An Effective Image Distortion Correction Method using Depth Sensor

YunSang Jeong[1], JiHye An[2] and Jinho Park[1]

[1]*Global School of Media, Soongsil University, Sangdo – ro 369,
Dongjak-gu, Seoul, Korea*
[2]*Department of ICMC Convergence Technology, Soongsil University, Sangdo – ro
369, Dongjak-gu, Seoul, Korea*
{kunka8271, comma4301, c2alpha}@gmail.com

Abstract

We utilize Kinect v2, which is a sensor deeply, to provide a way to recognize walls and interpolate mapping images accordingly. It recognizes the angle and shape of the wall and basically interpolates the recognized form. First, we will use depth sensors to receive wall depth data. The data received in this way is converted to primary usable data via preprocessing. By using subsequent deformed data, we calculate by deducing the extent and degree of coking of the wall. Using the calculated result, Unity 3D will correct the image and interpolate. We show that it is possible to propose a method which can more easily map through this process through research and experiment.

Keywords: *Projection Mapping, Depth Sensor, Interpolation, Surface recognition, Interpolation of Image*

1. Introduction

Projection mapping is the process of applying photographs and images to a real object using a projector. [1] Recently, it has developed into the field of new art. Narrowly it is made from simple objects such as cubes and walls mapped largely to buildings so that various people can see and enjoy it. In our case, such projection mapping is not limited to such art work, and research has proceeded so that general people can solve problems as if viewing images or installing projectors. Recently, projection mapping in various places will proceed in various ways. Various works by artists and projection mapping art [2-10] using various methods and techniques are being produced at different locations. Projection mapping will serve as a medium for delivering video, not just a display of work. However, when the projection mapping is performed, the shape of the screen varies depending on the position and direction of the projector. These problems cause designers and developers to need repetitive modifications and additions. We will use depth sensor to solve this image distortion. We propose a method of realigning the image by recognizing the angle of the wall through the depth sensor. We use inexpensive and easily available equipment and programs like Kinect and Unity. Even beginners of this design and mapping can easily use it.

2. Related Work

Research on existing projection mapping has focused on research on problem solving that occurs when mapping over a specific shape or color and mapping on a non-stable plane. A research conducted by Oliver Bimber has been undertaken to address the multi-focus problem that occurs when mapping to non-planar surfaces. [11-13] In addition to this, a

research was conducted to perform color correction by mapping on a specific color or texture. [14-15] Then, the work was conducted by integrating the above research into a single system. [16] Convergence research such as trekking [17-18] of objects is under way, not limited to video mapping. There is also research on mapping and correcting objects for the above-mentioned projection mapping. [19] These Researches are carried out to make mapping more convenient, and it helps the mapping to be carried out later and effective mapping in more various environments. When displaying projection mapping, there are many limitations such as light, plane, and fixed positions. Therefore, researches are underway to reduce these constraints and to enable projection mapping in a wider variety of environments. We conducted the study focused on reducing the environmental variables according to the location of the plane and the projector, and conducting the projection mapping effectively.

3. Our Method

We describe a method for plane recognition using a depth sensor and propose a mapping correction using the method. For depth sensors, data can be quantified by digitizing the depth of each pixel unit. The depth sensor uses the Kinect V2 (Kinect), which is the most easily accessible and well developed API and library. Preprocessing depth data is used to correct distortion. After the data to be used in the preprocessed depth data is classified, distortion correction of the image is performed.

3.1. Using Kinect

The reason for using kinect v2 is because it has inexpensive and powerful performance. For Kinect v2, it shows the cheapest but good performance among the depth sensors available on Windows. The depth value is accepted as a numerical value and resolution is supported by the RGB camera at 1920 * 1080 with FHD. In addition, it has a resolution of 512 * 424 depth sensor. This helps to recognize and map wider screens. The distance that you can find the depth is up to 8M and it is suitable for use with the projector. In addition, Kinect v1 can be used. Kinect v1 is cheaper than Kinect v2, but resolution is lower and accuracy drops. Additional cables are required to use Kinect v2 for Windows. After connecting Kinect v2 and PC, I received data of Kinect v2 using Unity 3d.

