

Unsupervised Discovery of Repetitive Objects

Jiwon Shin, Rudolph Triebel, and Roland Siegwart

Autonomous Systems Lab (ASL)

Swiss Federal Institute of Technology, Zurich (ETHZ)

{jiwon.shin, rudolph.triebel, rsiegwart}@mavt.ethz.ch

Abstract—We present a novel approach for unsupervised discovery of repetitive objects from 3D point clouds. Our method assumes that objects are geometrically consistent, and uses multiple occurrences of an object as the evidence for its existence. We segment input range data by superpixel segmentation, extract features for each segment, then find a set of segments that have a matching set using a joint compatibility test. The discovered objects are then verified by the Iterative Closest Point algorithm to remove false matches. The presented method was tested on real data of complex objects. The experiments demonstrate that the proposed approach is capable of finding objects that occur multiple times in a scene and distinguish apart those objects of different types.

I. INTRODUCTION

For a robot that interacts with people, it is essential to semantically analyze its surroundings. In particular, home environments usually contain various objects, which often define the particular location at which they are encountered (e.g. furniture). The ability to detect and distinguish objects autonomously is thus a key for a robots' independence when working in a home environment. For instance, if a robot can determine that a dining room contains a set of chairs, which are multiple occurrences of the same object, and a table, which is different from chairs, then it can use such information to classify a dining room as a place with two types of objects - many chairs and one table. Then, when it encounters an unfamiliar room, it can simply search for the characteristics of the room - many instances of one object type and one instance of a different object type - and the fingerprints of the objects found in the room. When both are verified, the robot can label the room as a dining room. Such an automatic process eliminates the necessity of training a robot with every object it is likely to find in the environment. Instead, we can simply label each type of object a robot finds in the appropriate language of the household, e.g. *chair* or *Sessel*. In this work, we investigate the possibility of unsupervised discovery of objects that occur multiple times, such as chairs in a dining room, from data taken with a 3D laser scanner.

Unsupervised discovery of repetitive objects in a given scene is a challenging task because we do not know a priori the definition of an object, the number of occurrences of a certain object type, nor the number of different object types present in the scene. In addition, the method must be able to distinguish real objects - chairs and couches - from walls, ground, and ceiling as we do not pre-segment them out. The method thus should be able to hypothesize on objects

Fig. 1. An example of a scene observed with a nodding SICK laser scanner. Objects that are discovered through the algorithm are colored, where all points which belong to the same object are assigned to one color. Arrows are drawn between two segments that match.

while handling lots of clutter in the scene. As an indoor robot can easily collect more evidence to support or refute a hypothesis without any harm, it is better for a robot to claim no knowledge than have a false belief when the uncertainty is high. To minimize false discovery, we take a conservative approach and only accept the output when the uncertainty is low.

Figure 1 depicts a typical scene of interest in this paper, which is captured using a nodding SICK laser scanner. The scene contains two working chairs and two arm chairs along with some ceiling light fixtures and a plant. Of these, we are interested in discovering the two types of repeating objects - the working chairs and the arm chairs. If the process is successful, each instance of the object gets its own color, and arrows are drawn between all segments that match.

We propose an approach to discover, without supervision, objects that occur multiple times in a scene. Using 3D point clouds from a laser scanner as input, we first segment the points according to their surface property using superpixel segmentation and extract features for each segment. We use an extended joint compatibility test to discover object models and their matching objects, and verify these objects by the Iterative Closest Point algorithm to remove false matches. Through this work, we demonstrate that repetition can aid the discovery of objects and define object models.

The organization of the paper is as follows. We discuss related work in Section II. Section III explains how the input scene is segmented and how features are extracted from each segment. In Section IV, we discuss the object discovery method and the verification step. Section V presents the experimental results. The paper concludes with Section VI.

II. RELATED WORK

Repetition detection has been well-explored in the field of image analysis. In particular, many authors have investigated methods for detecting regularly repeating patterns [1], [2]. More recently, Loy and Eklundh [3] focused on grouping of features based on symmetry, and Wenzel *et al.* [4] proposed an algorithm that uses symmetry to detect repetitive structures in facade images. They argued that symmetry is a strong

clue to group features together. Likewise, we group together segments upon a discovery of a matching set, but we do not explicitly search for the symmetric plane between the two objects. In this way, our approach is similar to Zeng and van Gool [5], where the authors employ point-wise repetition to improve segmentation results. They use mutual information to determine if two segments of an initially oversegmented image are of the same group. We instead extract features for every segment and compare these features to measure the similarity between two segments.

