International Journal of Science and Research (IJSR)
ISSN (Online): 2319-7064
Index Copernicus Value (2015): 78.96 | Impact Factor (2015): 6.391

A Method to Construct Automatic Object Bounding-Box Estimation System using 3D Cameras

Ngo Tung Son[1], Bui Ngoc Anh[2], Tran Quy Ban[3], Tran Binh Duong[4]

[1,2,3,4] ICT Department, Hoa Lac Campus, FPT University, Hoa Lac High-Tech Park, Thach That District, Hanoi, Vietnam

Abstract: ***The measurement of the bounding-box size of a particular object in logistic system is critical. It consumes very much effort of the agents when processing a new order. Therefore reducing the performance of the agents. This paper introduces a method for measuring size of object bounding-box using multiple 3D cameras. These cameras will be arranged around the object such that they can view the whole of object. The object then will be reconstructed in three dimensions using output data that captured by the sensors. There are some of 3D processing techniques used to do this. Finally a bounding box will be computed to illustrate the size of the object. Experimental results are presented to demonstrate the accuracy and the performance of the introduced method.***

Keywords: Camera calibration, object reconstruction, object segmentation, bounding box estimation

1. Introduction

In the operations of the logistics system, the workers need to measure to obtain size of the object manually before it is packed into parcels. This process reduces the automation ability as well as the productivity of the system. There are several researches have been done in object measurement such as: Y.M. Mustafah et al used multiple stereo cameras for measuring object size in [1]. Suraphol Laotrakunchai et al proposed to drag the mobile devices around objects to measure its size in [2]. 3D camera technology and 3D data processing techniques allow us to provide a measurement using multiple 3D views, which is totally automatic without mechanical support structure.

Figure 1: Example of depth image from Kinect

The introduced method is very easy and cheap to implement. Meanwhile, maintaining a good assurance and performance. We selected Microsoft Kinect for the implementation. Kinect captures two types of data in every frame: the RGB image and the Depth image. The Depth image [Figure 2] represents the distances of points to the camera, which used to provide the cloud of 3D points [3]. That means if we know the point in 2D image; we will know where it is in 3D cloud. For these two types of data, both of image processing and 3D processing methods can be applied.

There are three steps of the method showed in [Figure 2]: (1) 3D scene reconstruction: Multiple cameras capture different scenes. The technique of camera calibration with checkboard applied to transform the views of cameras to one standard view. The scene can be reconstructed by merging the 3D points of the views. (2) Object segmentation: In order to determine the main object from the scene, some of 3D processing techniques applied to segregate the object. (3) Bounding box estimation: The bounding box of the object then will be estimated. Its size will be obtained by measuring the directions of the bounding box.

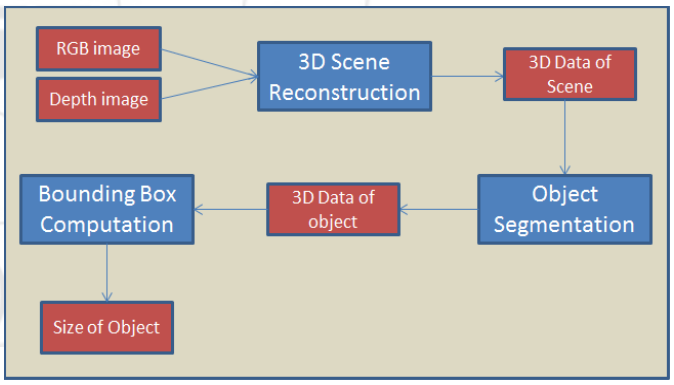

Figure 2: Three steps: 3D scene Reconstruction, Object Segmentation. Bounding Box Computation..

2. Implementation

2.1. 3D Scene Reconstruction

In order to merge the cameras views, we apply 3D camera calibration method with checkboard to transform the views of cameras. The basic theory can be found in opencv camera calibration [4]. Each camera has a different view so their generated 3D data point clouds are also in different coordinate systems. Therefore transform them to the same system. We detect the corners of the checkboard from each RGB image as the common points [5]. These points then will

be calculated to find the 3D transformation matrix. We have done the implementation using open source RGBDemo [6].
The selected Kinect version has **43°** vertical by **57°** horizontal field of view. We arrange Kinect around the checkboard such that all of Kinect can see the checkboard as we can see in [Figure 3]. In our project, we put the camera away from the center of the checkboard **1.3 meter**. The cameras generate the lower accuracy of depth image if we arrange it too far from the object.

