International Journal of Latest Research in Engineering and Technology (IJLRET)
ISSN: 2454-5031
www.ijlret.com‖ Volume 2 Issue 2‖ February 2016 ‖ PP 49-56

Upper-Nonconvergent Objects Segmentation Method on Single Depth Image for Rescue Robots

Hoang Vu, Seoungjae Cho, Seongjo Lee, Kyungeun Cho*
Department of Multimedia Engineering, Dongguk University, Republic of Korea
**Corresponding author*

ABSTRACT: In this study, we proposed an algorithm to segment a special type of objects which we called the Upper-Nonconvergent Objects (UNOs) from the depth image. This type of objects plays an important role in our system. In that system, we use a robust algorithm based on height map to segment objects in three dimensional data. However, this algorithm cannot be segment the Upper-Nonconvergent Objects. Our algorithm works on two dimensional depth image which produced from 3D data. We first segment all the objects from the depth image including the ground using splitting and merging approach, then use morphological erosion method to extract the boundary of every objects, finally, we extract the Upper-Nonconvergent Objects using mathematical requirements which we defined. Information extracted from this algorithm can be used as additional process to the whole algorithms works on three dimensional data.

KEYWORDS - Upper-Nonconvergent Objects; UNO; height map; depth image; splitting and merging;

I. INTRODUCTION

One of the important issues for autonomous robots lies on how to create an effective object segmentation algorithm. Many formal methods and techniques from computer science have been proposed to solve this problem.

Depth imaging technology has advanced dramatically over the last few years. Pixels in a depth image indicate the calibrated distance in meters of 3D points in the world from the imaging plane, rather than a measure of intensity or color. There are two types of depth maps. The first depth map shows luminance in proportion to the distance from the camera. Nearer surfaces are darker; further surfaces are lighter. The second depth map shows luminance in relation to the distances from a nominal focal plane. Surfaces closer to the focal plane are darker; surfaces further from the focal plane are lighter.

The increasing availability of fully three-dimensional sensor is enabling to scan 3D volumes resulting in a cloud of 3D points. This allows detection of all kinds of objects and the ground is also sampled. However, the vast amount of data poses a great challenge on the algorithms.

Our project is concerned with the problem of detecting objects based on three dimensional data which received from Velodyne sensor device. Our robust algorithm based on height map, and it works directly on three dimensional data. However, there is a special types of objects in the environment which we call the Upper-Nonconvergent Object (UNOs) cannot be fully recognized.

To overcome this problem, we proposed a new algorithm works only on two dimensional depth image. The depth image is produced from the 3D data and can be used as input. This algorithm can recognize the Upper-Nonconvergent Objects by analyzing the mathematical properties of the boundary of all the objects in the environment. The output of this algorithm and be used as additional information to segment all the objects in robust time.

II. RELATED WORK

In autonomous robots research area, a lot of segmentation algorithms based on stereo images or a combination of visual imagery and 3D scanning. A. Hoover, G. J.-Baptiste, X. Jiang, P. J. Flynn, H. Bunke, D. B. Goldgof, K. Bowyer, D. W. Eggert, A. Fitzgibbon, and R. B. Fisher et al [1] introduces an algorithm based on scan line segmentation and region growing algorithm; the seed region is a triple of line segments on three adjacent scan lines with a minimum length, a minimum overlap, and a maximum distance between the neighboring points on two adjacent scan lines. B. Enjarini, A. Gräser et al [2] apply planar segmentation directly on depth images using gradient of depth feature. However the clustering process take much time and cannot apply in real-time.

In related area, some research use combination of depth images and color image, C. Plagemann, V. Ganapathi, D. Koller, and S.Thrun et al [3] focus on the task of accurately detecting parts of the human body, such as the head, hands and feet, from a single color depth image. Their algorithm focuses on detecting detected interest points and estimates the local pose and does not suitable for extracting the geometric property the objects. J. Rock, T. Gupta, J. Thorsen, J. Gwak, D. Shin, D. Hoiem et al [4] try to recovery of 3D shape from a depth image by inferring from earlier views or handling and symmetries. They retrieve a similar 3D mesh exemplar based on depth image matching, then deform that mesh to fit the observed depth point and predict the shape of the object based on the depth and exemplar mesh from a database of 3D models. J. Salas and C. Tomasi et al [3] combines color and depth images to detect people in indoor environments using weighted graph. They model the background as a mixture Gaussian and extract the foreground by maximum a posteriori (MAP) estimation. They also use color images combine with depth information to model the background.

