Focal Pre-Correction of Projected Image for Deblurring Screen Image

Yuji OYAMADA and Hideo SAITO
Graduate School of Science and Technology, Keio University
3-14-1 Hiyoshi Kohoku-ku, Yokohama 223-8522, Japan
charmie@ozawa.ics.keio.ac.jp, saito@ozawa.ics.keio.ac.jp

Abstract

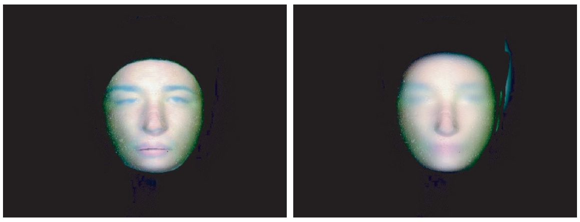
We propose a method for reducing out-of-focus blur caused by projector projection. In this method, we estimate the Point-Spread-Function (PSF) of the out-of-focus blur in the image projected onto the screen by comparing the screen image captured by a camera with the original image projected by the projector. According to the estimated PSF, the projected image is pre-corrected, so that the screen image can be deblurred. Experimental results show that our method can reduce out-of-focus projection blur.

1. Introduction

Augmented Reality (AR) and Mixed Reality (MR) are techniques for overlaying virtual objects onto a real world. AR has recently been applied to many kinds of applications including entertainment. In traditional AR applications, LCD displays and HMDs are generally used as output device. Recently, projectors are used for overlaying virtual objects onto real world objects, so that multiple users can experience the AR applications at the same time.

This type AR is called Projection Based AR. Yotsukura et al. have proposed a system that supports an actor wearing a face mask detected by infrared LEDs [10]. This system tracks a face mask with LEDs so that the projected image can always be attached onto the face mask surface. We have proposed 3D face display [7], in which we use a mannequin as a screen object and project face image onto the mannequin. The projector is used for various usages, but can be used in limited environments. The projector basically needs to project onto non-textured and non-colored planar screen, which should be perpendicular to the projection direction. To reduce such limitation for using projectors, a lot of methods have been proposed. Most research project feature images for measuring the display's geometric information [8, 9], display's and environmental radiometric information [1, 2, 3].

Applying these methods to Projection Based AR make displayed image more realistic. However, these methods ignore that projector's depth-of-field restricts the position of display objects. If the screen object has complex surface or movement, out-of-focus projection blur can occur as shown Fig.1.



(a) In-focus projection (b) Out-of-focus projection

Figure 1. Out-of-focus projection blur

To solve this problem, Bimber et al. proposed a method that uses multiple projectors with differently adjusted depth-of-field [5]. The out-of-focus blur on the projected display is estimated automatically for each projector pixel via camera feedback and feature image projection. Then each projector projects to make out-of-focus blur region minimal. Zhang et al. proposed iterative pre-correct algorithm using obtained depth map [11]. They project the feature images and pre-correct the projection image iteratively using camera feedback. Then the iteratively pre-corrected image projected onto the display is close to the original image. Brown et al. proposed another method that pre-corrects the projection image to reduce out-of-focus blur [4]. They also use a feature image for estimating the out-of-focus projection blur using the camera feedback. First, they detect the most in-focus region by comparing the captured image with the projection image, and define this region as an exemplar region. Second, they pre-correct each image region depending on the degree of blur from exemplar region. Projecting the precorrected image will reduce out-of-focus projection blur is reduced.

In the case of projection onto the moving display as [7], these three methods have a fatal problem. Since the degree of out-of-focus blur on the display changes every moment that the display moves, these methods [4, 5, 11] have to

1-4244-1180-7/07/$25.00 ©2007 IEEE

project the feature image to estimate the degree of out-of-focus blur at every moment. Projecting the feature image that actually should not be shown to the observers prevents realistic projection and loses advantage of Projection Based AR.

To achieve Projection Based AR that can be used in case of moving display object, we propose a new method that reduces out-of-focus blur on the projected display without projecting the feature image. We use the original image that should be displayed to the observers as projected image for estimating the degree of out-of-focus blur and the geometrical skew. In order to adapt the projector's slanted pose to the screen, we can also reduce gradually changing out-of-focus projection blur by using a camera to estimate a series of spatially varying degrees of blur. Then we pre-correct the original image by Wiener Filtering according to the estimated blur before projection.

