Implementation of 3D Object Recognition and Tracking

Pankaj Bongale, Anurag Ranjan, Sahil Anand
India Innovation Labs
Bangalore, India

***Abstract*—In this paper, we discuss an object recognition and tracking system that utilizes the depth information from a low-cost depth sensor. Conventional object recognition methods that utilize RGB cameras are unable to accurately identify objects in the real world since they do not take into consideration the shape and three-dimensional characteristics of the object. Another major factor determining the accuracy of recognition is the lighting conditions and object pose at the time of recognition. We discuss an approach making use of the depth information and 3d properties of objects inorder to accurately identify them independent of lighting conditions.**

Keywords-3D object recognition, viewpoint feature histogram, point cloud, flann.

I. INTRODUCTION

The image obtained by the depth camera can be represented in the form of Point Clouds. Point clouds are a set of points represented by XYZ co-ordinates which are used to model the environmentas three dimensional clouds. These depth points or Point Clouds are generated for further processing.This work focuses mainly on 3D features and not the 2D imagery. The color space is not considered during processing. Therefore, the recognition is based purely on the shape and structure of the object and not on its color.

This paper presents implementation of VFH descriptors for the deployment of an Object Recognition and Tracking system through depth images obtained by depth sensor camera.The depth sensors used in the implementations Microsoft Kinect that is capable of giving depth images at high frame rate which supports deployment of this system in practical scenario. The system is tested for 35 objects with over 250 different poses to check for its reliability and accuracy. We assume that the objects being recognized are not transparent or reflective. The objects are in light clutter so that they can be easily segmented. The objects can contain textures or fragments of different colors. The system is quite reliable to lighting conditions. The recognition accuracy, unlike in traditional image based recognition systems, is not affected by lighting conditions since a depth camera is utilized.

The structure of this paper is as follows: the related work in this area has been described in section II. The description of the feature used in recognition is given in section III. The detailed step-wise procedure of theimplementation of the system is discussed in section IV. The experimental set up and recognition performance are discussed under Experimental Results in section V. Conclusions and Future work are discussed in section VI.

II. RELATED WORK

The problem of Object recognition has been addressed since a long time in different ways. Several object recognition algorithms for 2D images have been successfully developed. However the data obtained from 2D RGB cameras is not sufficient for deployment in the real world.One example would be its inability to differentiate between an actual image and its picture. 3D processing is capable of overcoming such issues. This field of 3D image processing has attractedareas such as Robotics, Computer Graphics and Game Development. Several methods to obtain 3D scans of natural images are available. Laser scanners give very accurate results but are not economic for mass deployment. Other approaches include Laser rangefinders, 3D Flash LIDAR, 2D or 3D sonar sensors and multiple 2D cameras for stereovision.Recent advancement has taken place to introduce low cost depth sensing devices by Primesense, Asus etc. which are economic for deployment in robotic applications. One such example is Microsoft's Kinect sensor.

Point Feature Histograms which represent signature of a particular 3D image have been described in [1]. The pre-processing of Point Clouds involves filtering using a Pass through Filter. KdTree search techniques which are used for finding K nearest neighbors of a specific point or location have been described in [3]. The segmentation is necessary to extract a cluster which is to be recognized. Euclidean Cluster Extraction was implemented for this purpose. A novel feature called Viewpoint Feature Histograms (VFH) has been discussed in [2]. The use of this feature enables detection of different poses of similar objects and facilitates high recognition rates.

III. VIEWPOINT FEATURE HISTOGRAM

Point feature histogram (PFH) [1]is generated by binning the set of all quadruplets (α, Φ, θ, d) [1] generated for each query point p. The binning process divides each features value range into subdivisions, and counts the number of occurrences in each subinterval. In order to create the quadruplet, Darbouxuvw co-ordinates are used.

$$\alpha = \mathrm{v} \cdot \mathrm{nt} \tag{1}$$

$$\varphi = \frac{\mathrm{n} \cdot (\mathrm{pt} - \mathrm{ps})}{\mathrm{d}} \tag{2}$$

$$\theta = \tan^{-1} \frac{\mathrm{ns} \cdot \mathrm{v}}{(\mathrm{ns} \times \mathrm{v}) \cdot \mathrm{nt}} \tag{3}$$

978-1-4673-0255-5/12/$31.00 ©2012 IEEE

where

$$v = \frac{ns \times (pt - ps)}{d} \quad (4)$$

Here, d is Euclidian distance between the two points ps and pt, source point and the target point respectively. ns and nt are the normals at these points respectively.

