OBJECT CLASSIFICATION VIA PLANAR ABSTRACTION

Sven Oesau, Florent Lafarge and Pierre Alliez

INRIA Sophia Antipolis - Méditerranée, France
Email: Firstname.Lastname@inria.fr

KEY WORDS: object classification, point cloud processing, machine learning, planar abstraction

ABSTRACT:

We present a supervised machine learning approach for classification of objects from sampled point data. The main idea consists in first abstracting the input object into planar parts at several scales, then discriminate between the different classes of objects solely through features derived from these planar shapes. Abstracting into planar shapes provides a means to both reduce the computational complexity and improve robustness to defects inherent to the acquisition process. Measuring statistical properties and relationships between planar shapes offers invariance to scale and orientation. A random forest is then used for solving the multiclass classification problem. We demonstrate the potential of our approach on a set of indoor objects from the Princeton shape benchmark and on objects acquired from indoor scenes and compare the performance of our method with other point-based shape descriptors.

1. INTRODUCTION

Beyond geometric modeling, understanding 3D scenes is indispensable for a wide range of applications such as robotics, reverse architecture or augmented reality. While a geometric model of a 3D scene provides a means to navigate and locate surfaces for a robot, a semantic interpretation of this model is required to identify objects and better interact with the environment. The classification of 3D objects is an important facet of the scene understanding problem. While object classification from images has been a long standing research topic, the 3D instance of this problem has been less explored.

Our main motivation for object classification is the modeling and semantization of indoor scenes. Recent advances in acquisition technologies provide high accuracy and sampling rates that allow for an efficient recording of the entire inside of buildings within hours. The rapid evolution of low-cost handheld 3D scanners also provide real-time acquisition of 3D objects or small-scale scenes, in the form of unstructured point clouds. As a consequence, 3D point clouds moved into focus for object classification.

The scientific challenge is to extract high-level information from raw 3D point data. The high diversity of these data due to the wide range of objects and scales, adds further hurdles. Surface reconstructing methods for indoor scenes commonly perform a planar abstraction to reduce complexity and facilitate further processing (Boulch et al., 2014, Mura et al., 2014). Object classification methods instead commonly process the points directly and extract local features from key points. We depart from previous work by computing more global features through exploring the relationships between planar parts detected from the point data.

1.1 Related Work

We now review the two areas closely related to our approach: object classification and planar shape detection and abstraction.

Object classification. Image processing and machine learning have long been concerned by object classification. Supervised machine learning classifiers are often trained to build a model from labeled training data, then to predict labels for new unknown instances. A popular method for detecting and describing key feature points (keypoints) in images is the scale-invariant feature transform (SIFT) (Lowe, 1999). Keypoints for feature extraction are first located by searching for the scale-space of the image with high contrast. Features are then extracted from the neighborhood of each keypoint. Performing the feature extraction at the scale with highest signal range and extracting histograms aligned with the strongest signal peak provides invariance to rotation and scaling.

Several point-based features are used for object classification from point clouds. Rusu *et al.* propose the notion of *fast point feature histograms* (FPFH) (Rusu et al., 2008, Rusu et al., 2009) to capture local geometric properties based on normal information. Johnson *et al.* (Johnson and Hebert, 1999) introduce the *spin images* as a local point descriptor. Knopp *et al.* (Knopp et al., 2010) extend the SURF image descriptor to 3D representation. Based on a point-normal pair the neighboring points are mapped onto a pose-invariant 2D histogram. Common approaches, eg (Teran and Mordohai, 2014), combine several local point descriptors at many keypoints. Based on the resulting labels the classification hypotheses are verified by registering meshes or point clouds of known objects with the scene (Aldoma et al., 2012, Alexandre, 2012). While these approaches achieve good recognition rates, they are in general compute-intensive and have limited capability to classify unknown object instances of a class. More global descriptors are also used, e.g. (Osada et al., 2002, Wohlkinger and Vincze, 2011).

Golovinskiy *et al.* (Golovinskiy et al., 2009) introduce a segmentation and shape-based classification method for objects in urban environments. On large data sets they localize and segment potential objects. A small set of basic features such as estimated volume and spin images, combined with contextual features such as "located on the street", are used to discriminate the objects. After evaluating different machine learning methods they conclude that considering different segmentation methods and adding contextual information significantly improve the detection performance.

