Dynamic Correction of Image Distortions for a Kinect-Projector System

Jihoon Park	Seonghyeon Moon	Kwanghee Ko
Gwangju Institute of Science and Technology 123 Cheomdangwagiro, Bukgu Gwangju, Republic of Korea (61005) parkbi16@gist.ac.kr	Gwangju Institute of Science and Technology 123 Cheomdangwagiro, Bukgu Gwangju, Republic of Korea (61005) moonsh@gist.ac.kr	Gwangju Institute of Science and Technology 123 Cheomdangwagiro, Bukgu Gwangju, Republic of Korea (61005) khko@gist.ac.kr

ABSTRACT

This paper addresses the problem of correcting distortion in an image projected onto a target screen without using a camera. Unlike a camera-projector system that projects a special pattern on the screen and acquires it using a camera for distortion correction, the proposed system computes the amount of correction directly from the geometric shape of the screen, which is captured by a Kinect device, a scanner that produces 3D points of the screen shape. We modified the two-pass rendering method that has been used for the projector-camera system. An image is created on the Kinect plane. Next, the image is mapped to the 3D points of the screen shape obtained by the Kinect device using the ray-surface intersection method. Finally, a corrected image is obtained by transforming the image on the 3D point set to the projector plane. The proposed method does not require a marker or a pattern and can be used in a dynamic environment where the shape of the screen changes, or either the viewer's position and direction change. Various tests demonstrate the performance of the proposed method.

Keywords

Geometric Alignment, Projector-Kinect system, Projection in dynamic environment, C^2-continuous surface.

1 INTRODUCTION

Projection displays an image or information on the surface of an object such as a flat white screen or a wall. It has been used in various fields such as education, presentation, and performing arts. A projector is installed in such a way that the optical axis of the projector and the flat screen are perpendicular to each other. Sometimes a non-flat surface is considered such as displaying images or other contents on the outer surface of a building in a media-art performance show. In this case the images from the projector should be adjusted to minimize any distortion caused by the relative relation between the geometric shape of the surface, the positions and directions of the projector and the viewer. Many researchers have focused on methods for automatic correction of distortions in an image that is projected on the surface of an object considering the projector and the viewer.

Most methods use a projector-camera system, a system consisting of a projector for projecting an image and a camera that captures the geometric distortion. The distortion is then corrected based on the detected information [An16]. [Ahm13] are focused on the approximation of a shape using a higher order B-spline surface. The method corrects the distortion through the difference between the two sets of feature points; one contains the origin points on the projector plane, and the other has the extracted features from the pattern image. The extracted points are then transformed to the projector plane through homography. Thereby, the distortion in the image can be adjusted by warping the original image based on the difference between the origin and the corresponding transformed feature points. Kaneda *et al.* [Kan16] considers the case that the projector's optical axis and the screen are not perpendicular to each other when a planar screen is used. In this method, a distorted image and the geometry information of the screen are obtained through a camera and a Kinect device. The method also uses homography to correct the distorted image. Unlike the previous two methods, depth information obtained by the Kinect is used to decide the orientation of the planar screen. A

normal vector is estimated from the depth values. Then, two perpendicular vectors that define the plane of the screen are obtained using the normal vector. The shape of the corrected image is determined using the vectors by considering the relative geometric relation to the optical axis. The method adjusts the distorted image into a rectangular one with respect to the orientation of the screen. However, the method is only focused on the planar screen only.

Methods using surface reconstruction are similar to the aforementioned ones. The main concept is to transform a rendered image in the world space to the projector space. This process is called the two pass rendering method [Ras98]. First, an image is mapped to the surface of the screen model through rendering. Here, the screen model can be modeled directly or reconstructed from a set of measured points. Then, the image on the model surface is transformed to the projector plane. In [Bro05], [Ras99] and [Ras00], the two pass rendering method for a projector-camera system is used for correcting the distortion in an image. Here, a 3D screen model is obtained by the camera pair, and a projection matrix is established from the relation between the projector and the position of the viewer. Then, an image without distortion is created. The first pass is finished by rendering the desired image on the 3D screen model. Next, the rendered image is transformed to the projector plane through the projection matrix. The image transformed to the projector plane becomes the corrected image at the viewer's viewpoint. If the projection matrix and the position of the viewer are known, the method can correct the distorted image whenever the viewer moves.

