Assignment 4: Search and rescue

ETH Zurich

Due: Monday 16.12.2013 at 16:00

"Help! Help!" Your neighbor's house is on fire. You call the fire department, but you want to help. You want to deploy your robot to locate the victims in your neighbor's house. Can you ensure that your robot can locate the victims in the house?

1 Object recognition

1.1 Background

Object recognition is the process of identifying known objects in an environment. Various object recognition algorithms exist, including appearance-based algorithms and feature-based algorithms. Appearance-based algorithms use example images of the objects to perform recognition, which suffer from a severe limitation that objects look different under varying conditions. Feature-based algorithms, on the other hand, find objects by matching object features and image features.

Spin image [1] is a 3D feature descriptor, which encodes the distribution of points in a 2D histogram. For a given point with a normal $\vec{n}$, the spin image is computed by spinning a sheet around the normal and collecting the points into a 2D histogram. Given a 2D histogram indexed by α by β, the spin image is generated for a point by looping through all points within the support of the spin image, and for each point, incrementing the bin that corresponds to the coordinate (α, β) of the point. The resulting 2D histogram (or spin image) is then normalized so that it can be compared to the model spin images for recognition.

Several parameters affect the spin image. One parameter is bin size. The bin size is the geometric width of the bins in the spin image and is often set as a multiple of the resolution of the points in order to eliminate the dependence of setting bin size on object scale and resolution. Bin size determines the storage size of the spin image and has an effect on the descriptiveness of the spin images. Another parameter is image width, which is the number of rows or columns in a square spin image. Multiplying the image width by the bin size gives the spin image support distance. The support distance determines the amount of space swept out by a spin image. Last parameter is support angle. The support angle is the maximum angle between the direction of the oriented point basis of a spin image and the surface normal of points that are allowed to contribute to the spin image. A neighboring point with a normal $\mathbf{n'}$ makes a contribution towards the spin image with normal $\mathbf{n}$ if $acos(\mathbf{n'}, \mathbf{n}) < A_{threshold}$. Effectively, the support angle limits the effect of self occlusion and clutter during spin image matching. More detail on spin image generation can be found in chapter 2.3 of `http://www.ri.cmu.edu/pub_files/pub2/johnson_andrew_1997_3/johnson_andrew_1997_3.pdf`.

Object recognition can be performed as follows: first, we compute a set of spin images for the models. Then, for a given scene, we can first segment the scene to extract foreground. The foreground extraction enables us to focus our computational power only on most relevant areas of the scene. We can then compute spin images at some random points of the foreground object and match the extracted spin images to the model's spin images. We then filter the matches to find a plausible transformation from the model to the object. The match can then be verified by transforming all the points and evaluating the overlap or by ensuring that spin images from other points of the foreground object also match well to the model.

1.2 Task

Implement an object recognition algorithm using spin image as the feature descriptor. You can utilize the knowledge about the environment, i.e., the wall and ground are planar and white, to ease the segmentation and extract foreground object. The main algorithm to implement for the recognition is the spin image descriptor. You can use the Point Cloud Library (PCL) [2] for the other algorithms such as segmentation and correspondence computation.

1.2.1 Hints

To get started, you need to first subscribe to `/camera/depth_registered/points` of type `sensor_msgs::PointCloud2` and convert the data to PCL. http://wiki.ros.org/pcl/Tutorials has a tutorial on how to use PCL in ROS. Once the image is converted to PCL data type, you can utilize various functionalities that PCL provides to perform the necessary task. In particular, given that our environment is planar, plain model segmentation http://www.pointclouds.org/documentation/tutorials/planar_segmentation.php#planar-segmentation may be helpful in removing the background planes. Euclidean cluster extraction http://www.pointclouds.org/documentation/tutorials/cluster_extraction.php#cluster-extraction may also be helpful in getting foreground objects. Once the foreground objects are extracted, compute spin images on some random points on the foreground and find the best matching spin images in the model. If the set is geometrically consistent, then you can assign the object the label of the matching model.http://www.pointclouds.org/documentation/tutorials/correspondence_grouping.php#correspondence-grouping has an implementation of a recognition algorithm that is similar to what you will be writing in this assignment. You can start by copying the code and ensuring that you can run the code with your data before introducing foreground extraction and replacing the descriptor.

There are various ways to run the object recognition node. One is to implement it using the publish/subscribe model and try to recognize something at a regular interval. This will be easier to implement as you already know how to publish and subscribe to topics in ROS, but it will be computationally expensive. Another method is to run the recognition algorithm as a service. You can detect when the environment is likely to contain a foreground object and only run the recognition algorithm when the probability is high. You can either use 2D laser data and detect when the difference between the map and the scene is high, or directly use 3D data and check, for instance, if the color distribution of the scene deviates a lot from the expected background color distribution.

2 Grading

You will be given a map as a png file and three destinations to visit before coming back to the starting location. The environment will contain three figurines to be rescued. Each destination may contain zero, one, or two figurines. A figurine may be an adult, a child, or a duck. Your robot must visit all three destinations before returning to the starting position.

The goal is for your robot to identify all the figurines that are in the map. For every figurine the robot identifies, the robot should light up its circle LED light. In addition, when the robot identifies a figurine that is not a family member, it should turn on a red buttons LED. For instance, if the robot detects and recognizes a family member figurine, it should turn on a circle LED. Then when it detects and recognizes another family member figurine, it should turn on another circle LED. In addition, for every figurine your recognition system detects, you need to publish its location to RViz using `visualization_msgs::Marker::CYLINDER`. The center of a cylinder should be the corresponding object's center and the radius and the height should be the width and the height of the object. For more information on Marker type, read http://wiki.ros.org/rviz/Tutorials/Markers:BasicShapes and http://wiki.ros.org/rviz/DisplayTypes/Marker.

The demonstration grading for this assignment will be done in two parts. On Thursday, 12.12.2013, you need to demonstrate that you can recognize objects. We will place three figurines in front of your sensor, one at a time, and your algorithm needs to label each of them. You can either show it in RViz or print the label and the object's dimensions (center, radius, and height) in the terminal. On Monday, 16.12.2013, you will demonstrate that your whole system works. The software grading will be based on Monday's submission.

2.1 In-class demonstration (30 points)

- Thursday, 12.12.2013: Object recognition demonstration (9 points)
  - Every correct label (2 points)
  - Every correct bounding box (1 point)
- Monday, 16.12.2013: Search and rescue (21 points)
  - LED indicators (2 points)
  - Every correct object location (shown in RViz) (2 points)
  - Every correct object label (shown in RViz) (2 points)
  - Timing (7 points)
    * The fastest team (7 points)
    * Every up to 2 minute delay (-1 point)

2.2 Software quality (30 points)

- Choice of abstraction and relations (9 points)
- Correctness of implementation (12 points)
- Extendibility and reusability (6 points)
- Comments and documentation, including "README" (4 points)

References

[1] Johnson, A. and Herber, M. "Using Spin Images for Efficient Object Recognition in Cluttered 3D Scenes". IEEE Transactions on Pattern Analysis and Machine Intelligence, 21(5). 1999.

[2] http://wiki.ros.org/pcl and