Interactive System for Objects Recognition using 3D Camera, Point Cloud Segmentation and Augmented Reality User Interface

Matej Nikorovic[1,2], Radoslav Gargalik[1,2] and Zoltan Tomori[2]
[1]*Institute of Computer Science, Faculty of Natural Science, P. J. Safarik University, Kosice, Slovakia*
[2]*Institute of Experimental Physics, Slovak Academy of Science, Kosice, Slovakia*

1 ABSTRACT

Depth-sensing cameras are frequently used in computer vision and augmented reality applications as a key component of the point cloud acquisition system as well as a natural user interface tool. We integrated both these functions into the automatic objects recognition system based on the machine learning. Acquired point cloud is segmented by the region growing algorithm exploiting smoothness constraint as a homogeneity criterion. Segmented objects lying on the ground plane are recognized via the supervised machine learning and the corresponding label is is projected near to the object. Natural user interface controls the learning process as well as the mode of operation. System is proper for specific environments like e.g. science center (museum).

2 STAGE OF THE RESEARCH

Recently (September 2014), the first author of this paper became a PhD student at the Department of Computer Science, P. J. Safarik University, Kosice, Slovakia. This paper brings a review of literature, technical background and some preliminary results related to future PhD work.

3 OUTLINE OF OBJECTIVES

We plan to develop a simple and low-cost objects recognition system based on the supervised machine learning and install it in the science museum taking into account following specific conditions:

- Most of visitors are groups of children requiring simple, robust and self-explanatory control of the exhibits.
- No extra hardware such as keyboard, mouse, cables etc. is acceptable.
- The interaction with exhibits should exploit the virtual menu based on the same camera as the exhibit itself.
- The function of menu must be very simple – usually just select the mode of operation.
- Periodical innovation and upgrade of exhibits is the necessary condition to achieve repeating visits of the same people.
- The museum is open daily for more than 100 of visitors per day so the time for the installation and testing of exhibits in real conditions is limited.

Machine learning libraries became a part of popular open-source libraries (like OpenCV) which opened the new possibilities in analyzing and understanding images. New low-cost 3D cameras like Microsoft Kinect, Creative Senz3D or Asus Xtion can acquire both RGB and depth images at the same time. If these devices are properly calibrated, then we are able to create a set of 3D points (point cloud) from acquired images.

Our objective is to propose and implement a low-cost objects recognition system proper for the above mentioned specific conditions of science center (museum). It means the human interaction is required only in the supervised learning stage. Our goal is to implement the recognition system into two existing exhibits of ”Steel Park” science center – ”Interactive statue” and ”Interactive sandbox” described in (Tomori et al., 2015)

4 RESEARCH PROBLEM

To achieve the above mentioned goal, we have to deal with all stages of the process consisting of:

1. Acquisition of point cloud.
2. Segmentation in 3D.
3. Features extraction.
4. Recognition (classification).

Nikorovic M., Gargalik R. and Tomori Z..
Interactive System for Objects Recognition using 3D Camera, Point Cloud Segmentation and Augmented Reality User Interface.

5. Visualization and interactive control via natural user interface.

The critical part of our research is the segmentation as a process of grouping image pixels into a set of segments hopefully representing our objects of interest. Subsequent object recognition classifies the objects into some classes using pre-specified rules in the supervised learning stage. Natural User Interface is necessary for communication between the human and the machine.

The last stage is visualization which shows projected objects' labels on the ground-plane shown on Figure 1.

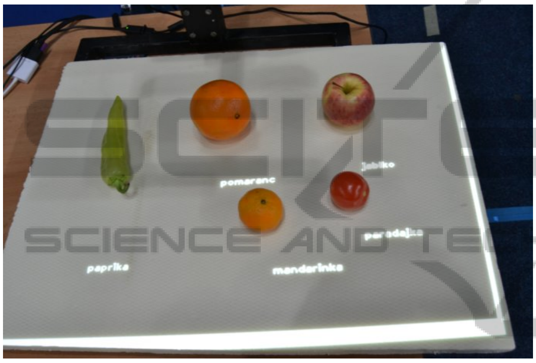

Figure 1: Objects' labels in Slovak language are shown on the ground-plane. We can see that although the dominant descriptor is color, due to other descriptors mentioned later it is able to distinguish orange from mandarin having the similar color.

5 STATE OF THE ART

The traditional RGB image segmentation was one of the most intensively studied part of computer vision. However, the importance of point cloud segmentation methods increases along with appearance of depth-sensing cameras.