III. UPPER-NONCONVERGENT OBJECT SEGMENTATION ALGORITHM

1. Upper-Nonconvergent Object

Segmentation algorithm applying to three dimensional data requires much cost also in resource and processing time. In our system, we develop a good algorithm based on heightmap. This algorithms works very well with three dimensional data, however it cannot fully recognize one type of objects which some part of them may obscured by another part from the top view. The result of this algorithm include some objects which their shape may not the right one.

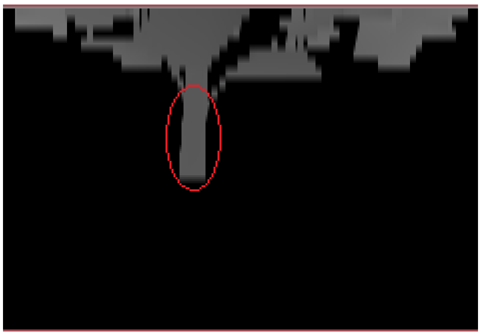

Figure 1. UNO object in depth image

As we see in Figure 1, the bottom part of the tree will completely be obscured from to top view, so that we cannot see that part from the top. The segmentation algorithm rely on height map cannot fully recognize this type of object.

We need to develop another algorithm which produce information about this type of objects in robust time, so we cannot apply directly to three dimensional data. Our approach is that we produce the depth image of the 3D dataset, and apply an algorithm to extract information about UNOs only on that depth image. The processing time is very good because of the small size of depth image.

At first, we need to formalize that Upper-Nonconvergent Object with a formal mathematical definition.

Definition 1:

The Upper-Nonconvergent Object is an object that has at least one of these two properties:

- $d_m > d_b$

d_m = maximum horizontal size of the object

d_b = horizontal size of the bottom part of the object

- Assume that:

Set B : all the boundary points of the object

p, q: any two points in set B such that $p_y = q_y$

Set $L = \{ d \mid d = t*p + (1-t)*q\}$, with every t in [0,1]

If $L \cap B = \{p,q\}$ so B is a boundary of an Upper-Nonconvergent Object.

Figure 2 show some examples of Upper-Nonconvergent Objects. All of these objects will lack some part of it if we see from the top view

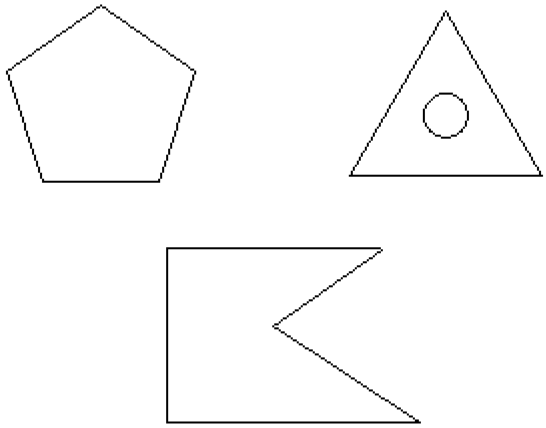

Figure 2. UNOs shape

The first condition is very simple and also easily to calculate and to check the size d_m and d_b. Figure 3 explain this condition with a simple shape.

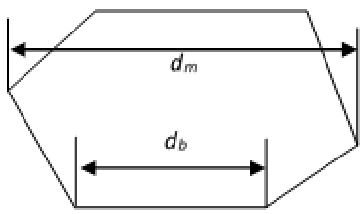

Figure 3. First condition of UNOs

The second condition is more difficult to check, we need to analyze the geometrical properties of the object. So the boundary of the object must be extracted first, then we need to check all the meeting points between every vertical line with the boundary. Figure 4 illustrates the second condition of the UNO.

