2016 IEEE International Conference on Robotics and Automation (ICRA)
Stockholm, Sweden, May 16-21, 2016

Learning to Generalize 3D Spatial Relationships

Jimmy Li[1] and David Meger[1] and Gregory Dudek[1]

***Abstract*— This paper presents an approach to learn meaningful spatial relationships in an unsupervised fashion from the distribution of 3D object poses in the real world. Our approach begins by extracting an over-complete set of features to describe the relative geometry of two objects. Each relationship type is modeled using a relevance-weighted distance over this feature space. This effectively ignores irrelevant feature dimensions. Our algorithm RANSEM for determining subsets of data that share a relationship as well as the model to describe each relationship is based on robust sample-based clustering. This approach combines the search for consistent groups of data with the extraction of models that precisely capture the geometry of those groups. An iterative refinement scheme has shown to be an effective approach for finding concepts of differing degrees of geometric specificity.**

Our results show that the models learned by our approach correlate strongly with the English labels that have been given by a human annotator to a set of validation data drawn from the NYUv2 real-world Kinect dataset, demonstrating that these concepts can be automatically acquired given sufficient experience. Additionally, the results of our method significantly out-perform K-means, a standard baseline for unsupervised cluster extraction.

I. INTRODUCTION

In this paper we present an approach to learn meaningful spatial relationships that govern the relative geometry of pairs of objects in natural scenes. Commonly referred to as "object contexts", these arise from effects ranging from laws of physics (*e.g.,* gravity causes a pillow to rest on top of a bed) to human customs and culture (*e.g.,* it is polite to leave one's chair tucked squarely under the table). Object context models are an important component in a robot's understanding of its environment. They have been used previously as a tool to regularize the perception of holistic 3D scenes [1] and to ensure that manipulated objects can be placed intelligently [2], among countless other examples.

Humans regularly reason about a wide variety of relationships, for example: *on*, *near*, *facing*, *inside*, *beside*, *aligned-with*, *perpendicular-to* and *covering*. Each relationship type has a precise meaning and humans will tend to agree with one another when describing the relationships that occur in a scene. Learning from data is the most promising approach to create such a rich and robust representation. In particular, the recent introduction of large datasets of realistic scenes and precise object labels, such as the NYUv2 Kinect dataset, include many object relationships of interest. These datasets are not annotated with relationship labels but this paper demonstrates that unsupervised learning is a feasible approach to extract meaningful relationships by finding geometric regularities in the data.

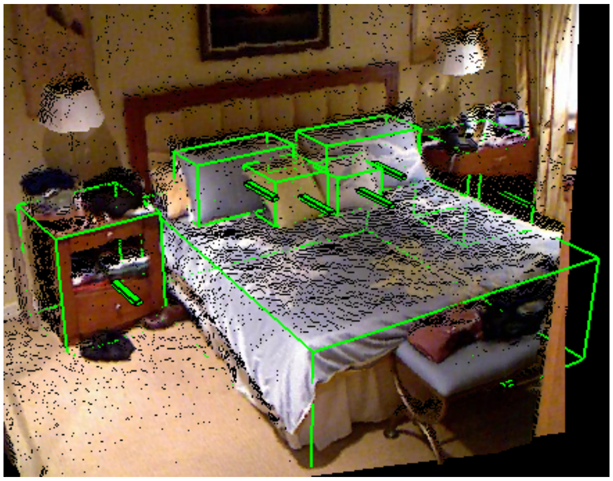


Fig. 1. An annotated scene with examples of night stands to the *left* and *right* of the bed, pillows *on* the bed, and the bed *supporting* pillows.

There have been many models proposed to represent spatial relationships. They commonly represent *relative* spatial information between the objects, as opposed to their absolute position or orientation, typically as a prior or constraint. The stronger a relationship, the more tightly it constrains the position of one object with respect to another. In particular, for many of the relationships that exist in man-made indoor scenes, this constraint is likely to exist in a subset of the spatial dimensions. For example, a *supporting* relation limits one object to lie in a narrow region of space at the surface of the other. This paper demonstrates an approach to automatically select the relevant feature dimensions that describe a relationship, driven by regularities in large datasets collected in indoor spaces.

