Implementation of 3D Object Reconstruction using a Pair of Kinect Cameras

Dong-Won Shin and Yo-Sung Ho
School of Information and Communication,
Gwangju Institute of Science and Technology, Gwangju, South Korea
E-mail: {dongwonshin, hoyo}@gist.ac.kr Tel: +82-62-715-2258

Abstract— **In this paper, we propose a process of 3D object reconstruction using a pair of Kinect cameras. After we refine raw depth images from two Kinect cameras using a joint bilateral filter, we find intrinsic and extrinsic parameters by camera calibration. Then, we apply 3D warping to obtain a point cloud model in the 3D space and acquire a smooth surface model of the 3D object. In order to accelerate a processing speed, we employ a CUDA framework for GPU parallel processing. We reconstruct depth data in the integrated 3D space and obtain a 3D object model at 5 fps.**

I. Introduction

Our lives in real-world consist of a 3-Dimensional (3D) space which has X, Y and Z axes. 3D reconstruction means that a reproduction of objects from real world to virtual environment like a computer. Recently, a 3D reconstruction has been taking a center stage in various filed such as a game, movie, advertisement, construction, surveying and art.

As a specific example, there is a 3D facial reconstruction. Human face has the most muscles in our body. We can take about 3000 kinds of expression by numerous muscles and it makes us efficiently communicate and convey our emotion to other people. We can make a figure which is like our own face and apply it to a mobile content industry that has grown recently. Moreover, in a plastic and cosmetic field, they try to use a 3D facial reconstruction to their medical counseling. Besides, via a 3D human body reconstruction which reconstruct a person's body into a virtual space, real-time cloth fitting service has also developing now.

This kind of 3D reconstruction technology can be applied not only 3D information on a human but also a reconstruction of an object. One of the example is a reconstruction of a cultural asset. When a fire had broken out at Sungnyemun Gate in 2008 which is South Korea's top cultural landmark, a pre-scanned 3D information of Sungnyemun Gate would be of a great help to rebuild Sungnyemun Gate. After the accident, the Cultural Properties Administration in South Korea felt keenly the importance of 3D virtual reconstruction and proceeded a 3D scanning of the main cultural assets. Moreover it doesn't end at here, it used for an investigation of a fire accident. Unlike existing way to record pictures or clips, this way to record a 3D information of the scene would be of a great help to a fire investigation.

There are various ways to acquire 3D information in order to 3D reconstruction, it is mainly divided to an active and a passive method. An active method is that an infrared projector scatter an infrared pattern to the scene and an infrared camera measures a distortion degree, and then acquire depth data of 3D objects or a laser beam projector shines on 3D objects and receiver measures how much time it takes to come back. These methods mainly project rays to the objects and get a 3D depth data on the scene. This kind of an active methods have a relatively high accuracy but its camera sensor is expensive and hard to handle it. A passive method is that color camera captures 2 views or more and we calculate disparity values between left and right images through a pattern matching. This kind of a passive method is cheaper than an active method because we can create this system by color cameras that have a reasonable price but it has a low resolution and takes much time to compute.

Recently, in the scheme of an active method, an optical device technology has developed very well and an active sensor's cost became reasonable. Moreover, in the scheme of a passive method, the research is lively undergoing due to an increasing of a computer performance and a development of a parallel processing technology [1].

The main issue in 3D reconstruction is firstly a camera calibration needs to be correctly measured. After the acquiring intrinsic and extrinsic parameters via the camera calibration and the system should correctly reconstruct objects from color and depth information obtained from each view in an integrated space by using these parameters. Next, the algorithm that can process a vast amount of 3D information in short time need to be developed. Currently, an improvement of computer processing performance and a propagation of a parallel processing technology help algorithms to accelerate. Lastly, the accuracy on depth images obtained from each view must be high. A raw depth image by using the methods that mentioned earlier has a low accuracy compared with representing real 3D world. Because of a sensor error, occlusion area, boundary mismatch, etc., there can exist holes which has no depth value. Moreover, a resolution matching process is also needed because a depth image has low resolution compared with a color image.