Simplified PFH or SPFH is generated when the quadruplets are calculated for each point and its neighbors only. Further, for each point its k neighbors are re-determined, and the neighboring SPFH values are used to weigh the final histogram called Fast Point Feature Histogram (FPFH)[1].

Viewpoint Feature Histogram [2] consists of two parts: a viewpoint direction component and a surface shape component comprised an extended Fast Point Feature Histogram (FPFH). The viewpoint component is computed by collecting a histogram of the angles that the viewpoint direction makes with each normal. The computational complexity of VFH as mentioned in [2] is O(n).

IV. Design And Implementation

A. Sensing Camera

The depth sensors of Microsoft Kinect are used as the sensing camera. The device includes an RGB camera, infra-red depth sensors and microphones. However, we use only the depth sensors of the device.

B. Interfacing

The device is connected to the computer and interfaced using OpenNI Grabber framework by Point Cloud Library. OpenNI provides an application programming interface (API) for writing applications utilizing the 3D image data. This API covers communication with both low level devices (e.g. vision and audio sensors), as well as high-level middleware solutions (e.g. for visual tracking using computer vision). The depth images obtained by the sensors are obtained in the form of 'Point Clouds'.

C. Recognition

The object recognition involves two distinct steps, training stage and testing stage.

1) Training Stage: During the training stage, we capture a 3D snapshot of the object (being trained for) that is kept on a plane background like floor or table which is uniformly illuminated. This snapshot taken from depth sensors is stored in the Point Cloud Data format.

The object of interest in the snapshot is extracted using Euclidean Cluster Extraction. This clustering results in a number of clusters ranging from 2 to a higher value like 20, depending on the uniformity of background, one of the clusters being the object itself. The cluster which represents the object of interest is chosen and the rest of them are discarded.

The Viewpoint Feature Histograms (VFH) of this cluster are then obtained. This process is repeated for different poses of the same object and for different objects. Thus all the poses and their VFH descriptors are obtained and stored in the dataset. All such histograms for different poses are converted into FLANN[3] format and then added to the dataset.

A K-d tree of the dataset being trained is obtained for recognition purpose. Every time a new object is added to the dataset, K-d tree is to be updated.

2) Testing Stage: The objects are kept in front of the 3D camera for recognition. These objects are similar to the ones that were used during training stage. The 3D image of the scene is obtained using OpenNI Grabber Framework. The obtained depth image is clustered using Euclidean Cluster Extraction. For every cluster in the depth image, Viewpoint Feature Histogram (VFH) is estimated. The VFH descriptors of each of the clusters are matched with VFH descriptors of the objects present in the dataset using the trained K-d tree. The closest neighbor from the dataset gives the object (and pose if required).

Figure 1: Test object samples

V. Experimental Results

Our system was trained for the following five objects: bottle, coffee mug, tissue box, book, soccer ball (figure 1). It is necessary that the objects used are almost non-reflective and non-transparent. Hence the reflective bottles were covered with opaque sheets during the training session. We used a huge set of each of these objects and trained the system for different poses of the objects, the number of poses for training varied from object to object based on their symmetry. Table I shows the details of the training stage. The database used for recognition consisted of a total of 11 objects trained for 33 different poses. However, it was necessary that the objects used during recognition should not merge with the other objects present in the surrounding since this might lead to improper segmentation during initial stages. The K-d tree created for this database during the training session was used to recognize the objects.

During the testing stage, the objects used for recognition purposes were similar to the ones from the dataset. The experiments showed accurate results giving the object name, pose and its co-ordinates (figures 2, 3, 4). Table II shows the results obtained during the testing stage. The system was

tested for a total of 34 objects. The experiment was performed in fairly uniform lighting conditions. Objects' reflective properties were not altered during this stage. We could successfully recognize multiple objects simultaneously provided they are not physically in contact.