Kim *et al.* (Kim et al., 2012) introduce a graph-based primitive matching approach to classify objects in an indoor environment captured by a hand-held scanner. During a learning phase, canonical geometric primitives (e.g., planes, boxes) are fitted to the training point data and a hierarchical primitive-joint graph is

built from the data. A joint herein denotes the type of junction between the primitives. During recognition primitives are fitted to the query point data. Guided by the learned hierarchical graph, the query data are iteratively segmented into objects.

Vosselmann (Vosselman, 2013) discusses different methods for abstraction of high density point data acquired from urban scenes to aid identification of typical urban object classes, i.e., buildings, vegetation, ground and water. Xu *et al.* (Xu et al., 2012) demonstrate the effectiveness of such abstraction via a context rules based classification in the urban environment. Mallet *et al.* (Chehata et al., 2009) demonstrate satisfying performance of the random forest machine learning method in feature based classification of photogrammetry data.

Mattausch *et al.* (Mattausch et al., 2014) introduce a unsupervised machine learning method for segmenting similar objects in indoor scenes. They perform a planar patch detection as preprocessing and categorize the patches into vertical and horizontal patches. A small set of geometric features is computed per patch and a similarity matrix is constructed, considering pairwise similarity between patches that share similar neighborhoods. Clustering under consideration of the similarity matrix yields a segmentation of patches into similar objects across the datasets. The detection results are in general satisfactory but some limitations remain. Only objects with the same upward orientation are clustered and it is unclear whether the method can cluster different types of the same class.

Structural considerations have been recently exploited for object recognition (Kalogerakis et al., 2012, Zheng et al., 2014). The notion of structure goes beyond the use of geometric features as it allows the analysis of an object as a set of connected parts where each part has a specific functionality. Extracting the structure from an object is however a difficult problem that restricts the generality of these methods.

Planar shape detection and abstraction. Related works differ greatly in the way they detect the planar shapes, depending on the defects in the input point data. Region growing is very efficient in point clouds structured as range images (Boulch et al., 2014, Holz and Behnke, 2012, Oesau et al., 2016), but are not suited to unstructured point clouds due to missing neighborhood linkage. The Hough transform (Hough, 1962, Davies, 2005), popular for detection of primitive shapes in images, is now commonly used for plane detection in point clouds. While this approach is robust against various defects such as occlusion and missing data, its memory requirements and computational complexity rapidly increase with the degrees of freedom of the shapes sought after and highly depend on the choice of parameters.
Schnabel *et al.* (Schnabel et al., 2007) proposed an efficient RANSAC method for detecting several primitive shapes in unstructured point data. This approach is robust to defect-laden inputs but does not scale well to complex scenes with many shapes. As shapes are detected under a user-specified tolerance error, we find it relevant to generate hierarchies of shapes detected at different tolerance errors. A recent shape abstraction approach (Mehra et al., 2009, Yumer and Kara, 2012) hinges on the idea that the scale space of objects must be explored for a better understanding of the structural dimension.

For object classification the extraction of local features of the point data from many keypoints requires three main steps. The locality requires the detection of keypoints followed by classification, then clustering in order to turn the labels of keypoints into an object label. As pointed out by Alexandre (Alexandre, 2012), the computational complexity is high. In addition, a point-based feature can only capture local shape properties and is therefore not easy to generalize from single object instances to object classes.

Furthermore, many previous approaches rely upon the knowledge of the up vector (Mattausch et al., 2014, Kim et al., 2012). While the latter helps simplifying the classification problem, it also restricts the detection to upward posed objects.

Positioning. We propose to classify objects based on features derived from planar shapes, themselves detected from the input point data. First, robust and efficient shape detection methods can abstract large point data into a set of planar shapes, at multiple scales. Second, the planar abstraction provides us with a means to extract more global information and capture common properties within object classes. Third, exploring the relationships between the planar shapes yields invariance to orientation and scale.

Contributions. We contribute a novel supervised machine learning method for the classification of objects acquired in indoor scenes. The key novelty of our approach is to derive from a multiscale planar abstraction a so-called feature vector. These features, extracted from a pre-labeled dataset of CAD models, are used to train a random forest classifier (Breiman, 2001) and to evaluate the performance. We then demonstrate the performance of our classifier on point data acquired from indoor scenes.

Our approach improves over previous work on two main aspects:

- **Robustness:** Performing the planar abstraction at different scales makes it possible to detect dominant properties at low scales while being robust to defects and variations in the acquisition process, as well as to capture the discriminative role of details at finer scales.