Our technical approach begins with the extraction of features from pairs of objects. We extract large feature vectors that contain both relevant and irrelevant dimensions, based on simple geometric computations about the pair being considered. Our relationship model contains a relevance weighting of these dimensions, which is effectively able to ignore irrelevant information. Models are parameterized and can potentially represent a diverse range of concepts, only a few of which correspond to meaningful relationships that actually occur in the world. We have designed an efficient unsupervised learning algorithm to determine a small set of active relationship models from training data. We evaluate this method on a set of object pairs that was manually labeled

[1] Mobile Robotics Lab, McGill University, Montreal, Canada {jimmyli,dmeger,dudek}@cim.mcgill.ca.

This work was supported by the Natural Sciences and Engineering Research Council (NSERC) through the NSERC Canadian Field Robotics Network (NCFRN).

978-1-4673-8026-3/16/$31.00 ©2016 IEEE

with the relationship name (*e.g.*, *on* or *left*). Our results demonstrate that our automated approach reliably recovers concepts that are meaningful to humans and we are able to introspect these models to find that the automatically selected feature dimensions are similar to those a human would understand as useful for a relationship type.

II. RELATED WORK

The ability of humans to perceive and reason about object relationships has been studied by numerous researchers (*e.g.,* Beiderman *et al.* [3]). In seminal work in human psychology, Landau and Jackendoff identified a minimal subset of spatial predicates that capture much of how naive humans describe spatial relationships at a qualitative level [4]. Exploiting the the naturalness of spatial relationships in human perception, Skubic *et al.* [5] investigated a human-robot interaction mechanism in which an evidence grid was used to compute pre-defined relationships that could be used for robot control. Even earlier, Zelek [6] developed a minimalist language over spatial relationships on a 2D maps that uses speech recognition to allow spatial predicates to be exploited in mobile robot control and navigation. The resulting SPOTT architecture allows a person to steer a robot in a simple indoor environment by using spatial predicates. In all this work on exploiting spatial relations computationally, the relationships were discrete and computed with respect to a 2D plane and/or the relationships were largely pre-coded. Our work, in contrast focuses on learning these relationships and examining their combinatorial structure.

Object context has been used as a sub-component of countless robotics algorithms for a variety of tasks, such as object recognition [7]. When labeling 3D point clouds with semantic object labels, spatial context is useful as a prior on label transitions, as has been demonstrated by [8]. When searching for an object in an environment, object context can be used as an informed heuristic to indicate promising locations, once an initial set of objects has been observed [9], [10], [11]. Context is also a guide to indicate where an object can be safely placed after manipulation [2].

We are strongly motivated by the potential use of spatial relationships in the task of generating natural language descriptions from sensory data. Recent work has shown that Deep Learning methods can extract features and objects from images [12] and to generate meaningful text resulting from these shared internal representations. In fact, the text descriptions often include reference to spatial context concepts, but these are not grounded in any physical understanding of the 3D space. That is, although the method outputs "the cup is on the table", it cannot direct a robot to pick up that cup. Our method is much more firmly grounded in the 3D scene model that defines the objects in a pair, but does share the component of unsupervised concept creation. This is an interesting approach for further work. The sentence-to-geometry alignment problem has previously been studied by [13], which demonstrates particular words can be aligned to objects in a scene. However, that work did not describe the context governing the layout of objects, or use spatially relevant language - a task we hope to pursue.

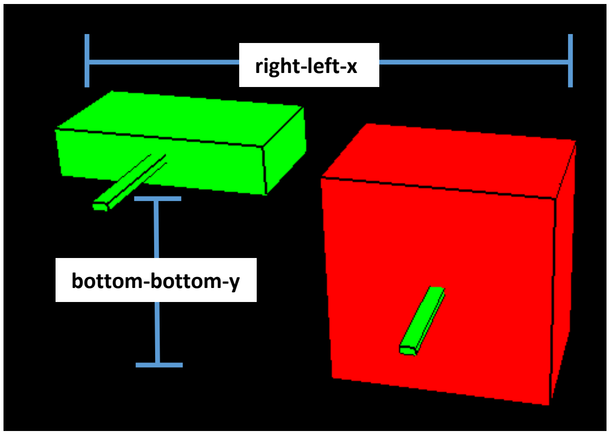


Fig. 2. An example of a follower object B (green) placed to the left of an anchor object A (red). Orientations of bounding volumes are indicated by a green bar extending out of the front center of the volumes. We illustrate two FGDs: right-left-x and bottom-bottom-y.