The methods based on the projector-camera system, however, do not extend well to the dynamic environment where the lighting condition, the shape of the screen, and the viewer's position and orientation change. The change of the shape of the screen may alter the reflection pattern of the screen; the intensity or color of the screen in the image may change. The same phenomenon happens when the view position changes. In such cases, the feature extraction step, which is used for estimating 3D points of the shape of the screen, may fail to detect the features for the estimation because image processing methods used in the feature detection are not stable to the lighting condition and therefore the 3D shape may not be obtained robustly. In addition, the projector-camera system requires projecting a pattern on the surface of the screen for generating the 3D shape of the screen. It means that whenever the shape of the screen or the position and the orientation of the viewer change, the pattern should be projected and processed to obtain the geometric shape, which means that we cannot expect a continuous projection of images on the screen.

Kundan and Reddy [Kun13] proposed a geometric compensation method of a non-planar surface with a Kinect device. The compensated image is obtained by warping a mesh model of the target surface that is calculated from a depth map generated by the Kinect. The method does not require any pattern for acquiring the 3D surface model because it can be directly obtained by the Kinect.

We propose a method for correcting the image distortion when an image is projected onto the surface of a screen. Here, we consider two cases: the change of the screen shape with a static viewer's position and orientation, and the change of the viewer's position and orientation with a static screen position and shape. The proposed method uses a Kinect v2 device for acquisition of the 3D shape of the screen and a projector for image projection on the screen. No projection and acquisition of a pattern are required and the method can obtain the 3D shape of the screen in real time.

The contributions of the proposed method are twofold. First, the proposed method can handle the cases mentioned above. In addition, it can process the dynamically changing environment in the distortion correction computation. Second, a registration based method is proposed to estimate the relative change of the viewer's position and orientation in the distortion correction step. Based on these technical contributions, the method is demonstrated with several examples.
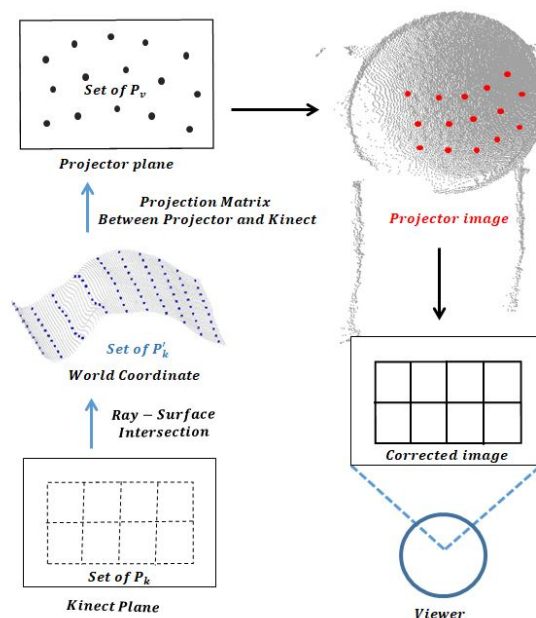
2 PROPOSED METHOD

Figure 1: Overview of correcting a distorted image.

In this work, we consider a projector-Kinect system and modify the two pass rendering method introduced

in [Ras98] for the proposed system. We take some assumptions. First, the Kinect coordinate coincides with the viewer's. Second, the screen is a C^2-continuous surface. In addition the screen is always in front of the Kinect. Figure 1 shows the overall procedure of the proposed method.

The 3D geometric shape of the screen is obtained using the Kinect device. Here, we assume that the viewer's position and direction are the same as those of the Kinect device. An image that is needed to be projected on the screen is assumed to be in the Kinect plane, a virtual 2D plane that the viewer watches. The pixels of the image in the Kinect plane are denoted P_k. They are mapped on the screen surface through the ray-surface intersection to produce P'_k, which correspond to the rendered image on the 3D surface. Then P'_k are transformed to the projector plane by a transformation matrix to yield P_v. When P_v is projected on the surface of the screen, an image whose distortion has been corrected is displayed on the surface.

2.1 Kinect Device

There are two types of Kinect, Kinect v1 and Kinect v2. Kinect v2 is an improved version of Kinect v1. The devices are affordable and easy to use. However, they have inherent noise in the measurement, which prevents them from being used in applications that require high accuracy. The quantitative analysis of the noise of Kinect v1 and Kinect v2 over the scan distance is made as shown in Fig. 2 [Pag15]. According to [Pag15], Kinect v2 has the noise of $0.02m \sim 0.03m$ in the maximum depth of $5m$, and the point cloud obtained by Kinect v2 becomes sparse as the depth increases.
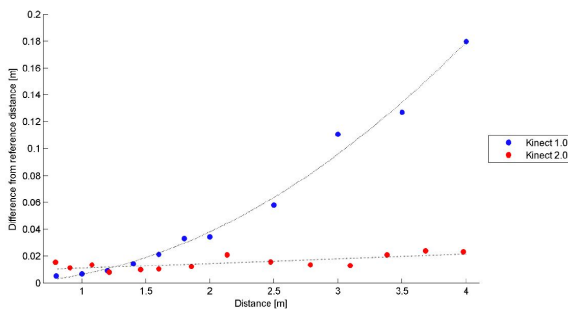