TABLE I. TRAINING STAGE

Object	*No. of training samples*	*No. of trained poses/sample*	*Total no. of trained poses*
Coffee Mug	4	8	32
Deodorant Bottle	2	8	16
Tissue box	2	8	16
Book	2	2	4
Soccer Ball	1	1	1

TABLE II. TESTING STAGE

Object	*No. of test samples*	*No. of tested poses/sample*	*Total no. of tested poses*	*Success rate*
Coffee Mug	8	8	64	98.43%
Deodorant Bottle	8	8	64	100%
Tissue box	8	8	64	100%
Book	8	2	16	100%
Soccer Ball	2	1	2	100%

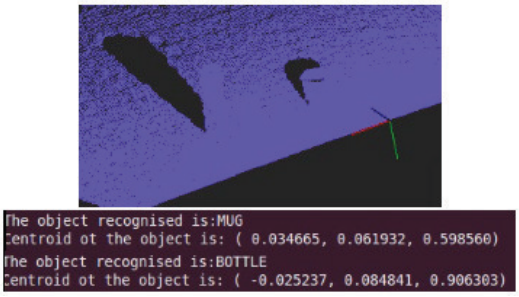

Figure 2: Test results – mug and bottle

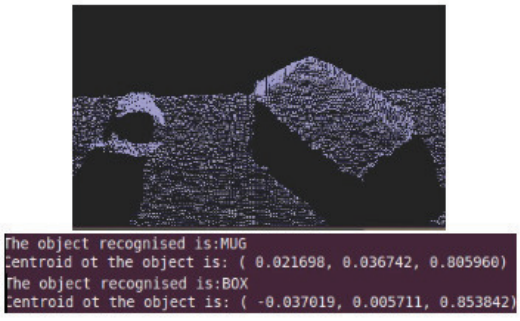

Figure 3: Test results – mug and box

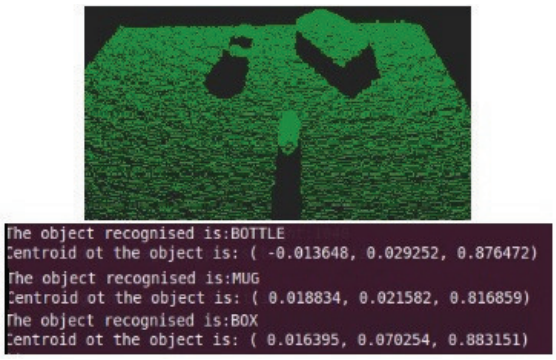

Figure 4: Test results – bottle, mug and box

VI. FUTURE WORK

We plan to improve the current system we have developed to gather more information on the objects. For this, we look up to exploitation of the RGB camera present on the Kinect also. Our work can be integrated with text recognition in Natural Images which would allow the system to differentiate between objects having similar shapes but different labels. Color recognition can also be employed in the system. This work can be carried forward to estimate shape, volume and other physical features of the objects. The inclusion of these features in the system can lead to deployment of the system in real time scenarios.

REFERENCES

[1] Radu Bogdan Rusu, "Semantic 3D object maps for everyday manipulation in human living environments," Artificial Intelligence (KI - Kuenstliche Intelligenz), 2010.

[2] Radu Bogdan Rusu, Gary Bradski, Romain Thibaux, John Hsu,Willow Garage, "Fast 3D recognition and pose using the viewpoint feature histogram", Proceedings of the 23rd IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS).

[3] Marius Muja and David G. Lowe, "Fast approximate nearest neighbors with automatic algorithm configuration," International Conference on Computer Vision Theory and Applications (VISAPP), Lisbon, Portugal (Feb 2009).

[4] Iryna Gordon and David G. Lowe, "What and where: 3D object recognition with accurate pose"in *Toward Category-Level Object Recognition,* eds. J. Ponce, M. Hebert, C. Schmid, and A. Zisserman, (Springer-Verlag, 2006), pp. 67-82..

[5] Vinod Nair and Geoffrey E. Hinton, "3D object recognition with deep belief nets," NIPS 2009: 1339-1347.

[6] A. K. Jain and C. Dorai, "3D object recognition: Representation and matching," Statistics and Computing, vol. 10, no. 2, pp. 167–182, 2000.

[7] G. Burel and H. H´enocq, "Three-dimensional invariants and their application to object recognition," Signal Process., vol. 45, no. 1, pp. 1–22, 1995.

[8] A. S. Mian, M. Bennamoun, and R. Owens, "Three-dimensional model-based object recognition and segmentation in cluttered scenes," IEEE Trans. Pattern Anal. Mach. Intell, vol. 28, pp. 1584–1601, 2006.