In this paper, as we consider these main issues, we explain how we reconstruct a 3D object by using a pair of Kinect cameras. In chapter 2, we describe a physical structure of a proposed 3D reconstruction system and in chapter 3, we illustrate a flowchart of proposed 3D reconstruction algorithm.

978-616-361-823-8 © 2014 APSIPA

In chapter 4, we show results by using a proposed method and lastly chapter 5, we discuss a conclusion and future work putting above contents together.

II. System Setting

The proposed method in this paper is about a reconstruction of 3D objects in real world by using a pair of Kinect cameras as a convergence form. Kinect is a 3D vision camera released from Microsoft and it has a high performance camera and many users but relatively low prices. Kinect equips a RGB camera, microphone, infrared projector and camera. Kinect is using a structured light method so that it projects a randomly arranged dotted pattern to the objects from an infrared projector and a projected pattern forms a distorted pattern depending on a distance from Kinect to objects. After the distorted pattern is captured by an infrared camera, depth values can be measured by a triangular surveying [2]. In this paper, we place a pair of Kinect cameras in front of an object like Fig. 1 and set it as a convergence form by rotating about 45 degrees toward an object.

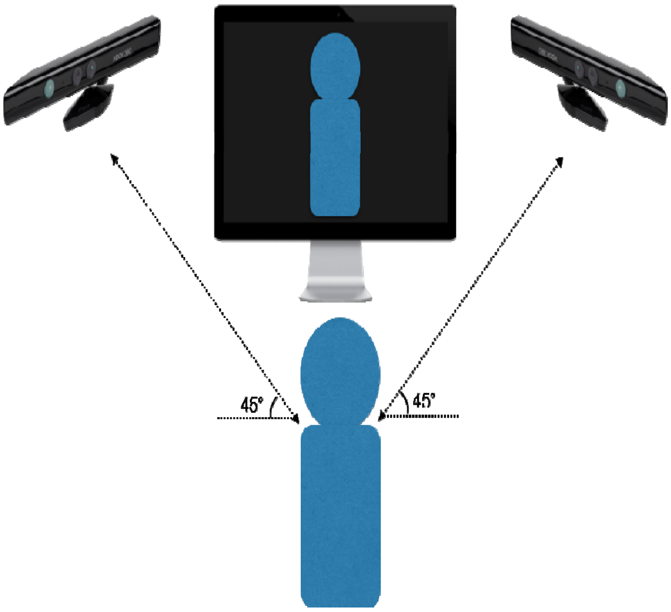

Fig. 1 Physical structure of a proposed method

After that, we place an object at a convergence point of a pair of Kinect cameras. In order to connect a plural Kinect to single computer, it needs to be connected to a USB hub which is directly connected to a mainboard. Hence, depending on the number of Kinect, we need to additionally equip USB hubs to a mainboard.

III. 3D Reconstruction

Fig. 2 shows the flowchart of the proposed 3D reconstruction.