Figure 2: Comparison of the noise levels of Kinect v1 and Kinect v2 with respect to the distance to the target. [Pag15]

Moreover, the maximum range that Kinect v2 can robustly cover is 5m according to the specification of the device and [Pag15]. Therefore, the effective space where the proposed system works would be limited.

2.2 Geometric Correction

The projection matrix is necessary to transform an image in the Kinect space to the plane of a viewer. We calculate the projection matrix $\mathbf{T}(x)$ relating the initial position of the viewer to the projector once before correction [Jon14]. Next, we obtain the shape of the screen for distortion correction. The point set representing the screen is extracted from the depth values measured by the Kinect. Grid points, which cover an image to be projected on the screen, are created on the Kinect plane. Next, rays are shot from the Kinect origin $(0,0,0)$ through each of the grid points. From the rays, a virtual frustum is created, which intersects the point set of the measured point set. Then, the points within the frustum are collected and used as the points that lie on the screen. For this process, the kd-tree data structure is used for an efficient computation [Pha10]. The angle of the line segment connecting a point in the measurement and the Kinect origin is considered. Namely, the angles of the segment with respect to the *xy*, *xz* and *yz* planes are computed. If the angles are close to those of a ray, then the point is considered to be within the frustum and taken as a point on the screen.

The Kinect device produces measurements with some noise. The noise may cause a serious problem in the distortion correction process. In particular computation of the intersection between the screen and the ray, one of the steps in the proposed method, is mostly compromised when the raw measured points are used directly. Therefore, the noise level in the measurement data set should be controlled. In this study, we use a hierarchical B-spline approximation method by [Lee97] to avoid such a problem. Here, the approximation of the points using a B-spline surface is used as a low pass filter. Suppose that we have a set of points with some noise. Representing the shape defined by the points accurately may require a function of high order or a B-spline surface with many control points because the high frequency components of the noise should be considered. Unless they are part of the surface, they do not have to be represented in the surface definition and should be smoothed out to obtain the underlying geometric structure. A B-spline surface with a reasonable number of control points can be used to filter out the high frequency noise components and to approximate the given points with satisfactory accuracy. For this purpose we consider the hierarchical B-spline approximation method. Approximation is started with a small number of control points such as 4×4. If the error of approximation is larger than the user defined tolerance, the control net is refined to be 8×8, and the points are approximated again. This refinement step is repeated until a surface with reasonable accuracy is obtained. The approximated surface is used as a virtual screen in the proposed method. Next, an image on the Kinect plane is rendered on the virtual screen to obtain the positions of the pixels of the image in the world coordinate space. The positions correspond to the intersections between the rays and the virtual surface. The virtual

screen is given as a cubic B-spline surface, and the intersection points P'_k between the rays and the surface are calculated through Newton-Raphson method [Pre92]. Finally, the corrected points P_v on the projector plane is obtained by transforming P'_k to the projector plane through the projection matrix $\mathbf{T}$. A viewer can see the desired image by projecting P_v to the screen.

The aforementioned process can correct the distortion of an image in a static environment, where the shape of the screen and the position and orientation of the viewer do not change. A new projection matrix must be computed to show a corrected image on the screen whenever either the shape or the position and direction change. This computation process requires the calibration process, which hinders the continuous projection of an image on the screen. A solution to this problem is proposed for two cases.

2.3 Case 1: Change of Screen Shape with Static View Position and Orientation

When the shape of a screen is changed, the parameters of the projection matrix are constant because the orientation and position of the projector and the viewer (Kinect) do not change. Instead, new intersection points with respect to the changed shape are computed. They are obtained in the same way as presented in the previous section. However, if the points are calculated at each frame, the computation time is increased, and the frame rate of projection is decreased. So, it is necessary to determine if the shape of the screen has been changed or not. First, the mean of the depth values is calculated at the current frame. Then, the difference between the current mean and the previous values is calculated. If the difference value is lower than the threshold, we determine that the current shape be equal to the previous one, do not perform the step of obtaining the new intersection points at the current frame and return to the acquisition step. Otherwise, we decide that the current shape be changed. Hence, new intersection points are calculated at the current frame. Then, a corrected image is obtained with respect to the changed shape by multiplying the projection matrix to the new intersection points. Figure 3 shows the overall process for Case 1.

