur kernel contains many zero-crossings in the Fourier domain, the kernel loses image information and the deconvolution becomes ill-conditioned. To address this issue, several attempts have been made in the field of computational photography to control the motion-blur PSF using special optics or hardware so that the PSF estimation and motion deblurring can be handled easily.

Raskar et al. [9] proposed a fluttering shutter method that modifies the motion-blur PSF to achieve a broader-band frequency response and avoid zero-crossings. Although the method does stabilize the deconvolution results, there is still a requirement for precise knowledge of motion segmentation and object velocities. Agrawal et al. [10] improved the fluttering shutter to estimate object motion or the deconvolution kernel robustly by modifying the shuttering pattern. There is, however, an intuitive disadvantage in terms of the image SNR when employing these shuttering methods, since half the incoming light is blocked in engineering the PSF.

Levin et al. [11] proposed parabolic-motion coding of the camera. The parabolic motion of a camera makes the motion blur invariant to the object speed and the kernel has a broadband frequency in the Fourier domain. This was the first proposal for motion-invariant photography. The method has an advantage in terms of the image SNR since it uses the camera motion to engineer the motion-blur PSF and the shutter is completely open during the exposure. However, the invariance and broadband properties of motion blur are only applicable to one-dimensional horizontal motion. In addition, this method requires a mechanical mechanism, such as the use of cams and gears, to implement the parabolic camera motion. This should be avoided because of practical implementation difficulties and the limitation on motion speed restricted by the inertia of the element.

[1] Kyushu University, Fukuoka 819–0395, Japan
[a)] sonoda@limu.ait.kyushu-u.ac.jp
[b)] nagahara@ait.kyushu-u.ac.jp
[c)] rin@ait.kyushu-u.ac.jp

McCloskey et al. [12] proposed an implementation that achieves motion-invariant photography using lens shifting. This method achieves motion-invariant photography more practically than using camera body motion. Cho et al. [13] extended Levin's parabolic-motion coding to two-dimensional object motion. Similarly, Bando et al. [14] extended the coding using a circular camera motion. Although these methods engineer a broadband frequency response of the PSFs, they require motion estimation since motion invariance is not realized, contrary to the method using Levin's parabolic motion.

In this paper, we propose a novel method that achieves motion-invariant PSF coding using a programmable aperture camera [15], [16]. The camera can change its aperture pattern at high speed using a liquid crystal on silicon device. The camera achieves virtual camera motion by translating patterns of an aperture opening. Hence, we realize Levin's motion-invariant photography without using a mechanical mechanism as in the original method in which the camera body or some of the elements are moved. Our method improves practicability and utility with respect to implementation of the coding. On the other hand, some limitations are present that do not exist in the previous methods. We model the projective geometry of the programmable aperture camera, the virtual camera motion, and the generated PSFs. We investigate the parameter settings for the aperture pattern and optical parameters of the camera through simulation experiments. Finally, we confirm that the proposed method realizes motion-invariant photography using a prototype camera in experiments.

This journal version extends our work which appeared on Ref. [17] with more comparisons via extensive simulations and experiments.

2. Motion-invariant Photography Using a Programmable Aperture Camera

2.1 Modeling Motion Blur

In a conventional photograph, objects moving at different speeds cause varying degrees of motion blur with different shapes and lengths. To remove such motion blur, we must estimate the speed of each object. To address this problem, Levin et al. [11] realized motion-invariant photography that makes the PSF invariant to motion through the use of parabolic camera motion during an exposure. Because of this invariance, we can remove all blur for all moving objects by deconvolution using a single PSF. In Levin's work [11], the obtained PSF that is invariant to motion is expressed as:

$$\phi(x) = \frac{\lambda(x)}{2T\sqrt{s_i^2 - 2a_i x}}, \tag{1}$$

$$\lambda(x) = \begin{cases} 2, & \frac{s_i^2}{2a_i} \le x < s_i T - a_i T^2, \\ 1, & s_i T - a_i T^2 < x < s_i T + a_i T^2, \\ 0, & \text{otherwise}, \end{cases}$$

assuming that the image has acceleration a_i derived from camera motion and object velocity s_i derived from object motion. Both a_i and s_i are described in the image space. Here x is the position in the image and $2T$ is the exposure time. These assumptions are expressed as:

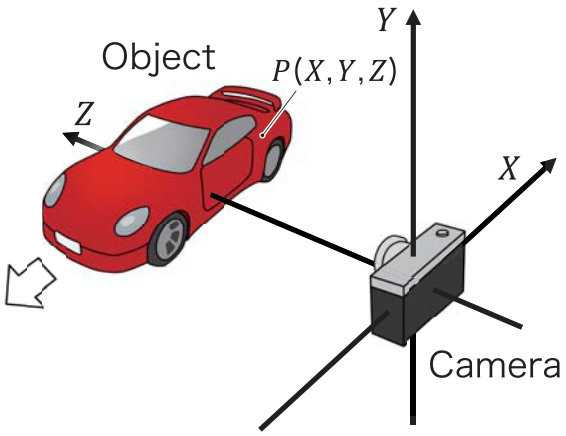


Fig. 1 Coordinate system of a camera.