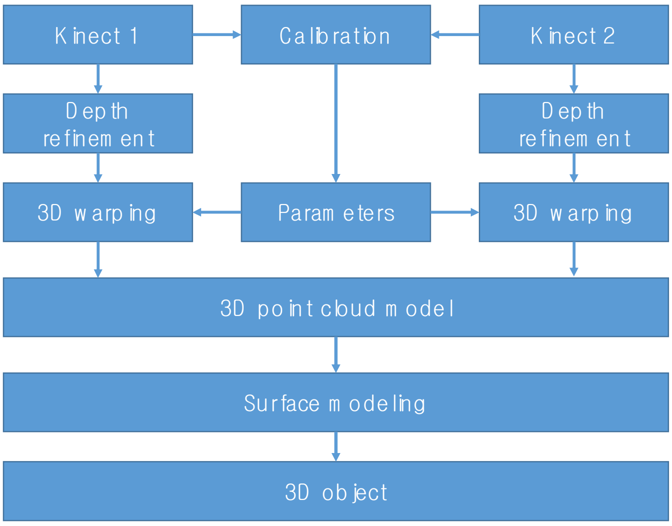

Fig. 2 Flowchart of proposed method

First of all, in order to obtain intrinsic and extrinsic parameters from Kinect, we perform a calibration step as an offline process. After that, we gather color and depth images from each Kinect camera and employee a depth image refinement to fill holes which has no depth value. Next, we send depth information from each view to integrated 3D space by using intrinsic and extrinsic parameters obtained before. As a result of this, a 3D point cloud model is created in a 3D space. Lastly, after a surface modeling, we can acquire a 3D object that has a smooth surface.

A. Calibration

In order to perform a 3D reconstruction using plural cameras, a calibration process is necessary. In this paper, we capture calibration sequences by using a planer checkerboard pattern that usually uses for a camera calibration and obtain intrinsic and extrinsic parameters by using Matlab calibration toolbox [3]. We use a 7 x 5 pattern which is 30x30mm on a single rectangular size. Fig. 3 shows checkerboard pattern sequences captured from a proposed camera system.

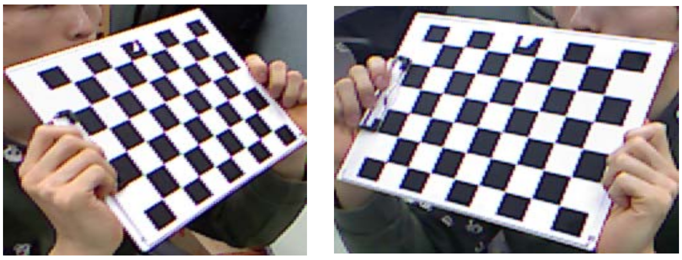

Fig. 3 Pattern sequence for camera calibration

A single scene for a pair of Kinects need to be captured at a same time and all feature points of a pattern should be shown clearly in a sequence. When we capture 10 pattern sequences from each Kinect, we get 20 pattern sequences in total. Through Matlab calibration toolbox, we perform a calibration step using by these pattern sequences. Finally we can get intrinsic and extrinsic parameters on each camera. By using these parameters, we employee a 3D warping, then we can represent depth information from each camera in an integrated 3D space.

B. Depth image refinement

An original depth image from Kinect has many holes that has no depth values due to sensor errors, occlusion areas, etc. In order to fill these holes as proper depth values, there are many ways to achieve this goal. In this paper, considering the real-time property, we employee a joint bilateral filter implemented by a parallel programming technique [4]. Moreover, even though an existing joint bilateral filter considered color and range differences between center and neighbor pixels in the kernel, we reflect a depth difference between them. Equation (1) shows an existing joint bilateral filter.

$$D_o(x,y) = \frac{\sum_{u \in \vec{u}_p} \sum_{v \in \vec{v}_p} W(u,v) D_i(u,v)}{\sum_{u \in \vec{u}_p} \sum_{v \in \vec{v}_p} W(u,v)} \tag{1}$$

D_o means a output depth image and *(x, y)* represents a position of center pixel in the kernel. D_i means an input depth image and *(u, v)* represents a position of neighbor pixel. *W* means a weighting factor. In this paper, we change *W* like (2) to reflect a depth difference.

$$W(u,v) = \begin{cases} 0 & D_i(u,v) = 0 \\ g(u,v) \cdot f(u,v) \cdot d(u,v) & otherwise \end{cases} \tag{2}$$

g(u, v) is a weighting factor of Gaussian filter reflecting a range difference between center and neighbor pixels and *f(u, v)* is an weighting factor of Euclidean filter reflecting a color difference. So far, it is the weighting factor of an existing joint bilateral filter. We added a weighting factor *d(u, v)* reflecting a depth difference between center and neighbor pixels. Equation (3) shows a weighting factor *d(u, v)*.

$$d(u,v) = \exp\left\{ -\frac{\left| D_i(x,y) - D_i(u,v) \right|^2}{2\sigma^2} \right\} \tag{3}$$

The depth weighting function d(u,v) calculates the weighting by substituting the depth difference into Gaussian function.