2.4 Case 2: Change of Position and Orientation of Viewer with Static Screen

When a viewer moves, the parameters of the projection matrix should be changed accordingly. The projection matrix consists of an intrinsic and an extrinsic matrices. The intrinsic matrix is related to the device properties such as the focal length, the principal point, etc. For this reason, it is not influenced by the position or orientation of the viewer. The extrinsic matrix, however, should be modified with respect to the new viewer's position and

P_t : Mean of the depth value at the current frame
P_{t-1}: Mean of the depth value at the previous frame
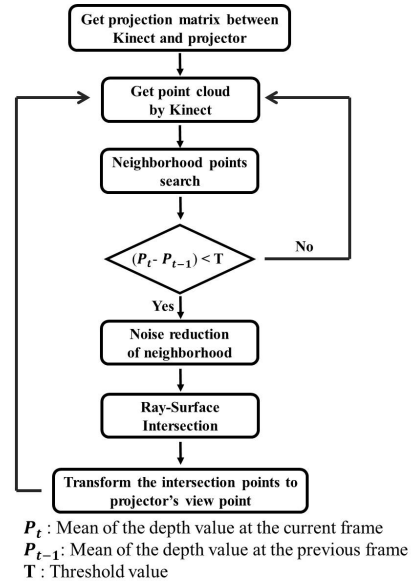$\mathbf{T}$: Threshold value

Figure 3: Flowchart for handling Case 1.

orientation because it captures the relation between the viewer and the projector. In this study, a registration algorithm is employed to estimate the relation.

Suppose that the viewer has moved from pos_1 to pos_2. At pos_1, the shape of the screen S_1 has been measured by the Kinect. After the movement, the screen shape S_2 is measured by the Kinect at pos_2. The relation of the viewer's position and orientation can be estimated from S_1 and S_2. Since S_1 and S_2 are point clouds with some overlap of the same shape, they can be registered to form one point cloud of the screen shape in the reference coordinate system, which produces the rigid body transformation that registers S_2 onto S_1 as closely as possible. This transformation provides the relative relation of the viewer at pos_1 and pos_2, which can be translated into the relation of the viewer at the new position to the projector. The point-to-plane algorithm [Low04] is used for computing the transformation matrix to register S_1 and S_2. The new projection matrix at pos_2 is calculated by multiplying the inverse of $\mathbf{M}$ to the extrinsic matrix at pos_1. The process should only be performed when the viewer's position and orientation have been changed much, which can be decided by checking the change of depth values as is performed for Case 1 because the change of the viewer's position and orientation is equivalent to the relative change of the shape of the screen. The step by step procedure of the proposed method for Case 2 is illustrated in Fig. 4.

3 RESULT AND DISCUSSIONS

The proposed method is implemented in C++. The workstation used for testing has a 4-GHz Intel Core i7 CPU with 8GB RAM. We use a Microsoft Kinect v2 to obtain a point cloud and a Panasonic PT-DX1000
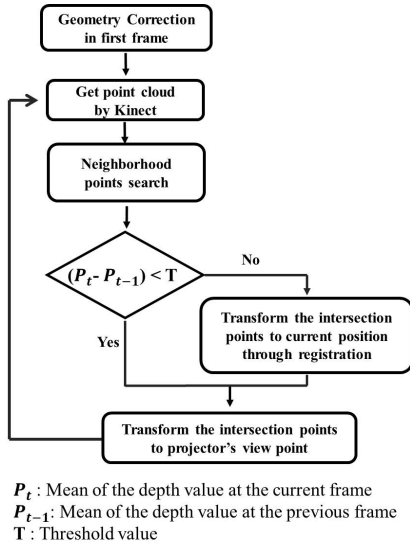
P_t : Mean of the depth value at the current frame
P_{t-1} : Mean of the depth value at the previous frame
T : Threshold value

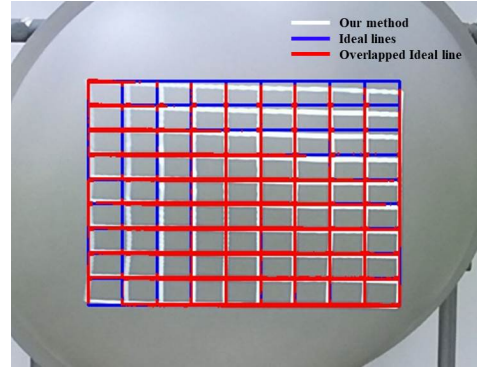Figure 4: Flowchart for handling Case 2.