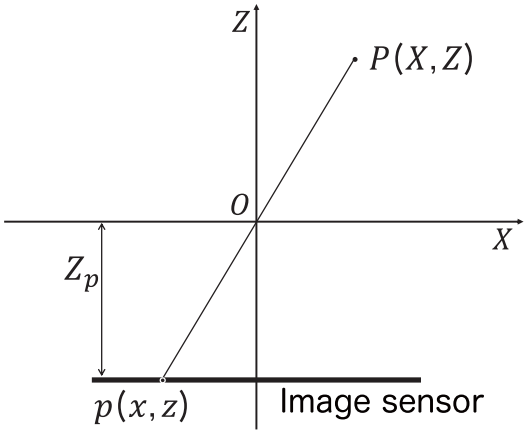


Fig. 2 Projective geometry of a normal camera.

$$x(t) = s_i t + \frac{a_i t^2}{2}. \tag{2}$$

For detailed derivations of Eqs. (1) and (2) the reader is referred to Ref. [11]. In this paper, we extend the concept of motion-invariant photography to be realized by a coded aperture.

Figure 1 shows the assumed coordinate system for modeling motion blur. The principal point of the lens is at the origin of the coordinate system, while the optical axis coincides with the Z-axis. The camera moves along the X-axis of the coordinate system during exposure, while an object point, denoted as $P(X, Y, Z)$, also moves along the X-axis. Point $P(X, Y, Z)$ is projected onto image point $p(x, y)$. **Figure 2** shows an X-Z slice of the projection for simplicity. $P(X, Z)$ is projected onto $p(x)$ on the imager plane ($Z = -Z_p$) by a pinhole camera model. This can be expressed as

$$x = \alpha X, \quad \alpha = \frac{-Z_p}{Z}, \tag{3}$$

where $P(X, Z)$ is a point on a moving object in a scene. We assume that the point moves parallel to the x-axis with the position expressed as $X(t)$. Similarly, if the camera moves parallel to the x-axis and the position is expressed as $X_c(t)$, the position of the projective point $p(x)$ on the image space relative to P can be expressed as

$$x(t) = \alpha(X(t) - X_c(t)). \tag{4}$$

It is shown that distance on the image space corresponds to that

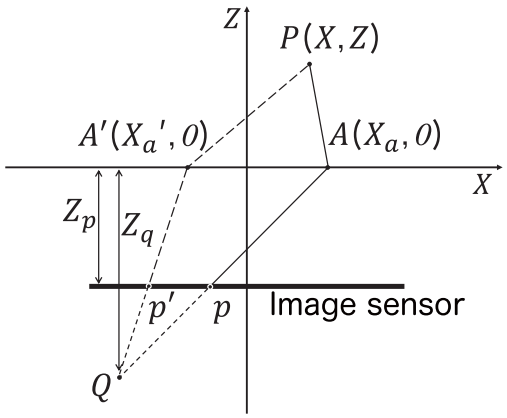


Fig. 3 Virtual camera motion model for a programmable aperture.

in the real scene with coefficient α from Eq. (4). Thus we obtain the following relations:

$$a_c = \frac{1}{\alpha} a_i, \tag{5}$$

$$s_o = \frac{1}{\alpha} s_i, \tag{6}$$

where the camera acceleration and object velocity in the real scene are denoted by a_c and s_o, respectively.

The center of the camera aperture is also the center of the projection in the projective geometry, and a projective change can thus be realized by motion of this aperture position. In this research, we realize the virtual camera motion needed for motion-invariant photography by temporally changing the aperture patterns. **Figure 3** shows the projective geometry of the programmable aperture camera. It is described with a similar form to that of Fig. 2. The geometry is on the X-Z space. An objective point in the scene is $P(X, Z)$, and it is projected to the image plane ($Z = -Z_p$) as $p(x)$. The aperture is assumed to lie on the plane $Z = 0$. The lens focal distance is denoted as f. The distance Z_p between the lens plane and focal point Q can then be expressed as

$$\frac{1}{f} = \frac{1}{Z} + \frac{1}{Z_q}. \tag{7}$$

By setting the position of the pinhole aperture as $A(X_a, 0)$, as shown in Fig. 3, the ray radiating from P goes towards the focal point Q through lens refraction via A and is projected onto point p on the image. The relation of the projection of projective point p can be modeled as

$$x(t) = \alpha X(t) - \beta X_a(t), \tag{8}$$

$$\beta = Z_p \left(\frac{1}{f} - \frac{1}{Z} - \frac{1}{Z_p} \right). \tag{9}$$

From Eqs. (4) and (8) it is found that the motion of the camera differs from that of the aperture, whereas the motion is the same as the object motion in the projection of the programmable aperture camera. Since it is also shown that the distance in the image corresponds to that in the real scene with coefficients α and β, we obtain the following relation:

$$a_a = \frac{1}{\beta} a_i. \tag{10}$$

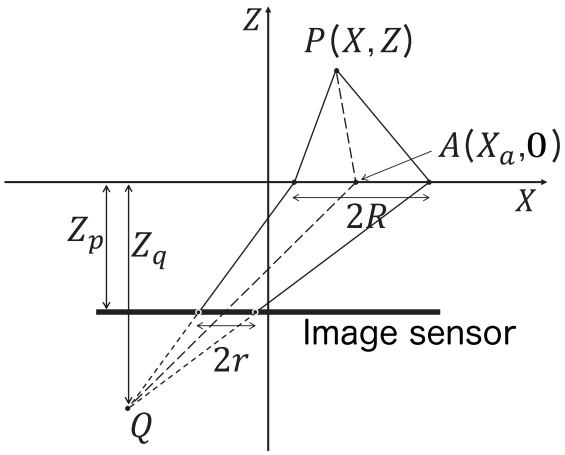


Fig. 4 Relation between the programmable aperture and static PSF.

