Depth Image Based 3D Object Model Generation

YANG-KEUN AHN, KWANG-SOON CHOI, YOUNG-CHOONG PARK
Realistic Media Platform Research Center
Korea Electronics Technology Institute
121-835, 8th Floor, #1599, Sangam-Dong, Mapo-Gu, Seoul
REPUBLIC OF KOREA
ykahn@keti.re.kr

Abstract: - This research studied the method of creating a complete 3D model by obtaining the location and r, g, b values of an object from the object's color and depth map images, and conducting the point based 3D modeling in OpenGL, and, finally, by filling the holes on the front or side according to the depth value. The most important aspect of the method is filling the front and sides of the object with differential hole-filling method. In addition, the 3D model could be viewed through 3D glasses by extracting left, right images from the 3D TV, and a view that corresponds to the location of the user could be supplied to the user by locating the user with an OptiTrack camera.

Key-Words: - 3D object, 3D modeling, 3D object model, 3D object model generation, Lenticular, hole-filling

1 Introduction

This study expanded from the conventional glasses-type multi-view generation system based on the location of its user, and focused on generating an effective real-time 3D model. The 3rd year system employed Depth-Image-Based Rendering, DIBR, method to generate diverse real-time multi-view images according to the location of the user. A texture image was rendered to generate a multi-view image in order to generate a virtual viewpoint according to the user's location from the RGB texture image and a depth image. When an image from a virtual viewpoint with respect to the user's location is generated through warping multi-view images according to the corresponding user locations, 3D effect of warping image moving with the movement of the user was achieved. However, the technology did not give the feeling that the user is looking at a 3D model. The dominant feeling from the technology was that a 2D image is moving with the user. Such feeling arises from the fact that a 3D model was not generated in the technology and it was just a moving 2D image. On the other hand, this 4th year's development actually generated a 3D model with a depth map image and a color image in OpenGL window, and provided an appropriate real-time image of the 3D model according to the each viewpoint of the user to give the user to get the feeling of the 3D model. The quality of the original image was maintained as much as possible for the user to view. Detailed explanations will be provided in 2. Main Body. 2.1 Corresponding-point-based camera calibration technology, 2.2 Depth map preprocessing, 2.3 Color-image-based Texture generation technology, 2.4 Color/Depth-image-based Mesh generation technology, 2.5 multi-view Color/Depth image registration technology, 2.6 will explain autostereoscopic 3D model generation employing 8-viewpoint-synthesized lenticular image in detail, and 3. Experiment and 4. Conclusion will summarize at the end. The references of this study are listed in 5. References.

2 Main Body

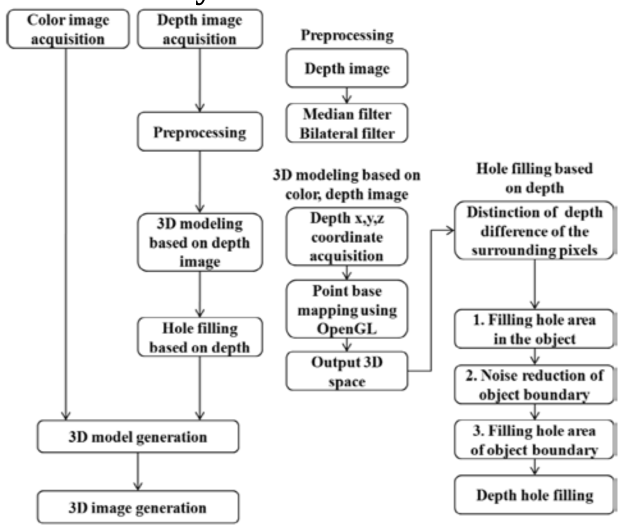

Color image acquisition
Depth image acquisition
Preprocessing
3D modeling based on depth image
Hole filling based on depth
3D model generation
3D image generation
Preprocessing
Depth image
Median filter
Bilateral filter
3D modeling based on color, depth image
Depth x,y,z coordinate acquisition
Point base mapping using OpenGL
Output 3D space
Hole filling based on depth
Distinction of depth difference of the surrounding pixels
1. Filling hole area in the object
2. Noise reduction of object boundary
3. Filling hole area of object boundary
Depth hole filling