Figure 5: Corrected image by the proposed method. The blue lines in the figure show an ideal grid and the white lines show a corrected grid. The overlapped ideal lines are represented in red color.

projector. Two types of screen are considered in the test. One is a spherical screen. The distance between the Kinect and the spherical screen is determined such that the projected image covers the maximum area of the screen. The other is a curtain screen the shape of which can be changed arbitrarily. Here, the distance between the Kinect and the screen is about 2 meters. To simulate the dynamic environment, the spherical screen is moved or rotated, and the shape of the curtain was changed with a hand by pushing or pulling it behind. A viewer also moves within a valid range of the projector.

We use the overlap ratio between the ideal and the corrected images for error evaluation, which is denoted R. The overlap ratio quantifies how much similar the corrected image is to the ideal one. The ideal image is represented as a grid whose numbers of the columns and rows are equal to the feature points. The ratio is calculated by dividing the number of the overlapped pixels by the number of the ideal pixels. It is expressed as

$$R = \frac{\sum_i P_{overlap}}{\sum_i P_{ideal}} \times 100, \qquad (1)$$

where $P_{overlap}$ are the pixels in the overlapping area and P_{ideal} are the ideal pixels. Figure 5 shows the corrected image by the proposed method.

The overlap ratios before and after noise reduction are 66.66% and 89.5%, respectively. The result without noise reduction shows that the grid lines are not straight due to the inaccurate intersection points as shown in Fig. 6(a). On the other hand, the result after noise reduction has a higher overlap ratio as 20% as shown in Fig. 6(b).

Table 1 shows the computation time of each step for Cases 1 and 2. The three steps of searching the adjacent points, generating a surface with noise reduction
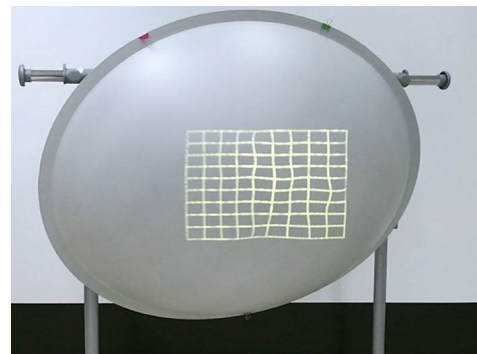
(a) Corrected image before noise reduction
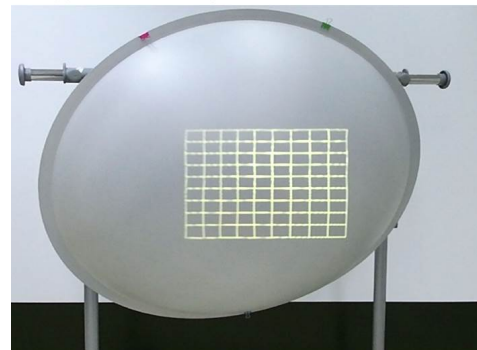
(b) Corrected image after noise reduction

Figure 6: Corrected images with and without noise reduction. The resolution of figures is [395 x 307]

and calculating intersection points through Newton-Raphson method are the same for the two cases. The process of searching and generating only takes less than 0.01s. Most of the computation time is spent for registering the two point clouds as 0.78s. The computation time is affected by the number of grid points and a generated cubic B-spline surface.

The number of grid points increases the computation time of the entire process but can contribute to the higher accuracy of correction. More grid points can obtain more accurate information about the distortion

Case 1	
Process	Computation time(second)
Search the region	0.011s
Noise reduction	0.003s
Ray-surface intersection	0.0571s
Total	Over 0.06s
Case 2	
Process	Computation time(second)
Search the region	0.011s
Noise reduction	0.003s
Registration	0.78s
Total	Under 1.00s

Table 1: Computation time of the proposed method

because more intersection points are used. However, a large number of grid points may result in a discontinuous surface and makes the intersection computation fail from time to time. Therefore, a tradeoff between the number of grid points, the computation time and the accuracy should be taken into account. We have chosen the 10×10 grid points empirically in the proposed method. We tested various numbers of grid points to analyze the influences to the distortion correction. After a series of experiments, we found that the accuracy is almost unchanged although the number of grid points is increased from 10×10. However, the computation time is quite sensitive to the number of grid points because intersection should be computed the number of times proportional to the number of grid points. For example, the accuracy of the result is converged to 89.5% although the number of grid points is increased from the 10×10 grid points in Fig. 6. However, the computation time is significantly increased. If the number of grid points is changed to 15×15 from 10×10, the computation time grows to $0.79s$ from $0.06s$.