(Sedlacek and Zara, 2009) and (Golovinskiy and Funkhouser, 2009) solve simplified point cloud segmentation task. The goal is to divide point cloud into the object of interest and the background. The authors developed an interactive application in which the user have to specify parts of point cloud belonging either to the object of interest or to the background. Remaining points are classified by the algorithm based on the energy minimization function using the weighted graph constructed from the point cloud. The user specified points influence weights in the graph and the algorithm searches for the minimal cut (Boykov and Kolmogorov, 2004). The resulting graph is divided into source part (object of interest) and sink part (background).

(Dal Mutto et al., 2010) described segmentation as grouping the data with similar attributes using machine learning principles. Six dimensional vectors, consisting of point location and point color in uniform color space, are used as the input of the k-means method (Arthur and Vassilvitskii, 2007) to find segments in a point cloud. The result of segmentation contains also a lot of noisy segments and therefore some post-processing method is used.

In contrast (Rabbani et al., 2006), (Wang and Shan, 2009), (Vosselman, 2013), (Castillo et al., 2013), (Dupuis et al., 2014), (Zhan et al., 2009) use seeded region-growing method to segment the point cloud. They choose the starting seeds and let segments grow respecting the homogenity criterion. Their approaches differ in the way how to select the starting seeds and which attributes are used in a homogenity criterion (e.g. close distance, similar color, similar surface normal, similar curvature or their combination between 2 points in a neighborhood). Our work is based on this principle too.

6 METHODOLOGY

Our concept of point cloud grouping into segments and subsequent recognition of the segments should respect specific conditions mentioned in the Outline of Objectives and should use low-cost hardware and open source software.

6.1 Hardware

Projection-based augmented reality (Mine et al., 2012) can project images onto the real surface using one or several projectors.

For the prototyping purposes and for experiments with projector-based augmented reality we constructed the setup shown on Figure 2. Projector P and 3D camera C share the same mount attached to the massive stand. A calibration matrix compensates the translation and possible rotation between them. The objects of interest are located on a solid planar plate, which is placed inside the field of view of the camera.

We exploited projector BENQ MX613ST and depth-sensing camera Microsoft Kinect for Xbox 360 (Windows v1). We used OpenNI library to acquire RGB and depth images. The library offers to transform these images into point cloud.

For experiments with ”Interactive statue” (Figure 3 left) we constructed its minimized version (Figure 3 right) using the identical hardware (Creative Senz3D

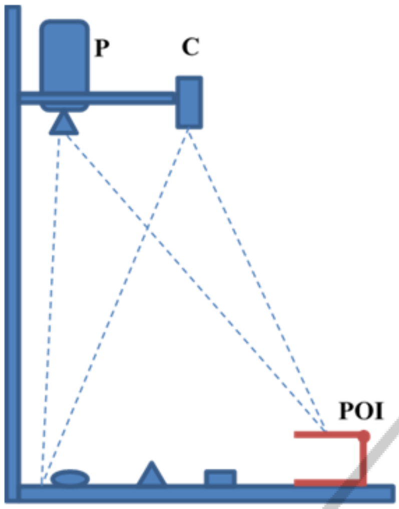

Figure 2: Setup for Projection-Based Augmented Reality Experiments. P – Projector, C – 3D Camera, POI – Plane of Interaction.

camera supplied by Intel along with PC SDK software, servomotors Hitec HS-645MG controlled by Phidgets servo motor controller).

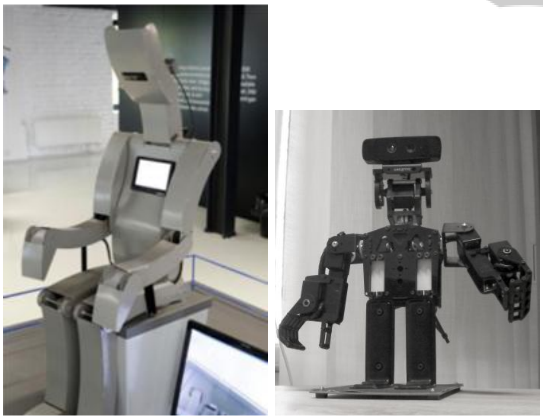

Figure 3: Small robot is used as a prototype of the bigger one located in science center.

6.2 Point Cloud Segmentation

Finding segments correlated with the real objects is a non-trivial process described in the following subsections.

6.2.1 Point Cloud Reduction

We have to process the large amounts of data (point cloud) and therefore it is important to reduce them. VoxelGrid filter divides 3D space containing the point cloud into small voxels (3D boxes in space) and every voxel is approximated by its centroid in a new point cloud. The new point cloud is the reduced version of the original one, points are uniformly distributed and additive noise is suppressed.

