AWERProcedia Information Technology & Computer Science

Vol 03 (2013) 1748-1754

3rd World Conference on Information Technology (WCIT-2012)

3D Reconstruction Technique with Kinect and Point Cloud Computing

Erdal Ozbay *, Faculty Of Engineering Computer Engineering, Firat University, Elazig 23119, Turkey.
Ahmet Cinar, Faculty Of Engineering Computer Engineering, Firat University, Elazig 23119, Turkey.

Suggested Citation:

Ozbay, E., & Cinar, A. 3D Reconstruction Technique with Kinect and Point Cloud Computing, *AWERProcedia Information Technology & Computer Science*. [Online]. 2013, 3, pp 1748-1754. Available from: http://www.world-education-center.org/index.php/P-ITCS *Proceedings of 3rd World Conference on Information Technology* (WCIT-2012), 14-16 November 2012, University of Barcelon, Barcelona, Spain.

Received 12 January, 2013; revised 13 May, 2013; accepted 03 September, 2013.
Selection and peer review under responsibility of Prof. Dr. Hafize Keser.

Abstract

The background of this study using a variety of methods lies in providing a new interaction between them as the basis of virtual space with objects in the physical world. In this respect, the purpose is to cover in detail a scene for the reconstruction of 3D real-world objects. This involves the creation of virtual spaces using a 3D standard camera, the object of which is to display a 3D model. In order to achieve this target, during data acquisition, examination of the available methods and integration of multiple scan-related research were undertaken. As a result, in summary, this study involves the discussion of real-world objects as solid on the planar surfaces, which have been active in more than one depth-scanned point clouds (PCL), applying various filters to the background of the object, and then combining these scans.

Keywords: 3D reconstruction, kinect, depth camera, image reconstruction, tracking, point cloud;

* ADDRESS FOR CORRESPONDENCE: **Erdal Ozbay,** Faculty Of Engineering Computer Engineering, Firat University, Elazig 23119, Turkey, *E-mail address*: erdalozbay@firat.edu.tr / Tel.: +90-424-237-0000

1. Introduction

Today, the human eye finds it extremely difficult to find solutions to restructuring real-world objects in three-dimensions when using a two-dimensional image structure and re-organizing it's content. In nature, the visual features of the human eye do not have to undertake a conversion from 2D to 3D. This property of the human eye is an exceptional characteristic which allows people to understand the visual world. Computer vision, virtual reality, object recognition, augmented reality, and the logical approach to restructuring in order to understand the visual world aims to emulate the visual capabilities of people. However, the restructuring of real world objects in a 3D computer environment is very cumbersome.

3D models of real-world objects play an important role in many areas. However, 3D modeling is not commonly used since it is time consuming and costly. For example, in archeology, only the most valuable objects are scanned. If we were able to simplify this process and accelerate the scanning process, we believe that through reduced time, cost and manpower, we would be able to minimise this problem.

Even though the concept of the depth camera is not new, the Kinect sensor has made it accessible to everyone. The quality of Kinect depth sensing instruments, their low cost and their use in real-time, that means the use of this sensor has become popular among the enthusiasts and researchers. In this study we aim to create a 3D model of an indoor environment by using the point cloud library. In this we show the capabilities of the Kinect camera. Using Kinect, more than one depth image frame is combined with the camera position, and rotation can be used to produce a dense 3D map. This map consists of large number of point clouds, which can be further processed to obtain a detailed surface. Many approaches have been developed for 3D reconstruction purposes [1]. Most approaches involve various sensors such as range scanners [2][3], stereo cameras [4] or monocular cameras [5].

During the implementation of this study, pre-existing technology and software tools are used. The software libraries, frameworks, and applications which are used in this study are all managed by the open source community. The Kinect sensor camera is a commercial product issued for use with Microsoft's Xbox360. There is also a Kinect Software Development Kit (SDK). In the software phase of this study we have used Kinect open source applications [6].

2. Kinect Camera

We use the Kinect sensor camera as a 3D measuring device. We analyzed the quantitative values with a Kinect sensor to examine the errors and to compare the results with experimental measurements of depth. Both 3D depth measurement and Kinect accurate calibration needs to have an accurate geometric model with Kinect. We benefited from the Kinect calibration feature of the 3D Kinect camera in terms of the object position in space, which is transformed into a common coordinate system.