A curtain is used to test the performance of the proposed method as shown in Fig. 7. In this test, the shape of the curtain is changed, and the position and orientation of the viewer are fixed. Two different types of shapes are considered for the experiments. Figures 7-(a) and (b) show a distorted and a corrected images on the curtain of one shape, respectively. Similarly, Figures 7-(c) and (d) show images before and after distortion correction on another shape of the curtain. Here, the resolution of the images is 376×297. As shown in the figures, the proposed method corrected the distortions of the images and produced corrected ones on the different shapes of the curtain, respectively. The correction was performed at the speed of 10 FPS (Frames Per Second).

Figure 8 shows the case when two different screen shapes are considered. In this test, a spherical and a curtain are used with a different image. As shown in the figure, the proposed method successfully corrects

the distortions and shows the corrected images on the screen.

Figure 9 shows the corrected images when the position and orientation of the viewer change. Here, the shape of the curtain is maintained. The method successfully produces the corrected images at the three different positions and orientations of the viewer (pos1, pos2, and pos3) as shown in Figs. 9-(a), (b), and (c). In this test, the resolution of the figures used in this test is 376×297. The process runs at about 1 FPS. The drop of FPS in this case mostly attributes to the estimation of transformation between two positions using the registration method.

4 CONCLUSION

In this study, we propose a method of correcting the distorted projector image in a dynamic environment using a Kinect device. The dynamic environment includes the two cases: that the shape of the screen changes and that the position and orientation of the viewer change. Additionally, the proposed method can compensate the distortion of the two cases simultaneously during execution.

The proposed method uses the Kinect for obtaining the 3D shape of the screen in real time, which is an advantage of the proposed method over others that use a camera-projector configuration. Therefore, the acquisition step is not influenced by lighting conditions. Moreover, when the position and orientation of the viewer change, the proposed method estimates the projection matrix only by considering the relative motion between the positions and orientations before and after the viewer moves, which is computed by a registration method.

However, there are a few limitations of the proposed method. The method cannot differentiate Cases 1 and 2 automatically because they address the same problem from the theoretical viewpoint. It means that we must select one of the two cases before executing the proposed method. As a possible solution, an additional sensor such as an accelerometer or a gyro sensor can be employed to detect the position or the orientation change of the viewer. Moreover, the implementation of the method needs to be refined for improving computation time to yield a higher frame rate for real-time operation. Alternatively, a parallel computation scheme can be introduced in the intersection computation between a ray and a surface for reducing the computation time. Finally, the proposed system has been designed to consider one projector, which limits the screen area that the system can cover. Overcoming the range of one projector can be achieved by using multiple projectors, and each of the projectors can be handled individually using the multiple thread framework. These problems need to be solved before the proposed method is used in practice, which is recommended for future work.

5   ACKNOWLEDGMENTS

6   REFERENCES

[Ahm13] A. Ahmed, R. Hafiz, M. M. Khan, Y. Cho and J. Cha, Geometric Correction for Uneven Quadric Projection Surfaces Using Recursive Subdivision of Bezier Patches, ETRI Journal 35, 6 (2013)

[An16] H. An, Geometrical Correction for Arbitrarily Curved Projection Surface By Using B-Spline Surface Fitting, Master's thesis, Gwangju institute of Science and Technology, 2016

[Kan16] T. Kaneda, N. Hamada and Y. Mitsukura, Automatic Alignment Method for Projection Mapping on Planes with Depth, 2016 IEEE 12th International Colloquium on Signal Processing Its Applications (CSPA), pp.111-114, 2016.

[Ras98] R. Raskar, G. Welch, M. Cutts, A. Lake, L. Stesin and H. Fuchs, The Office of the Future: A Unified Approach to Image-based Modeling and Spatially Immersive Displays, SIGGRAPH '98, pp.179-188, 1998.

[Ras00] R. Raskar, Immersive planar display using roughly aligned projectors, Proceedings IEEE Virtual Reality 2000 (Cat. No.00CB37048), New Brunswick, NJ, pp.109-115, 2000.

[Ras99] R. Raskar, M. S. Brown, R. Yang, W.-C. Chen, G. Welch, H. Towles, B. Scales and H. Fuchs, Multi-projector displays using camera-based registration, Visualization '99. Proceedings, San Francisco, CA, USA, pp.161-522, 1999.