6.2.2 Nearest Neighbors Search

Local neighborhood of a single point consists of all surrounding points that are closer than a given small radius. Two approaches can be used to obtain it:

1. select k-nearest neighbors (kNN),
2. select fixed distance nearest neighbors (FDNN).

We have uniformly distributed points, therefore FDNN is the better choice (in contrast to (Rabbani et al., 2006)).

Clustering method is the effective implementation of the nearest neighbors search. The method uses VoxelGrid to group the points into clusters omitting far points as neighbors (it is sufficient to check only the surrounding voxels to find the local neighborhood of the point).

6.2.3 Surface Normal Estimation

We can estimate the surface normal in a point by its local neighborhood. Paper (Klasing et al., 2009) compare surface normal estimation methods. The most robust one is PlanePCA based on Principal component analysis (PCA), which is the statistical procedure to find perpendicular vectors of the biggest variance of data. PCA will return 3 eigenvectors if data is 3 dimensional. Two of them having the greatest eigenvalue are direction vectors of estimated plane and the last one is plane (surface) normal.

Accuracy of the surface normal estimation depends on the size of its local neighborhood. Too few points means that the surface normal estimation will be inaccurate (under-fitting). The opposite state is over-fitting. The compromise between under-fitting and over-fitting was experimentally tested in (Rabbani et al., 2006). Good estimation is approx. 50 points in a local neighborhood.

Automatic selection of the neighborhood radius (our approach) consists of:

1. Select *k* points by the point cloud randomly uniformly.
2. Create a new distance list.
3. For each point find 50th nearest neighbor and add its distance to the list.
4. Return upper quartile from the list.

6.2.4 Region-growing

We try to find independent segments in point cloud by a seeded region-growing algorithm (in graph theory it is also called breadth-first search). Growing is the process of adding actual seed's neighbors with similar attributes (using homogeneity criterion) to the same segment.

Smoothness constraint by (Rabbani et al., 2006) modifies homogeneity criterion to allow region-growing algorithm to find smooth connected objects. The constraint is based on the small difference between surface normals of adjacent points. A new seed point is selected according to the lowest surface curvature of the non-visited point in the point cloud. We approximate surface curvature in the point by mean square distance (1, 2) between the points in the neighborhood and a plane created by the point's location and surface normal.

$$distance(\rho,P) = \frac{Ax+By+Cz+D}{\sqrt{A^2+B^2+C^2}}, \quad (1)$$

where $P=(x,y,z)$ is point, ρ is plane and (A,B,C,D) are ρ's general plane equation's parameters.

$$\text{mean square distance} = \frac{1}{n}\sum_{i=1}^{n} distance(\rho,P_i)^2 \quad (2)$$

6.2.5 Plane Detection

RANSAC (RAndom SAmple Consensus) is iterative algorithm to find the best model in data. It is frequently used to detect planes but in large point clouds it requires a lots of iterations to work properly.

We can detect all planes in the point cloud much quicker by using region-growing algorithm. If we add plane distance constraint to the first seed point into homogeneity criterion then it will able to detect all planes in point cloud.

6.2.6 Ground-plane Removal

Ground-plane is usually the biggest plane of point clouds acquired by a depth-sensing camera and therefore we simply remove the biggest plane from the point cloud (similar idea was mentioned in (Dupuis et al., 2014).

The first seed point can contain additive noise and therefore it is necessary to refine the plane. Averaging surface normals of the segment is one possibility how to create plane more precise and to add next points lying on the plane.

6.2.7 Segment Detection and Noise Removal

After removing the ground-plane we can find segments in point cloud by region-growing algorithm as defined above.

The result of segmentation is a list of segments and its list of the points. A lot of segments contain only a few points (less than a threshold) and we should remove them as a noise. The selected threshold is the average point count of in the segments. Result is shown on Figure 4.

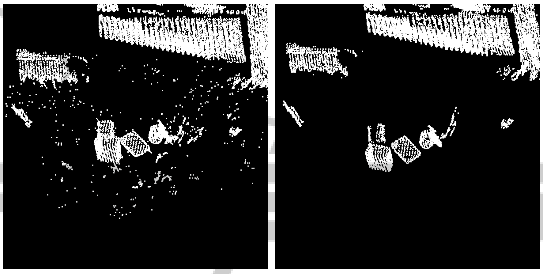

Figure 4: Removing noise segments using thresholding.