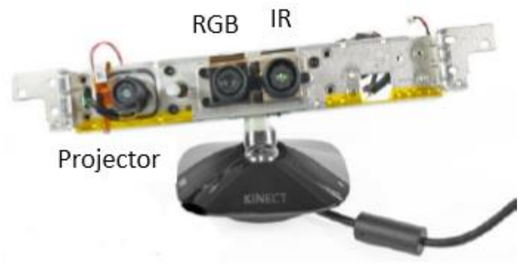


Figure 1. Kinect consists of an Infra-red (IR) projector, an IR camera and a RGB camera [7].

In addition, Kinect sensor camera is produced for ‘Xbox360’, it consists of the standard RGB camera, a depth camera and a microphone. Kinect may be used for research purposes because it is relatively inexpensive. Kinect has open-source drivers, so can be used outside of the Xbox360. Kinect camera has an IR Projector, which allows us to access the depth information of each pixel in the image. Kinect and its drivers perform calculations for access to raw data from the system.

The depth and RGB images taken by the Kinect camera are shown in Figure 2. The RGB camera captures the image at a resolution 300x300 pixels at 30-31 changeable frames per second. The depth measurement parameters can be computed using camera calibration methods. The Kinect camera uses a light technique to create a depth map that contains a range of discrete measurements of a physical scene [8]. These data can be adjusted as a set of discrete 3D point clouds. Kinect depth data measurement contains noise fluctuations and a new depth map can contain numerous 'holes' where there are no readings [9].

Kinect camera’s raw depth range is a maximum 2^{11}. During the measurement, metric depth is necessary to convert the raw depth. In terms of using the metric depth calculated using the following formulae of each pixel depth position from the IR camera angle:

$$X_{ir} = \frac{f_{x\,ir}}{(x - c_{xir})dm} \quad Y_{ir} = \frac{f_{x\,ir}}{(y - c_{yir})dm} \quad Z_{ir} = dm \qquad (1)$$

$$\begin{pmatrix} X_{rgb} \\ Y_{rgb} \\ Z_{rgb} \end{pmatrix} = \begin{pmatrix} X_{ir} \\ Y_{ir} \\ Z_{ir} \end{pmatrix} R + T \qquad (2)$$

$$X_{rgb} = \frac{X_{rgb} \;\; f_{xrgb}}{Z_{rgb}} + c_{xrgb} \quad Y_{rgb} = \frac{Y_{rgb} \;\; f_{yrgb}}{Z_{rgb}} + c_{yrgb} \qquad (3)$$

x, y are the positions of the image depth pixels, fxir. fyir is the focal length, cxir. cyir is the position of the principal point of the IR camera, and dm is depth in meters. fir and cir are estimated by calibration. The mapping between the colour image and the depth image can be expressed between the RGB and the IR camera rotation R and translation T [1].

Figure 2. Left side Kinect depth image. Right side is captured by Kinect RGB camera.

3. Data Acquisition

Depth cameras usually use a structured light method, which involves infrared light spread in an orderly manner, together with the RGB camera. These features are contained in the cameras, often referred to as an RGB-D camera. Also Microsoft Kinect Sensor camera is incorporated. We propose a method for capturing the individual objects and creating 3D models of a scene using the Microsoft Kinect camera.

A complete reconstruction of a physical object always requires multiple frames captured from different angles. Whatever the reason for this camera position, one or more of the captured faces of the object always remain occluded. For this reason, some method is required for tracking the conversions between the object captures [10]. For this purpose, several methods used by researchers, they have focused on conversions for tracking and calculating transformations.

3.1 Point Clouds Library

A point cloud is often used in order to obtain 3D point data. A data set is defined as a collection of multi-dimensional points. In a 3D point cloud, points are often above the sample surface. x, y and z coordinates are defined geometrically. Point clouds can be obtained by hardware sensors such as stereo cameras, 3D scanners or time-of-flight cameras. PCL supports the OpenNI and it can acquire and process data such as from Microsoft Kinect and PrimeSensor 3D camera devices [11].

The Point Cloud Library (PCL) is a Berkeley Software Distribution (BSD) licensed software library built for three-dimensional point clouds and can be used for 3D geometry calculation [11]. The PCL supports state-of-the-art algorithms, feature estimation, surface reconstruction, registration, model fitting and segmentation for filtering. The PCL was used during the implementation of this study. Because of the PCL, filters and algorithms that are required for the reconstruction of a 3D object can be applied. Data which is taken from the PCL can be executed in MeshLab format [12].