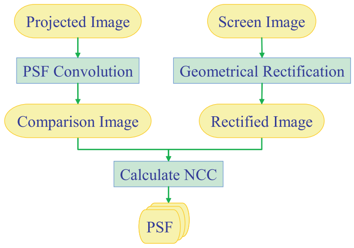
2. Proposed Method

When we use a projector located at a slanted position to a screen, non-perpendicular projection makes the displayed image on the screen partially blurred. We can represent this out-of-focus projection blur with the PSF. In the case of a perpendicular projection to the screen, all of the regions are uniformly blurred. On the other hand, in the case of a slanted projection to the screen, blur on the screen is not uniform. To handle the case of the slanted projection blur, we have to estimate a series of spatially varying PSFs of blur in the captured image. Fig.2 shows that our method is divided into two main phases, estimation of PSFs as shown in Fig.2(a) and pre-correction as shown in Fig.2(b). First, we project an image on the screen, and then capture the displayed image on the screen by a camera. Here we have two input images, one is what we would like to project onto the screen as shown in Fig.2(c) (a projected image), and the other is the displayed image captured by the camera as shown in Fig.2(d) (a screen image). Then we rectify the geometrical skew of the screen image as shown in Fig.2(f) (a rectified image). In order to estimate the degree of out-of-focus blur included in the rectified image, we generate a comparison image as shown in Fig.2(e). We estimate the PSFs on the screen image by comparing the rectified image with the different comparison images blurred by different PSFs. Finally, we pre-correct an original image according to the estimated PSFs.

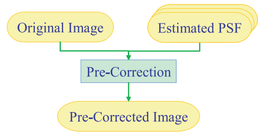
2.1. Image Blurring and Deblurring

The out-of-focus projection blur can be represented by the 2D circular disk type PSF $h(x,y)$ with the radius r[pixel] as Eq.1

$$h\left(x,y\right)=\begin{cases}1 & \text{if}\left(\sqrt{x^2+y^2}\leq r\right)\\ 0 & \text{if}\left(\sqrt{x^2+y^2}>r\right)\end{cases}\qquad(1)$$



(a) Overview of PSFs estimation



(b) Overview of pre-correction

(c) Projected image

(d) Screen image

(e) Comparison image

(f) Rectified image

Figure 2. Overview of proposed method

A blurred image $g(x,y)$ can be represented with a convolution of PSF $h(x,y)$ on the original image $f(x,y)$.

$$g\left(x,y\right)=f\left(x,y\right)*h\left(x,y\right)\qquad(2)$$

Based on the traditional image processing technology, we can restore the unknown original image by convolution with an inverse function $h^{-1}(x,y)$ on the blurred image. The main topic of traditional image restoring is how to estimate the PSF and how to restore the unknown original image using estimated PSF. In the case of out-of-focus projection blur, we know the type of PSF, and we also know the original image that we would like to show to the observers. Therefore, we can display the original image by projecting a pre-corrected image $\tilde{f}(x,y)$, in which the out-of-focus blur is previously deblurred.

$$\begin{aligned}f\left(x,y\right)&\approx\tilde{f}\left(x,y\right)*h\left(x,y\right)\\&=\left[f\left(x,y\right)*h^{-1}\left(x,y\right)\right]*h\left(x,y\right)\end{aligned}\qquad(3)$$

We can represent convolution in the spatial domain as the product in the frequency domain, where the blurring is represented as

$$G\left(u,v\right)=F\left(u,v\right)H\left(u,v\right)\qquad(4)$$

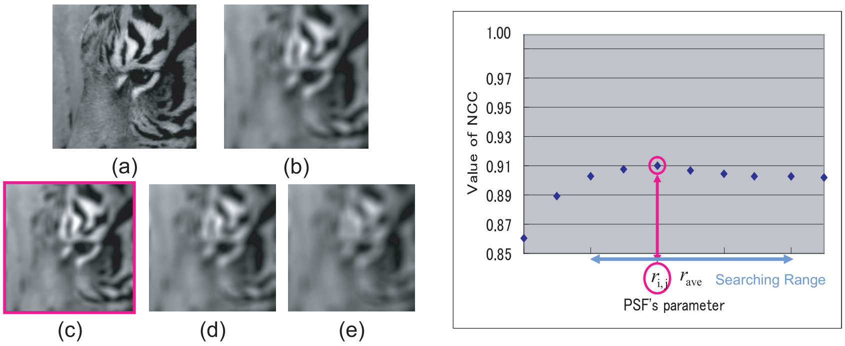


Figure 3. Estimation of PSF:(a)Original image;(b)Screen image;(c)$g_{r_{\text{ave-1.0}}}(x, y)$;(d)$g_{r_{\text{ave}}}(x, y)$;(e)$g_{r_{\text{ave+1.0}}}(x, y)$