The learning of object relationship models has been previously studied. In particular, the 3D Geometric Phrases (3DGP) model [1] is the most similar approach to our own. That method learned relationships for specific types of objects, such as the pattern of chairs around a dining table using training data that contained annotated 3D objects. However, while we learn concepts that generalize across pairs made up of many different categories, such as *supported-by*, a 3DGP is highly specialized to one specific set of object labels. This allows us to more easily scale up to a large set of object types, and we show results on a larger set of realistic data.

III. REPRESENTATION

We propose a set of face-centric geometric descriptors (FGDs) that measure the relative positions of points on the boundaries of two objects. An FGD is extracted from two objects represented using oriented rectangular bounding volumes, where the orientation denotes the *front* face of the object. We allow bounding volumes to yaw, but not roll and pitch. This is a typical approach since household objects are usually all aligned with the ground plane, and thus do not roll or pitch with respect to one another. An anchor object A and a follower object B are both represented in the reference frame of A. We then extract the center point of each of the six surfaces of the rectangular bounding volumes, which we call the key points. For example, the key points of A are A-top, A-bottom, A-left, A-right, A-front, A-back. Each point is a 3-element vector consisting of the (x,y,z) position of the point. A feature dimension can be constructed by subtracting the corresponding vector element of any point on A from any point on B. For example, we can construct the dimension right-left-x by subtracting the x vector element of A-right from that of B-left (see Figure 2 for an illustration). We perform only the comparisons that are not uniformly redundant which leaves 12 continuous dimensions.

Many semantic concepts rely heavily on discrete features. The *left* and *right* relationships, for example, do not depend crucially on the distance between objects. Rather, one object simply needs to be on the *left* or *right* side of the other. For each continuous FGD, we introduce a discrete descriptor, which is either 1 or -1 depending on whether the continuous feature is positive or negative. During training, a weighting is chosen for the discrete features so that they are comparable with the continuous features.

The full list of discrete and continuous features we use is shown in Table III. Only a subset of the 24 original feature dimensions are likely to be relevant when representing any given relationship type, so we will continue in the next section by describing a method to determine relevant dimensions from training data.

IV. Method

We seek to recover prominent geometric relationship types by analyzing the distribution of realistic data and recovering models for prevalent concepts. Our process begins by extracting an FGD descriptor, x_i, for each pair of objects, i, that occur in the same scene. We construct an unordered dataset that groups together descriptors of pairs formed by many different categories of objects. If we were given all inliers for a concept, fitting its model would be trivial. Likewise, given a model, computing inliers is simple. However, the unsupervised learning problem requires simultaneously estimating inliers and performing model fitting. The brute-force approach of attempting all subsets of data is computationally infeasible. Therefore, we adopt an approach based on sampling and refinement to robustly approximate useful concepts. The pseudo-code for the algorithm we describe in this section is presented in Algorithm 1.

A. Spatial Relationship Model

Our model for a spatial relationship is a parameterized, relevance-weighted distance function, $D_m(x)$. We employ the Mahalanobis distance parameterized by a model m consisting of a mean μ and a covariance matrix Σ. We can compute the distance of a descriptor x under the model as

$$D_m(x) = \sqrt{(x-\mu)^T \Sigma^{-1} (x-\mu)} \tag{1}$$

In order to capture a prominent relationship type, the model parameters should minimize the distance to a set of inlier samples, while maximizing the size of the sample set, and describing a geometric relationship that is as constrained as possible. This objective is intractable to compute exactly, so we propose an approximation algorithm based loosely on RANSAC, which we will refer to as RANdomized SEmantic Modeling (RANSEM), for extracting semantically meaningful concepts from the training data.