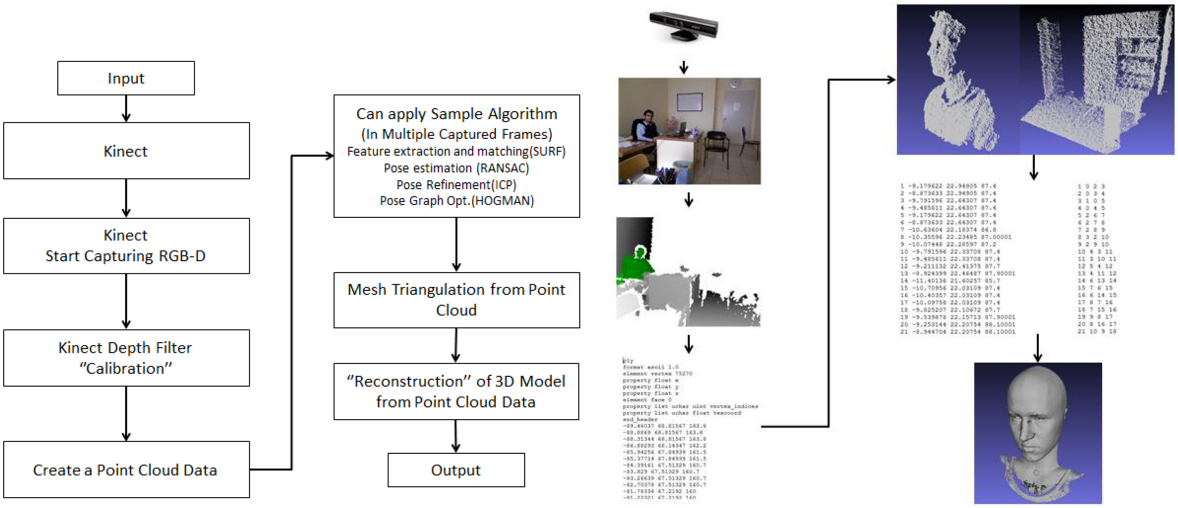
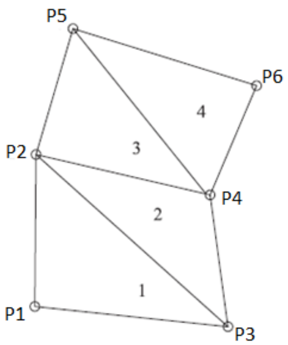


Figure 3. Implementation FlowChart

Primarily, the program needs to install the driver and the SDK to use it properly in terms of our Kinect sensor device. We get this from the official website of the Kinect developer KinectSDK-v1.0-beta2-x86 version of the SDK [13]. The program was developed in visual studio 2010 .Net_4 version. In addition, OpenNI Win32-1.3.2.3-Dev OpenNI version of the SDK was installed, and finally, the PCL-1.6.0-AllInOne-msvc2010-win32 version was installed to use the point cloud library [14].



	X	Y	Z
P1	-9.179622	22.94905	87.4
P2	-10.63604	22.18374	86.8
P3	-9.791596	22.33708	87.4
P4	-8.924399	22.46487	87.9
P5	-11.40136	21.60257	85.7
P6	-10.70956	22.03109	87.4

Field
1- P1 P2 P3
2- P2 P3 P4
3- P2 P4 P5
4- P4 P5 P6

Figure 4. Triangulation of the data points.

After generating the point cloud data from the physical object's surface, we used the Delaunay triangulation method to determine the mesh, which is known to combine the coordinate data points with its nodes [15].

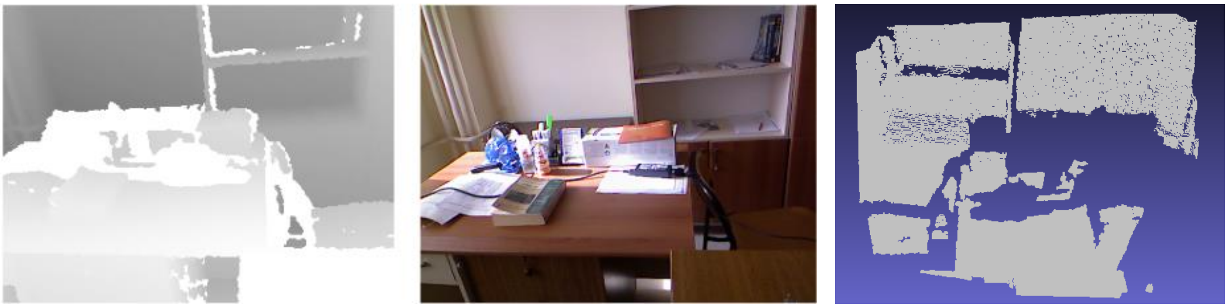


Figure 5. Left: Kinect depth image.   Middle: captured by Kinect RGB camera. Right: Point Cloud Data.