Removed noise segments left holes in the segments created by inaccurate surface normal estimation. We proposed method to fill holes in segments inspired by median filter:

1. Find all points removed by thresholding only.
2. For each removed point find the most frequent segment in local neighborhood and reassign point into this segment.
3. Repeat steps 1-2 until no change occurs.

Result is shown on Figure 5.

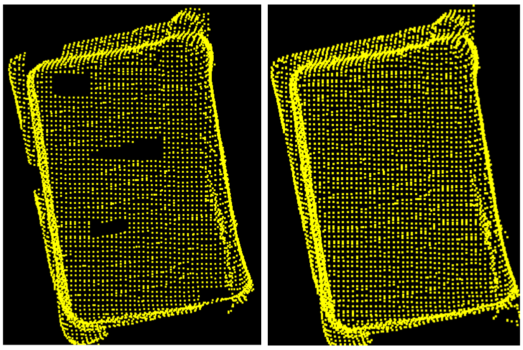

Figure 5: Filling holes in segments by reassigning removed points.

6.3 Object Recognition

The result of segmentation is a list of the segments which we consider as a list of the objects. Object recognition is based on the proper choice of descriptor and classifier ("No free lunch" theorem). For our purposes we use k-nearest neighbors classifier and 100 dimensional vector described in further subsection.

6.3.1 Descriptor Extraction

Successful object recognition requires a powerful descriptor covering color and geometric attributes of the object. Traditional features like average color, roundness or size do not exploit full power of three-dimensional information available in the object's point cloud. Moreover, the descriptor has to be robust, because Microsoft Kinect for Windows v1 is not very accurate (was designed for other purposes).

(As'ari et al., 2013) studied the problems of descriptors for object recognition based on depth-sensing devices. We proposed a 100 dimensional vector based on their idea. The first 50 numbers contain shape histogram (geometric properties) and the others contain color histogram (color properties).

Shape histogram is formed by counting the object's points' distances to the object's centroid. At first, we have to determine the maximum recognition distance. We uniformly divide the distance into 50 intervals. If an acquired distance (from a point to the centroid) is within the pre-defined interval then the interval counter will be incremented (see Figure 6). Then the resulting vector is normalized to unit length.

Color histogram is formed by counting the object's points' hue similarly as in the shape histogram. At first, we transform points' colors from the RGB color space into the HSL color space. Subsequently we remove under-lighted (first quartile) and over-lighted (fourth quartile) points to minimize the sensor distortions. The other colors (second and third quartile) we take into the histogram and normalize to the unit length too.

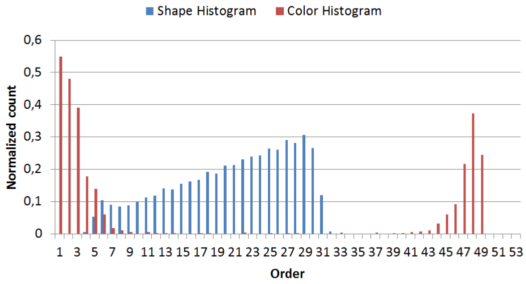

Figure 6: The shape and color histogram.

6.3.2 Learning Process

During supervised learning we assume only one segmented object lying on the ground-plane. The objects not lying on the ground-plane are identified as background. We can choose the object closest to the center of the ground-plane as the object of interest (remaining objects are background). The user should identify the selected object of interest and label it by using virtual keyboard menu options as described later.

6.4 Natural User Interface

It is difficult to control our system by mouse and keyboard (see specifics in the initial chapter). This situation can be solved by augmented reality "virtual keyboard" system consisting of projector and camera. A part of ground-plane is defined as Plane of Interaction (POI) shown on Figure 2), which is able to interact with the user as can be seen on Figure 7.

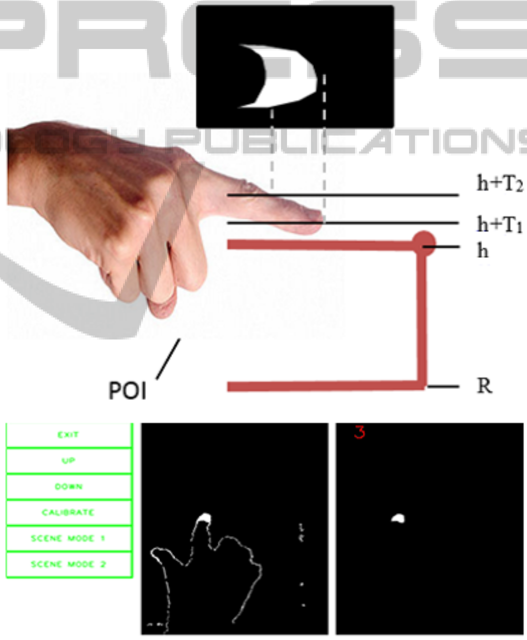

Figure 7: Virtual keyboard projected onto the plane of interaction. Fingertip detection principle (top). In-memory image of menu buttons (left), projected binary image of detected fingertip with noise (middle) and detected fingertip after noise removal (right).