Setting the aperture motion to a constant acceleration as in the equation below obtained from Eqs. (4) and (8), we can represent motion-invariant photography with aperture motion.

$$X_a(t) = \frac{\alpha}{\beta} X_c(t) = \frac{\alpha}{\beta} a_c t^2 \tag{11}$$

Since the proposed method imitates the camera motion in Levin's work [11], the limitation that the object motion must have constant acceleration and one-dimensional horizontal motion corresponding to the camera motion is also the same as in that work.

2.2 Relation between Aperture and Static PSF Size

An actual camera aperture is an opening of finite radius $R > 0$ and not a pinhole like that depicted in Fig. 3. Radius r generated by the static PSF on the image space is proportional to R, as shown in **Fig. 4**. Radius r of the static PSF projected onto projective point p on the image can be modeled as

$$\begin{aligned} r &= -Z_p \left(\frac{1}{f} - \frac{1}{Z} - \frac{1}{Z_p} \right) R \\ &= -\beta R. \end{aligned} \tag{12}$$

If coefficient β is zero, there is no parallax. Therefore, we must accept depth blur in generating a parallax to make $|\beta| > 0$, since the static PSF is no longer an impulse function ($r > 0$). The generated static PSF can be modeled as a pillbox function with radius r,

$$\psi(x, y) = \begin{cases} \frac{1}{\pi r^2}, & x^2 + y^2 \le r^2, \\ 0, & \text{otherwise.} \end{cases} \tag{13}$$

The motion blur caused by our proposed method is modeled using a combination of static PSF and temporal PSF, and is expressed as:

$$\Phi(x, y) = \phi(x) * \psi(x, y). \tag{14}$$

We use $\Phi(x, y)$ to restore motion blur as a deconvolution PSF.

2.3 Relation between Aperture Size and Object Speed

The aperture position must be moved to realize motion-invariant photography using a programmable aperture camera. The maximum size of the aperture radius of the $R_{\max}$ is restricted

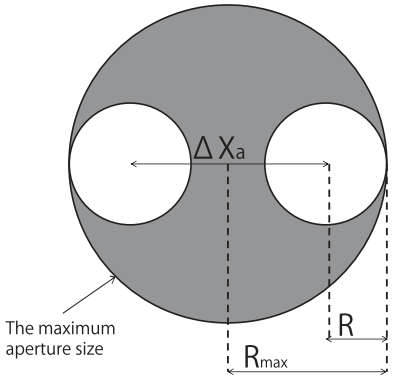


Fig. 5 Relation between aperture size and length of the aperture motion.

by the lens and the caliber of the optical system. When a large motion of aperture ΔX_a is required, the size of aperture R must be smaller than $R_{\max}$, as shown in **Fig. 5**.

$$\Delta X_a = 2(R_{\max} - R) \tag{15}$$

As a result, when using a certain exposure time T, a larger motion of aperture ΔX_a produces greater acceleration:

$$a_a = \frac{2\Delta X_a}{T^2}. \tag{16}$$

As explained in Levin's work [11], large aperture acceleration a_a is needed to restore large motion blur. When setting a large acceleration a_a, radius R becomes small and the amount of light decreases simultaneously. This means that there is a trade-off between acceleration a_a and the SNR ratio of the captured image.

$$a_a = \frac{4}{T^2}(R_{\max} - R) \tag{17}$$

We have explained that acceleration depends on the size of radius R, however, the motion invariance of the proposed method depends more directly on the acceleration in the image space a_i than the physical aperture acceleration a_a, and therefore we rewrite Eq. (17) as Eq. (18).

$$a_i = \frac{4}{T^2}\beta(R_{\max} - R) \tag{18}$$

From this equation, if we set β to a larger value, acceleration a_i that is presented in the image space also increases. β can be considered to be the difference between position Z_q where an object is focused and the imager position Z_p from Eq. (9). In short, β can be regarded as the amount that the image is out-of-focus. Furthermore, as shown in Eq. (12), a greater β produces a greater static PSF for a given aperture size. A greater static PSF results in a worse quality of the restored image. Hence, there is another trade-off. If we would like to use greater aperture acceleration in the image space a_i, we must accept that the quality of the restored image will be worse as a result of the greater static PSF. The PSF size is also affected by the object positions, although we can set the fixed size of the aperture. In Section 4.2, we evaluate the optimal PSF size for recovering an image.