C. 3D warping

Next, we perform a 3D warping by using color and depth images. 3D warping is the algorithm sending depth information from each view to an integrated space by using intrinsic and extrinsic parameters from each camera. Through this algorithm we can get a synthesized 3D point cloud model. Equation (4) shows a 3D warping [5].

$$M_W = R^{-1} A^{-1} \begin{bmatrix} d \cdot x \\ d \cdot y \\ d \end{bmatrix} - R^{-1} t \tag{4}$$

R means a rotation matrix of extrinsic parameters and *A* means intrinsic parameter matrix. *x* and *y* means a position in the image and *d* means a depth value at (x, y). Lastly, t represents a translation matrix of extrinsic parameters. By using (4), we can calculate a vector M_W having x, y, z value.

Fig. 4 shows an object getting through a 3D warping.

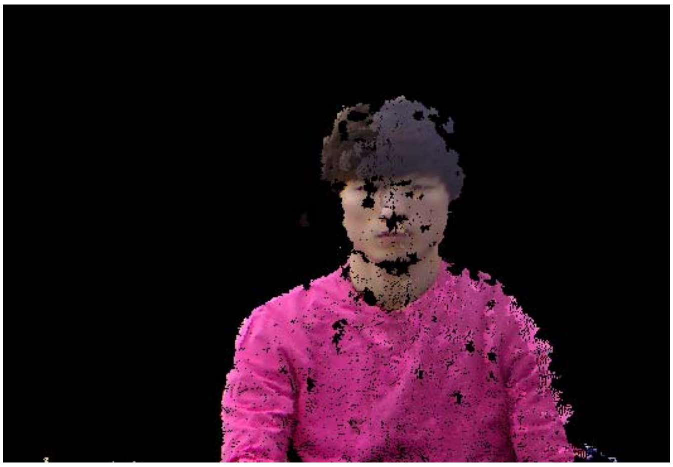

Fig. 4 Result of 3D warping

Because we represent an object in a virtual 3D space, we can alter views freely. However, because this is point cloud model, there are some empty space between points and points. In order to solve this problem, we need to turn a 3D point cloud model into a surface model. Fig. 5 shows the surface modeling.

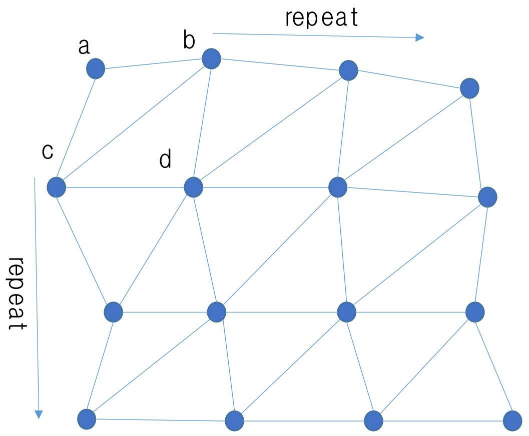

Fig. 5 Surface modeling

We construct a triangle with point *a, b, c*; *a* is a base point, *b* is a point on the right and *c* is a point on the below. Next, we construct a triangle with point *d* and a leg of triangle *abc.* By using this way, we construct surfaces repeatedly. However, it could connects a point far away from a base point, so we construct surfaces in the range of a specific threshold value.

IV. Experimental Results

We used Intel Xeon 2.53Ghz CPU and Nvidia Geforce GTX Titan GPU for this experiment. In a joint bilateral filter, the sigma value of Gaussian filter is 3, sigma value of Euclidean filter is 100 and threshold value of depth weighting filter is 100, filter radius is 3. Moreover, we set a threshold following a distance as 10 in a surface modeling. Fig. 6 shows a 3D object via a proposed 3D reconstruction method.

