ISSN XXXX XXXX © 2017 IJESC

Research Article Volume 7 Issue No.5

Object Recognition using Feature Based Technique

Swati G. Shanbhag[1], Razia de Loyola Furtado E Sardinha[2], Cassandra .C. Fernandes[3]
M.Tech Student[1], Assistant Professor[2, 3]
Department of Information Technology [1, 2], Department of Computer Engineering[3]
PCCE, Goa, India

Abstract:
Object Recognition is an application in the field of Computer Vision for identifying objects in the real world. Humans can recognize objects with little effort despite the fact that they may vary when captured from different viewpoints. Objects can even be recognized when they are partially obstructed from the view. This task is still a challenge for computer vision systems. In this paper we present an efficient algorithm that recognizes the objects using its feature points and the Euclidean distance between them. First, the point cloud of the Object is partitioned using k-means algorithm then 20 points are selected as feature points from each cluster. These feature points represent the 3D object and are stored along with the object name in a repository. The target object is then matched with all the objects from the repository. This technique is invariant to rotation and translation.

Keywords: 3D scanners, Feature Points, k-means, Object Recognition, Point cloud.

I. INTRODUCTION

Objects can be represented in different ways. They can be represented as a collection of multiple images captured at different angles. We get the accurate model if we can capture more number of images per object but we need a lot of space to store information about a single object. Therefore this is not an efficient representation. Another way to represent 3D object is using Constructive Solid Geometry technique. It is based on the notion that a physical object can be divided into a set of primitives. These primitives can be combined in a certain order following a set of rules (Boolean operations) to create the object. The CSG object can be represented using a Binary Tree as shown in Figure 1. CSG representation is of considerable importance for manufacturing 3D Objects.

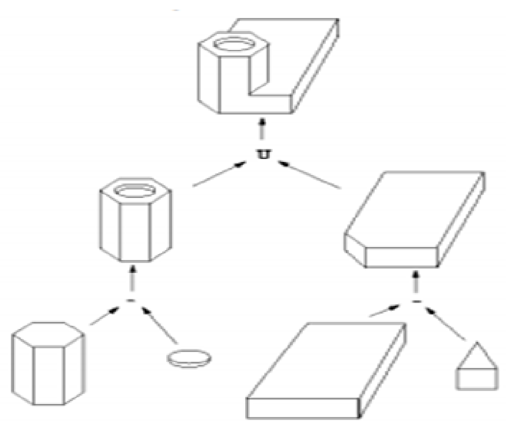

Figure.1. Binary tree of the csg object

Octree is a best technique to represent moving objects. Octree is a tree data structure in which each internal node has exactly 8 children. Octree partitions the three- dimensional space by recursively subdividing it into 8 octants. . If the block under consideration is completely contained within the object it is left alone; otherwise, it is divided into eight octants each of which is treated similarly. The blocks are divided until they lie completely within or completely outside the object until the block size is minimum. The octree representation for the object is shown in Figure 2a, 2b and 2c.

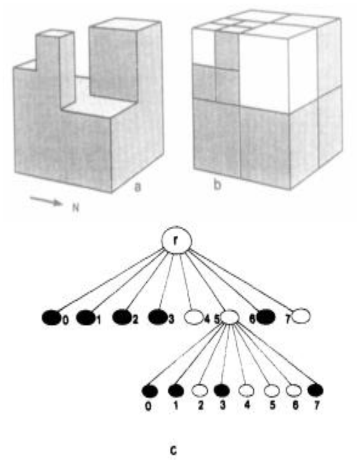

Figure.2.a. Simple Object b. Recursive subdivision of space superimposed on the object in 2a. To obtain its representation c. Octree for the object

In our research we used polygon mesh to represent a three dimensional object. We chose this representation because they are easy to represent, transform and can be used to model any object. Polygon meshes are collection of polygons or surfaces that forms the surface of the object. For a mesh model we specify a set of points in space and we connect various pairs of these points to form edges. A number of polygons may form a surface. And a number of surfaces will form an object. Polygon mesh of a dolphin is shown in Figure 3.

