

Mined Object and Relational Data for Sets of Locations

J. Timothy Balint

Rafael Bidarra

1 Introduction

In order to procedurally generate a given location, the common objects as well as how those objects are placed in relation to one another, must be known before generation. For example, when generating a *bedroom-like* location, it should be expected that there are one or more *beds*, and each are placed in some proximity to a *wall*. However, the types, frequency, and positioning of objects in a location are variable. In the above example, terms such as “*or more*” and “*in some proximity*” are not well defined, and can cause (desired) variations in generated locations. To understand the overall pattern (what we call *motif*) of a location, there have been attempts to learn the object distribution and relationships from examples (either images or 3D environments).

The purpose of this report is to describe the methods that we used to obtain the data-set for *Mined Object and Relational Data for Sets of Locations*¹. We did this not to replicate their work exactly, but to get a baseline set of motifs that could be programmatically altered to fit the authorial requirements of a scene designer. We first describe the two data-sets that we used, before discussing the relationship mining methods. Specifically, we detail when and where we diverged from the baseline methods, mainly from Kermani et al. [1]. Finally, we quickly discuss the results that we obtained from our two data-sets.

2 Data Sets Used

We learn motifs from two types of data-sets: *annotated images* [2] and *annotated 3D environments* [3]. Both data-sets have objects labeled with their name, as well as 3D positional information for a variety of example scenes. For the annotated images, we used the 3D positional information with no additional modifications (see [2] for more details on their generation methods). For the annotated 3D environment, walls were generated as the convex hull of the accompanying obj file attached to each example.

¹<http://doi.org/10.4121/uuid:1fbfd4a0-1b7f-4dec-8097-617fea87cde5>

There are locational differences between the two data-sets, as well as different sets of objects used in the two rooms. Furthermore, using annotated images creates a certain perspective such that the entire room is not captured in a single photograph. While each photograph is treated as a single example, it is not an example of the entire room. Conversely, the annotated environments do contain the entire information attached to the room. For *open concept* rooms that contain two or more rooms conjoined together, we simply concatenate those locations and treat them as a separate type of location (e.g a *Dining Room/Office* location is combined and treated as a different location than a *Dining Room* or an *Office*).

3 Mining Rooms for Motifs

Motif Mining attempts to create a structured understanding, in the form of relationships, between objects. In this context, relationships are loosely defined spatial distributions. Similarly, the set of objects that can be found in a given location create a distribution of their probability of appearance.

We learn two types of relationships from each location: *High Level Relationships* and *Low Level Relationships*. We define a low level relationship as a numeric pairwise relation between two objects, regardless of their location. Statistics between the distance and orientation of two objects make up the low level relation, and location is disregarded to provide a higher base frequency between object pairs. One interesting extension of this work would be to compare the difference between low level relationships among locations in a data-set.

High level relationships are semantic relations such as *Parallel Orientation* or *Vertical Support*. Each of these are general ideas of placement and assist in defining functions between objects. In many cases (such as orientation), low level relationships are used to define the bounds of high level relationships. Additionally, the fact that there is a high level relationship between two objects makes them a candidate to determine low level relationships between them.

To determine the high level relationships, we implemented methods from three sources, primarily consisting of the method of Kermani et al. [1]. We did this due to the public availability of the base data-sets that they used (the SUN RGB-D [2] data-set and NYU V2 [4] data-set). We also implemented part of the higher order relationships found in Savva et al. [5]. This was done in order to capture support relationships from the SUN-CG data-set. Finally, inspiration for determining the closest points for Savva et al. came from Liang et al. [6].

Low level relationships were generated using two unsupervised learning methods: a K-Means clustering method outlined in Kermani et al. and a Gaussian Mixture Model method outlined in Fisher et al. [7]. Each of these two methods have a different search model for determining the correct number of cluster centers (K-Means) or components (Gaussian Mixture Model). To keep each search method from getting stuck in a local minimum, we further augment the search with Beam Search.