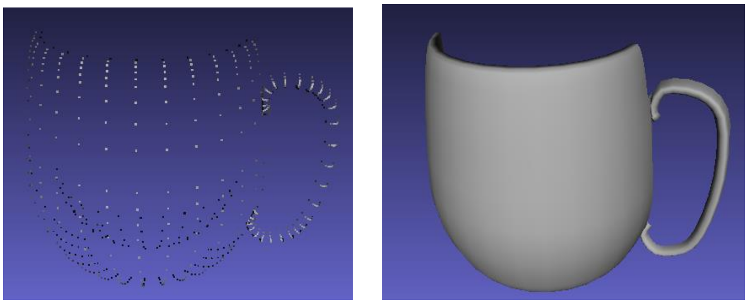


Figure 6. Left: Point Cloud Data.   Right: Mesh Triangle (Delaunay) Point Cloud Data.

4. Conclusion and Future Work

The aim of this study was perform to 3D point cloud configuration and reconstruct a model from real-world physical objects. Segments to be built outside of the desired objects were not considered for this purpose. In this study we used a single Kinect depth camera. Although the software allows multiple Kinect studies, when we want to create data for 360 degree objects, we would have to merge multiple captures and the apply various filters into a single point cloud presentation.

In this study, our application shows that we can generate to capture a set of point cloud a sufficient amount of capacity. But we can't create a complete model of 360 degrees real world objects yet. For this process to be successful we have to use various filters and they have to merge them which are captured more than one point clouds. We talked about point cloud reconstruction as a part of the project we are working on. During this study, we learned the difficulty of implementing the 3D modelling reconstruction using point cloud library, various method, algorithm and filters. As a continuation of this study, our goal is to place clothes on the reconstructed model. For this purpose, pre-designed fixed clothes will be better together with our model.

References

[1] Solony, M. Scene Reconstruction From Kinect Motion, Doctoral Degree Programme (2), FIT BUT.

[2] Thrun, S., Burgard, W. & Fox, D. A Real-Time Algorithm for Mobile Robot Mapping with Applications to Multi-Robot and 3D Mapping. *Proc. of the IEEE International Conference on Robotics Automation (ICRA),* 2000.

[3] Triebel, R. & Burgard, W. Improving Simultaneous Mapping and Localization in 3D using Globalconstraints. *Proc. of the National Conference on Artificial Intelligence (AAAI),* 2005.

[4] Konolige, K. & Agrawal, M. FrameSLAM: From Bundle Adjustment to Real-Time Visual Mapping. *IEEE Transactions on Robotics*, 2008, 25 (5).

[5] Lemaire, T., Berger, C., Jung, I.-K., & Lacroix, S. Vision-Based SLAM: Stereo and Monocular Approaches. *International Journal of Computer Vision,* 2007, 74 (3), pp. 343–364.

[6] Miika, S. *3D Content Capturing and Reconstruction Using Microsoft Kinect Depth Camera,* 2012.

[7] Jan, S., Michal, J. & Tomas, P. *2011 IEEE International Conference on Computer Vision Workshops* 978-1-4673-0063-6/11

[8] Freedman, B., Shpunt, A., Machline, M. & Arieli, Y. *Depth Mapping Using Projected Patterns.* Patent Application, 10 2008. WO 2008/120217 A2.

[9] KinectFusion: Real-time 3D Reconstruction and Interaction Using a Moving Depth Camera*, *UIST'11,* October 16-19, 2011, Santa, B., CA, USA. Copyright 2011 ACM 978-1-4503-0716-1/11/10

[10] Bernardini, F. & Rushmeier, H. The 3D Model Acquisition Pipeline. *Computer Graphics Forum,* 2002, 21 (2).

[11] 3D is here: Point Cloud Library (PCL). Robotics and Automation (ICRA), IEEE International Conference, 2011.

[12] MeshLab. [Online] Available from: http://meshlab.sourceforge.net/ [Accessed 19th October 2012].

[13] Kinect for Windows Developer Toolkit. [Online] Available from: http://www.microsoft.com/en-us/kinectforwindows/develop/developer-downloads.aspx [Accessed 19th October 2012].

[14] PCL Prebuilt Binaries for Windows. [Online] Available from: http://pointclouds.org/downloads/windows.html [Accessed 19th October 2012] .

[15] Jonathan, R. & Delaunay, S. Refinement Algorithms for Triangular Mesh Generation; *Computational Geometry,* 2002, 22 (1–3), pp. 21–74.