where G, F and H are Fourier transforms of g, f and h respectively. If we know the PSF, we can apply Wiener Filtering, which is one of the popular solutions that minimizes the effect of deconvoluted noise. The Wiener Filter H_{w} is modeled as

$$H_{\text{w}} = \frac{1}{H} \frac{|H|^2}{|H|^2 + \gamma} \tag{5}$$

where γ is the signal-to-noise ratio in power.

2.2. Geometrical Rectification

We estimate the PSF by comparing the projected image with the screen image, but the screen image is geometrically skewed to the screen surface. So we have to perform geometric calibration between these two images [6, 8, 9]. We model this relationship of these two images as the 3 × 3 planar perspective transformation matrix (homography). A pixel in the screen image x_{c} is rectified to a pixel in the projected image x_{p} using homography H_{cp} as

$$x_{\text{p}} \cong \text{H}_{\text{cp}} x_{\text{c}} \tag{6}$$

To calculate the homography, we need at least 4 corresponding points. We project feature rectangle images and capture the screen image. Then, we calculate homography by comparing the screen image with the projected feature rectangle image.

2.3. Estimation of PSF

As mentioned Sec.2.1, we can estimate PSF on a rectified image by comparing with the different comparison images with different PSFs. First, we generate multiple comparison images $g_r(x, y)$ by convolution different PSFs on the projected image. Then we calculate NCCs R_{NCC} (Normalized Cross Correlation) between each comparison image $g_r(x, y)$ and the captured image $g(x, y)$. The PSF of the comparison image with highest correlation to the captured image is $h_r(x, y)$.

$$R_{\text{NCC}} = \frac{\displaystyle\sum_{x=0}^{N-1}\sum_{y=0}^{M-1} \left(g(x, y) - g_r(x, y)\right)}{\sqrt{\displaystyle\sum_{x=0}^{N-1}\sum_{y=0}^{M-1} g^2(x, y) \sum_{x=0}^{N-1}\sum_{y=0}^{M-1} g_r^2(x, y)}} \tag{7}$$

To handle to the case of projection from slanted position, we estimate a series of spatially varying PSFs of out-of-focus blur. Experience shows that spatially varying PSFs does not drastically fluctuate, but are close in value to the PSF that estimated from whole images. Therefore we first estimate PSF on the whole rectified image r_{ave}, and then estimate each varying PSF $r_{\text{i,j}}$ close to r_{ave}. Applying to every sub-image, we can estimate PSFs.

Fig.3 shows example of estimation. We first estimate r_{ave} and then estimate each $r_{\text{i,j}}$. Pink-framed image is closest image with the captured image.

2.4. Pre-correction

We apply Wiener Filtering using estimated PSFs as a pre-correction. However Wiener Filtering has two problems. The first one concerns the degree of out-of-focus blur on the captured image that is not piecewise-constant but smoothly changes. Pre-correction on each sub-image respectively causes boundary between adjacent sub-images. So we have to pre-correct sub-image using adjacent sub-image's PSF as shown in Fig.4. We denote these four neighbors's parameter $r_{\text{i,j}}, r_{\text{i,j+1}}, r_{\text{i+1,j}}, r_{\text{i+1,j+1}}$, and let $\tilde{f}_{\text{i,j},r_{\text{i,j}}}$ refers to the pre-corrected sub-image using parameter $r_{i,j}$. The pre-corrected sub-image $\tilde{f}_{\text{i,j}}(x, y)$ is written as

$$\begin{aligned}
\tilde{f}_{\text{i,j}}(x, y) \quad = \quad & (1 - s_x)(1 - s_y)\,\tilde{f}_{\text{i,j},r_{\text{i,j}}}(x, y) \\
& + s_x (1 - s_y)\,\tilde{f}_{\text{i,j},r_{\text{i+1,j}}}(x, y) \\
& + (1 - s_x)\, s_y \tilde{f}_{\text{i,j},r_{\text{i,j+1}}}(x, y) \\
& + s_x s_y \tilde{f}_{\text{i,j},r_{\text{i+1,j+1}}}(x, y)
\end{aligned} \tag{8}$$

where $s_x, 1-s_x, s_y, 1-s_y$ are linear interpolation coefficients in the x and y axis.