[Kun13] S. D. Kundan and G. R. M. Reddy, Projection and Interaction with Ad-hoc Interfaces on Non-planar Surfaces, 2013 2nd International Conference on Advanced Computing, Networking and Security, Mangalore, pp.1-6, 2013.

[Bro05] M. Brown, A. Majumder and R. Yang, Camera-based calibration techniques for seamless multiprojector displays, in IEEE Transactions on Visualization and Computer Graphics, vol. 11, no. 2, pp.193-206, March-April 2005.

[Jon14] B. Jones, R. Sodhi, M. Murdock, R. Mehra, H. Benko, A. Wilson, E. Ofek, B. MacIntyre, N. Raghuvanshi and L. Shapira, RoomAlive: Magical Experiences Enabled by Scalable, Adaptive Projector-camera Units, Proceedings of the 27th Annual ACM Symposium on User Interface Software and Technology, Honolulu, Hawaii, USA, pp.637-644, 2014.

[Pha10] M. Pharr and G. Humphreys, Physically Based Rendering, Second Edition: From Theory To Implementation, 2nd ed. Morgan Kaufmann Publishers Inc., 2010.

[Pag15] D. Pagliari and L. Pinto, Calibration of Kinect for Xbox One and Comparison between the Two Generations of Microsoft Sensors. Zlatanova S, ed. Sensors (Basel, Switzerland). 2015.

[Lee97] S. Lee, G. Wolberg and S. Y. Shin, Scattered data interpolation with multilevel B-splines, IEEE Transactions on Visualization and Computer Graphics, vol. 3, no. 3, pp.228-244, 1997.

[Pre92] W. H. Press, S. A. Teukolsky, W. T. Vetterling and B. P. Flannery, Numerical Recipes in C, 2nd ed. Cambridge University Press, 1992.

[Low04] K. L. Low, Linear Least-Squares Optimization for Point-to-Plane ICP Surface Registration. Chapel Hill, University of North Carolina, vol. 4, 2004.

(a) Corrected image on the spherical screen. The resolution : [427,326]

(b) Corrected image on the curtain. The resolution : [676 x 512]

Figure 8: Results of our method using the curtain and the spherical screen.

(a) Distorted image on shape$_1$

(b) Corrected image on shape$_1$

(c) Distorted image on shape$_2$
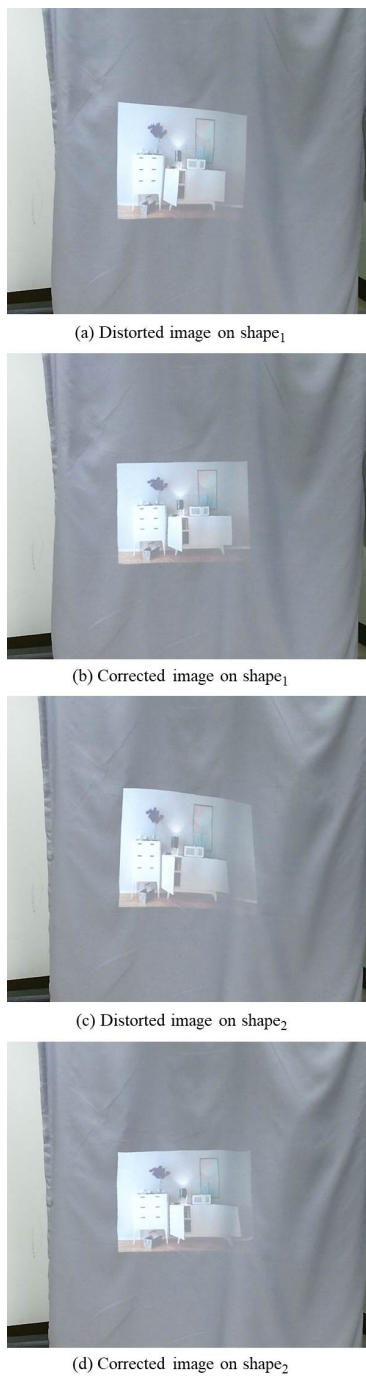

(d) Corrected image on shape$_2$

Figure 7: Distorted and the corrected images by our method on the different shapes of a curtain. The images of (a)-(b) and (c)-(d) use the same shape of the screen, respectively. (a) and (c) show the distorted images with respect to the shape and (b) and (d) show the corrected images, respectively. The resolution of figures is 376×297.

(a) Corrected image at pos$_1$

(b) Corrected image at pos$_2$

(c) Corrected image at pos$_3$

Figure 9: Results of the distortion correction when the position and orientation of the viewer change. (a), (b) and (c) show the corrected images at three different positions and orientations of the viewer. The resolution of the figures used in this test is 376×297.