Table 1. Sensor Comparison

	Kinect v1	Kinect v2
Resolution(RGB)	640 x 480	1920 x 1080
Frame Rate (RGB)	30fps	30fps * 3
Resolution(Depth)	320 x 240	512 x 424
Frame Rate(Depth)	30fps	30fps
Range of Depth	0.8 ~ 4.0m	0.5 ~ 8.0m
Fov(Horizontal, Vertival)	57 , 43	70 , 60
Various applications	X	O

Table 1 summarizes the detailed functions of Kinect v1 and Kinect v2. It can be seen that Kinect v2 can distinguish between wider resolution and larger depth than Kinect v1. Also, the Field of View is larger and it is possible to recognize wider surfaces and improve accuracy.

3.2. Connect to Unity 3d

Unity3D used the Personal version. After receiving the data of the depth of the kinetic directly from Unity, it can process and map. There is a plug-in that helps you to use Kinect v2 with Unity. This plugin continues to give Unity and Kinect v2 and calls RGB and depth data in a one-dimensional array. Also, Kinect v2's RGB camera can be used by using Unity's Open CV. We did not use RGB, so we proceeded. When Unity receives the Depth data using the corresponding plug-in, it will receive the value in the primary array.

In the case of an array to be received, the size of the array is 0 to 1024, which is 0.5 to 8.0 m. A preprocessing process is required to use the received data. We proceeded to map the image using the Unity3D Plane object. Plane objects are mapped and the objects are modified using the corrected values. Unity 3D provides a lot of features that are easy to use and convenient for designers, making it a great tool for designers who are new to mapping

3.2. Data Preprocessing

During the process of linking the Kinect with the Unity3d engine, the basic data of Kinect was pre-processed. This is because the data is likely to come into a single dimension and include noise. The first thing we did was to remove noise. In order to remove the noise, that was removed by setting the limit value on the data. Second, we have reduced the size of the data in a one-dimensional array to a desired size. This is because the size of the existing data is very large. Set the maximum value to $2^n(n = 8)$. Finally, a single dimension was replaced and visualized to the image color values of the two-dimensional array in order to verify that the data was properly entered and the depth of the data was normally entered.
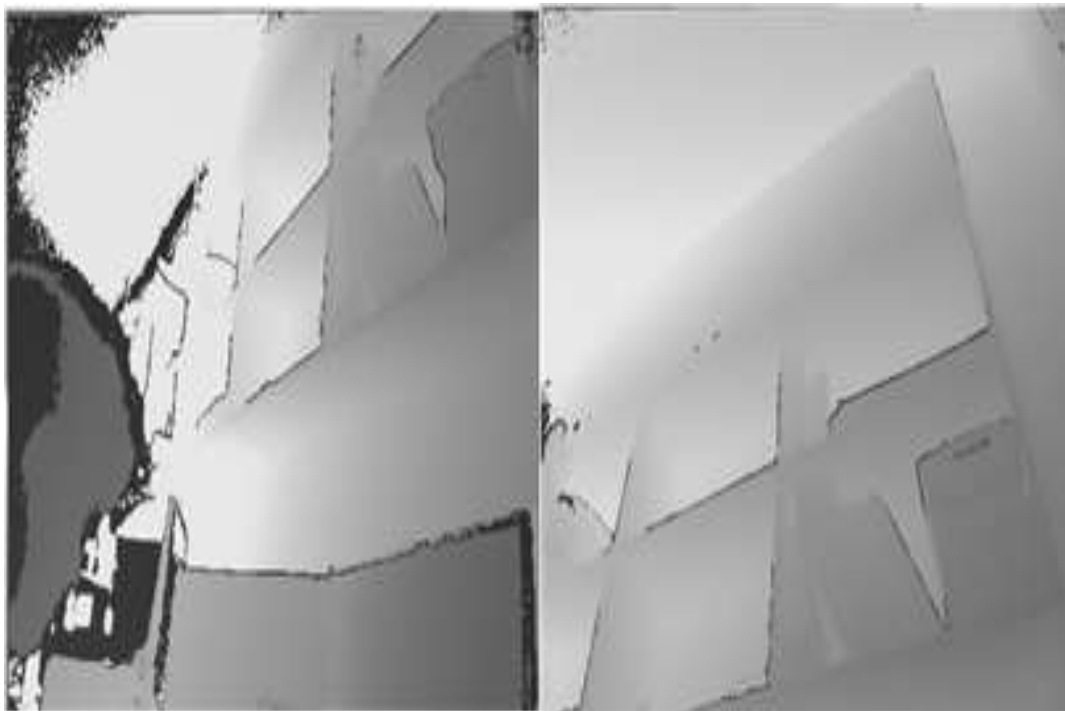