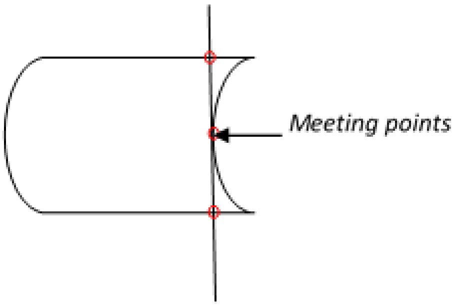

Figure 4. Second condition of UNOs

The special case that the objects pass the second condition but not an UNO also must be considered. As in Figure 5, the object’s boundary is the *ABC* triangle, the line through A and B meet the boundary at every points between A and B, so this object passes the second condition, but it is not an Upper-Nonconvergent Object.

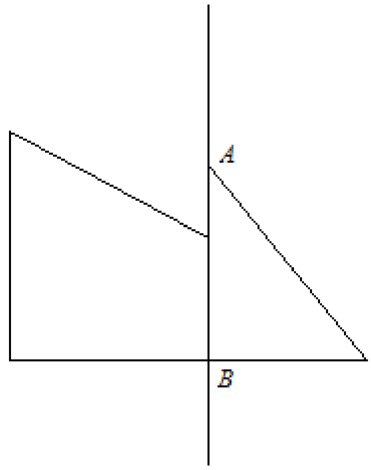

Figure 5. Special case of second condition

2. Objects segmentation

In this stage, we will segment all the objects of the depth image based on the depth information, each object will be labeled with unique integer ID. The ground also must be removed.

Most of the segmentation algorithms are based on one of two basic properties of intensity values: discontinuity and similarity. We use region splitting and merging approach which partition the depth image in to regions that are similar.

Our algorithm based on analyzing the properties of the depth image which has already produced from 3D cloud data.

Figure 6. Depth map

Figure 6 is an example of a depth map. As we see, every object is placed on the ground or on top of another. One object can be recognized by its gray value (depth information), the depth value almost constant inside the object. Two different objects have different depth value, it turns out that at the boundary points there are some significant changes of the depth value. Another important thing to extract the ground is that the boundary points of objects placed on the ground has smaller depth value than the ground under it. The depth information of the ground points are almost constant, that mean two horizontal adjacent points have similar gray value. In vertical comparison, the depth value is different but the changes is more smoothly. And that is the most important aspect that we will use to remove the ground.

We start by splitting the 2D depth image into small region with the horizontal size equal 1 that we call the initial region. Our dataset produce a depth image with the size 1080 x 32 which we separate into 1080 initial regions. Each initial region has size 1 x 32. We just only need to care about vertical differences of depth value between the adjacent points.

Figure 7 illustrates one initial region with the horizontal size equal 1, in vertical it can be segmented by different parts based on their different gray value. This figure is seemed unreasonable, because its size is not actually equal 1, we create that figure with the bigger size in horizontal to clarify the difference between depth values of adjacent part in vertical.

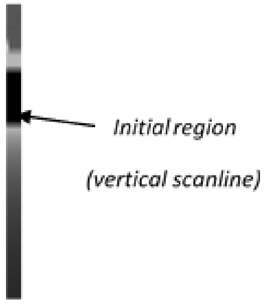

Figure 7. Splitting region

Our algorithm segment the initial region into separate parts by keep track all their boundary points. All points inside one part must has the difference of depth information under the threshold value. At boundary points there's a significant changes in depth. This process also label each separate part of initial regions with a unique ID.

Figure 8 shows the initial region with its separated parts, the boundary points denote by red circle. The mean depth value of each part also be calculated for later merging process.

In this figure, we can see the difference of the ground part with object part easily.

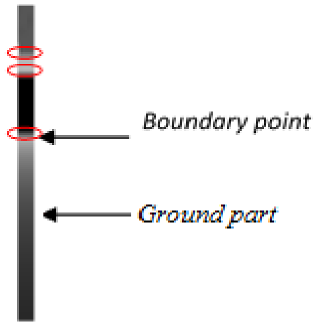

Figure 8. Boundary points

We do not consider the area that has depth value equal zero. At that point there is no information about objects in that area, we call all these points as infinite points. Technically, we do not have responded laser beam from those objects in scanning process.