B. RANSEM Algorithm

RANSEM is an algorithm for discovering semantic concepts that exist in a dataset. Concepts are extracted iteratively, and the inliers of each concept are removed sequentially.

Each concept is found using a consensus-search loop similar to RANSAC. We repeatedly draw k samples from the training data, fit a model m_1 using the samples, evaluate the inliers of this model and refine the model by refitting using the full set of inliers. Fitting a model involves computing the mean and covariance of the given samples. Intuitively, the covariance Σ_1 of m_1 can be viewed as a hypothesis of the relevance weights for the feature dimensions. By evaluating inliers using the distance function D_{m_1}, we identify training examples whose distances to the mean μ_1 of m_1 is small under this relevance weighting. We then refine the model by fitting a new model, referred to as m_2, using the inliers. The consensus-search loop continues to discover models until an m_2 model achieves a sufficiently high cluster quality score, which we will define below. At this point we extract m_2, which we regard as a learned semantic concept, and remove the inliers of m_2 from the training data. We can now start a new consensus-search loop to seek another concept from the training data that remains. This is repeated until there is not enough training examples left to sample from, at which point we return the set of discovered concepts.

In our implementation, we discard covariance values during model fitting, resulting in a diagonal Σ. To regularize Σ^{-1}, we find it helpful to compute Φ^{-1} defined as

$$\Phi^{-1} = (\Sigma + \epsilon I)^{-1} \tag{2}$$

and use Φ^{-1} instead of Σ^{-1}. By choosing a positive ϵ, we ensure that elements in $diag(\Phi^{-1})$ do not receive extremely high values when elements of $diag(\Sigma)$ are close to zero.

C. Computing Model Quality

We assess the quality of a model m_2 using

$$Q_{m_2} = inliers(m_2) \cdot separation(m_2) \cdot ||\Phi^{-1}||_1 \tag{3}$$

where $inliers(m_2)$ is the number of inliers of m_2, $separation(m_2)$ is the difference in score between the lowest-scoring inlier and the highest-scoring outlier, and $||\Phi^{-1}||_1$ is the L-1 norm of Φ^{-1}. These three factors reward models that generalize well across the data, that can be clearly identified without confusion, and that have a tight geometric constraint. A tighter geometric constraint is represented by a larger $||\Phi^{-1}||_1$, and is referred to as being more specific.

D. Threshold Refinement

Different semantic concepts vary in their prevalence, separation from surrounding data, and geometric specificity. Thus, each discovered concept may exhibit different cluster quality scores. To facilitate learning multiple semantic concepts, we employ a progressively decreasing thresholding schedule. Let Q^* be the threshold, such that a model m_2 is extracted if $Q_{m_2} > Q^*$. We lower Q^* if the consensus-search loop does not discover any model after the maximum number M of iterations. The subsequent consensus-search loop can then discover less prominent or less specific concepts.

The initial value of Q^*, referred to as Q_{init}, can be determined in a data-driven manner. Our schedule smoothly decreases the threshold starting incrementally above the score of the highest scoring concept we find. An important free parameter is the maximum number of iterations, and this roughly determines the final quality of resulting clusters. If too low a value is set, the threshold decreases quickly and the algorithm is more likely to emit sub-optimal clusters. However, as long as a sufficiently large number of iterations is allowed, we observe stable performance with valuable concepts being learned in all cases.

Algorithm 1 RANSEM

```
procedure RANSEM
    $Q^* \leftarrow Q_{init}$
    Concepts ← empty list
    $T \leftarrow$ training examples
    while $size(T) > k$ do
        $m$, $I \leftarrow$ CONSENSUSSEARCH($Q^*$, $T$)
        if $m = null$ then decrease $Q^*$
        else
            Append $m$ to Concepts
            Remove $I$ from $T$
        end if
    end while
    return Concepts
end procedure
procedure CONSENSUSSEARCH($Q^*$, $T$)
    loop at most $M$ times
        $S \leftarrow$ Sample $k$ examples from $T$
        $m_1 \leftarrow mean(S), cov(S)$
        $I_1 \leftarrow$ inliers of $m_1$ under $D_{m_1}$
        $m_2 \leftarrow mean(I_1), cov(I_1)$
        $I_2 \leftarrow$ inliers of $m_2$ under $D_{m_2}$
        if $Q_{m_2} > Q^*$ then return $m_2$, $I_2$
        end if
    end loop
    return $null, null$
end procedure
```