Figure 3: Setup Kinects around measurement area

The particular Kinect gets its different view with other as shown in [Figure 4]. However, we pick one of the arranged Kinects to take the standard view, other cameras output point clouds will be transformed to the coordinates system of the standard one.

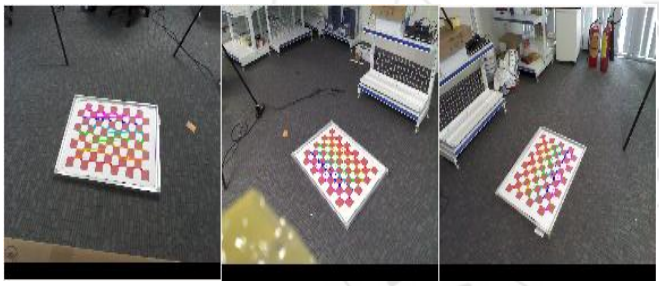

Figure 4: How Kinects see the checkboard

When deploying the actual system, the user only needs to do the work with the check board for the first time. After that the retrieved transformation matrix stored for future uses. However, if cameras are moved then we have to calibrate again. Therefore, we consider this calibration as the configuration system phase. For next steps, the system will be completely automatic. [Figure 5] illustrates an example of the merged point cloud from 3 different cameras using checkboard calibration.

2.2. Object Segmentation

This step begins with replacing the checkboard by the object. We use the output of calibration to reconstruct the scene. [Figure 6] shows the reconstructed scene with object. We can see that the reconstructed 3D scene contains a lot of un-useful data such as other objects that captured by the cameras, noise, floor. Those things must be removed before starting object measurement.

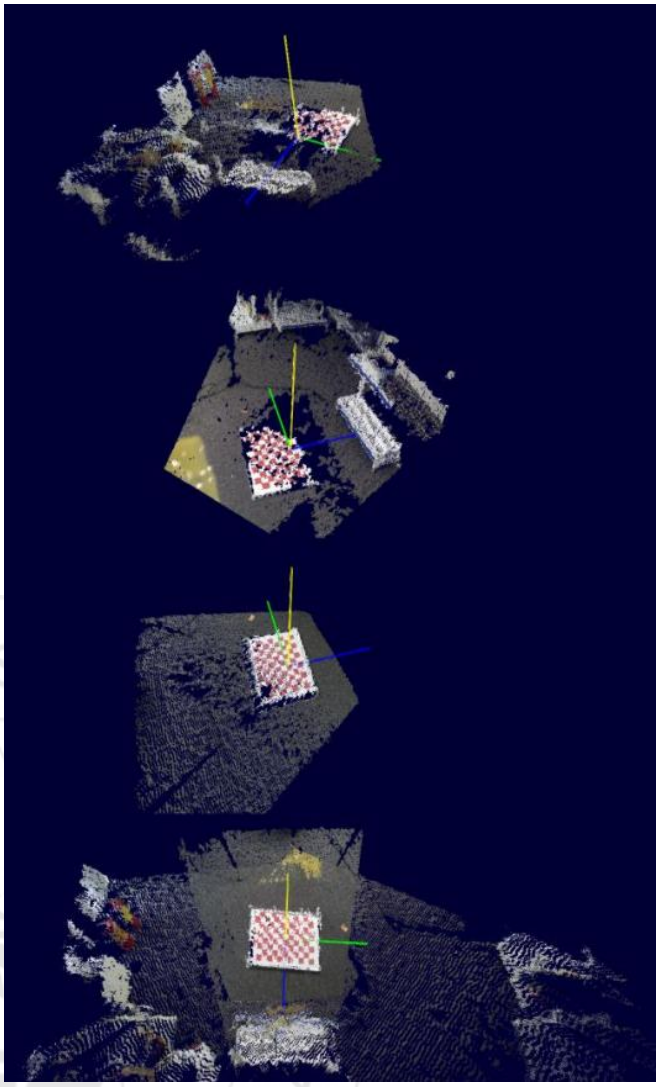

Figure 5: Merged point cloud from three different cameras

For more convenient in next steps, we transform the point cloud such that the center of checkboard duplicates to the origin of coordinate system, then store the 3D scene in a K-D tree structure that strongly support the nearest neighbors search. In order to remove the noise (the outlier points that generated by the camera). The number of neighbors to analyze for each point is set to k, and the standard deviation multiplier to m. All points that have a distance larger than m standard deviation of the mean distance to the query point will be marked as outliers and removed. In our experiment $\boldsymbol{k}$ **= 50** and $\boldsymbol{m}$ **= 1** give the accepted result while maintaining the performance.