Fig. 1 3D model generation algorithm

2.1 Corresponding-point-based Camera calibration technology

A depth map image and a color image from DS325 could not be utilized immediately because the depth map camera and color camera are separated from each other. First of all, the ratio and location of a color image and a depth map image has to be adjusted. Here, we calibrated with respect to the color image instead of the depth map image. The resolution of the depth map image supported by the ds325 softkinetic was 320*240 whereas the resolution of the r, g, b camera was 640*480. Therefore, when a color image is adjusted according to a depth map, there will be a resolution reduction in the color image. In addition, the resolution reduction will be observable when the image is viewed from a tv. Therefore, the depth map was resized with respect to the color image of 640*480 for the calibration.Here, holes are generated due to the size difference when the image is simply resized, and bicubic interpolation was employed to deal with the holes. The interpolation method employs 3rd order polynomials. The assumption is that two points and the slopes of the tangent lines at the two points are known. When a 2nd order polynomial is employed it results in an awkward curve or no curve, but the employment of a 3rd order polynomial results in a very natural curve. In addition, connecting each section also results in smooth connects without breaks. In enlarging an original image, bicubic interpolation employs 16 adjacent pixel values for each pixel of the target image when the coordinate of the original image has real values.

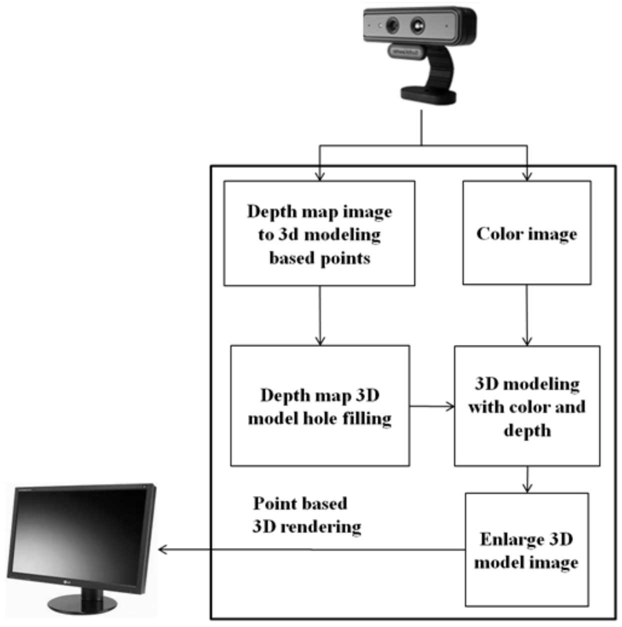

Depth map image to 3d modeling based points
Color image
Depth map 3D model hole filling
3D modeling with color and depth
Point based 3D rendering
Enlarge 3D model image

Fig. 2 Color/Depth value acquisition method

2.2 Depth map preprocessing

Enlarging a depth map according to a color image results in larger hole and disocclusion regions. Median filter and bilateral filter were employed to resolve this problem. Median filter was employed to remove the hole visible from the front side of the object, and bilateral filter was employed to maintain the edge of the depth map object.