3.1 Learning Higher Order Relationships

We learned higher order relationships independently of lower order relationships on both of our target data-sets, mainly using Kermani et al. Their work utilizes percent thresholds for each relationship to determine what are *salient* relations (ones that should be kept) compared to noise. These thresholds can be thought of as tunable parameters, with specific parameterization provided in the original work. We found that when applied to the entire set of locations, these tunable parameters more generally allowed in noisy relations. We therefore changed the parameters to those seen in Table 3.1. It should be noted the only upward change was to the *symmetry* parameter, which was increased. This was done due to noticed noise from locations that contained a high number of examples. For example, the bedroom motif contains 564 rooms. At Kermani et al.’s original definition, a salient symmetry relationship must appear in 2.82 (3) examples². Furthermore, we ensure that any of the relationships that employ numerical data (orientation, symmetry, side-to-side) have a error threshold that is used to determine if the relationship exists or does not. For example, the error threshold for the difference in size between two objects for the symmetric relation is 0.1 unit (10 cm). While not reported in Kermani et al., it is necessary for allowing measurement and processing errors to exist in the data-sets.

Relationship	Threshold (Kermani et al.)	Threshold (ours)
Support	10%	10%
Symmetry	0.5%	1%
Proximity	5%	5%
Orientation (Perpendicular)	5%	5%
Orientation (Parallel)	5%	1%
Side-To-Side	5%	5%

Table 1: The percent threshold reported in Kermani et al. and the thresholds we used to determine saliency in high level relationships.

In addition to changing the tunable parameters, we also performed a number of preprocessing steps in order to reduce noise in the learned relations. We created a threshold for locations that removed any location that did not have enough example scenes. We also performed a renaming in order to create consistency between object labels (e.g *bathroom_vanity* and *bathroomvanity*).

Finally, we make a slight change to the learning method for *proximity* relations. In Kermani et al.’s original definition, Proximity relations are learned from using G-Span [8] on a minimum spanning tree of each example scene, where objects are the nodes and the distances between objects are the edges. While the original definition uses all objects in the scene as nodes, we only use objects whose frequency of occurrence in all example scenes are greater than the percent threshold for *proximity*. This means that we parse out our non-salient objects

²It should be noted that our interpretation is based on each example and not on simply on its co-occurrence

from the scene before building the minimum spanning tree, causing our edges to always be between two salient nodes. This increases the likelihood that we find a relation between two objects.

3.2 Learning Lower Order Relationships

Our low level relations are based on our high level relations. After learning high level relationships, we use the object pairs to determine numerical distance relationships between them. Each data-point is generated from an example location that has both objects in them, as the relative distance and difference in rotation between them. As stated previously, we use two different disjoint methods that produce two data-sets: the k-means method outlined in Kermani et al. and the Gaussian Mixture method described in Fisher et al. As each of these methods requires a seed for the number of clusters or Gaussian, we use beam search. Each method provides a heuristic of improvement, which we also use to guide the search.

4 Results and Discussion from the Data-Sets

From our examination of the SUN-RGBD data-set, we find that there are 48 possible locations in the data-set, and we found salient relations in all of them. This is due to the fact that we did not remove locations that had only a few examples for this data-set (which was only done in the larger SUNCG data-set). Furthermore, we find that our system is able to determine salient relationships for 270 out of the 855 objects in our data. This means that there were many objects that appeared in very few example scenes specifically for locations that had several example scenes. This is an indication of the amount of noise images inherent in the data-set. Even when only requiring an object to appear in 1% of scenes for it to be considered salient, 70% of objects did not meet that threshold and therefore do not add to the overall understanding of a location.