3. Simulation Experiments

We investigated the limitations of our proposed method and optimization of the parameters through simulation experiments. We

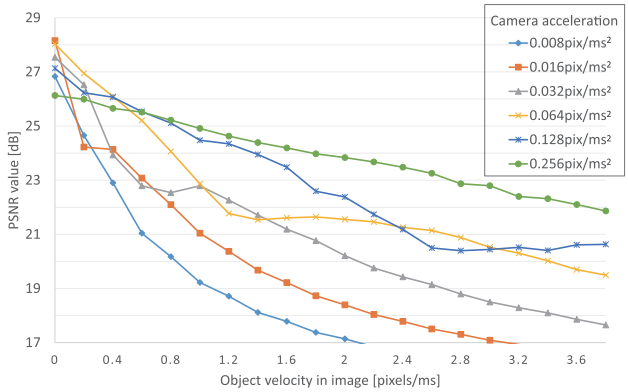


Fig. 6 Relation between PSNR and object velocity with varying camera acceleration.

used 30 natural images downloaded from Flickr as scene textures, and generated artificially captured images including motion blur, thereby realizing the image acquisition process using our motion-invariant coding. We added Gaussian noise with zero mean and standard deviation of 0.01 to the images to emulate readout noise as a standard of the experiment. We used Weiner deconvolution, since it is simple and suitable for analysis.

3.1 Camera Acceleration and Object Velocity

First we evaluated how much acceleration to apply to recover the motion blur with the object speed. In the experiment we set the camera acceleration in the image a_i to 0.008, 0.016, 0.032, 0.064, 0.128, 0.256, and generated coded images in which the captured scenes contained objects with varying motion. Since we assumed motion invariance, we used the PSF for an object with zero speed ($s_i = 0$) for the deconvolution. We calculated PSNR between the original image and the deconvolved image, and evaluated the image quality and invariance of the deconvolution according to the PSNR.

In **Fig. 6**, PSNR is plotted against object speed. We set s_i to 0.01 to 4.0 pixels/ms to emulate a moving object, as indicated on the horizontal axis of this figure. Figure 6 shows that the PSNR is high when object velocity s_i is low. If we use a higher setting for the acceleration of the camera a_i, PSNR becomes flatter across the object speeds. This shows that a higher a_i gives more invariance to a wide range of object speeds, although there is a trade-off between acceleration a_i and the peak quality of the restored image, since greater acceleration yields larger motion coding and makes the deconvolution more difficult. Hence, we should set a_i according to the maximum velocity anticipated in the scene.

3.2 Effect of Static PSF Size

Our method requires that the focusal point Z_q is displaced from the imager position Z_p to realize a parallax for coding. This means that the coded image has defocus blur as well as motion blur. Both types of blur can be recovered simultaneously by deconvolution, but the quality of the restored image will be worse than that of Levin's method, since our coded image has defocus blur while Levin's method assumes an impulse as the static PSF. In this section, we examine how static PSF size affects the quality of the restored image by means of a simulation experiment.

Figure 7 shows PSNR across the static PSF r with different

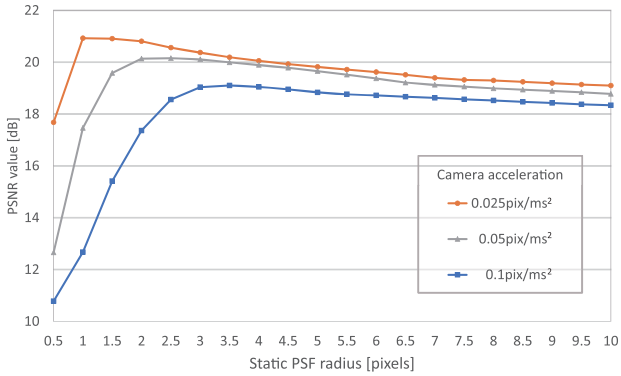


Fig. 7 Relation between PSNR and the static PSF radius with varying camera acceleration.

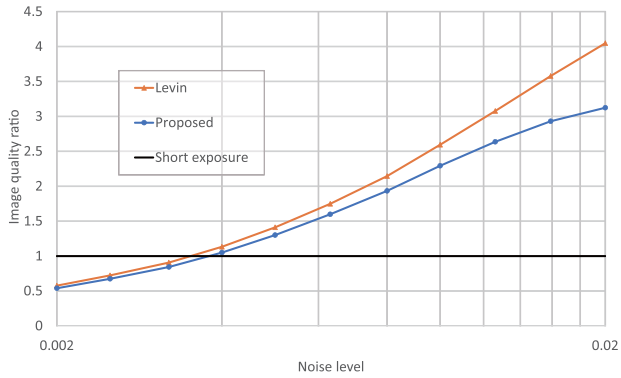


Fig. 8 Image quality ratio compared with short exposure imaging for varying noise levels.

camera acceleration in image a_i. In Fig. 7, the PSNR curve has a peak at 2.5 pixels for $a_i = 0.05$. The figure also shows peaks at 1 and 3.5 pixels for $a_i = 0.025$ and $a_i = 0.1$, respectively. This confirms that the optimal setting of r differs from the coding acceleration, which handles the different object speed ranges. Thus, r should be set considering the acceleration that will be used, which is similar to the finding in Section 3.1.

3.3 Effect of Noise Levels

We often use a high ISO setting to capture images in a dark scene. However, a higher ISO setting yields a higher level of noise in the image. We expect that conventional short exposure is preferable for application in a bright scene, but motion-invariant coding has the advantage in a dark scene. We evaluated under which conditions higher quality could be obtained by coding as opposed to short exposure imaging.

