Object Relational Anomaly Detection for Scene Analysis

Master Thesis Proposal

Sashidhar Reddy Kanuboddi

January 8, 2019

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 which objects should be in the scene and 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, to extract features relevant to each object (such as its dimensions), the positional and angular bearings of each object with respect to other objects; learns from these features; and finally, is able to identify anomalous scenes 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

The idea, at the moment, is to capture the point cloud of a shelf filled with objects in order to learn about the underlying structure of which objects go where on the shelf.

Let S_t be a set of n scenes $\{s_1, s_2, ..., s_n\}$, where each scene is that of a shelf with a few objects. For example, in s_1, the shelf may contain just books, in s_2, the shelf may contain a small mug along with some books and so on. The scene is always that of the same shelf, but just like in the real world, although most of the times the shelf will have the same objects at the same locations, sometimes some additional objects may be present or sometimes, an object can be placed at a different location than its usual one.

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 using a tool developed by the STRANDS project 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.

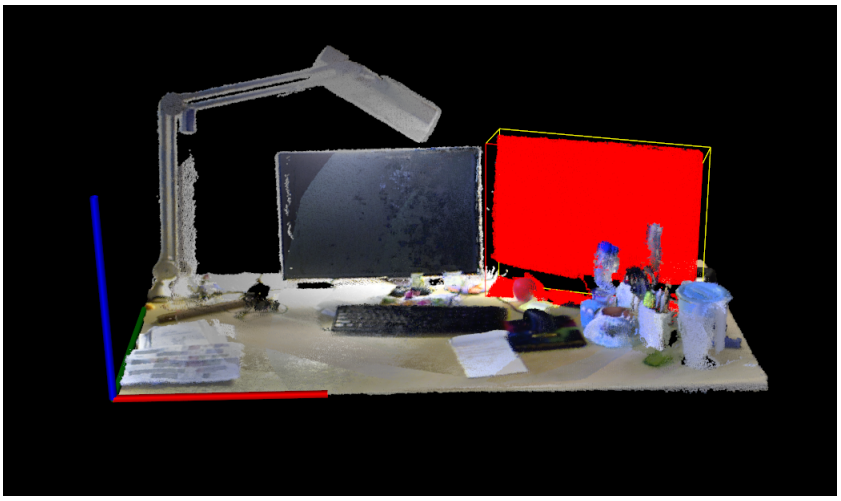

Figure 2: Annotated Point Cloud with Bounding Box over a Monitor [3]

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], w.r.t a fixed frame of reference:

Single Object Features (f_{o_i})

- 3D position of the object centroid (1 x 3)
- 2D (horizontal) bearing of object centroid, θ (1 x 1)
- Volume of the object (1 x 1)

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 (1 x 1)
- Ratio of the object volumes (1 x 1)
- $d_z(C_{o_i}, C_{o_j})$, where d_z if the vertical displacement between the two objects (1 x 1)

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

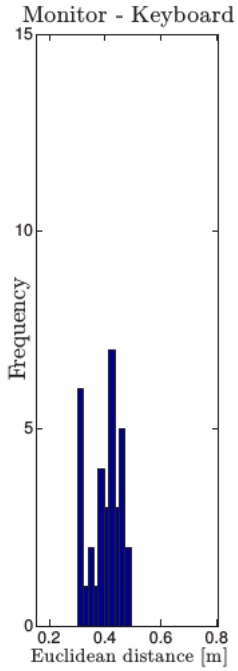

Figure 3: Histogram of one of the features, the Euclidean Distance between two objects [3]

In this way, each of the scenes from the set S_t, will be represented by a set of feature vectors. For example, if s_1 is a scene of a shelf with a book and a mug, then s_1 would be represented by $\{f_{book}, f_{mug}, f_{book,mug}\}$.

3.4 Fitting a Model

Gaussian Mixture Models (GMMs) shall be used to model the aforementioned features.

Modeling Single Object Features: To model the single object features, 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.

Continuing on our example from earlier, we fit a GMM over all feature vectors of the object 'book' extracted from all scenes i.e. GMM_{book} over $\{f_{book,s_1}, f_{book,s_2}, ...f_{book,s_n}\}$ and similarly, for other objects.

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.

In other words, we fit $GMM_{book,mug}$ over $\{f_{book,mug,s_1}, f_{book,mug,s_2}, ...f_{book,mug,s_n}\}$ collected from all scenes.

Essentially, now we have condensed the scene type S_t of that of a shelf to a set of GMMs: $\{GMM_{book}, GMM_{mug}, GMM_{book,mug}....\}$.

3.5 Computation of Scene Similarity Score

Now that we have modeled our scene with a set of GMMs, we present the program with a new unknown scene s_u of a shelf with a few objects and the question becomes: how similar is this new unknown scene to our set of training scenes? Is there something off in this new scene?