V. DATA

NYUv2 [14] is a state-of-the-art dataset containing Kinect scans of real 3D homes. We extended this dataset by annotating oriented rectangular bounding volumes of several kinds of household objects, and labeling each pair of co-occurring objects with either *left*, *right*, *on*, *supporting* or *none*. The label *none* is given to object pairs for which the four semantic concepts are not applicable. We chose these four concepts because they are the most prevalent in the dataset, as perceived by the human annotator. In bedroom scenes, we annotated pillow, bed, night stand, dresser, tv, tv stand, desk, and monitor. In kitchen scenes, our labels were stove, dishwasher, and microwave. We annotated all bedroom and kitchen scenes in the dataset.

Table I shows the number of examples of each semantic label in the dataset. *On* and *supporting* relationships are primarily demonstrated by monitors and desks, pillows and beds, tvs and dressers, and tvs and night stands. *Left* and *right* relationships can be primarily seen in the relationships bed-night stand, pillow-pillow, and combinations of fridges, microwaves, dishwashers, and stoves.

TABLE I
SEMANTIC LABEL STATISTICS

	left	right	supporting	on	none	total
occurrences	441	446	285	285	1121	2578

TABLE II
AVERAGE PRECISION

	left	right	supporting	on	overall
RANSEM	.868	.889	.982	.997	.934
RANSEM Euclidean	.729	.846	.100	.114	.447
RANSEM w/o disc	.303	.295	.663	.662	.481
K-means	.634	.706	.976	.993	.827
K-means w/o disc	.661	.608	.125	.144	.385

VI. EVALUATION

Due to their unsupervised nature, our methods have never seen the English labels for concepts, such as *on*. Rather, they are performing concept discovery. In order to evaluate the output of such unsupervised models, the typical practice is to first seek correlation between the discovered models and the labeled concepts in the ground truth. For example, we may find that our system's first model almost always gives a strong score for data items that a human has labeled *on*. We implement this concept translation as a maximal matching between learned models and ground truth labels. It is important that we restrict the number of concepts generated automatically to be equal to the number of ground truth labels, to avoid introducing an unfair bias through the matching process. For RANSEM, we achieve this by keeping only the n highest-scoring discovered concepts, where n is the number of ground truth labels. In our case, there are four ground truth labels: *left, right, on,* and *supporting*. Note that RANSEM may discover additional concepts such as *in front of*, but we expect them to receive lower scores than the four most prominent concepts.

We compare the performance of different variants of RANSEM against a baseline unsupervised concept discovery method, K-means. The same evaluation procedure is applied to all methods. The results of each method are matched with the ground truth labels. The scores that the method assigns to each test data element can then be thought of as a prediction of that pair's relationship type. Precision-recall statistics are generated to evaluate the model's classification performance on each of our four relationship concepts.

VII. RESULTS

A. Quantitative Results

Precision-recall curves for RANSEM and K-means are shown in Figure 3. We evaluate the proposed RANSEM algorithm, along with two variants to assess the impact of using Euclidean distances instead of Mahalanobis distances, and omitting discrete descriptors. We also run two versions of K-means, one using discrete features and the other without. Table II shows the average precision of each method.