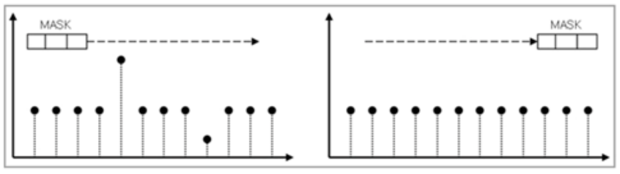


Fig. 3 Median filter

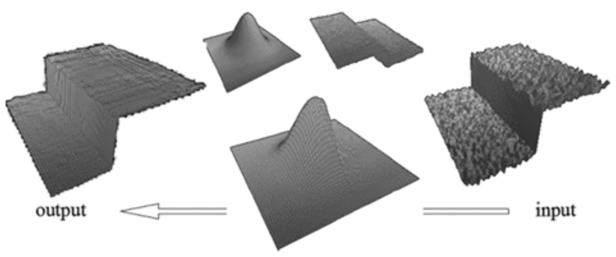


Fig. 4 Bilateral filter

2.3 Color image based Texture generation technology

Depth values from the depth map was loaded based on the color image, and x-coordinates, y-coordinates, depth values, and R, G, B values of the object recorded with ds325 were loaded. If the color values of the color image were loaded according to the depth map, every color value will not be loaded regardless of the interpolation and preprocessing because, unlike color image values, the depth values of a depth map image do not exist continuously. Therefore, the 3D modeling in OpenGL window will result in low resolution and holes. However, when the depth values are loaded with respect to the color values, every depth value could be utilized while maintaining the quality of the resolution of the color image. The values are obtained from the color image with respect to the x, y coordinates of r, g, b values, and obtaining depth values corresponding to the x, y coordinates results in x, y, z (depth), and r, g, b values. Rendering could be conducted for point based 3D modeling with respect to the x, y values in the order of the depth values in OpenGL.
In loading the point values of the corresponding model, extraction of x, y, z values with respect to

forward mapping from the depth map image, and the extraction of b, g, r from the color image could result in overlapping of the pixels corresponding to the output model and it could also result in empty pixel, or hole. Therefore, every pixel of the output model corresponds to a new pixel of the input image through backward mapping, and the depth map values are loaded with respect to the color image during the mapping. This partially solves the overlapping and hole generation in the output image from the forward mapping.

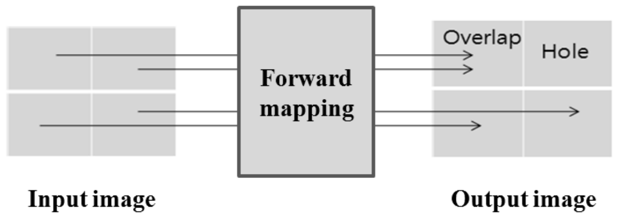


Fig. 5 Forward mapping

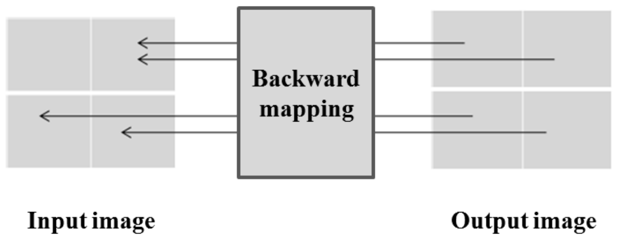


Fig. 6 Backward mapping

2.4 Color/Depth-image-based Mesh generation technology

A 3D model rendering with the information obtained from the color and depth map results in point-cloud-based 3D model. However, the 3D model does not constitute a complete 3D model due to the hole and disocclusion space. Values have to be assigned to the hole and disocclusion section to resolve the problem. However, filling the holes on the front, the sides, and the background with a same method does not fill the holes perfectly. When the holes are filled with respect to the front side, only the holes on the front side are filled and rotating the object will reveal the remaining holes. Such holes generated due to the difference in the depth values have to filled with respect to the depth. Fig. 7 explains the method.

As the above figure explains, the holes generated at the edge or on the sides of an object are generated due to the difference in the depth values. When points with depth values are drawn in the 3-dimensional space of OpenGL, a space is generated between the points, unless there is a sequential difference between the points, Preventing the generation of the spaces between the points could fill the holes and disocclusion sections on the edge and sides of the object. A point-based-3D rendering model forms a perfect 3D mesh when the holes and disocclusion sections of the model are completely filled. Meaning it becomes a perfect 3D model. This study focused on the interior and edges of an object. The holes on the interior of an object are mostly filled during the preprocessing and it has the appearance of a perfect model. The holes on the edges are not generated from the lose of the points, but rather they arise from the differences between the depth values of larger than 1, and because the points do not exist in a sequential manner. A perfect 3D model cannot be generated when the identifications and fillings of such sections are neglected. In addition, the infrared rays diffuse at the edges, and the points exist at the locations lower than where they are supposed to be. Neglecting such aspects could lead to the feeling that the object is diffused to the background when the 3D model is rotated. Such problem could be resolved by setting a threshold value of each point by averaging the depth values of the adjacent points, and cutting the values far from the threshold value. The points within the allowable range are pushed towards the object by the distance from the threshold value. Cutting and moving points in such manner resulted in more smooth and natural edges of a 3D object.