Two threshold levels $T1$ and $T2$ represent the range of sensitive distances from POI. 3D points having z-coordinate (depth) from the interval $< h + T1, h + T2 >$ create a binary image representing a rough approximation of the fingertip. To find out which button is clicked, we first project point cloud of the fingertip approximation into the ground-plane. The result of this step is binary image (middle on Figure 7).

To find out which virtual button has been pushed, we count projected points in each virtual button rectangle from the point cloud and we select the one, which has the maximum number of projected points in its rectangle. If the button is selected during few frames, then we declare the button as clicked.

It should be noted that the orientation of the hand is not critical to this approach, which is clearly an advantage.

The precise projection into the ground-plane requires to calibrate projector with camera by defining ground plane in projector's coordinate system. For simple tasks (like projection of labels into the neighborhood of objects) a 3-point calibration is sufficient. The more complicated tasks will require more sophisticated geometric transformation exploiting a grid of calibration points.

7 EXPECTED OUTCOME

Currently, we experiment with algorithms using prototype hardware. We expect the successful results of experiments with objects recognition system would increase the "intelligence" (and attractiveness) of exhibits in science center (technological museum) in our city.

ACKNOWLEDGEMENT

This work has supported by Slovak research grant agency APVV (grant 0526-11).

REFERENCES

Arthur, D. and Vassilvitskii, S. (2007). k-means++: The advantages of careful seeding. In *SODA '07 Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*.

As'ari, M., Sheikh, U., and Supriyanto, E. (2013). 3d shape descriptor for object recognition based on kinect-like depth image. In *Image and Vision Computing*. ScienceDirect.

Boykov, Y. and Kolmogorov, V. (2004). An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. In *Pattern Analysis and Machine Intelligence, IEEE Transactions*.

Castillo, E., Liang, J., and Zhao, H. (2013). Point cloud segmentation and denoising via constrained nonlinear least squares normal estimates. In *Innovations for Shape Analysis*. Springer Berlin Heidelberg.

Dal Mutto, C., Zanuttigh, P., and Cortelazzo, G. M. (2010). Scene segmentation by color and depth information and its applications.

Dupuis, J., Paulus, S., Behmann, J., Plumer, L., and Kuhlmann, H. (2014). A multi-resolution approach for an automated fusion of different low-cost 3d sensors. In *Sensors 2014*.

Golovinskiy, A. and Funkhouser, T. (2009). Min-cut based segmentation of point clouds. In *IEEE Workshop on Search in 3D and Video (S3DV) at ICCV*.

Klasing, K., Althoff, D., Wollherr, D., and Buss, M. (2009). Comparison of surface normal estimation methods for range sensing applications. In *Robotics and Automation*. IEEE.

Mine, M., Rose, D., Yang, B., Vanbaar, J., and Grundhofer, A. (2012). Projection-based augmented reality in disney theme parks. In *Computer 45, 7*, pages 32–40.

Rabbani, T., van den Heuvel, F. A., and Vosselman, G. (2006). Segmentation of point clouds using smoothness constraint. In *Robotics and Automation*. ISPRS Commission V Symposium 'Image Engineering and Vision Metrology'.

Sedlacek, D. and Zara, J. (2009). Graph cut based point-cloud segmentation for polygonal reconstruction. In *Advances in Visual Computing*.

Tomori, Z., Vanko, P., and Vaitovic, B. (2015). Using of low-cost 3d cameras to control interactive exhibits in science center. In *Sincak, P., Hartono, P., Vircikova, M., Vascak, J., and Jaksa, R. (Eds.): 'Emergent Trends in Robotics and Intelligent Systems: Where is the role of intelligent technologies in the next generation of robots?* Springer.

Vosselman, G. (2013). Point cloud segmentation for urban scene classification. In *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*. ISPRS2013-SSG.

Wang, J. and Shan, J. (2009). Segmentation of lidar point clouds for building extraction. ASPRS 2009 Annual Conference.

Zhan, Q., Liang, Y., and Xiao, Y. (2009). Color-based segmentation of point clouds. In *Laser scanning 2009*. IAPRS.