The results show that the models learned by RANSEM better match with human labelings of each relationship

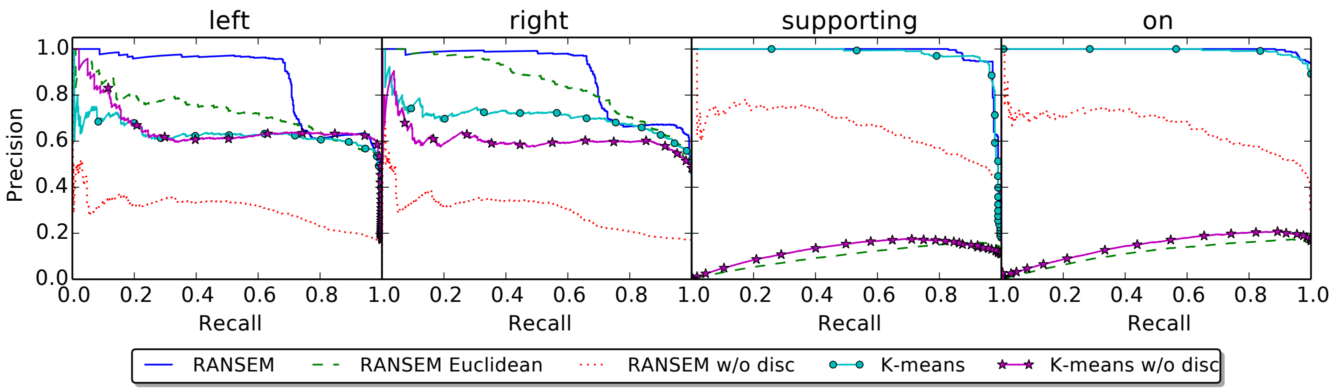


Fig. 3. Precision-recall curves for the semantic concepts *left*, *right*, *supporting*, and *on*. We show the performance of the RANSEM algorithm, RANSEM using Euclidean distances instead of Mahalanobis distances (RANSEM Euclidean), RANSEM without using discrete features (RANSEM w/o disc), K-means, and K-means without using discrete features (K-means w/o disc)

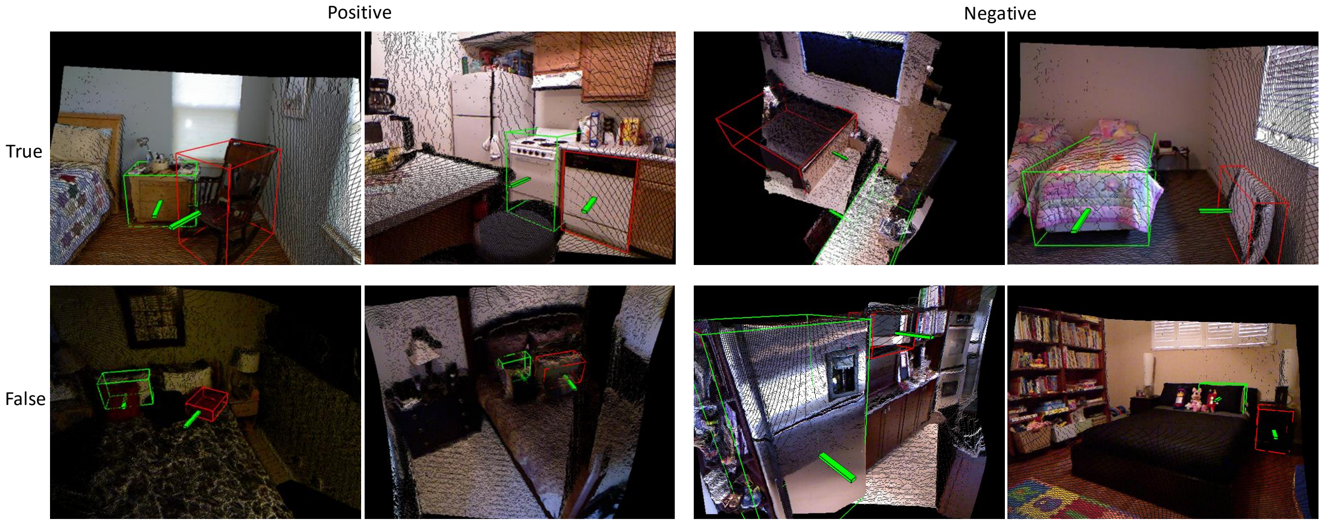


Fig. 4. Examples of true positives, true negatives, false positives and false negatives at precision = 0.85 for the *left* model learned by RANSEM. Red wireframes are the anchor objects and the green wireframes are the follower objects.

