Object Relational Anomaly Detection for Scene Analysis

Master Thesis Proposal

Sashidhar Reddy Kanuboddi

October 23, 2018

Contents

1 Motivation

Autonomous mobile robots typically operate in dynamic environments that consist of multiple types of objects. These objects are usually scattered all over the environment. As humans, we have an expectation of where each object should ideally be located, based on our countless observations of similar environments beforehand.

The motivation of this thesis is to enable robots to have a similar understanding of its surroundings. The objective is to implement a learning framework that uses annotated 3D point clouds of different configurations of a particular type of scene, say a living room or an office, and extracts features relevant to the positional and angular bearings of each object with respect to other objects, and finally, is able to identify anomalies when presented with new unseen instances of the scene based on what it has learnt from all the training instances.

2 Modeling Spatial Relations of Objects in a Scene

This thesis will essentially implement a framework from STRANDS, a collaborative European research project [1]. The steps to implement the framework are as follows:

1. Collecting a dataset of point clouds, consisting of different configurations of a particular type of scene.
2. Manually annotating each point cloud with 3D bounding boxes placed over each object.
3. Extracting features from this annotated point cloud dataset, such as pose and angular bearing of each object w.r.t a fixed frame of reference and w.r.t other objects, volume of each object etc.
4. Fitting a model over the extracted features, eg. a Gaussian Mixture Model.
5. Using the fitted model to compute scene similarity score for new scenes. Scenes with a score below a certain threshold would be declared anomalous.

3 Thesis

Each of the above mentioned steps is described in more detail below:

3.1 Dataset of Point Clouds

High resolution point clouds of a scene with different configurations shall be collected. The idea, at the moment, is to capture the point cloud of a bookshelf filled with objects in order to learn about the underlying structure of which objects go on which shelf. By different configurations, it is meant that the bookshelf will not always contain the same objects, but much like a bookshelf in a house, it can contain different objects at different times, and sometimes the same object maybe on different shelves.

Capturing the point clouds is not a trivial task, as we need high resolution point clouds of the entire scene, which means scanning with a 3D sensor (Microsoft Kinect, Asus Xtion) over multiple passes and fusing the point cloud from each pass to form one dense high resolution point cloud. A software called KinectFusion from Microsoft Research[2] claims to accomplish this task in real time with the help of heavy graphics hardware and it is proposed to use this tool in this thesis.

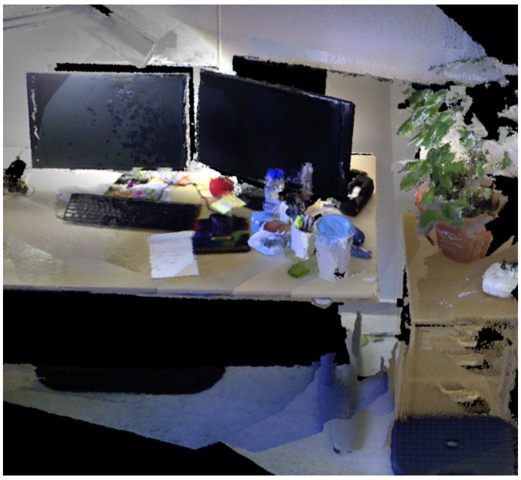


Figure 1: High Resolution Point Cloud of an Office Desk [3]

3.2 Manual Annotation of the Dataset

Once the point clouds have been gathered, each of them will be manually annotated by using a tool developed by KTH Sweden for their research. This tool allows us to graphically put 3D bounding boxes over each object in the scene and export information about the pose (w.r.t a fixed frame of reference) and dimensions of each bounding box to an XML file.

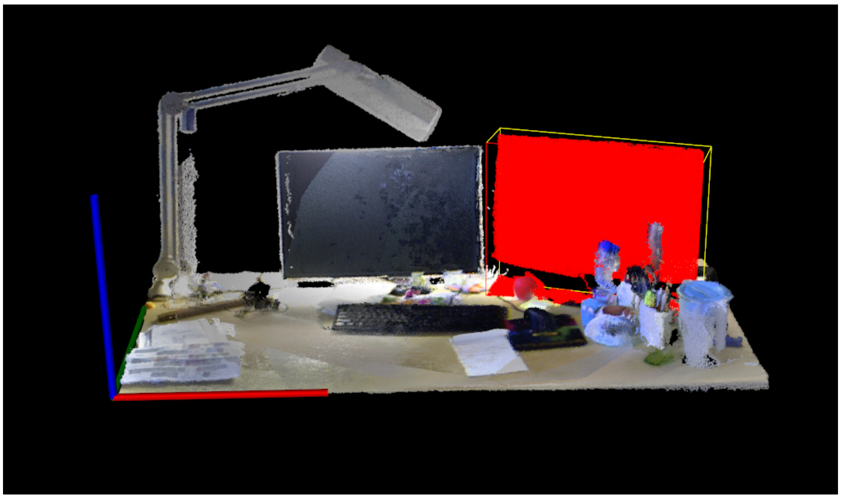


Figure 2: Annotated Point Cloud with Bounding Box over a Monitor [3]