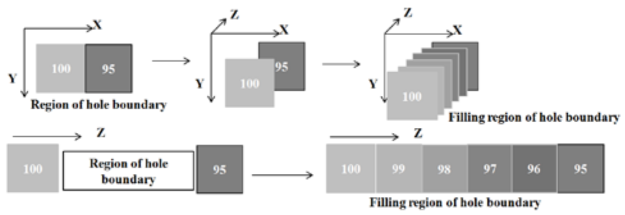


Fig. 7 Depth-map-based hole filling

2.5 Multi-view Color/Depth image registration technology

There are various methods of viewing a 3D image from a 3D TV. The method this research employs is by showing images with different left and right view points and registering them to show left side image to the left eye and show right side image to the right eye. Human sense of the distance, or depth perception, comes from the binocular disparity. In a similar manner, when a 3D model generated from OpenGL is rendered to different left and right images, rotating about the y-axis of the object to show the different image induces a depth perception. The program identifies the locations of the user's eyes with an optitrack camera, and rotated the 3D models of the left and right sides about the y-axis by the distance that corresponds to the locations of the eyes. The rotation by the location of the user's eyes provide real-time multi-point of view to give the user the experience of viewing the 3D object while the user walks around the object.

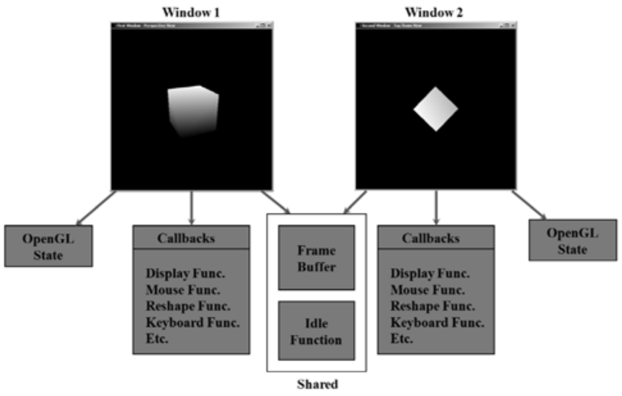


Fig. 8 OpenGL Multi window

In OpenGL window, there are methods of opening a single window and multiwindow. The method necessary for this project is the method of opening both left and right images. Therefore, subwindow and multiwindow methods of OpenGL were employed. The model was implemented according to multiwindow method in 3DModel-DS325. The resulting values of the object of the corresponding window were copied into opencv window and the output was loaded onto 1920*1080 TV. There is a caution to be taken when two or more windows are drawn or when a 3D model is continuously drawn in a subwindow. OpenGL has the characteristic of sequentially drawing a point or an object that it could distort the previously drawn image. Due to the sequential property, the points or models of the previously drawn model and a model drawn afterwards are mixed with the priority on the point order rather than their depth values.

glEnable(GL_DEPTH_TEST); is used to solve such problem. Even if OpenGL mixes 3D models with respect to the order, the function rearranges the 3D models according to the depth values. Therefore, the problem of a distorted 3D model object could be resolved through multiple windows.