In terms of 3D, discovery and utilization of repetition has been addressed in computer aided design and other synthetic models [6], [7], [8]. They focused on detection of symmetry or regular patterns in 3D with applications in graphics and image compression. Work of Bokeloh *et al.* [9] is more closely related to this work. The authors proposed an algorithm for detecting structural redundancy by matching symmetric constellations of feature lines. We also search for a collection of elements that repeat as a group, but we do not assume symmetry as the repetition pattern. To our knowledge, no work has dealt with discovery of objects by repetition in laser data.

In unsupervised object detection, several have proposed adaptation of text analysis methods in image analysis. For example, Liu and Chen [10] has proposed a modified probabilistic latent semantic analysis method to detect foreground objects from images. In [11], Endres *et al.* use Latent Dirichlet Allocation to detect object classes from range data without supervision. While this approach can classify objects of multiple classes, they assume that a ground plane and walls are extracted a priori and the objects are spatially disconnected. In our work, we do not make such assumptions. We consider every segment as a potential object part and test them to determine if they belong to an object.

The way we define an object is parts-based. We search for objects using the joint compatibility branch-and-bound algorithm [12]. Shin *et al.* [13] has shown that objects defined by parts can be represented by a grammar and recognized using a joint compatibility test. In our work, we do not perform a separate parts detection, nor require object parts to have physical meanings.

We employ feature-based approaches to recognize objects. Among various feature descriptors for 3D data, spin images have been shown to be successful and popular [14], [15]. Other features of interest for this work are shape distribution [16] and shape factors [17].

III. SEGMENTATION AND FEATURE EXTRACTION

The proposed algorithm is a three-step process. First, we extract segments from the input point cloud and extract features for every segment. We apply a joint compatibility test on these segments to detect objects and then verify them using the Iterative Closest Point algorithm. In this section, we describe the segmentation method and shape descriptors.

A. Range Data Segmentation

The goal of segmentation is to find labels $L(\mathbf{x})$ for all data points \mathbf{x} , where points that are close to each other and

similar in some predefined way, should have the same label. We use the superpixel segmentation method by Felzenszwalb and Huttenlocher [18], originally proposed for 2D images, to group together similar points. This algorithm creates a graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$ of vertices \mathcal{V} and edges \mathcal{E} , where each pixel in a given image corresponds to a vertex and the edges connect adjacent image pixels. Each edge $\mathbf{e} = (v_i, v_j)$ has an associated weight $w(\mathbf{e})$ representing the dissimilarity of the connected vertices v_i and v_j . In the case of an image, this can be, for example, the difference of the pixel intensities. The algorithm starts with a segmentation where each vertex is its own segment. Then, the edges are processed by increasing weights and the two segments C_i and C_j connected by a given edge \mathbf{e} are merged whenever

$$w(\mathbf{e}) \leq \min\left(d(C_i) + \frac{k}{|C_i|}, d(C_j) + \frac{k}{|C_j|}\right),$$

where $d(C)$ is the *internal difference* function defined by the maximal edge weight of all edges in the minimum spanning tree of the segment $C \subseteq \mathcal{V}$, and k is a consistency parameter that influences the granularity of the segmentation: a low value of k requires segments to be more consistent and thus produces more but smaller segments. The internal difference function ensures that two segments are merged only when the difference between the two is smaller than the difference within each segment with some tolerance.

In this work, we define each point \mathbf{x} of a 3D point cloud \mathcal{X} as a vertex and form an edge between two neighboring vertices, where neighbors are determined by a triangular mesh built on the data. We use the dot product $\mathbf{n}_i \cdot \mathbf{n}_j$ as edge weight where \mathbf{n}_i is the surface normal vector computed at point \mathbf{x}_i . Thus, regions with a smooth surface, e.g. a plane or a sphere, are segmented as one region while surfaces with sharp edges, e.g. between two sides of a box, are segmented into two regions. As a modification of the original algorithm, we do not force every point to be in a segment. This is because we cannot calculate the normal for the points with an insufficient number of neighboring points. For these isolated points, no vertices are generated in the graph, and thus no label is assigned. In addition, after termination we remove segments that contain less points than a given minimal value m_{size} . Such small segments are often caused by sensor imperfections or occlusions and do not reveal enough information for the later matching process.