than the clusters generated by K-means. Both methods perform quite strongly on the *on* and *supporting* relations, demonstrating that there was sufficient separation in the data for each of those concepts to appear even without feature relevance selection. For *left* and *right*, K-means performs significantly worse than our method since separation is only evident when the correct features are selected. We also observe that RANSEM using Euclidean distances shows worse performance than K-means. This is because when using Euclidean distances, the various semantic concepts are less separable, causing the quality metric to be based solely on the number of inliers. This often leads to less meaningful, over-sized clusters. K-means is more robust to this since it performs expectation maximization over all four clusters simultaneously. The use of discrete features appears to be crucial for both RANSEM and K-means. We observe that without discrete features, RANSEM cannot find a clear separation between inliers and outliers.

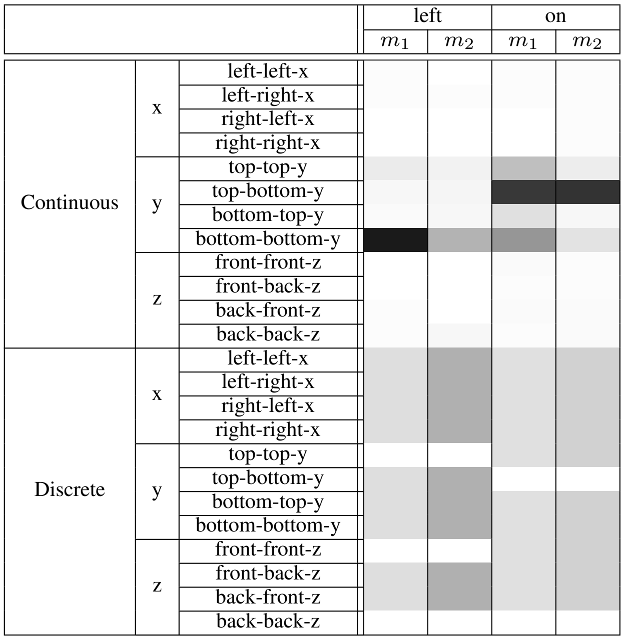
The weighting of dimensions by their relevance is clearly an important aspect of RANSEM. Removing this component causes a drop in performance across the board. Using Euclidean distances leads to a more challenging data distribution, so the algorithm is less able to effectively find meaningful concepts.

B. *Qualitative Results*

Figure 4 illustrates several correct and incorrect classifications of our model's predictions of *left* pairs. When objects fall within the 5-sided infinite volume defined by the left side of the anchor object and its rectangular hull, they are confidently and correctly predicted by this model. Our method produces false negatives on objects that a human would still label *left*, in spite of several of the corners or surfaces extending outside the bounds of this volume. Our analysis indicates that this was a relatively uncommon trend in the data, and that these instances more closely resemble pairs from irrelevant relationships. So, the unsupervised method did not group them into the concept. Our method does correctly disregard objects that face each other and those in front of one another. These fall far from the learned

TABLE III

FEATURE DIMENSION WEIGHTING



			left		on	
			m_1	m_2	m_1	m_2
Continuous	x	left-left-x				
		left-right-x				
		right-left-x				
		right-right-x				
	y	top-top-y				
		top-bottom-y				
		bottom-top-y				
		bottom-bottom-y				
	z	front-front-z				
		front-back-z				
		back-front-z				
		back-back-z				
Discrete	x	left-left-x				
		left-right-x				
		right-left-x				
		right-right-x				
	y	top-top-y				
		top-bottom-y				
		bottom-top-y				
		bottom-bottom-y				
	z	front-front-z				
		front-back-z				
		back-front-z				
		back-back-z				

model and are assigned a large weighted distance.

In addition to improving performance, relevant feature selection is an important aspect in making learned models interpretable by humans. Table III visualizes the weights for the learned models of *left* and *on*. The magnitude of the weights is illustrated by the darkness of the corresponding cell in the table. We show this for both m_1 learned using the sample, and m_2 learned from the inliers. It is interesting to note that the refitting to inliers generally reduces the weights given to continuous dimensions. This can be understood based upon the fact that a small sample of data is likely to accidentally align in non-meaningful subspaces (*e.g.*, be co-linear or co-planar within 3D space). Our approach is able to correct this during its refinement step.