Figure 8 shows the image quality ratio of the proposed aperture coding to short exposure imaging against the noise level. The graph also shows the ratio for Levin's method. The ratios were calculated as

$$Ratio = \sqrt{\frac{MSE_{short}}{MSE}}, \tag{19}$$

where MSE_{short} is the mean-squared error (MSE) for short exposure imaging, while MSE is that for the proposed method or Levin's method. A similar evaluation criterion was using in Ref. [18]. In the simulation, a_i was set to 0.025, 0.05, and 0.1 pixels/ms^2.

This figure shows that the proposed method and Levin's method have an advantage over short exposure in the case where the noise level is greater than 0.004, which denotes dark scene illumination. We can also see that Levin's method shows a much higher quality ratio than the proposed method, since the former method does not include depth blur for coding. However, this method requires physical coding motion such as camera shifting.

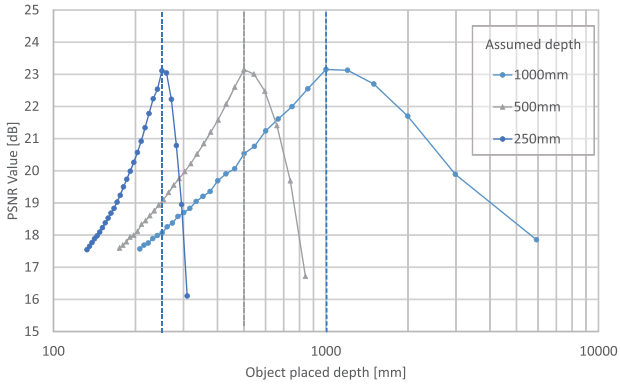


Fig. 9 Relation between PSNR and the displacement from the assumed object depth.

3.4 Effect of Scene Depth

Thus far we have assumed that the scene has a single depth and thus scene depth has been ignored when multiple objects are placed at various depths in the scene. This is an unrealistic assumption. As already defined in Eq. (12), the static PSF size r changes according to the object depth. The image quality decreases as a result of deconvolution error if the captured PSF size differs from the deconvolved one given by the assumed object depth. Camera acceleration a_i for motion-invariant coding also changes as defined in Eq. (18), if the object depth differs from the assumed one. This may degrade the invariance of motion blur. We evaluated the range of depths in which we can apply our method using a single PSF by means of a simulation experiment. To ensure that the evaluation was focused on this purpose, we set those parameters not affecting the evaluation to the values used in Sections 3.1 and 3.2.

Figure 9 shows the PSNR against scene depth for different assumed depths. The dotted line denotes the assumed object depth. Note that the horizontal axis is calibrated using a logarithmic scale. The PSNR reached its highest value with a zero displacement, which means that the captured PSF matches the deconvolved one. The PSNR becomes worse as the object is displaced close to or far from the assumed depth, since the PSF changes with a change in depth. If we allow some degradation of the restored image from the peak of the PSNR curve, it can be considered that some range of depths is acceptable and the depth difference is negligible. These plots show that the range is narrower when the assumed depth is close to the camera, and is wider when it is far from the camera, similar to the depth of field (DOF) of a normal camera.

4. Real Experiments

We carried out some real experiments. We constructed a prototype camera which has a similar implementation to the camera in the paper [15], [16]. The overview of the camera is shown in

Fig. 10 Prototype camera.

Table 1 Specifications of the prototype camera.

Image resolution	1,280 × 960
Image acquisition frame rate	15 fps
Aperture resolution	1,280 × 1,024
Aperture frame rate	365 fps
Minimum F-number	2.8
Field of view	46°

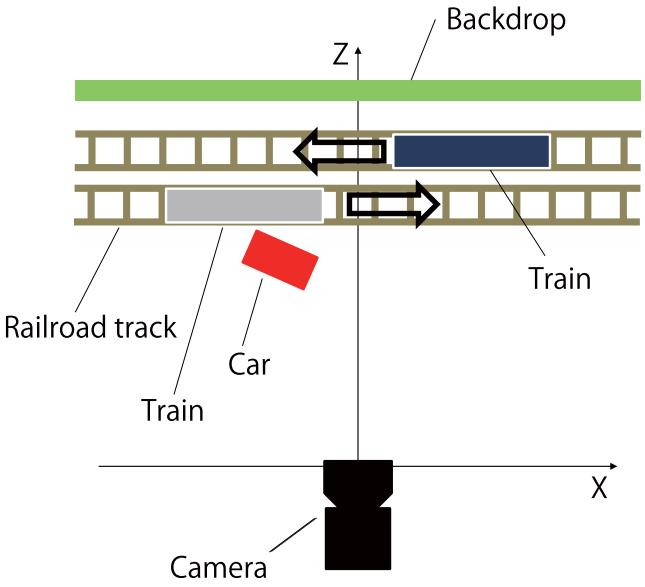


Fig. 11 Target scene.

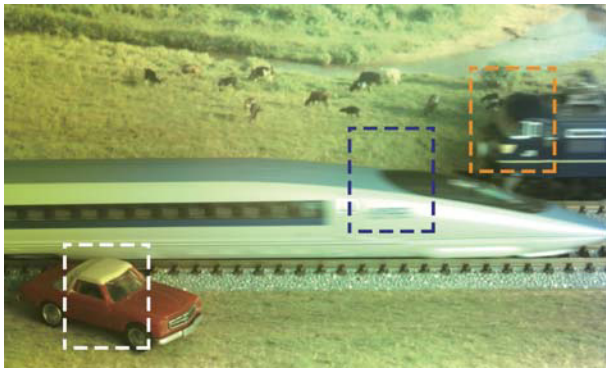
Fig. 10. The camera consists of LCoS (Forth dimension display SXGA-3DM), CCD (Point Grey Grasshopper GRAS-14S5C-C), polarizing beam splitter (Edmund Optics #49002), Primary lens (Nikon Rayfact 25 mm F1.4 SF2514MC) and custom made relay lenses (f = 27 mm, F/# = 2). The specifications of the prototype are given in **Table 1**.