A caution has to be taken when a point or an object is drawn in OpenGL using glbegin(); and glend();. Prior to draw an object in OpenGL, this function has to call glbegin(); and call glend(); at the end. However, a reckless use of the function results in the reduction in the performance of the program. The point cloud during point-based 3D rendering contains a significant number of points. Using glbegin(); and glend(); for every point in the cloud will result in a clear performance reduction. Using glbeing(); before the execution of a for-loop of the point cloud and using glend(); after every point has been drawn could resolve the problem. There is an immense difference between calling glbegin(); and glend(); six hundred thousand times and calling them once to draw minimum of ninety thousand to six hundred thousand points.

The presences of left and right images to give the stereoscopic 3D image effect do not immediately give the effect. The left and right images have to be the left and right images with binocular disparities when a person actually looks at them. Left and right images of a 3D model from same viewpoint will result in an image with no 3-dimensional effect. Therefore, the left and right images have to the image view from left and right eyes, respectively. Each of the 3D models in the left and right windows has to move about y-axis to generate images with different viewpoints from one model.

Therefore, the left image has to rotate about the y-axis to the right direction and the right image has to rotate about the y-axis to the left direction to generate left and right images with an actual left and right binocular disparity.

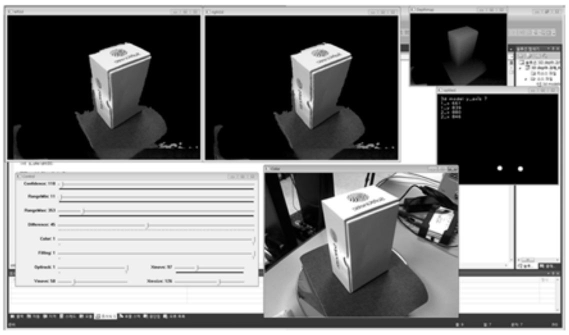


Fig. 9 PC screen UI

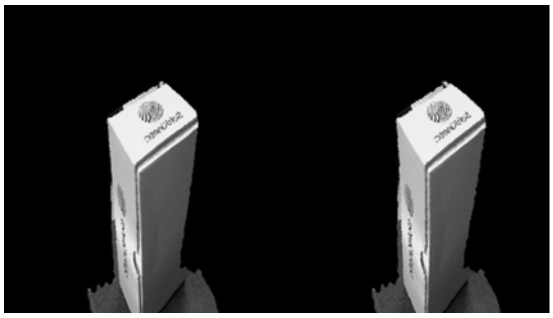


Fig. 10 3D TV UI

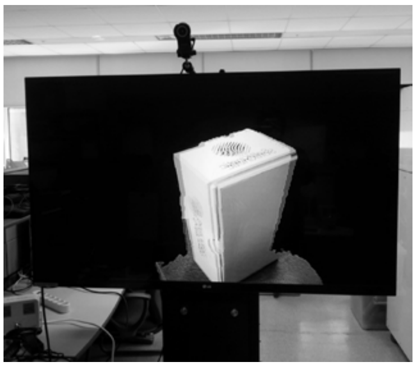


Fig. 11 3D mode applied image on TV

2.6 8-view point-synthesized lenticular image for an autostereoscopic 3D model generation

Lenticular lens is used view a 3D model image with no glasses. Lenticular lens shows an image of the viewpoint that corresponds to the direction of the viewer. The lenticular lens used for lenticular 3D display in this study is Alioscopy3DHD24. The display automatically shows the maximum of 8 viewpoints in the stereoscopic 3D display format. An image appropriate to the viewer's location is provided because lenticule lens have curved surfaces and one of the previously synthesized 8-viewpoints is supplied to the viewer as a curve.