B. Shape Descriptors

As shape descriptors, we use spin images [14], shape distributions [16], and shape factors [17], and weigh them accordingly. For a given point \mathbf{x} with normal vector \mathbf{n} , a *spin image* is defined as a 2D histogram H^s oriented along the line l through \mathbf{x} with direction \mathbf{n} . Each bin of H^s counts the points with a certain distance to l and the plane through \mathbf{x} with normal vector \mathbf{n} . For the spin image descriptor of a segment C , we form vectors \mathbf{h}_i^s of stacked lines of the histograms H_i^s for all points $\mathbf{x}_i \in C$ and compute the average $\bar{\mathbf{h}}^s$ over all \mathbf{h}_i^s .

A *shape distribution* is defined as a histogram of values of a predefined function $f : \mathcal{C}^r \rightarrow \mathbb{R}$, where r is the arity of \mathbf{f} and is usually a value between 1 and 4. In our implementation, we use two binary functions $f_d(\mathbf{x}_i, \mathbf{x}_j)$ and $f_a(\mathbf{x}_i, \mathbf{x}_j)$, namely the Euclidean distance of the points \mathbf{x}_i and \mathbf{x}_j and the scalar product of their normal vectors \mathbf{n}_i and \mathbf{n}_j . The resulting histogram vectors \mathbf{h}^d and \mathbf{h}^a are computed by evaluating f_d and f_a on all pairs of points in a segment \mathcal{C} . To make the feature vectors invariant with respect to the sample density, we normalize the histograms \mathbf{h}^d and \mathbf{h}^a by the total number of bin entries. A normalization with respect to the maximum distance encountered in a segment is not done, as this would result in scale-invariant features, and we consider scale as a feature to be distinguished between objects.

Finally, we compute *shape factors* per segment, i.e. the normalized eigenvalues of the covariance matrix C_i of all points in segment C_i , collected in a vector \mathbf{h}^f . All individual descriptors are used to define a distance metric d_c on segments as

$$d_c(C_i, C_j) = \lambda_1 \Delta \bar{h}^s + \lambda_2 \Delta h^d + \lambda_3 \Delta h^a + \lambda_4 \Delta h^f,$$

where the λ_i are weight factors and Δh is the Euclidean distance between two feature vectors $\mathbf{h}(C_i)$ and $\mathbf{h}(C_j)$.

IV. OBJECT DISCOVERY

The challenge of unsupervised discovery of repetitive objects is that we have neither an a-priori definition of an object, nor the number of occurrences per object type. Without such information, we cannot determine for each segment if the segment is an instance of an object. To overcome this problem, we search for only those objects that occur multiple times in the scene. The multiplicity allows us to reason on the object by comparing it against another instance of the same object. In addition, we only focus on complex objects and define an object as a collection of segments. Discovering objects composed of only one segment requires us to rely entirely on the shape descriptors for matching. The minimum segment constraint allows us to use physical constraints as an additional evidence for an object. Therefore, we consider an object hypothesis valid only when it is composed of at least two segments. To reduce false matches, we verify the hypotheses for objects by finding correspondences between the point clouds of discovered objects.

A. Repetitive Object Discovery

To find repeating objects, we use a joint compatibility test with branch-and-bound [12], a popular solution for data association problems. Data association is a well-known problem in robotics. The joint compatibility test addresses the data association problem by finding test points that not only correspond to the model points individually but also match well as a set. The branch-and-bound aspect enables the algorithm to search efficiently by growing a hypothesis when necessary and terminating one when no appropriate part is found.

In an ordinary data association problem, the model set is predetermined, and the goal is to find the best mapping from

Algorithm 1: JCBB The joint compatibility branch-and-bound test for discovering a pair of repetitive objects.