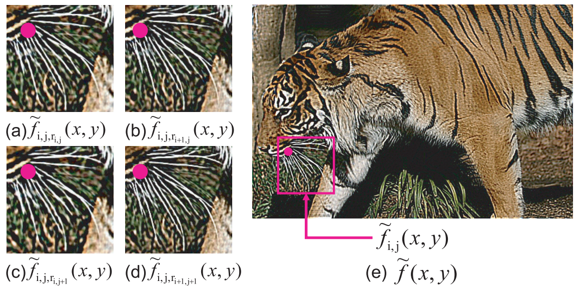


Figure 4. Pre-correction using bi-linear interpolation

The second problem is the Wiener Filtered image value. Indeed, these images may contain both negative values and higher values than the maximum allowed in the original image (i.e.255). Since the projector can only display images with pixel value with in the range of 0∼255, we have to normalize the output image, so that the range of values of the output image can be fit to the range of projected image (0∼255). To handle these outlying values, we test two normalization methods. One clamps the outlying values as in Eq.9, and the other scales the whole value as in Eq.10.

$$f_{\text{clamp}}(x,y) = \begin{cases} 0 & \text{if } (f(x,y) \leq 0) \\ 255 & \text{if } (f(x,y) \geq 255) \\ f(x,y) & \text{otherwise} \end{cases} \tag{9}$$

$$f_{\text{scale}}(x,y) = \frac{255}{f_{\max} - f_{\min}} (f(x,y) - f_{\min}) \tag{10}$$

$$\left(\begin{array}{l} f : \text{original pixel value} \\ f_{\text{clamp}} : \text{clamped pixel value} \\ f_{\text{scale}} : \text{scaled pixel value} \\ f_{\min} : \text{minimum value in the original image} \\ f_{\max} : \text{maximum value in the original image} \end{array} \right)$$

3. Experiment

Our experiments are performed using an EPSON ELP7600 LCD projector with a 1024×768 resolution placed in a slanted direction in respect to the screen. A SONY XCDC710CR camera with a 1024×768 resolution captures screen images to estimate PSFs. In our experiments, projected images are 960×640 resolution and captured screen images with 1024×768 resolution is rectified to the projected images. These rectified images are divided into 160×160 images for PSF estimation and pre-correction. During the Wiener Filtering computation, γ in Eq.5 is set to 0.01. Fig.5 shows experimental environment.

3.1. Geometrical Rectification

We examine the accuracy of homography between the projected image and the screen image at slanted projection.

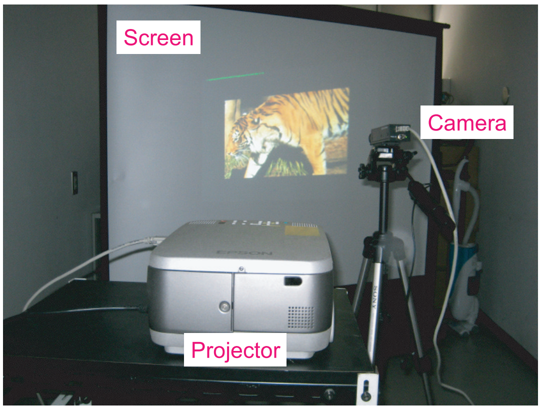
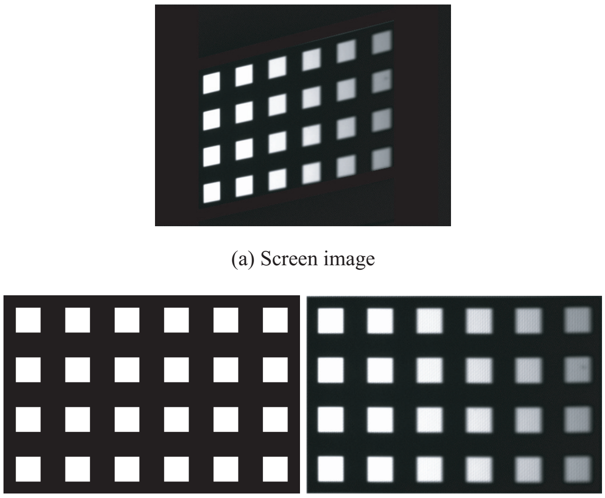