One more step to segment object is to extract the floor from the scene by doing a segmentation operation for planar surfaces, using a RANSAC model [7]. In our case the plane that consist the maximum area of point cloud will be considered as the floor and removed. Other methods to extract floor could be considered: Remove the plane that contains the checkboard, rotate the cloud such that its normal vector is parallel with any coordinate-axis and make a range filter to remove estimated floor. Note that: the floor points are not always lie on the same plane. For this install, we consider the thickness of the floor is up to **4 cm**. [Figure 7]

shows the scene still contains many objects (including target extracted object) without floor.

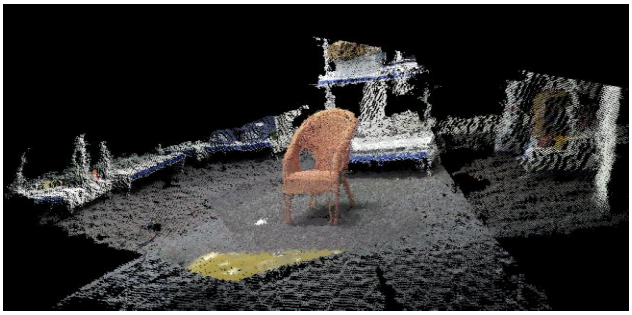

Figure 6: Scene Reconstruction

In order to remove those unintended objects, we use Euclidean Cluster Extraction algorithm that can be found in the Rusu PhD thesis [8], where the points that lie close together will be consider as belonging to the same cluster. The radius what we scan for each point to find its neighbors is **2 cm**.

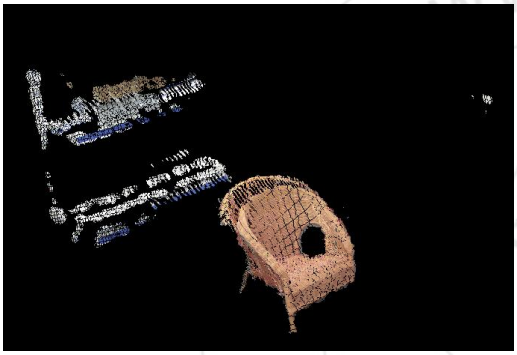

Figure 7: Scene without floor

Most of our tests, each cluster represents an object. So we collect many clusters that includes the target extracted object and others. However only the cluster that has center point closest to the origin of coordinate system will be considered as the measuring object. Finally the target object extracted as shown in [Figure 8]. Because we translate the origin to the center of checkboard and the object now replace the checkboard. Therefore, this is an important step in the user manual guide.

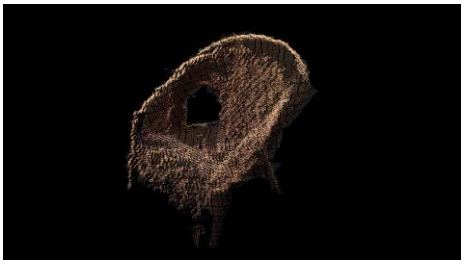

Figure 8: Segmented Object

2.3. Bounding Box Computation

The last phase is to estimate a good bounding-box of the object. The box can provide the size of object as well as to compute the transportation-fee based on the box's volume. There are different ways to compute the bounding box. For example: In [9], Darko et al applied PCA to find principal components as the direction vectors of the bounding box. This seems to be a good approach, because some of other open source library that we found implemented this as a feature of bounding box computation. However, over 60% of cases we have tested did not give the good results due to the quality of point clouds. One of the principal components may be parallel with the diagonal of the expected bounding box as shown in [Figure 9]. It led to the output to be very different with the hypothesis. Therefore we decided to use the simple minimum bounding boxes for 3-dimensional polytopes approach that introduced in [10].

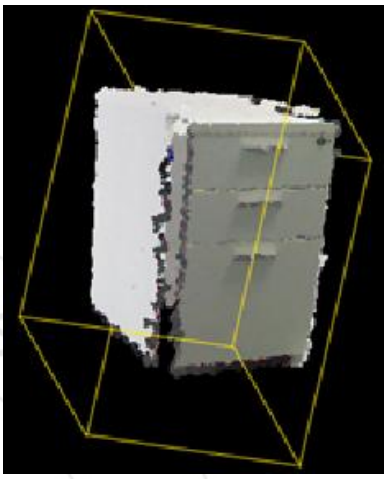

Figure 9: Example of an error when using PCA to construct Bounding Box

Firstly, we have to find the convex polyhedron of the point cloud. The polyhedron is represented by polygonal surfaces. [Figure 10] illustrate the polyhedron.