3.3 Extraction of Features

Using the information in the XML files obtained from the point cloud annotation tool, the following features shall be extracted [1]:

Single Object Features (f_{o_i})

- 3D position of the object centroid
- 2D (horizontal) bearing of object centroid from the origin
- Volume of the object

Object Pair Features (f_{o_i,o_j})

- d(C_{o_i}, C_{o_j}), where d is the Euclidean distance and C_{o_i}, C_{o_j} are the centroids of objects o_i and o_j respectively
- Ratio of the object volumes
- d_z(C_{o_i}, C_{o_j}), where d_z if the vertical displacement between the two objects

The possibility of adding more features which can improve the model shall also be explored in this thesis.

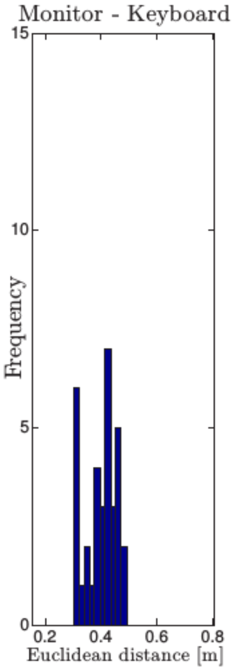


Figure 3: Histogram of one of the features, the Euclidean Distance between two objects [3]

3.4 Fitting a Model

A Gaussian Mixture Model (GMM) shall be used to model the aforementioned features.

Object class Modeling To model the object class category, we fit a GMM over each set of single object features collected from all the scenes:

$$GMM(f_{o_i}, \mu^z_{o_i}, \Sigma^z_{o_i}) = \sum_{z=1}^{n_c} \pi_z \frac{1}{K} exp(-\frac{1}{2}\zeta^z_{o_i})$$

where:

$$\zeta^z_{o_i} = (f_{o_i} - \mu^z_{o_i})^T \Sigma^{z-1}_{o_i} (f_{o_i} - \mu^z_{o_i}))$$

$$K = \sqrt{(2\pi)^{dim}|\Sigma^z_{o_i}|} \ , \ \pi_z \geq 0 \ , \ \sum_{z=1}^{d} \pi_z = 1$$

n_c is the number of mixtures, π_z is the weight of the z^{th} mixture, $\mu^z_{o_i}$ is the mean of the normal distribution, $\Sigma^z_{o_i}$ is the covariance matrix and *dim* is the dimensionality of the feature space.

Learning object pair relationships To learn these relationships, we fit a GMM again over each set of object pair features collected from all the scenes.

3.5 Computation of Scene Similarity Score

Once the model is trained on multiple instances of a scene type, a new instance is given to the program and the question now becomes: how similar is this new instance to our set of trained scenes? Is there something wrong with this new instance?

For computing the scene similarity score, we extract the feature vectors from the new instance and use the fitted model to compute the probability of these features belonging to the same distribution and hence, the same type of scene.

Ideally, for implementation on a real robot, the feature extraction from the new instance step should be performed automatically with the help of a perception software that is able to place 3D bounding boxes around objects, but that is outside the focus of this thesis, hence the new instance shall have a manually annotated point cloud as well.

Using the learned models, the scene similarity score is predicted as follows:

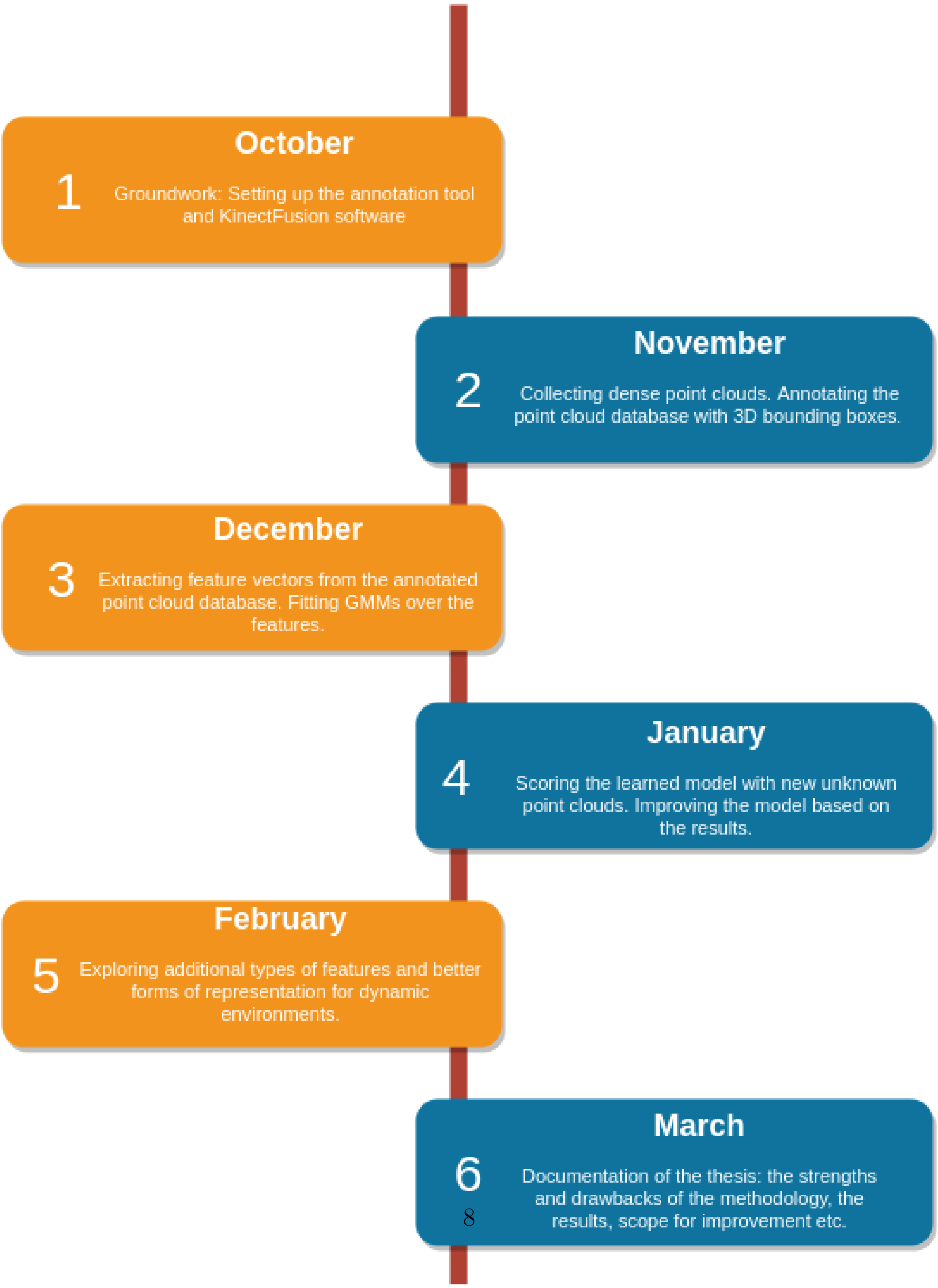
$$sim(s_u, S_t) = \sum_{o_i, o_j \in s_u} P(f_{o_i,o_j}|S_t)P(o_i, o_j) + \sum_{o_i \in s_u} P(f_{o_i}|S_t)P(o_i)$$

where $sim(s_u, S_t)$ is the scene similarity score of a new unknown instance s_u with respect to the modeled scene class type t, S_t and f_{o_i}, f_{o_i,o_j} are feature vectors from the new instance.

A threshold shall be fixed empirically after running the algorithm through some known anomalous scenes and this threshold shall then be used to detect future anomalous scenes.

Furthermore, to identify the source of the anomaly in the scene, having thresholds for individual terms in the summation shall also be explored.

4 Timeline



October

1 Groundwork: Setting up the annotation tool and KinectFusion software

November

2 Collecting dense point clouds. Annotating the point cloud database with 3D bounding boxes.

December

3 Extracting feature vectors from the annotated point cloud database. Fitting GMMs over the features.

January

4 Scoring the learned model with new unknown point clouds. Improving the model based on the results.

February

5 Exploring additional types of features and better forms of representation for dynamic environments.

March

6 Documentation of the thesis: the strengths and drawbacks of the methodology, the results, scope for improvement etc.

5 References

[1] M. Alberti, J. Folkesson and P. Jensfelt. *Relational Approaches for Joint Object Classification and Scene Similarity Measurement in Indoor Environments*. AAAI Spring Symposium, 2014.

[2] R. Newcombe, et al. *KinectFusion: Real-Time Dense Surface Mapping and Tracking.* ISMAR, 2011.

[3] A. Thippur, et al. *KTH-3D-TOTAL: A 3D Datset for Discovering Spatial Structures for Long-Term Autonomous Learning.* ICARCV, 2014.