Figure 5. Experimental environment

First we calculate homography using rectangle images. To confirm the accuracy of homography, we test projecting a rectangle array image as shown in Fig.6(b). Fig.6(a) is the screen image and Fig.6(c) is the rectified image using calculated homography. We calculate the center of gravity coordinates of every white rectangle in each image. Comparing these two images indicates that there is only a difference in the brightness of the image. This difference in the brightness is caused by the positional relationship between the projector and the screen. The darker region is farther from the projector than the brighter region. The error of mean square between these two center of gravity coordinates is 0.8pixel ($x$axis) and 0.1pixel ($y$axis). This rectified error appears vanishingly small.

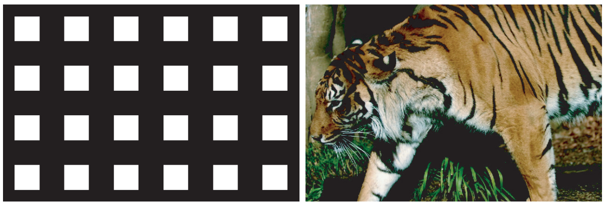

(a) Screen image

(b) Projected image (c) Rectified (a)

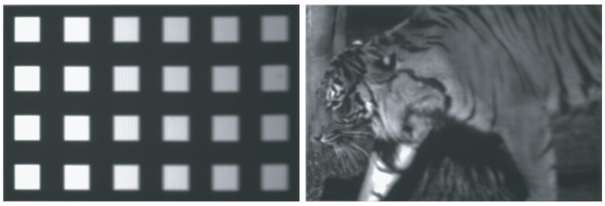
Figure 6. Rectification using homography

3.2. PSFs Estimation

Next, we examine the accuracy of PSFs estimation by comparing the result of the rectangle array image and the input image as shown in Fig.7(a) and Fig.7(b). Respectively Fig.7(c) and Fig.7(d) are screen images captured by the camera. Fig.7(c) indicates that the entire image is blurred and the degree of blur gradually increases from left to right. This is caused by the projector's position. The right side of the screen is nearer the projector, so the right side of the captured images is more blurred than the left side area of the captured image. The rectangle array image as shown in Fig.7(a) is suitable to estimate the degree of out-of-focus blur on the screen image. In the estimated PSF's parameter map of the rectangle array image (Fig.7(e)), the parameter of the right side area is larger than the left side. This means that PSF's parameter is correctly estimated because the screen is slanted within the horizontal direction. On the other hand, the animal picture (Fig.7(f)) includes error region in the bottom-right area. This error is caused by two main reasons. One reason is that the region of the projected image does not have many textures. When the sub-image with less texture is blurred, the estimated PSF of the out-of-focus blur can be less accurate. Another reason is that the pixel value of the projected image is almost saturated.



(a) Rectangle array image (b) Animal image



(c) Captured (a) (d) Captured (b)

4.0	6.0	7.0	9.0	10.0	10.0
4.0	6.0	7.0	9.0	10.0	10.0
4.0	6.0	7.0	9.0	10.0	10.0
4.0	5.0	7.0	9.0	10.0	12.0

(e) Parameter map of (c)

5.0	5.0	5.0	7.0	10.0	9.0
5.0	5.0	6.0	9.0	10.0	10.0
5.0	5.0	5.0	5.0	9.0	5.0
5.0	5.0	5.0	8.0	5.0	5.0

(f) Parameter map of (d)

Figure 7. Captured images and estimated PSFs

3.3. Normalization of Wiener Filtered image

As mentioned in Sec.2.4, we have to normalize Wiener Filtered image and we propose two normalization methods. We test these two methods. Fig8 are normalized images. Fig8(b) indicates scaled normalization loses image contrast. When the outlying values caused by Wiener Filtering is considerable big, almost pixel values are drastically scaled down. On the other hand, Fig8(a) shows that the clamp normalization is scarcely affected by clamping the outlying values. To preserve image contrast, we choose the clamp normalization.