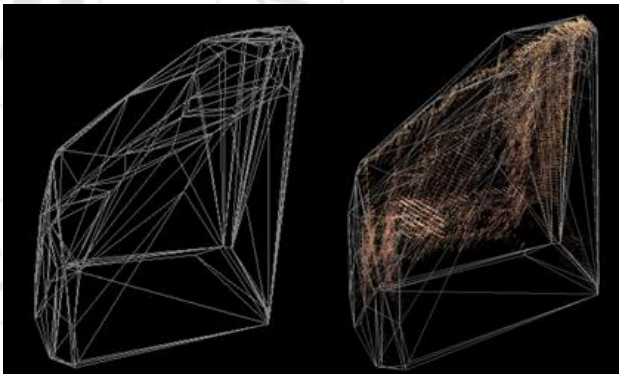

Figure 10: Convex polyhedron and its Faces (Vertices) of the object

After retrieving the convex polyhedron, for each of its surfaces we will find a bounding-box with a face belongs to the same plane with this polygon. The box that has minimum volume represents the size of the object. [Figure 11] gives the projection of output bounding box on the actual object.

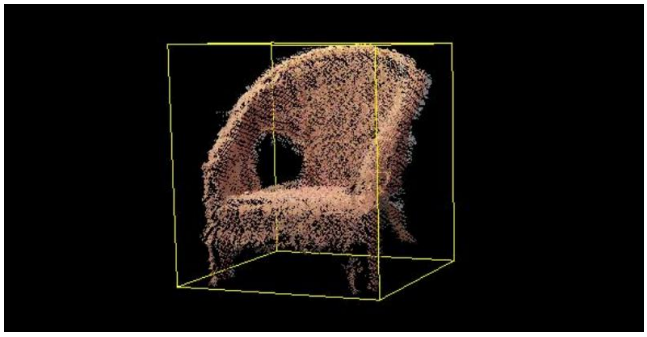

Figure 11: Bounding-box of the measured object

3. Experiment

We have tested with several objects with many types of objects and obtained some positive results (see [Figure 12 and 13]). [Figure 12] shows that our methods can provide better bounding box for low quality point cloud than the method of Darko Dimitrov in [9].

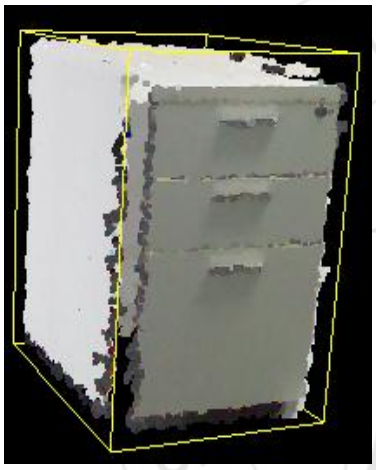

Figure 12: Better Bounding Box than Using PCA

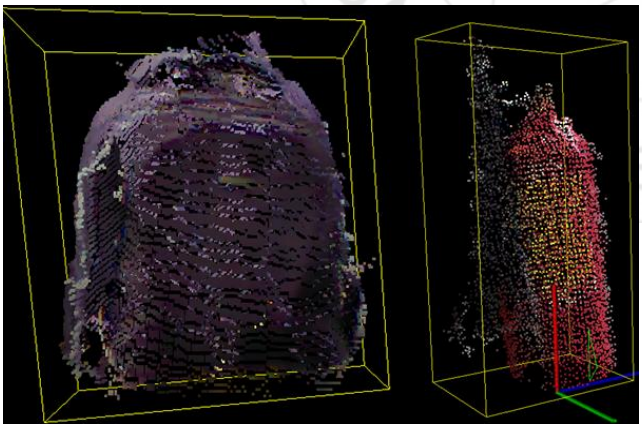

Figure 13: Computed Bounding Box of other objects

We have done a lot of experiments, but in this section we only report the last 10 measurements on three objects. We observed the average number of errors are around **+- 2cm** for each field of the size. [Table I] shows the detail of results, we denote R_ is the real value of the field and E_ is the average of errors. W, H and L are respectively width, height and length.

Table 1: Average number of errors recorded after 10 measuring times of 3 objects.