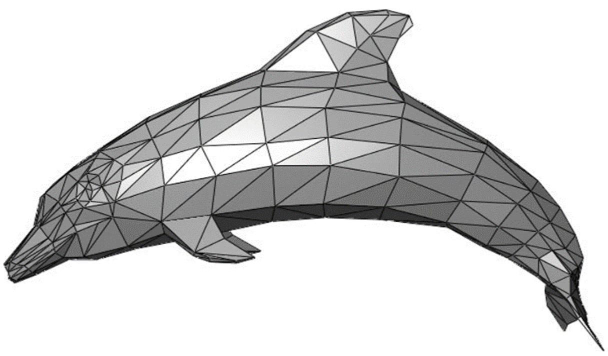

Figure. 3. Polygon Mesh of a Dolphin

We have used a ply file which stores 3dimensional data from the 3D scanners. Ply file represents the polygon mesh of a 3D object. From the ply file we have extracted the point cloud. The Point clouds are a set of points that are defined by the x, y and z coordinates. Since the 3D scanners are inexpensive devices we can easily generate the ply files from the data obtained from 3D scanners. We used the datasets of John Burkardt, (Department of Scientific Computing, Florida) to test our algorithm.

II. RELATED WORK

There are many object recognition techniques for detecting 2D objects in an image. SIFT (Scale Invariant and Feature Transform) and SURF (Speeded Up Robust Features) are the most leading techniques. SURF algorithm is divided into three main parts: Interest points extraction, neighborhood description of the interest point by a feature vector and matching the descriptor vectors of different images. In the matching process, the Euclidean distance between the descriptor of the trained image and all the other descriptors of the test image is computed.

If the variance between the two nearest neighbors is 0.7 than this pair is selected for matching. This technique is rotation and scale invariant. Since 2D images do not have depth information, the images cannot be rotated along the z axis. So if we say that SURF object detection is rotation invariant, we can have the rotations along the x-axis and y-axis. Shah, Bennanoum and Boussaid proposed a technique that uses keypoint based surface representations to detect 3D objects. This technique succeeded in recognizing occluded objects. The algorithm has two phases. Phase1 computes the keypoints based surface representations (KSR) for 3D Object.

KSR first identifies the keypoints and then computes the distances between these points to measure the geometrical relationships between them. Phase 2 is an online phase where the KSR's are computed for the 3D objects to be recognize. Then the two KSR's are matched using linear correlation coefficient.

III. PROPOSED METHOD

The System is divided into three steps namely point cloud extraction, Feature point detection and Object recognition. In step1 we extract the point cloud from the ply file. The ply file has a list of vertices and faces.

Therefore point cloud extraction would be the process of extracting the list of vertices from the ply file. In the second step, the key feature points will be found using k-means clustering algorithm.

The point cloud of a 3D object will be partitioned into clusters. n points from each cluster will be used as a feature points. A 3D Object represented as a set of these feature points will be stored in a database.

The algorithm runs accurately for n=20. In the final step, the target object will then be matched with all the objects stored in the database. Euclidean distance between the feature points and their neighborhood points of the stored object will be compared with that of the target object. If a match is found for the key feature points, the target object will be recognized to be the stored object it matches. The data flow diagram of the system is shown in Figure 4.

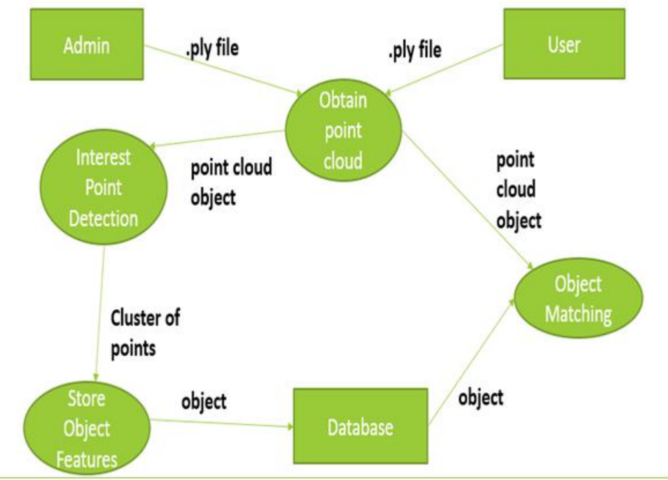

Figure. 4.Data Flow Diagram

IV. ALGORITHMS

We use k-means clustering algorithm to partition the point cloud and a matching algorithm for recognizing the target object. The matching algorithm will compare the target object with the objects stored in the database. If a match is found it will display the name of the object.