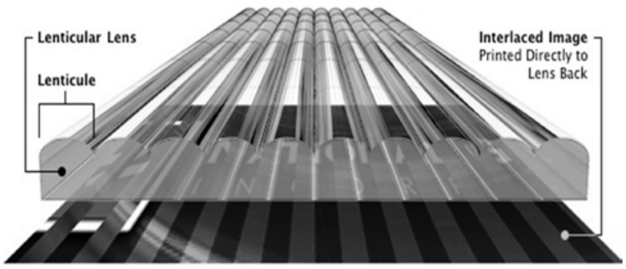


Fig. 12 Sectional view of lenticular lens

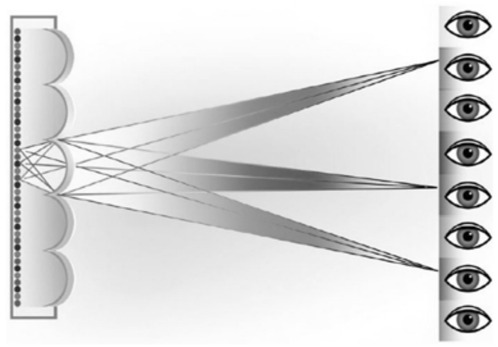


Fig. 13 Viewing an image from 8 viewpoints through lenticular lens

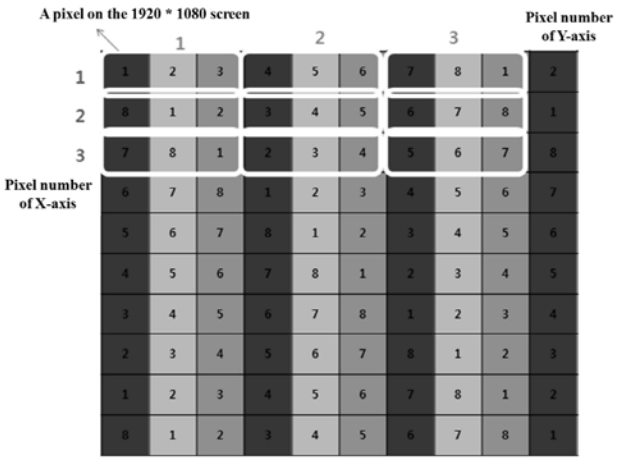


Fig. 14 8-viewpoint image synthesis method

The images from 8 viewpoints have to be synthesized as one image to look at the 3D model on a multi-view lenticular 3D display without glasses. The difference between an 8-viewpoint synthesized image and a conventional single-view image is that in an 8-view point synthesized image, the R, G, B values assigned to a pixel are not obtained from the single-view but sequentially obtained from the 8 viewpoints. For example, the Red value of pixel 1 is the Red value of the viewpoint 1 image, the Green value of pixel 1 is the Green value from viewpoint 2, the Blue value of pixel 1 is the Blue value from viewpoint 3, the Red value of pixel 2 is the Red value from viewpoint 4, the Green value of pixel 2 is the Green value from viewpoint 5 ~ the Green value of pixel 3 is the Green value from viewpoint 8, and the Blue value of pixel 3 is the Blue value from the Blue value from viewpoint 1. Such method assigns images from 8 viewpoints to one image and the images from 8 viewpoints could be synthesized with the method of the above.