We also examined low level behaviors for object pairs in the SUN-RGBD data-set. From a total of 1559 unique pairwise relationships that could have existed in our data-set, we find that both Kermani et al. and Fisher et al. find around 900 unique relationships between object pairs (941 for Kermani et al. and 938 for Fisher et al.). Pairwise objects that did not discover a low-level relationship either only had one example (which was the majority of the cases) or did not converge using their clustering method. While they found similar unique relationships, Fisher et al. found many more possible relationships than Kermani et al. (13277 vs 1902). A closer inspection shows that there were several area peaks with no spread between them, which means there is a failure somewhere in the search for the correct number of gaussians. This could be due to using Gaussian Mixture Models or the difference in heuristics used by each method. While we did not perform further testing, it would be interesting future work to examine which aspect of Fisher et al. caused this increase.

We further compare our implementation and assumption against what was

reported in Kermani et al. Specifically, Kermani et al. report on the number of objects and relationship that they are able to extract from bedroom scenes. Examining our data-set, we are also able to extract around 30 objects and 140 relationships between them (similar to their results). However, we did not replicate their results exactly, specifically the symmetric relationships that they found between night-stands. This is due, in part, to the change in the percent threshold for the symmetry relationship in Table 3.1. However, in our tests, simply tuning that parameter to what they reported was not sufficient to match their work. In addition to changing the parameters, we also would have had to increase the size bounds of what is considered a *symmetric* object from a 10cm margin of error between the two similar object sizes to about a 50cm summed margin of error (over all three dimensions)³. The tolerance error for similar sizes (or orientation) is not mentioned within Kermani et al. and therefore becomes another tunable parameter in our system.

The SUNCG data-set has a maximum of 72 locations, of which all were used to generate high level relationships in Savva et al. However, for Kermani et al., when we applied the threshold requiring at least 20 rooms per location, the number of locations dropped to 54 locations. Upon further inspection, it appears that the locations that were not considered are all agglomerations of multiple rooms (*open-design concepts*). Furthermore, the total number of objects that were available from the SUNCG data-set were calculated to be 185 types of objects. When no frequency threshold was used (i.e Savva et al.) 156 distinct object types were found to have some high level relationships between them. When using Kermani et al., that value dropped to 44 objects, approximately 20% of the original possible number of types. This is similar to the 30% of salient objects from the total set found in Kermani et al.. Finally, Kermani et al. generated 792 distinct high level pairing for the SUNCG data-set, and 3478 when only examining the support relationships with no threshold value for support (i.e. Savva et al.). This shows that the threshold and preprocessing parameters removes several relationships that were not often seen.

References

- [1] Z. Sadeghipour Kermani, Z. Liao, P. Tan, and H. Zhang. Learning 3d Scene Synthesis from Annotated RGB-D Images. *Computer Graphics Forum*, 35(5):197–206, 2016.
- [2] Shuran Song, Samuel P Lichtenberg, and Jianxiong Xiao. Sun rgb-d: A rgb-d scene understanding benchmark suite. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 567–576, 2015.
- [3] Shuran Song, Fisher Yu, Andy Zeng, Angel X Chang, Manolis Savva, and Thomas Funkhouser. Semantic Scene Completion from a Single Depth Image. *IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

³This is based on the assumption that the default unit in the SUN-RGBD data-set is 1m, which appears to be so upon inspection.

- [4] Nathan Silberman, Derek Hoiem, Pushmeet Kohli, and Rob Fergus. Indoor segmentation and support inference from rgbd images. *Computer Vision–ECCV 2012*, pages 746–760, 2012.
- [5] Manolis Savva, Angel X Chang, and Pat Hanrahan. Semantically-enriched 3d models for common-sense knowledge. In *the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 24–31, 2015.
- [6] Yuan Liang, Song-Hai Zhang, and Ralph Robert Martin. Learning guidelines for automatic indoor scene design. *Multimedia Tools and Applications*, pages 1–21, 2018.
- [7] Matthew Fisher, Daniel Ritchie, Manolis Savva, Thomas Funkhouser, and Pat Hanrahan. Example-based synthesis of 3d object arrangements. *ACM Transactions on Graphics (TOG)*, 31(6):135, 2012.
- [8] Xifeng Yan and Jiawei Han. gspan: Graph-based substructure pattern mining. In *IEEE International Conference on Data Mining*, pages 721–724. IEEE, 2002.