However, it easy to see that all the points has depth value upper nearby the infinite points always the boundary points of one object and cannot be the ground points.

The process continue by merging all the part of consecutive initial object which already be segmented.

Two parts of two consecutive initial region can be merge into one if they have both of these two conditions:

- The different of the mean of the depth value of them is smaller than a threshold value
- Any point in one part has at least one neighbor which is the point belong to the part need to be merged with. That is the different of the depth value between that two points is smaller the threshold value.

That mean two parts can be merged if they have at least one neighbor. The similarity in this situation is the depth distance.

If two parts can be merged, we label both of them with the id of the smaller one. Figure 9 illustrates merging process.

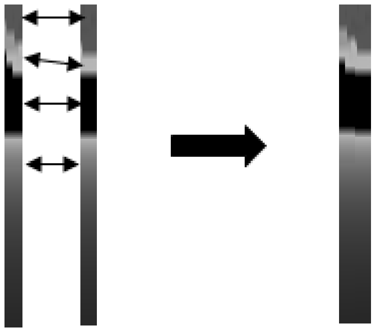

Figure 9. Merging process

After merging process, all the objects of the scene has its own unique id

In this Figure 10, we print out the medium result by assigning each object one color. And the ground had removed

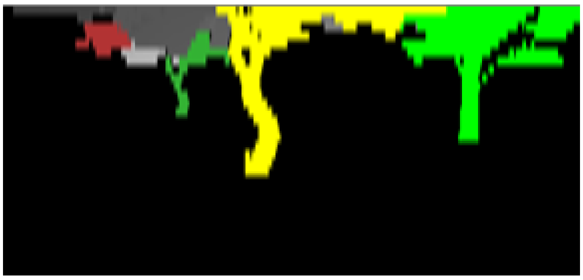

Figure 10. Labeling result

3. *Upper-Nonconvergent Object recognition*

We have segmented all the objects, with its shape. But we just only interest the Upper-Nonconvergent Objects. This process can be done in second stage of this algorithm in which we use some techniques to apply the mathematical concepts which we have already defined.

In this stage, we follow strictly the formal properties of UNOs. The first condition is easily to check. The second one needs some additional works to extract the boundary of objects that we solve by morphological erosion technique.

Morphological techniques is a tool for extracting image component that are useful in the representation and description of region shape.

As we defined, the Upper-Nonconvergent Object is decided by the geometrical properties of its boundary. There are many techniques to extract the boundary of an object. We chose morphological technique because of its simplicity and robustness.

The boundary of a set A, denoted by *b(A)* can be obtained by first eroding A by *B*, and then performing the set difference between A and its erosion. That is:

- $b(A) = A - (A \ominus B)$

Where B is a suitable structuring element (SE).

Structuring element is small set or sub-image used to probe an image under study for properties of interest.

Figure 7 illustrates the chosen structuring element in our method. This 3x3 structuring element has the origin in the center.

The operation of creating a new set using structuring element B from set A as follow: Create a new set by running B over A so that the origin of B visits every element of A. At each location of the origin of B, if B is completely contained in A, mark that location as a member of the new set.

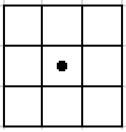

Figure 7. Structuring element in erosion process.

Figure 11 show the medium result of this stage. Different objects denote by different color, also their ground was extracted by morphological erosion technique.

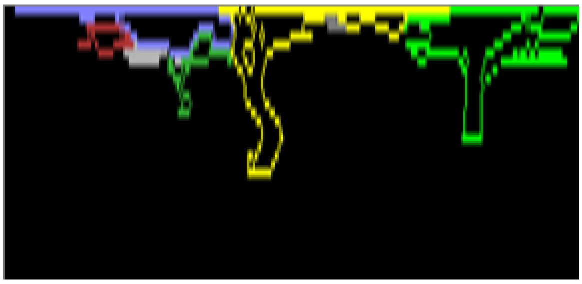

Figure 11. Boundary extracting

According to the mathematical definition of the Upper-Nonconvergent Objects, we can extract them using two conditions

The first condition related to the boundary size in x-y plane. A naïve approach is just looping through all the point inside an object. Practically, we can keeps track the size of objects in boundary extracting process. The number of points need to check is significant decreased after we extracted the boundary. The Upper-Nonconvergent Object has the maximum horizontal size higher than the bottom size.