For computing the scene similarity score, we extract the feature vectors from the new scene and use the fitted GMMs to compute the relative likelihood of these features belonging to the same distributions and hence, the same type of scene.

Ideally, for implementation on a real robot, the feature extraction from the new scene 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 scene shall have a manually annotated point cloud as well.

Using the learned GMMs, the scene similarity score is predicted as a weighted sum 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 the new unknown scene s_u with respect to the modeled scene class type S_t and f_{o_i}, f_{o_i,o_j} are feature vectors from the new scene.

$P(f_{o_i,o_j}; S_t)$ and $P(f_{o_i}; S_t)$ are probability densities from the learned GMMs. $P(o_i, o_j)$ is computed from the frequency of co-occurence of objects i and j in all the scenes as:

$$P(o_i, o_j) = \frac{N_{o_i,o_j}}{N_{tot}}$$

where N_{o_i,o_j} is the number of training scenes where both o_i and o_j are present and N_{total} is the total number of training scenes. Similarly, $P(o_i)$ is computed from the frequency of occurence of object o_i in all training scenes.

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.

$f_{plate,s_1}, f_{plate,s_2}, ..., f_{plate,s_8}$

$GMM_{plate,dinner\_table}$

$GMM_{cup,dinner\_table}, GMM_{spoon,dinner\_table}$

$f_{plate,cup,s_1}, f_{plate,cup,s_2}, ..., f_{plate,cup,s_8}$

$GMM_{plate,cup,dinner\_table}$

$GMM_{cup,spoon,dinner\_table}, GMM_{spoon,plate,dinner\_table}$

$GMM_{plate,dinner\_table}, GMM_{cup,dinner\_table}, GMM_{spoon,dinner\_table}$

$GMM_{plate,cup,dinner\_table}, GMM_{cup,spoon,dinner\_table}, GMM_{spoon,plate,dinner\_table}$

n_c = number of mixtures

π_z = weight of the z^{th} mixture

$\Sigma^z_{o_i}$ = covariance matrix

dim = dimensionality of the feature space

At $x = \mu$

$$log(f(x|\mu, \sigma^2)) = log(\frac{1}{\sqrt{(2\pi\sigma^2)}})$$

$$sim\_score(plate, s_u, dinner\_table) = GMM_{plate,dinner\_table}(f_{plate,s_u})$$

$$sim\_score(cup\_plate, s_u, dinner\_table) = GMM_{cup\_plate,dinner\_table}(f_{cup\_plate,s_u})$$

	plate_x	plate_y	plate_z	plate_length	plate_width	plate_height
table_1	0.3235	0.2085	0.021	0.215	0.215	0.042
table_2	0.3245	0.2265	0.021	0.215	0.215	0.042
table_3	0.3230	0.1745	0.021	0.214	0.215	0.042
table_4	0.3210	0.1745	0.021	0.214	0.215	0.042
table_5	0.3195	0.1715	0.021	0.215	0.215	0.042
table_6	0.3335	0.1975	0.021	0.215	0.215	0.042
table_7	0.3155	0.2375	0.021	0.215	0.215	0.042
table_8	0.3105	0.2365	0.021	0.215	0.215	0.042

	x_diff	y_diff	z_diff	length_ratio	width_ratio	height_ratio
table_1	0.1805	0.0825	0.016	0.372093	0.372093	1.761905
table_2	0.1835	0.0535	0.016	0.372093	0.372093	1.761905
table_3	0.1790	0.1035	0.016	0.373832	0.372093	1.761905
table_4	0.1590	0.1335	0.016	0.373832	0.372093	1.761905
table_5	0.1595	0.1315	0.016	0.372093	0.372093	1.761905
table_6	0.1695	0.0915	0.016	0.372093	0.372093	1.761905
table_7	0.1865	0.0505	0.016	0.372093	0.372093	1.761905
table_8	0.1895	0.0355	0.016	0.372093	0.372093	1.761905

	plate_sim_scores
table_1	30.709087
table_2	30.043966
table_3	30.133565
table_4	30.023998
table_5	29.566471
table_6	28.882619
table_7	29.895145
table_8	29.185859

	cup_plate_sim_scores
table_1	30.228529
table_2	30.322540
table_3	28.462048
table_4	29.377041
table_5	29.520910
table_6	30.225157
table_7	30.345241
table_8	29.740833

4 Timeline


October
Groundwork: Setting up the annotation tool and KinectFusion software
November
Collecting dense point clouds. Annotating the point cloud database with 3D bounding boxes.
December
Extracting feature vectors from the annotated point cloud database. Fitting GMMs over the features.
January
Scoring the learned model with new unknown point clouds. Improving the model based on the results.
February
Exploring additional types of features and better forms of representation for dynamic environments.
March
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.