Data: Segments \mathcal{C} of the scene

Input: Current *model* hypothesis $\mathcal{H}_M \subset \mathcal{C}$ and its matching *test* hypothesis $\mathcal{H}_T \subset \mathcal{C}$

Output: A pair of hypotheses $\mathcal{B}_M \subset \mathcal{C}$ and $\mathcal{B}_T \subset \mathcal{C}$ that yield the highest matching score.

Procedure:

if $|\mathcal{H}_M| \geq |\mathcal{B}_M|$ **and** $d_c(\mathcal{H}_M, \mathcal{H}_T) > d_c(\mathcal{B}_M, \mathcal{B}_T)$ **then**

$\mathcal{B}_M \leftarrow \mathcal{H}_M$;

$\mathcal{B}_T \leftarrow \mathcal{H}_T$;

end

$I \leftarrow |\mathcal{C}|$;

for $i = 1$ **to** I **do**

$C_M^i \leftarrow \text{random_select_from}(\mathcal{C})$;

$C_T^i \leftarrow \text{random_select_from}(\mathcal{C} \setminus \{C_M^i\})$;

if *individual_match*(C_M^i, C_T^i) **and**

relation_match($\mathcal{H}_M \cup C_M^i, \mathcal{H}_T \cup C_T^i$) **then**

 JCBB ($\mathcal{H}_M \cup C_M^i, \mathcal{H}_T \cup C_T^i, \mathcal{C} \setminus \{C_M^i, C_T^i\}$) ;

end

end

the test set to the model set. In our framework, however, we do not have a model. Our goal is to discover a model through the detection of matching pairs of segments. We thus propose a modification to the algorithm that is capable of extracting an object model and its matching test object from the input segments.

The overall algorithm is shown in Algorithm 1. Given segments as input to the algorithm, we search for a set of segments that occur multiple times in the scene. We perform the search in two steps. In the first step, we discover an object model, i.e. a collection of segments, and its matching object. Since the only evidence we have for an object is the presence of a matching object, the process will always return two hypotheses. In the second step, the algorithm searches for the remaining occurrences of the object using as a model, the objects found in the first step.

The first step is as follows: We begin with a randomly selected segment C_M^0 and look for a segment C_T^0 in the scene that match well with it. If we find C_T^0 , then we begin two hypotheses, \mathcal{H}_M and \mathcal{H}_T , one for the *model* set and the other for the *test* set. The distinction of *model* and *test* hypotheses is arbitrary as one hypothesis is only valid with the existence of a matching hypothesis. Therefore, there is no definitive model hypothesis to which a test hypothesis must match. Rather, a pair of segments must be similar enough to support each other's validity.

The hypotheses \mathcal{H}_M and \mathcal{H}_T grow as we select a new segment C_M^1 and search for C_T^1 that is individually compatible to C_M^1 , and its combination with \mathcal{H}_T is jointly compatible to $\mathcal{H}_M \cup C_M^1$. The hypotheses \mathcal{H}_M and \mathcal{H}_T continue to grow until the n -th model segment C_M^n no longer finds a compatible

(a) Without initialization
(b) With initialization

Fig. 2. Correspondences between two point clouds without the spin image initialization 2(a) and with the initialization 2(b). Without the initialization, ICP performs poorly when the two point clouds have a high rotational transformation as shown in 2(a). 2(b) shows that the same two sets match well with the initialization.

test segment C_T^n . The algorithm then starts a new pair of hypotheses with a different seed segment pairs, in search of the best pair of hypotheses \mathcal{B}_M and \mathcal{B}_T . The best pair of hypotheses contains the most number of segments with the smallest distance between the hypotheses. At the end of the process, we label \mathcal{B}_M and \mathcal{B}_T as objects O_1 and O_2 of type \mathcal{D} .

Upon the discovery of an object type \mathcal{D} , we begin the second step. To find the remaining instances of \mathcal{D} in the scene, we apply the algorithm again, but this time, using O_1 and O_2 as the model. Now the goal is to find a set of segments that matches the model best. Each time we find such a hypothesis \mathcal{H}_k , we label it as an object O_k of the type \mathcal{D} . The search for an object of type \mathcal{D} ends when we no longer find a hypothesis that matches either O_1 or O_2 . We repeat this two-step process of finding a pair of hypotheses and detecting other instances of the object until we no longer find a valid hypothesis.