Figure 1. Visualize of Depth Data

Figure 1 shows the result of visualizing depth data. This is an image captured after mapping the texture created by setting the depth data through the preprocessing process to the color value of the pixel unit image using the Unity3d engine and mapping it to the object. The image color values of the preprocessed depth data were set in the red and green fields except blue of the image color. The lower the depth, the more black is represented. The deeper the depth, the closer the yellow color is expressed. In the case of an outline where the depth sharply changes, there are cases where the depth is expressed as 0 which is not normally inputted and appears in black.

3.2. Set Location for Depth Data Usage

In order to use preprocessed depth data, you have to set the location of use. This is because two depth points are set to calculate the depth data of the corresponding point and to correct the image distortion. We adopted the manual setting method as a way to set the point. The manual setting method is a method of setting the depth data position to be used for the distortion correction directly by the user. The manual configuration method must be performed each time the mapping is performed. The reasons and advantages of adopting the manual setting method are as follows. First, it has the advantage of using the depth data at the correct position. It is difficult to set the precise position to be mapped in case of a preset point or automatic setting. Further, there is a problem that the additional calculation of the depth data and the criteria are not clear. Second, it can be set using only depth data without RGB data. Since RGB data is not used, it is not possible to find a mapped position using RGB data and set the depth value of the corresponding position. The manual setting method can effectively solve the problem and perform the distortion correction. If RGB data is available, a monochrome mapping image can be recognized and a specific location for using depth data based on the data can be set. Since the resolution of the depth data is different from the resolution of the RGB data, it is essential to perform calibration between the two.

3.3. Image Distortion Correction

Data averaging was performed in preparation for situations where noise was not sufficiently filtered through preprocessing of the set depth data. The depth data of 9 pixels around the set two points were read and averaged. And the image distortion correction is performed using the depth data of the two averaged positions. The depth data comes in the size of the linear length of the wall to be mapped with the depth sensor. In other words, if the plane and depth sensors are not horizontal, the depth data of the two points are different. The image distortion correction operation is performed using the difference between the two depth data. Depending on the distance between these two depths, it may be difficult to determine the depth only by the difference. You have to set the weights according to these distances. In this research, we set the weight according to the X distance of two points.

$$R = I(\Phi_a, \Phi_b) \tag{1}$$

R in Equation (1) means a result of image distortion correction. In function I, depth data Φ_a, Φ_b is needed. Φ_a and Φ_b are the averaged depth data determined by the two points respectively selected by the manual setting method. I function does not need the image data to be mapped because it uses the depth data to judge how to distort the image and to correct it. The resulting value R is the degree of distortion of the image, which is in degrees (°).

$$I(\Phi_a, \Phi_b) = K(\Phi_a - \Phi_b) + \alpha \tag{2}$$

Equation (2) describes a method of calculating the function I. Basically, it is calculated using the depth difference between two points a and b. K is a proportional constant used to

determine the angle along the depth data. α is a parameter to reduce the difference between the correction result and the distortion that occurs when the actual mapping is performed, and is determined by the vertical distance between the depth sensor and the wall.