Figure 11 shows the arrangement of the target scene in the experiment, in which we use two trains going right and left as moving objects. We set a backdrop for the background namely, the far side railroad track, the near side railroad track and a miniature car at a distance of 505 mm, 500 mm, 495 mm and 490 mm, respectively, so that these objects appeared in the DOF. When the image was captured using normal photography, the focal point was set on the object. All captured images (**Figs. 12–15**) were adjusted to almost the same intensity. We captured the scene and obtained the image shown in Fig. 12. For this experimental setup, the motion of the far side train appeared as linear motion of 1.0 pixels/ms in the image space, and the near side train is 1.4 pixels/ms. Since we set the camera exposure time to 45 ms, motion blur appeared with a length of about 45 pixels (the far side train) and 63 pixels (the near side train) in a normal photograph (The length was obtained from Fig. 12). The frame rate of the aperture change is 365 fps for the prototype camera, which means that 16 aperture patterns can be displayed during the exposure time. We set the radius of the displaying aperture pattern R as 145.5 pixels (F9.85) against the width of the aperture pattern display of 1,280 pixels to make the static PSF size $r = 2$ pixels. Under these conditions, acceleration of 0.089 pixels/ms^2 was presented in the imager plane. We coded the motion blur using this acceleration. Figure 13 shows the image captured by our motion-invariant photography. In this figure, both the moving trains and the static background are blurred. We used the measured PSF, which was captured in advance, for deconvolution (For deconvolution, we used BM3D deconvolution proposed by Dabov et al. [19]). Figure 14 shows the restoration result obtained by deconvolution of the captured image (Fig. 13) with the single measured PSF. This figure shows that we can reduce motion blur through deconvolution since the edge of the image is sharper than that captured by normal photography (Fig. 12). In addition, we reduced blur of the static background equally without motion estimation or segmentation. We also compared our result with a simple short exposure photograph. Figure 15 depicts the image obtained through short exposure imaging with the exposure time set to 0.679 ms so as to ignore all motion in the scene. In this case we used F2.8, which is the maximum radius setting of the lens. It can be seen that the short exposure results in a noisy image with loss of gradation, because the amount of light is reduced and noise is emphasized by the intensity adjustment. Our coded and deconvolved result yields a better image, which is both sharper than the blurred image and brighter than the short exposure image.

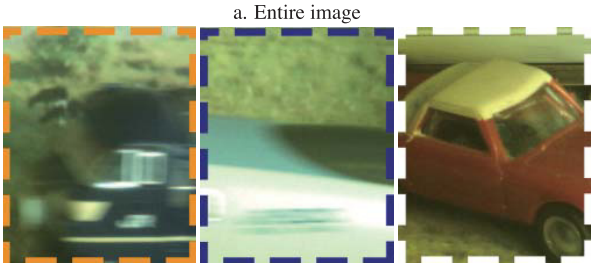
In addition, we showed other application results. A soccer ball is moving left to right by hand in **Fig. 16**, and a man is passing right to left in **Fig. 17**. The images on the top row were captured by normal photography, and the bottom row were deconvolved. The scenes has been captured as different depth ranges to the train scene (Figs. 12–15: 500 mm, Fig. 16: 1,000 mm and Fig. 17: 3,000 mm). We can confirmed that our method works in different depth ranges and settings as well. Note that these blurred images and coded images for deconvolution were captured individually. We took care so that the moving objects have the same velocity, but these cannot be retained strictly as well as the experiment in Figs. 12–15.

5. Conclusion

In this research, we proposed a novel method for coding a motion-invariant PSF using a programmable aperture camera. The camera can dynamically change the aperture pattern at a high frame rate and realizes virtual camera motion by translating the opening. As a result, we can obtain a coded image in which motion blur is invariant with respect to the object velocity. Thus, we can recover motion blur without estimating motion blur kernels or requiring knowledge of the object speeds. To realize this, we

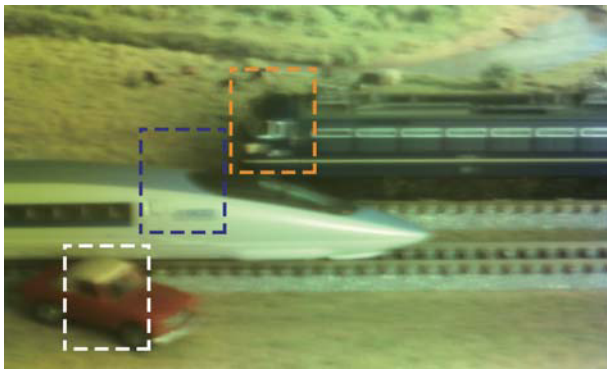


a. Entire image

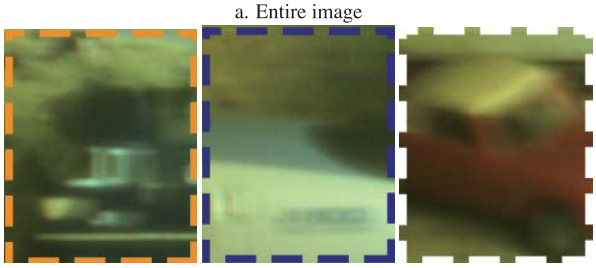


b. Magnified images
(left and center: moving object, right: static object)

Fig. 12 Image captured by normal photography.