A. K-means algorithm

Let X = $\{(x_1,y_1,z_1),(x_2,y_2,z_2),\ldots(x_n,y_n,z_n)\}$ be the set of data points and V = $\{v_1,v_2,\ldots\ldots,v_c\}$ be the set of centers.

1. Randomly select *'c'* cluster centers.

2. Calculate the distance between each point and cluster centers.

3. Assign the data point to the cluster center whose distance from the cluster center is minimum of all the cluster centers.

4. Recalculate the new cluster center using:

$$V_i(x_i,y_i,z_i) = \frac{1}{ci}\left(\sum_{j=1}^{ci} xi, \sum_{j=1}^{ci} yi, \sum_{j=1}^{ci} zi\right)$$

5. Calculate the distance between each point and the new cluster centers.

6. If no data point was reassigned then stop, otherwise repeat from step 3.

B. Matching algorithm

Object I- object from the database.
Object J- object to be recognized.

```
Match(object I, object J)
{
Match_found=0;
for every point xi in I
do
neighbouri[]= neighbours of xi;
for every point yj in J
do
neighbourj[]= neighbours of yj;
```

if(distance between xi and all the points in neighbouri[] == distance between yj and all the points in neighbourj[])
then
increment Match_found; break;
if (Match_found == c)
then
print "Object recognized: I.name";
}

V. EXPERIMENTAL RESULTS

The System was trained with 10 objects and was tested against 52 objects. The test objects were captured from different angles. The algorithm could recognize same objects and objects with different viewpoints.

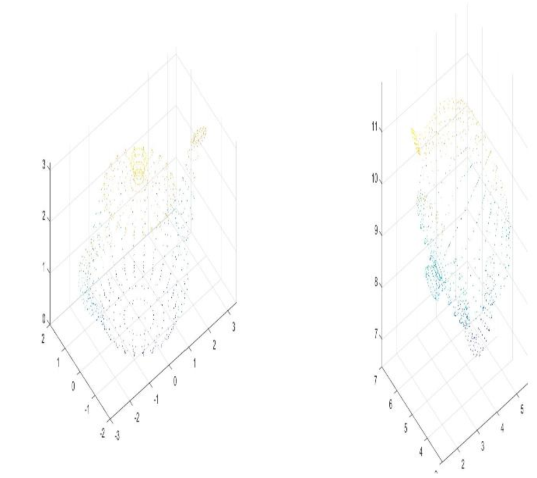

Figure.5. Objects with different view points recognized.

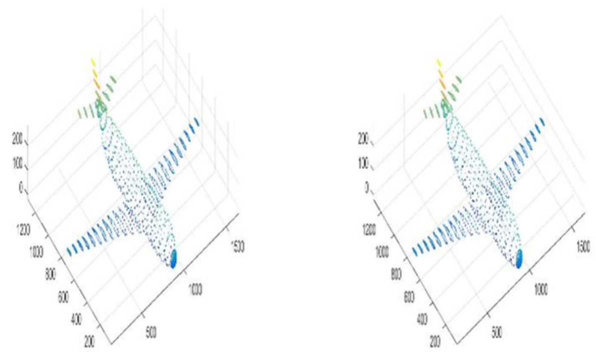

Figure .6. Same objects recognized.

VI. REFERENCES

[1]. S. M. Metev and V. P. Veiko, Laser Assisted Microtechnology, 2nd ed., R. M. Osgood, Jr., Ed. Berlin, Germany: Springer-Verlag, 1998. Syed Afaq Ali Shaha, Mohammed Bennamouna, Farid Boussaidb. (2017). "Keypoints -based surface representation for 3D modeling and 3D object recognition" Elseveir, Pattern Recognition , vol. 64, pp 29-38.

[2]. Ingo Rentschler, Markus Gschwind , Hans Brettel , Erol Osmana , Terry Caelli.(August 2008). "Structural and view-specific representations for the categorization of three-dimensional objects", Elseveir, Vision Research , vol. 48, pp 2501-2508.

[3]. M. Hassaballah, Aly Amin Abdelmgeid and Hammam A. Alshazly.(2016). "Image Features Detection, Description and Matching", Springer, International Publishing Switzerland.

[4]. Ivan Sipiran and Benjamin Bustos.(2011), "Harris 3D: a robust extension of the Harris operator for interest point detection on 3D meshes", Springer-Verlag.

[5]. https://people.sc.fsu.edu/~jburkardt/data/ply/ply.html

[6]. http://courses.cs.washington.edu/courses/cse576/book/ch14.pdf