To check the second condition, we find all the meeting points of every vertical line which go through any points of the object with the boundary of an object.

In special case that we have mentioned, it need more works to compare the number of meeting points and the number of points lying on AB. In theory, the number of points lying on AB is infinite, but in practical implementation it is countable.

IV. EXPERIMENT AND PERFORMANCE ANALYSIS

Our test system using the depth image get from laser sensor with size 1080x32

The laser sensor has FOV in vertical from +10.67 to -30.67 degree, and 360 degree FOV in horizontal.

Our data is urban area, almost the objects are Upper-Nonconvergent Objects, as we see in the figure 11, there are two non UNOs objects denoted by red circle which completely removed from Figure 12 – the result after using our algorithm.

With the size of the mentioned data, the processing time is about 45ms per one depth map with 1080 x 32 size.

The result of this algorithm can be used as additional information in complicated segmentation algorithm on three dimensional data.

Figure 12 shows the input depth image with a Non-Convergent Object, and this object was removed in the result of segmentation process as in Figure 13

The processing time is robust enough which can be apply to real-time system

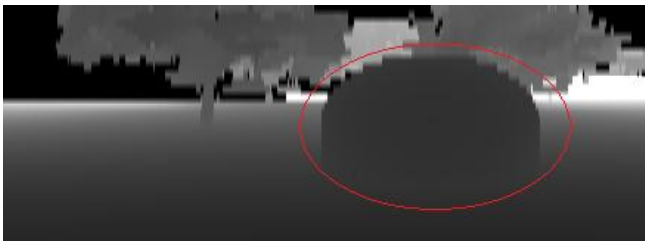

Figure 12. Input depth image

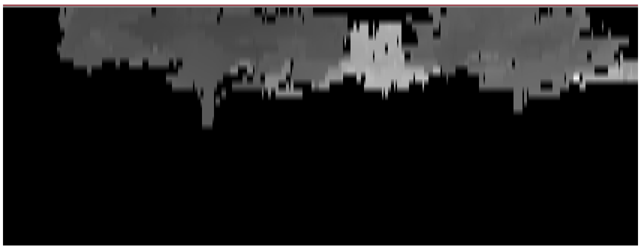

Figure 13. Extracting result

V. Acknowledgements

This research was supported by Basic Science Research Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Science, ICT and future Planning (NRF-2015R1A2A2A01003779).

REFERENCES

[1]. A. Hoover, G. Baptiste, X. Jiang, P. J. Flynn, H. Bunke, D. B. Goldgof, K. Bowyer , D. W. Eggert, A. Fitzgibbon, R. B. Fisher, "An experimental comparison of range image segmentation algorithms," *IEEE Transactions on pattern analysis and machine intelligence, 1996*

[2]. B. Enjarini, A. Gräser, "Planar Segmentation from Depth Images using Gradient of Depth Feature" *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS); 2012*

[3]. C. Plagemann, V. Ganapathi, D. Koller, and S. Thrun, "Real-time Identification and Localization of Body Parts from Depth Images" *Robotics and Automation (ICRA), 2010 IEEE International Conference on, 3108-3113*

[4]. J. Rock, T. Gupta, J. Thorsen, J. Gwak, D. Shin, D. Hoiem, "Completing 3D Object Shape from One Depth Image" *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (**CVPR**), 2015*

[5]. J. Salas and C. Tomasi, "People Detection using Color and Depth Images" *Proc. of Mexican Conference on Pattern Recognition (MCPR), pp. 127 – 135 (2011)*

[6]. S. Song, J. Xiao, "Sliding Shapes for 3D Object Detection in Depth Images" *13th European Conference on Computer Vision (ECCV2014)*

[7]. L. Chen, X.L Nguyen, C. Liang, "Object segmentation method using depth slicing and region growing algorithms" *Report NTUT*

[8]. Rafael C. Gonzalez, Richard E. Woods, *Digital Image Processing* (Prentice Hall, 3rd edition, 2007)