4. Experiment

We have experimented with a proposal for our method of recognizing walls using depth data and correcting and mapping images. The image was mapped onto a white wall and the experiment was carried out by excluding external factors. This is to recognize the wall and only to proceed with the calibration of the recognized part. The research was conducted on Windows 10, Intel® Core ™ i7-6700 CPU, RAM 16GB, Nvida GeForce 1070 (The experiment was conducted on Windows 10, Intel® Pentium G4600, RAM 8GB, Nvidia GeForece 1050Ti) and was implemented in parallel with the Unity3d engine. The depth sensor is Kinect. It also supports maximum distance 8M and receives depth data from 0 to 1024 data. We tested as many angles as possible. Normal distortion correction was applied at most angles, but when the angle was changed too much, the depth value did not come in normally.

Figure 2. Working Environment: Projector and Kinect

Figure 2 is a photograph of the work environment in which the experiment was conducted. The projector and the Kinect were bundled together. We have placed both devices in the same direction as possible, in order to minimize errors. In addition, the area that Kinect detects during the experiment is smaller than the area it maps to the projector, so the image was cut and used.

Figure 3. Visualized Depth Data and Distortion Correction Video

Figure 3. is a visualization image of the depth data taken at two angles and a distortion-compensated image. The image placed on the left is the depth data visualization image, and the image arranged on the right is the resulting image. In the case of the above image, the difference between the depth images is not so large, and it can be confirmed that the distortion of the result image is not large. However, in the case of the lower image, the shallower depth is black and the deep is closer to yellow. Based on this, it can be confirmed that the distortion of the result image is more severe than the above image. When mapping is performed, the more the distance between the projector and the wall to be mapped, the more the light spreads. That is, the size of the image increases. Therefore, it corrects the deep left side. That is, the deeper the depth, the more distorted the resulting image becomes. On the contrary, in the case of an image having a shallow depth, image distortion is not applied because there is a limitation on the image size of the projector. Because of these limitations, we tried to solve this problem by distorting deep images more strongly without distorting images in a shallow depth direction.

Figure 4. Mapping of Distorted Images

Figure 4 is an image of the actual mapping of the resulting images in Figure 3. Image was filmed with 3 angles in order, left side after projection mapping was not performed, right side after projection mapping. We proceeded with video shooting so that a difference of 15 to 90 degrees going down from the beginning appears. The first picture was taken at 30 degrees counterclockwise and the second picture was taken at 20 degrees clockwise in the case of the last picture at 45 degrees clockwise. After shooting the uncorrected video first, after correcting in real time, the right video was continuously shot.

During the mapping experiment, the experiment was conducted using the notebook. In the process, it is possible to perform high-speed image correction processing with a low specification with almost no time difference with the PC. It can be said that this is very useful in projection mapping work where video is the core.

Table 2. Operation Speed by Experiment Environment

	I7-Core (PC)	**G4600 (Note Book)**
Picture 1	76ms	98ms
Picture 2	68ms	101ms
Picture 3	77ms	94ms

The table below is a table derived based on Figure 4. Each picture of Figure 4 is set from Picture 1 to Picture 3 from the top to the bottom. It is a table for each computation speed in PC, which is research and development environment, and notebook, which is experimental environment. With this, it is expected that it can be fully used in video, not one picture that needs to be corrected in real time and mapped forward.

5. Conclusion

We propose that the image distortion correction method using depth sensor can effectively perform work in most situations. Correction can be made from a small angle to an angle reaching approximately 90 degrees, and it was confirmed that mapping is impossible due to the limitation of the projector by a method different from the calibration data. There are also some technical limitations such as the inability to focus the result of the correction. To solve these problems, there are also studies that attempt to perform multi-projection mapping, but it is currently impossible to take various focuses on a single projection. There is a disadvantage in that the user's manipulation is required during the execution of the experiment. It is also necessary to recalculate the constants according to the distance from the wall or the error caused by the noise of the depth sensor. Our research and experiments have been carried out to solve the problem that occurs in a bent state. Calibration and mapping of complexly curved surfaces is impossible. We expect that it will be possible to develop not only flat surfaces but also flattening and distortion correction of complex surfaces. In addition, it can develop various researches that can recognize and solve various situations using Kinect and Unity. Overall, it seems necessary to research multiple focal problems and mapping of more complicated and uneven surfaces. We will be able to provide a comprehensive solution system that is very convenient for designers and developers mapping when combining with mapping research on textures in the future. This can be expected to lead to superior results from the commercial point of view.