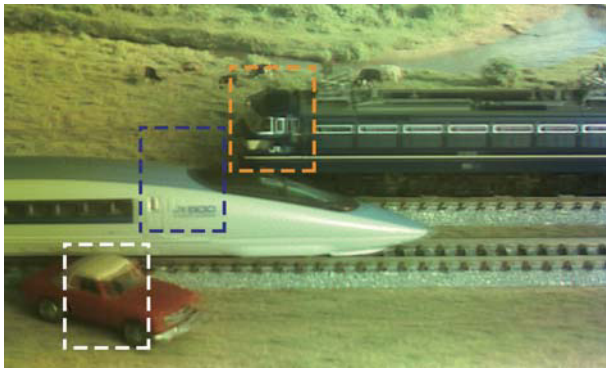


a. Entire image



b. Magnified images
(left and center: moving object, right: static object)

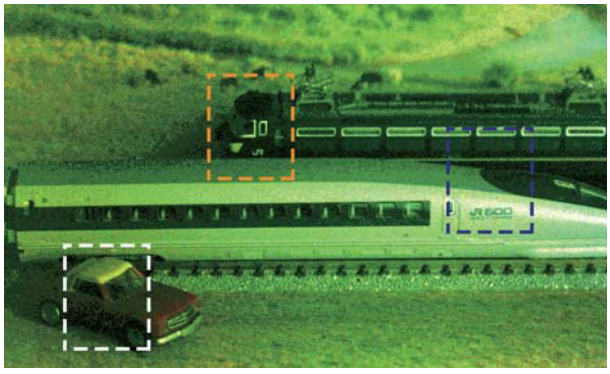
Fig. 13 Blurred image recorded by motion-invariant photography.



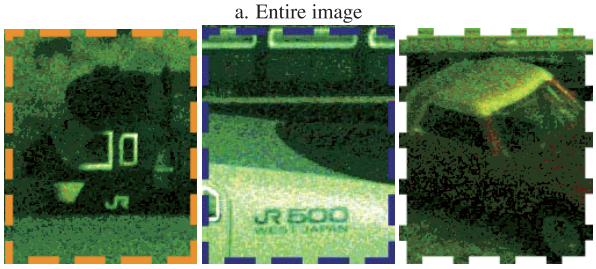
a. Entire image

b. Magnified images
(left and center: moving object, right: static object)

Fig. 14 Deconvolved image.



a. Entire image



b. Magnified images
(left and center: moving object, right: static object)

Fig. 15 Image captured using short exposure.

modeled the projective geometry of the programmable aperture camera, virtual motion of the camera, and the generated PSFs. We analyzed the parameter settings and optical parameters required for the proposed motion-invariant photography and discussed the range of parameters for which the proposed method is superior to short exposure in a simulation experiment. Moreover, we experimentally demonstrated that our proposed coding works with the prototype camera.

References

[1] Canon, EF Lens Work III, The Eyes of EOS, Canon Inc. Lens Product Group.

[2] Jansson, P.: *Deconvolution of Image and Spectra*, Academic Press, 2nd edition (1997).

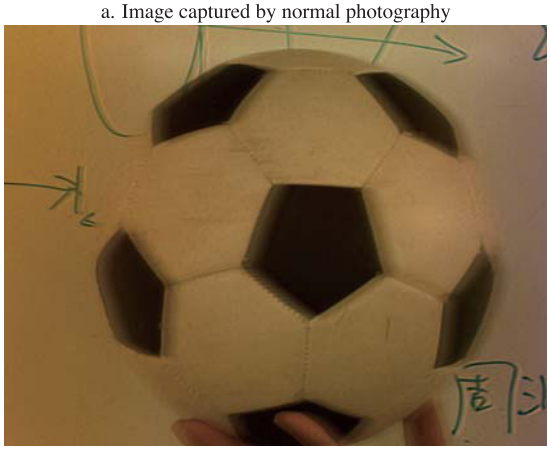
[3] Fergus, R., Singh, B., Hertzmann, A., Roweis, S.T. and Freeman, W.T.: Removing camera shake from a single photograph, *ACM Trans. Graph.*, Vol.25, No.3, pp.787–794 (2006).

[4] Shan, Q., Jia, J. and Agarwala, A.: High-quality motion deblurring from a single image, *ACM Trans. Graph.*, Vol.27, No.3, pp.1–10 (2008).

[5] Yuan, L., Sun, J., Quan, L. and Shum, H.-Y.: Progressive inter-scale and intra-scale non-blind image deconvolution, *ACM Trans. Graph.*, Vol.27, No.3, pp.1–10 (2008).