In the presented algorithm, the individual and the joint match score play a crucial role in deciding on a match. We use the shape descriptors as described in Section III-B to evaluate a match. For an individual match, we consider a pair of segments C_M and C_T compatible if

$$d_c(C_M, C_T) < T_i,$$

where T_i is a thresholding value for individual compatibility. For the joint compatibility, in addition to calculating $d_c(\mathcal{H}_M \cup C_M, \mathcal{H}_T \cup C_T)$, we compute the Mahalanobis distance $d_m(\mathcal{V}^{C:\mathcal{H}}_M, \mathcal{V}^{C:\mathcal{H}}_T)$ between the new segment pairs to the segments in their corresponding hypotheses, where $\mathcal{V}^{C:\mathcal{H}}_M$ indicates a vector from the center of the input segment C_M to a segment in the hypothesis \mathcal{H}_M . We require that for all segments in \mathcal{H}_M and \mathcal{H}_T ,

$$d_m(\mathcal{V}^{C:\mathcal{H}}_M, \mathcal{V}^{C:\mathcal{H}}_M) < T_j.$$

The physical constraints enable us to reject segments that are similar in features but are inconsistent with the hypotheses in their arrangement.

B. Match Verification

The goal of the verification step is to minimize falsely discovered objects by confirming that the discovered objects are consistent among themselves. We achieve this by mapping all points of an object O_i to the points that belong to its matching pair O_j by the Iterative Closest Point (ICP) algorithm [19].

ICP, often used in localization, finds the transformation from one point cloud to the other by minimizing the difference between the two sets. Since ICP finds a local minimum, it works well only when the initial correspondence between two point clouds is close to the global minimum. As the objects O_i and O_j can be in any orientation, the initial estimation cannot rely purely on the nearest neighbors in the Euclidean space. We instead estimate the initial transformation by computing features at various randomly selected points in O_i and finding their corresponding points from O_j in the feature space. We use spin images as the features, as presented in [14]. Figure 2 shows the effect of the initialization by the feature-space correspondence. As the figure indicates, without the initialization, the verification step performs poorly when the objects are mirrored.

The initialization is as follows: Given two objects O_i and O_j , we first center them with their respective mean values \bar{O}_i and \bar{O}_j in x- and y-direction, and randomly select a subset of points $(\mathbf{x}_1, \dots, \mathbf{x}_n)$ in O_i . We assume that objects are in their natural vertical position and do not center the points in z-direction. This helps us eliminate wall-ceiling, wall-floor, and ceiling-floor matches. For each point \mathbf{x}_k , we calculate its spin image and search for all points $(\mathbf{y}_1^k, \dots, \mathbf{y}_m^k)$ in O_j , whose spin image is similar to \mathbf{x}_k . These points are then used as the initial correspondence points for ICP. Once the transformation between the two point clouds is found, we count all the points in O_i that have a corresponding point in O_j and vice versa. We consider the two objects O_i and O_j matched if the total number of matched points is greater than 70 percent of the sum of points in O_i and O_j .

V. RESULTS

In this section, we test the algorithm on scans from real world scenes. We took data using a nodding SICK laser with a width of 100 degrees and a height of 90 degrees. Each set was captured at the horizontal resolution of 0.25 degrees and the vertical resolution of 15 degrees a second. The test set was composed of 55 data sets from four different rooms. Overall, the scenes had four types of working chairs and one type of arm chairs along with trash cans, a flip chart, and a plant as background. Objects were placed up to 90 degrees of rotation from each other. Most scenes contained two or three objects of the same type, but some scenes contained two objects of two kinds. Four scenes contained contained no repeating objects. In total, there were 138 instances of objects that the algorithm could discover.

We evaluate the algorithm by the rate of discovery and precision. The discovery rate is the number of objects the algorithm found over the number of objects we expect it to discover. We calculate precision as the number of correctly discovered objects over the number of correctly and incorrectly discovered objects. The rates for object types are computed likewise. For example, if a scene contains three chairs of type A and one of type B, then we define the ground truth as three chairs and one object type. As mentioned earlier, the program does not detect objects of single occurrence.