VIII. CONCLUSIONS

In this paper we have described and evaluated a method for learning object-to-object spatial relationships from publicly available Kinect data with 3D object annotations, but *without* annotations of the relationship types. We show that an unsupervised method is able to automatically output concepts which correlate well with a human's description. Our method uses intuitive geometric features extracted from the relative pose of the objects and their shapes. We model relationships with a form that encourages precise geometric descriptions. Model parameters are determined by sampling-based approximate search for prominent concepts occurring in the data. Our results demonstrate that our system's concepts correlate with human labels more strongly than a base-line approach, and validate the importance of using relevance-weighted distance models.

There are many outstanding problems remaining in the area of learning spatial relationships. In complex scenes, objects are often found in piles, stacked in shelves and combined to form complex groupings. The details of these relations are often important for a robot to understand. For example, when cleaning a kitchen, one cannot simply leave objects in open space on the counter, but rather they must be put in their assigned places in the crowded shelves. Due to its unsupervised learning nature, our method is well suited to scaling to such complex scenes. Rather than requiring tedious programming effort, we would simply need datasets whose annotated objects exhibit semantically meaningful concepts, which is becoming increasingly available with the Kinect and online labelers.

REFERENCES

[1] W. Choi, Y. W. Chao, C. Pantofaru, and S. Savarese, "Understanding indoor scenes using 3d geometric phrases," in *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013.

[2] Y. Jiang, M. Lim, C. Zheng, and A. Saxena, "Learning to place new objects in a scene," *The International Journal of Robotics Research*, vol. 31, no. 9, pp. 1021–1043, 2012.

[3] I. Biederman, R. J. Mezzanotte, and J. C. Rabinowitz, "Scene perception: Detecting and judging objects undergoing relational violations," *Cognitive Psychology*, vol. 14, no. 2, pp. 143–177, April 1982.

[4] B. Landau and R. Jackendoff, "Whence and whither in spatial language and spatial cognition?" *Behavioral and brain sciences*, vol. 16, no. 02, pp. 255–265, 1993.

[5] M. Skubic, D. Perzanowski, S. Blisard, A. Schultz, W. Adams, M. Bugajska, and D. Brock, "Spatial language for human-robot dialogs," *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, vol. 34, no. 2, pp. 154–167, May 2004.

[6] J. Zelek, "Human-robot interaction with minimal spanning natural language template for autonomous and tele-operated control," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, vol. 1, September 1997.

[7] V. Hedau, D. Hoiem, and D. Forsyth, "Thinking inside the box: Using appearance models and context based on room geometry," in *Proceedings of the 11th European Conference on Computer Vision: Part VI*, 2010.

[8] A. Anand, H. S. Koppula, T. Joachims, and A. Saxena, "Contextually guided semantic labeling and search for three-dimensional point clouds," *The International Journal of Robotics Research*, vol. 32, no. 1, pp. 19–34, 2013.

[9] J. Vogel and N. de Freitas, "Target-directed attention: Sequential decision-making for gaze planning," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2008.

[10] P. Viswanathan, D. Meger, T. Southey, J. J. Little, and A. Mackworth, "Automated spatial-semantic modeling with applications to place labeling and informed search," in *Proceedings of the Computer and Robot Vision (CRV)*, 2009.

[11] S. Vasudevan, S. Gächter, V. Nguyen, and R. Siegwart, "Cognitive maps for mobile robotsan object based approach," *Robotics and Autonomous Systems*, vol. 55, no. 5, pp. 359 – 371, 2007.

[12] A. Karpathy and L. Fei-Fei, "Deep visual-semantic alignments for generating image descriptions," in *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[13] C. Kong, D. Lin, M. Bansal, R. Urtasun, and S. Fidler, "What are you talking about? text-to-image coreference," in *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014.

[14] P. K. Nathan Silberman, Derek Hoiem and R. Fergus, "Indoor segmentation and support inference from rgbd images," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2012.