[6] Nayar, S.K. and Ben-Ezra, M.: Motion-based motion deblurring, *IEEE Trans. Pattern Recognition and Machine Intelligence*, Vol.26,

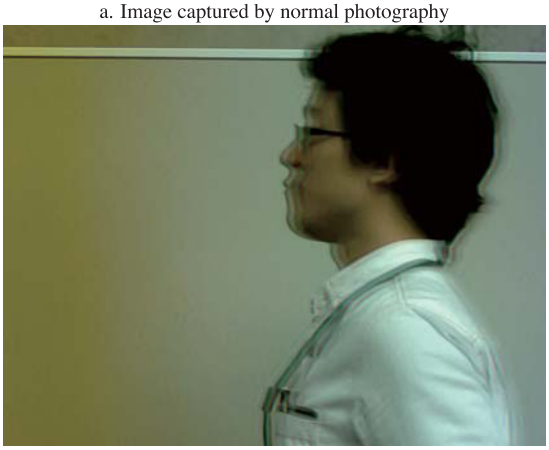
a. Image captured by normal photography



b. Deconvolved image

Fig. 16 A soccer ball is moving left to right by hand. The distance to the soccer ball is set as 1,000 mm.

a. Image captured by normal photography



b. Deconvolved image

Fig. 17 A man is walking right to left. The distance to the man is set as 3,000 mm.

Issue 6, pp.689–698 (2004).
[7] Yuan, L., Sun, J., Quan, L. and Shum, H.-Y.: Image deblurring with blurred/noisy image pairs, *ACM Trans. Graph.*, Vol.26, Issue 3, No.1 (2007).
[8] Bar, L., Berkels, B., Sapiro, G. and Rump, M.: A variational framework for simultaneous motion estimation and restoration of motion-blurred video, *Proc. ICCV 2007* (2007).
[9] Raskar, R., Agrawal, A. and Tumblin, J.: Coded exposure photography: Motion deblurring using fluttered shutter, *ACM Trans. Graph.*, Vol.25, No.3, pp.795–804 (2006).
[10] Agrawal, A. and Xu, Y.: Coded Exposure Deblurring: Optimized Codes for PSF Estimation and Invertibility, *Conf. CVPR* (2009).
[11] Levin, A., Sand, P., Cho, T.S., Durand, F. and Freeman, W.T.: Motion-Invariant Photography, *ACM Trans. Graph.*, Vol.27, Issue 3, No.71 (2008).
[12] McCloskey, S., Muldoon, K. and Venkatesha, S.: Motion invariance and custom blur from lens motion, *Proc. ICCP* (2011).
[13] Cho, T.S., Levin, A., Durand, F. and Freeman, W.T.: Motion blur removal with orthogonal parabolic exposures, *Proc. ICCP* (2010).
[14] Bando, Y., Chen, B.Y. and Nishita, T.: Motion Deblurring from a Single Image using Circular Sensor Motion, *Computer Graphics Forum*, Vol.30, No.7 (2011).
[15] Nagahara, H., Zhou, C., Watanabe, T., Ishiguro, H. and Nayar, S.K.: Programmable Aperture Camera Using LCoS, *Proc. ECCV* (2010).
[16] Nagahara, H., Zhou, C., Watanabe, T., Ishiguro, H. and Nayar, S.K.: Programmable Aperture Camera Using LCoS, *IPSJ Trans. CVA*, Vol.4, pp.1–11 (2012).
[17] Sonoda, T., Nagahara, H. and Taniguchi, R.: Motion-Invariant Coding Using a Programmable Aperture Camera, *Proc. ACCV* (2012).
[18] Cossairt, O., Gupta, M. and Nayar, S.K.: When Does Computational Imaging Improve Performance?, *IEEE Trans. Image Processing* (2012).
[19] Dabov, K., Foi, A. and Egiazarian, K.: Image restoration by sparse 3D transform-domain collaborative filtering, *Proc. SPIE* (2008).

Toshiki Sonoda received his B.E. and M.E. degrees from Kyushu University in 2012 and 2014. He is a Ph.D. student at Kyushu University. He has been engaged in computational photography.

Hajime Nagahara received his B.E. and M.E. degrees in electrical and electronic engineering from Yamaguchi University in 1996 and 1998, respectively. He received his Ph.D. in system engineering from Osaka University in 2001. He was a Research Associate of Japan Society for the Promotion of Science in 2001–2003. He was a Research Associate of the Graduate School of Engineering Science, Osaka University, in 2003–2006. He was a Visiting Associate Professor at CREA University of Picardie Jules Verns, France in 2005. He was an Assistant Professor of Graduate School of Engineering Science in 2007–2010. He was a Visiting researcher at Columbia University, USA in 2007–2008. He is an Associate Professor of Faculty of Information Science and Electrical Engineering, Kyushu University, since 2010. Computational photograhy, Image processing, Computer vision and Virtual reality are his research subjects. He got an ACM VRST2003 Honorable Mention Award in 2003.

Rin-ichiro Taniguchi received his B.E., M.E., and D.E. degrees from Kyushu University in 1978, 1980, and 1986. Since 1996, he has been a Professor in the Graduate School of Information Science and Electrical Engineering at Kyushu University, where he directs several projects including multiview image analysis and software architecture for cooperative distributed vision systems. His current research interests include computer vision, image processing, and parallel and distributed computation of vision-related applications.

(Communicated by *Ko Nishino*)