Fig. 3. Some test scenes with discovered objects in color. Points that belong to the same object have the same color

Fig. 4. A test scene with no repeating object. The algorithm discovers no object.

TABLE I

THE EFFECT OF SEGMENTATION PARAMETERS ON THE OBJECT DISCOVERY RATE

m_{size}	50	75	100	120	150
$k = 6$	56%	48%	51%	45%	40%
$k = 9$	59%	48%	59%	45%	40%
$k = 12$	42%	34%	36%	36%	36%
$k = 15$	47%	47%	36%	36%	36%

Figure 3 and Figure 4 contain some of the results of the presented algorithm. All points that belong to the same object have the same color, and an arrow connects two matching segments. The arrow starts from a *model* segment and points at the corresponding *test* segment. The overall rate of object discovery is 59% and that of object types is 68%. The precision is 98% for objects and 97% for object types. The precision is high because our method eliminates every uncertain object. In a home environment, it is better for a robot to take more data when it is uncertain about the environment than to make a false assumption about its surroundings. If we set the minimum segment requirement to one, i.e. an object is composed of one or more segments, then the discovery rate goes up to 76% for objects and 84% for object types, but the precision drops to 51% for objects and 49% for object types. The drastic decrease of precision is due to the false matches among segments that belong to wall, ceiling, and floor.

Our method does not assume a perfect segmentation. However, the final outcome is affected by the quality of segmentation. Table I shows the rate of discovery against the consistency parameter k and the minimum segment size parameter m_{size} . Our experiment revealed that $k = 9$ and $m_{size} = 100$ yields the highest discovery rate and precision. This is partially due to our assumption that an object is composed of at least two segments. The requirement naturally favors objects that are segmented into multiple segments. Therefore, for a high discovery rate without suffering the precision, it is crucial that the segmentation is done in such a way to allow multiple parts per object while each segment being large enough to be discriminative. One major source of no discovery was the lack of sufficient object segments. When an object is segmented into a single segment, the program fails to discover the object as it is invalid according to our definition of object, as shown in Figure 5. Another

Fig. 5. No object is discovered due to sufficient number of segments. When an object is segmented as one segment (top), the program fails to discover it as an object (bottom).

Fig. 6. No object is discovered because of a failure in the verification step. Objects that are found in the discovery phase (top) are falsely eliminated during the verification phase, yielding no detection (bottom).

source of no detection was the lack of sufficient points on objects. In addition to occlusion and the natural sparsity of data, the incident angle limits the number of usable points in laser data. Without sufficient points, however, we cannot extract segments and features reliably. This causes a lack of object discovery. Lastly, while the verification step most often improved the quality of results, it sometimes eliminated correct hypotheses due to incorrect initialization of point clouds, shown in Figure 6.

VI. CONCLUSION AND OUTLOOK

We presented an approach for unsupervised discovery of repeated objects in range data without a prior knowledge on parts, location, or the number of occurrences. It determines potential object parts by applying a modified superpixel segmentation on the point cloud and extracts features on these segments using spin images, shape distributions, and shape factors. It then discovers objects by finding a set of segments that has a matching set using a joint compatibility test. The objects are verified by the Iterative Closest Point algorithm to minimize false matches. We tested the algorithm on real world data sets to demonstrate its ability to detect repeated objects. The whole process is performed without any supervision and without presegmentation of the background.

There are several avenues for improvement. Work presented in this paper has so far only been tested indoor. While outdoor also contains repetitive structure, using the current algorithm for outdoor scenes poses challenges because outdoor objects are often much bigger than indoor objects. A single scan of an outdoor scene often fails to capture multiple instances of the same object at the level of detail necessary for the algorithm. To overcome this problem, it is necessary to merge several images together to obtain more dense data. Such utilization of a robot's mobility would also improve the indoor results as some objects were undiscovered due to an insufficient number of points on the object. The ultimate goal is to enable a robot to learn the characteristics of a place, which requires to extend the approach as to find matches among several places of the same type.

VII. ACKNOWLEDGMENTS

This work was partially supported by the EC in the project robots@home under contract number FP6-IST-045350, and in the project BACS under contract number FP6-IST-027140.