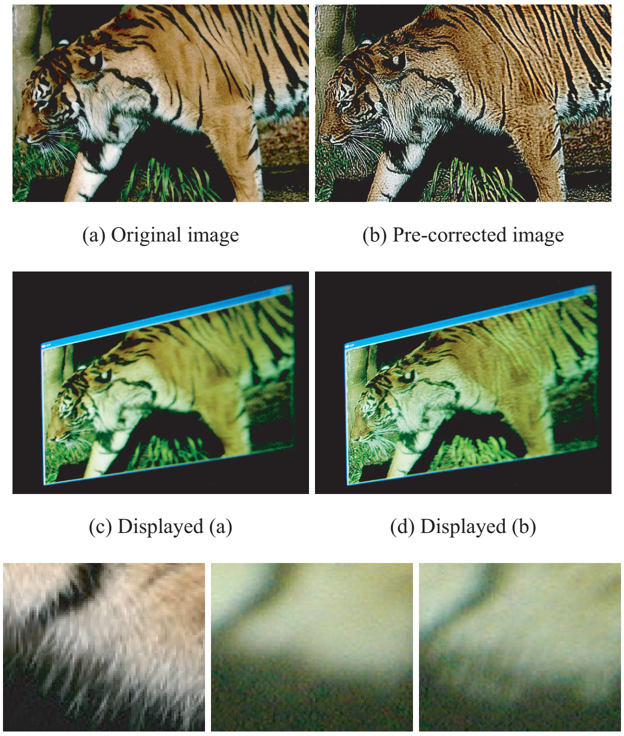
(a) Clamp normalization (b) Scale normalization

Figure 8. Normalized image

3.4. Projection pre-corrected image

We compare the screen images of the pre-corrected image with that of the original image. Comparing Fig.9(b) with Fig.9(a), all regions of the pre-corrected image are sharper than the original image. Especially, the stripe of the tiger's fur is emphasized. Fig.9(c) and Fig.9(d) are the screen images, and Fig.9(e)∼Fig.9(g) are the zoomed images. Zoomed images indicate that the projection result of the original image is blurred and lose texture of tiger's fur. On the other hand, the projection result of the pre-corrected image reduces the effect of out-of-focus blur. These images indicate that these results show that the screen image of the pre-corrected image is closer to the original image.

Next, we compare our method with Brown's method. In our method, PSFs are estimated by comparing the original image with the screen image captured by projecting the original image. In Brown's method, PSFs are estimated by finding the most in-focus region in the screen image of the rectangle array image. Fig.10 shows the projected images and the screen images, and Fig.11 shows specified regions of the each screen image. All region of the projection image pre-corrected by our method is sharper than the original image as shown in Fig.10(e). As shown in Fig.10(c), the left side of the projection image is non-corrected and the degree of correction is less than that of our method. Non-corrected region is referring to the exemplar region. Fig.11(c)∼Fig.11(e) refer to the exemplar region of each screen image. Comparing these images with Fig.11(a), the screen image of our method, especially lion's whiskers, is sharpest. As a result of non-correction, Fig.11(d) is as blurred as Fig.11(c). And Fig.11(f)∼Fig.11(h) refer to the lion's fur of each screen image. In this region, both images are corrected as shown in Fig.10(c) and Fig.10(e). Comparing Fig.11(h) with Fig.11(g), fur in the screen image of our method is sharper than Brown's one. These results indicate that our method can reduce the out-of-focus blur more than Brown's method.



(a) Original image (b) Pre-corrected image

(c) Displayed (a) (d) Displayed (b)

(e) Zoomed (a) (f) Zoomed (c) (g) Zoomed (d)

Figure 9. Experimental result of pre-correction

4. Conclusion

We propose pre-correction method to reduce out-of-focus projection blur without projecting the feature image. By pre-correcting every projected sub-image, we can reduce out-of-focus projection blur. Experimental results show that the our pre-correct method is successfully reduce the effects of out-of-focus projection blur, even though the screen image includes spatially varying blur without projecting the feature image.

5. Future work

As mentioned in Sec.1, our final goal is to achieve Projection Based AR that can handle the case of complex surface display and moving display. In this paper, we propose as a trial. Proposed method only performed the experiments under the assumption that the display is planar screen and does not move. And the computation time for PSF estimation and image pre-correction takes about 8 seconds respectively. To handle the case of non-planar display, we have to know the display's 3D shape and generate 3D PSF map from known the display's 3D shape information. And to handle the case of display moving, we have to estimate and pre-correct in real-time. We are considering to construct the PSF map database as a pre-process. The PSF map database knows how blur the displayed image is corresponding to the display's position. After constructing the PSF map database, only we have to do is tracking the display. When we handle this problem, we can experience Projection Based AR in the real sense of the term.