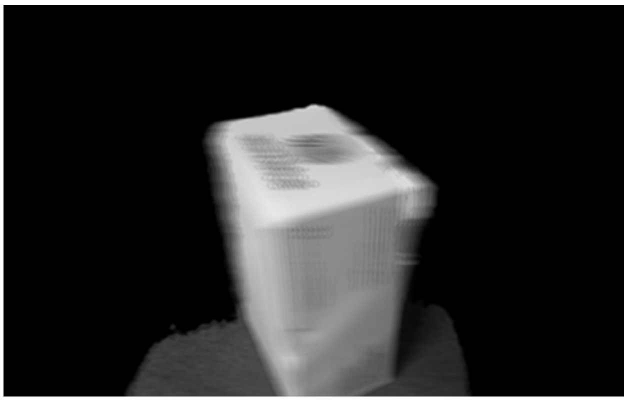


Fig. 15 8-viewpoint synthesized lenticular image

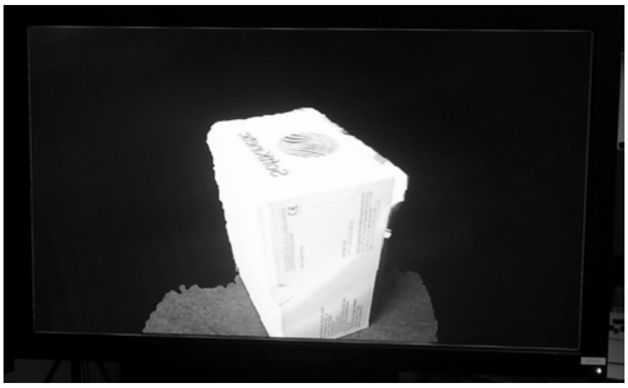
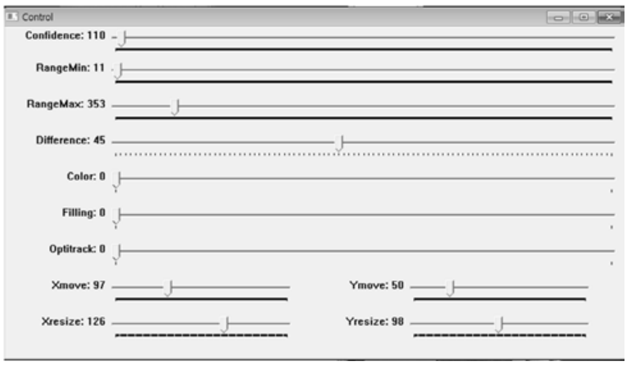


Fig. 16 8-viewpoint synthesized image viewed with Alioscopy3DHD24



Control

Confidence: 110

RangeMin: 11

RangeMax: 353

Difference: 45

Color: 0

Filling: 0

Optitrack: 0

Xmove: 97

Ymove: 50

Xresize: 126

Yresize: 98

Fig. 17 Control UI

3 Experiment

OS Windows7 , Tool Visual Studio 2010, and C++ constituted the development environment, and Reflective Marker 3D glasses, Optitrack Flex 13 infrared camera, DS325 camera, 3D display, and desktop Intel i7-2600K, RAM 3.48GB constituted the hardwares of the development environment.

PSNR (Peak Signal-to-Noise Ratio) value was used as the measurement for the image quality comparative evaluation. PSNR represents the power of the noise regarding the maximum power a signal could have, and it is often used to evaluate the image quality loss information from the image or video lossy compression. It is a value that quantitatively represents the difference between two different images. PSNR values were computed from the equation 2. MSE represents the interframe variance of a corresponding image, and MAX represents the maximum value of the image computed from (maximum value - minimum value) of the corresponding image channel. The unit of PSNR is dB, and since it is measured in the log scale, a small loss results in a higher value. 30dB was determined to be an appropriate value from the results.

The 3D model generation speed was 0.125 seconds, which is faster than the 3D model generation speed, 5 seconds, of the 4th year evaluation. In addition, the program updated a 3D model on average of 20~26 fps prior to the hole-filling, and it updated the 3D model at the rate of 10 fps after the hole-filling.

$$(1)\ \mathrm{MSE} = \frac{1}{\mathrm{mn}} \sum_{\mathrm{i}=0}^{\mathrm{m}-1} \sum_{\mathrm{j}=0}^{\mathrm{n}-1} [\mathrm{I}(\mathrm{i},\mathrm{j}) - \mathrm{K}(\mathrm{i},\mathrm{j})]^2$$

$$(2)\ \mathrm{PSNR} = 10 \cdot \log_{10}\left(\frac{\mathrm{MAX}_{\mathrm{I}}^2}{\mathrm{MSE}}\right) = 20 \cdot \log_{10}\left(\frac{\mathrm{MAX}_{\mathrm{I}}}{\sqrt{\mathrm{MSE}}}\right)$$