[KJ13]   Nilay Khatri and Manjunath V. Joshi. Image super-resolution: Use of self-learning and gabor prior. In Kyoung Mu Lee, Yasuyuki Matsushita, James M. Rehg, and Zhanyi Hu, editors, *Computer Vision – ACCV 2012*, pages 413–424, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg.

[KJ14]   N. Khatri and M.V Joshi. Efficient self-learning for single image upsampling. In *22nd International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision (WSCG 2014)*, pages 1–8, 2014.

[LHG+10]   Xuelong Li, Yanting Hu, Xinbo Gao, Dacheng Tao, and Beijia Ning. A multiframe image super-resolution method. *Signal Processing*, 90(2):405 – 414, 2010.

[MPSC09]   Dennis Mitzel, Thomas Pock, Thomas Schoenemann, and Daniel Cremers. Video super resolution using duality based tv-l1 optical flow. *Proceedings of the 31st DAGM Symposium on Pattern Recognition*, pages 432–441, 2009.

[NM14]   Kamal Nasrollahi and Thomas B. Moeslund. Super-resolution: A comprehensive survey. *Machine Vision Applications*, 25(6):1423–1468, August 2014.

[NMG01]   Nhat Nguyen, P. Milanfar, and G. Golub. A computationally efficient superresolution image reconstruction algorithm. *IEEE Transactions on Image Processing*, 10(4):573–583, Apr 2001.

[NTP17]   Mattia Natali, Giulio Tagliafico, and Giuseppe Patan. Local up-sampling and morphological analysis of low-resolution magnetic resonance images. *Neurocomput.*, 265(C):42–56, November 2017.

[PC12]   P. Purkait and B. Chanda. Super resolution image reconstruction through bregman iteration using morphologic regularization. *IEEE Transactions on Image Processing*, 21(9):4029–4039, Sept 2012.

[RIM17]   Y. Romano, J. Isidoro, and P. Milanfar. Raisr: Rapid and accurate image super resolution. *IEEE Transactions on Computational Imaging*, 3(1):110–125, March 2017.

[RU90]   S. P. Raya and J. K. Udupa. Shape-based interpolation of multidimensional objects. *IEEE Transactions on Medical Imaging*, 9(1):32–42, Mar 1990.

[SLJT08]   Qi Shan, Zhaorong Li, Jiaya Jia, and Chi-Keung Tang. Fast image/video upsampling. *ACM Transactions on Graphics*,

27(5):153:1–153:7, December 2008.

[Sob68]   Feldman G. Sobel, I. A 3x3 isotropic gradient operator for image processing. *Stanford Artificial Intelligence Project*, 1968.

[TAe17]   R. Timofte, E. Agustsson, and L. V. Gool et.al. Ntire 2017 challenge on single image super-resolution: Methods and results. In *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 1110–1121, July 2017.

[TDSVG15]   Radu Timofte, Vincent De Smet, and Luc Van Gool. A+: Adjusted anchored neighborhood regression for fast super-resolution. *Computer Vision – ACCV 2014: 12th Asian Conference on Computer Vision, Singapore, Singapore, November 1-5, 2014, Revised Selected Papers, Part IV*, pages 111–126, 2015.

[War03]   Greg Ward. Fast, robust image registration for compositing high dynamic range photographs from handheld exposures. *Journal of Graphics Tools*, 8:17–30, 2003.

[YMY14]   Chih-Yuan Yang, Chao Ma, and Ming-Hsuan Yang. Single-image super-resolution: A benchmark. In *Proceedings of European Conference on Computer Vision*, 2014.

[YSL+16]   Linwei Yue, Huanfeng Shen, Jie Li, Qiangqiang Yuan, Hongyan Zhang, and Liangpei Zhang. Image super-resolution: The techniques, applications, and future. *Signal Processing*, 128:389 – 408, 2016.

[YZS12]   Q. Yuan, L. Zhang, and H. Shen. Multiframe super-resolution employing a spatially weighted total variation model. *IEEE Transactions on Circuits and Systems for Video Technology*, 22(3):379–392, March 2012.

[ZC14]   H. Zhang and L. Carin. Multi-shot imaging: Joint alignment, deblurring, and resolution-enhancement. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 2925–2932, June 2014.

[ZWZ13]   H. Zhang, D. Wipf, and Y. Zhang. Multi-image blind deblurring using a coupled adaptive sparse prior. In *2013 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1051–1058, June 2013.