Acknowledgments

References

[1]   https://en.wikipedia.org/wiki/Projection_mapping
[2]   https://www.youtube.com/watch?v=lX6JcybgDFo
[3]   https://www.youtube.com/watch?v=CpRLwLcLHNA
[4]   https://www.youtube.com/watch?v=HB5nJB9R8Qw
[5]   https://www.youtube.com/watch?v=a1v4W95wJnM

[6]   https://www.youtube.com/watch?v=doHXNtTsQWk
[7]   https://www.youtube.com/watch?v=Q1F0kt3AidE
[8]   https://www.youtube.com/watch?v=O_BfeXbpUQs
[9]   https://www.youtube.com/watch?v=A-La3NgO9Bw
[10]   https://www.youtube.com/watch?v=19X5izWHBmc
[11]   O. Bimber and A. Emmerling, "Multifocal Projection: A Multiprojector Technique for Increasing Focal Depth", IEEE Transactions on Visualization and Computer Graphics, vol. 12, no. 4, **(2006)** pp. 658-667
[12]   M. Brown, A. Majumder and R. Yang, "Camera-Based Calibration Techniques for Seamless Multiprojector Displays", IEEE Transactions on Visualization and Computer Graphics, vol. 11, no. 2, **(2005)**.
[13]   A. Grundh¨ofer, "Practical Non-linear Photometric Projector Compensation", IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Portland, USA, **(2013)**.
[14]   A. Grundh¨ofer and O. Bimber, "Real-Time Adaptive Radiometric Compensation", IEEE Transactions on Visualization and Computer Graphics, vol. 14, no. 1, **(2008)**, pp. 97-108.
[15]   K. Fujii, M. D. Grossberg and S. K. Nayar, "A Projector-Camera System with Real-Time Photometric Adaptation for Dynamic Environments", IEEE Computer Society Conference on Computer Vision and Pattern Recognition, San Diego, USA, **(2005)**.
[16]   O. Bimber, D. Iwai, G. Wetzstein and A. Grundh¨ofer, "The Visual Computing of Projector-Camera Systems", Computer Graphics Forum, vol. 27, no. 8, **(2008)**, pp. 2219-2245.
[17]   S. Akkaladevi, M. Ankerl and C. Heindl, "Tracking multiple rigid symmetric and non-symmetric objects in real-time using depth data", IEEE International Conference on Robotics and Automation (ICRA), Sweden, Stockholm, **(2016)**.
[18]   D. J. Tan, F. Tombari, S. Ilic and N. Navab, "A Versatile Learning-Based 3D Temporal Tracker: Scalable, Robust, Online", The IEEE International Conference on Computer Vision (ICCV), USA, Santiago, **(2015)**.
[19]   D. G. Aliaga, Y. H. Yeung, A. Law, B. Sajadi and A. Majumder, "Fast High-Resolution Appearance Editing Using Superimposed Projections", ACM Transactions on Graphics (TOG), vol 31, no. 13 **(2012)**
[20]   Y. Jeong, J. An and J. Park, "Surface recognition and Interpolation of Image with Depth Sensor", Future Generation Information Technology, December 21 – 23 Korea, Daejeon, **(2017)**

Authors

YunSang Jeong, master's degree at Soongsil University and majoring in computer graphics, deep learning, and game programming.

JiHye An, bachelor's degree at Department of Multimedia, Namseoul University (2013), master's degree at Soongsil University (2015), Ph.D in Media Studies, Soongsil University

Research: Computer Graphics, Emotional Data Visualization, and Projection Mapping

Jinho Park, Associate Professor at Soongsil University, Korea, South Korea