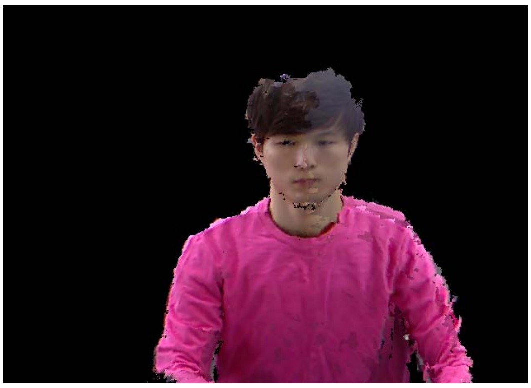

Fig. 6 Result of proposed method

In comparison with Fig. 4, a lot of holes are relatively filled with proper depth values. Fig. 7 shows the object at several views. We can observe the object in various angle views and it operates as 5 fps.

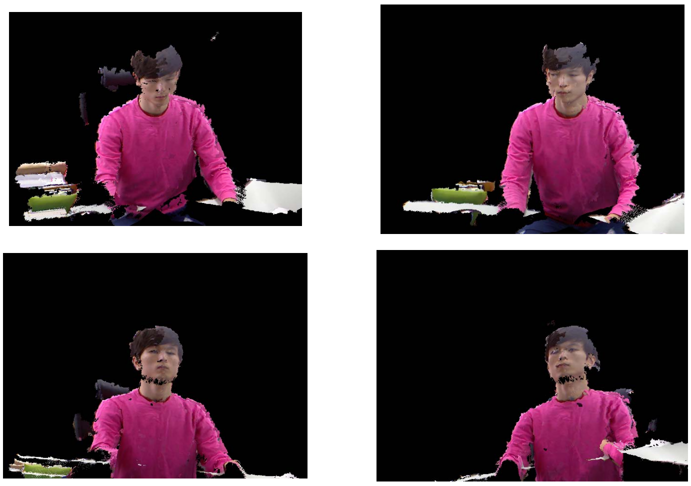

Fig. 7 3D object at each view

V. Conclusions and Future work

In this paper, we explained a 3D reconstruction method using a pair of Kinect cameras. We placed a pair of Kinect cameras in front of an object. After getting intrinsic and extrinsic parameters by using a camera calibration, we refined raw depth images via a joint bilateral filter with an additional depth weighting factor. Then we represented depth images from each view in an integrated 3D space via 3D warping. Finally, we reconstructed a 3D object having a smooth surface through a surface modeling. However, because we couldn't acquire a complete object model, we need to study to get more accurate depth values and to achieve 30 fps as a real-time also.

Acknowledgement

This research was supported by the 'Cross-Ministry Giga KOREA Project' of the Ministry of Science, ICT and Future Planning, Republic of Korea(ROK). [GK13C0100, Development of Interactive and Realistic Massive Giga-Content Technology]

References

[1] S. Yoon, B. Hwang, K. Kim, S. Lim, J. Choi, B. Koo, "A Survey and Trends on 3D Face Reconstruction Technologies," *Electronics and telecommunications trends*, vol. 27, no. 3, pp. 12-21, June 2012.

[2] J. Chang, M. Ryu, S. Park, "Technology Trends of Range Image based Gesture Recognition," *Electronics and telecommunications trends*, vol. 29, no. 1, pp. 11-20, Feburary 2014.

[3] Camera Calibration Toolbox for MATLAB: http://www.vision.caltech.edu/bouguetj.

[4] D. Shin, Y. Ho, "Real-time Depth Image Refinement using Hierarchical Joint Bilateral Filter," *Journal of The Korean Society of Broadcast Engineers*, pp. 140-147, March 2014.

[5] W. R. Mark, L. McMillan, and G. Bishop, "Post-rendering 3D Warping," *in Proc. of Symposium on Interactive 3D Graphics*, pp. 7-16, April 1997.