References

[1] M. Ashdown, T. Okabe, I. Sato, and Y. Sato. Robust content-dependent photometric projector compensation. In *CVPRW '06: Proceedings of the 2006 Conference on Computer Vision and Pattern Recognition Workshop*, page 6, 2006. 1

[2] O. Bimber, A. Grundhofer, T. Zeidler, D. Danch, and P. Kapakos. Compensating indirect scattering for immersive and semi-immersive projection displays. In *Proceedings of the IEEE Virtual Reality Conference (VR 2006)*, page 21, 2006. 1

[3] O. Bimber, G. Wetzstein, A. Emmerling, and C. Nitschke. Enabling view-dependent stereoscopic projection in real environments. In *Proceedings of the Fourth IEEE and ACM International Symposium on Mixed and Augmented Reality(ISMAR'05)*, pages 14–23, 2005. 1

[4] M. S. Brown, P. Song, and T.-J. Cham. Image preconditioning for out-of-focus projector blur. *Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition(CVPR'06)*, pages 1956–1963, 2006. 1

[5] A. Emmerling. Multifocal projection: A multiprojector technique for increasing focal depth. *IEEE Transactions on Visualization and Computer Graphics*, 12(4):658–667, 2006. Member-Oliver Bimber. 1

[6] S. Nayar, H. Peri, M. Grossberg, and P. Belhumeur. A Projection System with Radiometric Compensation for Screen Imperfections. In *ICCV Workshop on Projector-Camera Systems (PROCAMS)*, Oct 2003. 3

[7] Y. Oyamada and H. Saito. 3d face display with projection based augmenget reality. In *Proceegings of the 2006 IEICE General Conference(In Japanese)*, 2006. 1

[8] R. Raskar, J. van Baar, T. Willwacher, and S. Rao. Quadric transfer for immersive curved screen displays, 2004. 1, 3

[9] J.-P. Tardif, S. Roy, and M. Trudeau. Multi-projectors for arbitrary surfaces without explicit calibration nor reconstruction. *Proceedings of the Fourth International Conference on 3-D Digital Imaging and Modeling(3DIM2003)*, page 217, 2003. 1, 3

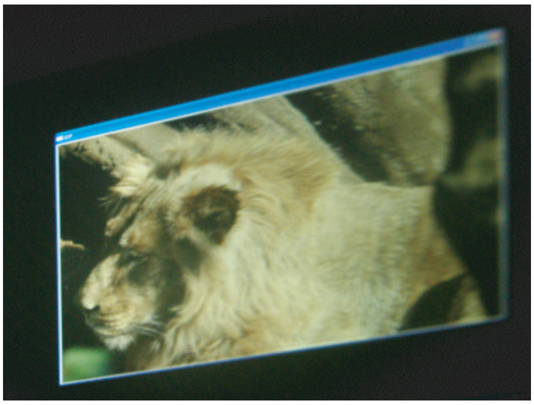
[10] T. Yotsukura, F. Nielsen, K. Binsted, S. Morishima, and C. S. Pinhanez. Hypermask: Talking head projected onto real object. *The Visual Computer*, 18(2):111–120, 2002. 1

[11] L. Zhang and S. Nayar. Projection defocus analysis for scene capture and image display. In *SIGGRAPH '06: ACM SIGGRAPH 2006 Papers*, pages 907–915, 2006. 1

(a) Original image

(b) Displayed (a)

(c) Projected image pre-corrected by Brown's method



(d) Displayed (c)

(e) Projected image pre-corrected by our method

(f) Displayed (e)

Figure 10. Projected images and screen images

(Left column) Projected images
(Right column) Screen images captured by the camera

(a) Lion’s whiskers in Fig.10(a)   (b) Lion’s fur in Fig.10(a)

(c) Lion’s whiskers in Fig.10(b)   (d) Lion’s whiskers in Fig.10(d)   (e) Lion’s whiskers in Fig.10(f)

(f) Lion’s fur in Fig.10(b)   (g) Lion’s fur in Fig.10(d)   (h) Lion’s fur in Fig.10(f)

Figure 11. Specified regions of the each screen image and the original image

(Top row) Lion’s whiskers and furs of the original image
(Middle row) Lion’s whiskers of the screen images
(Bottom row) Lion’s fur of the screen images