REFERENCES

- [1] T. Tuytelaars, A. Turina, and L. V. Gool, "Noncombinatorial detection of regular repetitions under perspective skew," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 25, no. 4, pp. 418–432, 2003.

- [2] Y. Liu, R. T. Collins, and Y. Tsin, "A computational model for periodic pattern perception based on frieze and wallpaper groups," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 26, pp. 354–371, 2004.
- [3] G. Loy and J. Olof Eklundh, "Detecting symmetry and symmetric constellations of features," in *ECCV*, 2006, pp. 508–521.
- [4] S. Wenzel, M. Drauschke, and W. Förstner, "Detection of repeated structures in facade images," *Pattern Recognition and Image Analysis*, vol. 18, no. 3, pp. 406–411, 2008.
- [5] G. Zeng and L. V. Gool, "Multi-label image segmentation via point-wise repetition," in *CVPR*, Alaska, USA, June 2008.
- [6] D. Shikhare, S. Bhakar, and S. Mudur, "Compression of large 3d engineering models using automatic discovery of repeating geometric features," in *Vision, Modeling, and Visualization*, 2001.
- [7] J. Podolak, P. Shilane, A. Golovinskiy, S. Rusinkiewicz, and T. Funkhouser, "A planar-reflective symmetry transform for 3d shapes," in *SIGGRAPH '06: ACM SIGGRAPH 2006 Papers*. New York, NY, USA: ACM, 2006, pp. 549–559.
- [8] M. Pauly, N. J. Mitra, J. Wallner, H. Pottmann, and L. J. Guibas, "Discovering structural regularity in 3d geometry," *ACM Trans. Graph.*, vol. 27, no. 3, pp. 1–11, 2008.
- [9] M. Bokeloh, A. Berner, M. Wand, H. Seidel, and A. Schilling, "Symmetry detection using feature lines," *Computer Graphics Forum (Proc. of Eurographics)*, vol. 28, no. 2, pp. 697–706(10), April 2009.
- [10] D. Liu and T. Chen, "Semantic-shift for unsupervised object detection," in *CVPRW '06: Proceedings of the 2006 Conference on Computer Vision and Pattern Recognition Workshop*. Washington, DC, USA: IEEE Computer Society, 2006, p. 16.
- [11] F. Endres, C. Plagemann, C. Stachniss, and W. Burgard, "Unsupervised discovery of object classes from range data using latent dirichlet allocation," in *Proc. of Robotics: Science and Systems*, 2009.
- [12] J. Neira and J. D. Tardos, "Data association in stochastic mapping using the jointcompatibility test," *IEEE Trans. Robotics and Automation*, vol. 17, no. 6, pp. 890–897, 2001.
- [13] J. Shin, S. Gächter, A. Harati, C. Pradelier, and R. Siegwart, "Object classification based on a geometric grammar with a range camera," in *IEEE Int. Conf. Robotics and Automation*, 2009.
- [14] A. Johnson, "Spin-images: A representation for 3-d surface matching," Ph.D. dissertation, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, August 1997.
- [15] R. Triebel, R. Schmidt, O. M. Mozos, and W. Burgard, "Instance-based amn classification for improved object recognition in 2d and 3d laser range data," in *Proc. of the Intern. Joint Conf. on Artif. Intell.*, 2007.
- [16] R. Osada, T. Funkhouser, B. Chazelle, and D. Dobkin, "Shape distributions," *ACM Trans. on Graphics*, vol. 21, no. 4, pp. 807–832, 2002.
- [17] C.-F. Westin, S. Peled, H. Gudbjartsson, R. Kikinis, and F. A. Jolesz, "Geometrical diffusion measures for MRI from tensor basis analysis," in *ISMRM '97*, Vancouver Canada, April 1997, p. 1742.
- [18] P. F. Felzenszwalb and D. P. Huttenlocher, "Efficient graph-based image segmentation," *Int. J. Comput. Vision*, vol. 59, no. 2, pp. 167–181, 2004.
- [19] Z. Zhang, "Iterative point matching for registration of free-form curves and surfaces," *Int. J. Comput. Vision*, vol. 13, no. 2, pp. 119–152, 1994.