Object ID	**R_W**	**R_H**	**R_L**	**E_W**	**E_H**	**E_L**
1	45	35	55	+3.21	-2.23	-2.65
2	35	30	20	+1.24	1.54	-2.44
3	35	27	43	+2.25	-2.08	-2.45

Three tested objects are different shapes so the size of point cloud we got from the scene was different. We have also measured the speed of processing on each object. The experiments done using computer: core i3 6300, 4GB RAM. The results of the processing time are shown in [Table II].

Table 2: Average number of retrieved points of the scene and corresponding processing time

Object ID	size of the scene (points)	Time (seconds)
1	~2,000,000	~3
2	~1,850,000	~2.7
3	~1,928,000	~2.75

4. Discussion

An automated method of measuring the size of object bounding-box was introduced. The method includes the combination of several image processing techniques, 3D data processing techniques and machine learning techniques. It is entirely feasible for industrial applications, not only the logistics system but also other fields. The results from the experiments show that the system can operate with high performance, can completely replace the human effort. Although the outputs still contain a certain error, we still believe that these errors can be improved through the use of 3D industrial cameras that are better than the Kinects. The calibration algorithm has not been optimized yet, because we have re-used all the default parameters of the author. We will carry out the measurements needed to optimize this in the future work

References

[1] Y.M. Mustafah, R. Noor, H. Hasbi, and A.W. Azma, “Stereo Vision Images Processing for Real-time Object Distance and Size Measurements,” International Conference on Computer and Communication Engineering, pp. 659-663, 2012.

[2] Suraphol Laotrakunchai, Akarapas Wongkaew, Karn Patanukhom, “Measurement of Size and Distance of Objects Using Mobile Devices”, 2013 International Conference on Signal-Image Technology & Internet Based Systems.

[3] S. Rodríguez-Jiménez, N. Burrus and M. Abderrahim. "3D Object Reconstruction with a Single RGB-Depth Image". International Conference on Computer Vision Theory and Applications (VISAPP). Barcelona, Spain, February, 2013.

[4] “Camera Calibration and 3D Reconstruction”, http://docs.opencv.org/modules/calib3d/doc/camera_calibration_and_3d_reconstruction.html Last view 18.05.2015.

[5] M. Rufli, D. Scaramuzza, and R. Siegwart. "Automatic detection of checkerboards on blurred and distorted images." IEEE/RSJ International Conference on Intelligent Robots and Systems. (2008)
[6] "Demo software to visualize, calibrate and process Kinect cameras output"http://nicolas.burrus.name/index.php/Research/KinectRgbDemoV4?from=Research.KinectRgbDemo#tocLink10. Last view 14.03.2017
[7] A. M. Fischler and C. R. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," Communications of the ACM, vol. 24, no. 6, pp. 381–395, June 1981.
[8] Gill Barequet and Sariel Har-Peled, "Efficiently Approximating the Minimum-Volume Bounding Box of a Point Set in Three Dimensions" , June 30, 2001.
[9] Darko Dimitrov, Christian Knauer, Klaus Kriegel, G¨unter Rote, "On the Bounding Boxes Obtained by Principal Component Analysis"
[10] J. O'Rourke, "Finding minimal enclosing boxes," International Journal of Parallel Programming, vol. 14, pp. 183–199, 1985. 10.1007/BF00991005

Author Profile

Ngo Tung Son is currently working at Computing Fundamental Department, FPT University as a Lecturer and R&D engineer at Panasonic Hanoi Laboratory, Vietnam .He graduated a Bsc in Computing since 2011 and master degree in computer science at University of Avignon, France since 2014. His researches mainly focus on intelligent and adaptive system.

Bui Ngoc Anh is an IT specialist. He is working at FPT University as IT lecturer. He had more than 15 years of experience in IT System consultant and development. He graduated a master degree in computer science at Hanoi University of Technology, Vietnam. His researches lies on Mobility Technology, Information Assurance, IoT applications.

Tran Quy Ban received his engineer degree in computer science at Hanoi University of Technology, Vietnam, since 2004 and master degree in ICT at University of Poitiers since 2015, France. Currently he is working at FPT University as lecturer. His research interests include industrial application, software architecture, software testing.

Tran Binh Duong is an IT Lecturer at FPT University. He has 10 years of experience in software development. He gained an engineer degree at Hanoi University of Technology since 2005 and a master degree in ICT at University of Poitiers, France since 2016. His research interests focus on Image processing, datamining and especially recommender system.