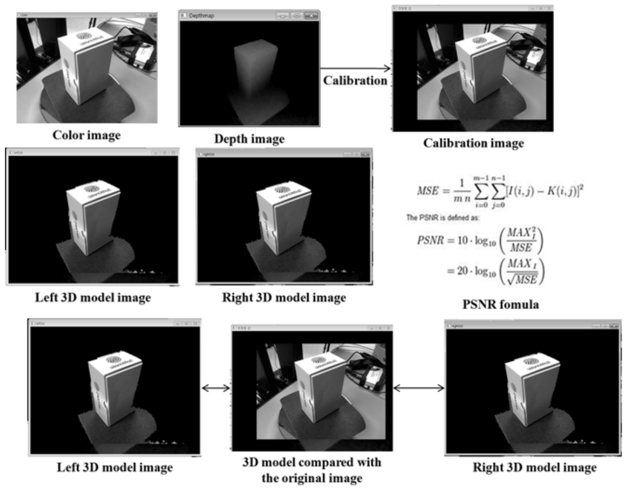

Color image
Depth image
Calibration
Calibration image
Left 3D model image
Right 3D model image
$MSE = \frac{1}{m\,n}\sum_{i=0}^{m-1}\sum_{j=0}^{n-1}[I(i,j) - K(i,j)]^2$
The PSNR is defined as:
$PSNR = 10 \cdot \log_{10}\left(\frac{MAX_I^2}{MSE}\right)$
$= 20 \cdot \log_{10}\left(\frac{MAX_I}{\sqrt{MSE}}\right)$
PSNR fomula
Left 3D model image
3D model compared with the original image
Right 3D model image

Fig. 18 PSNR experiment process

4 Conclusion

This project has yielded the result of 26fps before hole-filling and 10fps after hole-filling. Measuring the time of each section and function revealed that the repetitive computation during hole-filling is an issue. Optimizing the structure to sequentially compute the corresponding section and save the values in a different array and referring to the necessary parts would result in faster computations. The main purpose of this project is to give a user real-time 3D space and object experiences through 3D modeling. Therefore, efforts were made to provide 3D space and object experiences to a user, and 3D space and object experiences more real than those of the 3rd year project was successfully provided. Further development of the technology and improvement in the speed would provide quality 3D contents and service as a user records an object.

References:

[1] Wei Wang, Longshe Huo, Wei Zeng, Qingming Huang, Wen Gao, "Depth Image Segmentation For Improved Virtual View Image Quality In 3-DTV," Proceedings of 2007 International Symposium on Intelligent Signal Processing and Communication Systems, Dec 2007, pp. 300-303.

[2] Liang Zhang and Wa James Tam "Stereoscopic Image Generation Based on Depth Image for 3D TV," IEEE TRANSACATION ON BROCASTING, Vol. 51, No2, June 2005, pp. 191-199.

[3] Wenxiu SUN,Lingfeng XU, Oscar C. AU, Sung Him CHUI, Chun Wing KWOK The Hong Kong University of Science and Technology, "An overview of free viewpoint Depth-Image-Based Rendering (DIBR)," Asia Pacific Signal and Information Processing Association(APSIPA'10), December, 2010, pp. 1023-1030.

[4] Sang-Beom Lee, Yo-Sung Ho, “Discontinuity-adaptive Depth Filtering for 3D View Generation,” The Korean Institute of Communications and Information Sciences, November 2008, pp. 358-361.

[5] Jeeho Hyun, Jaeyoung Han, Jongpil Won, Jisang Yoo, “Generation of an eye-contacted view using color and depth cameras,” Journal of the Korea Institute of Information and Communication Engineering, Vol. 16, No. 8, 2012, pp. 1642-1652.

[6] Ryusuke Sagawa, member, IEEE, and Katsushi Ikeuchi, Fellow, IEEE "Hole filling of a 3D model by Flipping Signs of a Signed Distance Field in Adaptive Resolution," IEEE TRANSACTIONS ON PATTERN ALALYSIS AND MACHINE INTELLIGENCE, Vol. 30, No.4 , April 2008, pp. 686-699.

[7] David Doria Rensselaer Polytechnic Institute Troy, NY, Richard j. Radke Rensselaer Polytechnic Institute Troy, NY "Filling Large Holes in LIDAR Data By Inpainting Depth Gradients," Computer Vision and Pattern Recognition Workshops (CVPRW), June 2012, pp. 65-72.