- **Invariance:** We require no assumptions on orientation or scale. Our approach classifies objects independently from their orientation by using both unoriented features and features that are automatically registered on a detected reference direction.

2. OVERVIEW

Our method takes as input a set of point clouds with unoriented normals, sampled from objects. When normal attributes are not available we estimate them using a principal component analysis in a local neighborhood. For training and evaluation of the classifier a set of ground-truth object labels of the input point clouds is required. We assume that the scene has already been segmented into objects and focus on the classification of objects. Some previous works perform segmentation of objects in a 3D scene (Silberman et al., 2012, Knopp et al., 2011) or perform clustering in feature space in order to segment similar objects in an indoor scan (Mattausch et al., 2014).

Our method generates as output a classifier, ready to predict a trained object class from a feature vector. Our method comprises three main steps: (i) Preprocessing, i.e. multiscale planar abstraction and adjacency detection, (ii) Feature computation, and (iii) Training.

3. MULTISCALE PLANAR ABSTRACTION

The input point data are abstracted by planar shapes using an efficient RANSAC approach (Schnabel et al., 2007), with a range of three fitting tolerances to capture the variation of the extracted

shapes at different scales. The feature vector, computed in following step, aggregates all scales. More specifically, the largest fitting tolerance ϵ is chosen as 2% of the longest bounding box diagonal, then each following scale ϵ is halved. The main reasons for proceeding in a multi-scale fashion are the following. A detailed abstraction by a large number of small planar shapes obfuscates the dominant surfaces of the object. Conversely, choosing a large fitting tolerance captures well the dominant shapes but obfuscates the details. In addition, curved objects behaves differently, as the abstractions differ for each value fitting tolerance, see Fig. 1.

4. FEATURES

Classification through machine learning requires a meaningful description of an object represented by a feature vector:

$$x = (x_1, x_2, .., x_n) \in \mathbb{R}^n, \tag{1}$$

where n denotes the dimension, similar for all feature vectors. In our approach we compute one feature vector per object, and the features are derived solely from the planar shapes. The main rational behind our choice of feature vectors is that the function of an object – class in our context – constrains the shape. As the number of planar shapes detected from a single object depends on the object and detection parameters, we represent *distributions* of features computed for the whole set of planar shapes detected for each object. Each bin of the distribution represents one element of the feature vector, and the distributions are normalized to ensure comparability.

Most features describe distributions: areas, orientations, and relationships between pairs of shapes: pairwise orientation, pairwise orientation restricted to adjacent shapes, transversality. We also add feature elements measuring the global aspect ratio of the object.

Prior to computing the feature vectors we compute for each shape a planar polygon derived from the 2D alpha-shape of the associated point cloud, projected in the detected plane. A planar polygon makes it easy to compute geometric properties such as areas and pairwise orientation.

Note that the random forest approach is oblivious to the relations between the elements of the feature vector, so that a series of elements that belong to the same distribution is unknown to the classifier. In general each element of the feature vector is compared to the same element of other feature vectors. The number of bins of the distributions is thus kept low to avoid increasing the sensitivity of the classifier and to separate objects of the same type. We detail next the features used for training and classification.
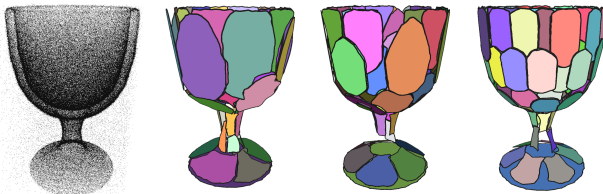

Figure 1: **Multiscale Planar Abstraction.** Left: Input point cloud of a goblet with outliers and noise. Mid to Right: Planar abstraction with varying fitting tolerance from coarse to fine: 1%, 0.5% and 0.25% of bounding box diagonal.

4.1 Area Fragmentation

We compute the distribution of shape areas, normalized to sum up to 1. More specifically, we accumulate the shape area within each bin of the distribution, instead of counting the shapes within a specific area range. The fragmentation of shape areas reflects whether the surface of an object is composed of few large shapes or many smaller planar shapes, or anything in-between such as for a curved surface with a wide range of curvatures. We observed that using a linear scale for the bins of the distribution leads to a poor discriminative capability for the shapes with small areas: We can have either very few large shapes, or many small shapes. We thus use a logarithmic scale of base 2 to provide a higher resolution for the small area bins.
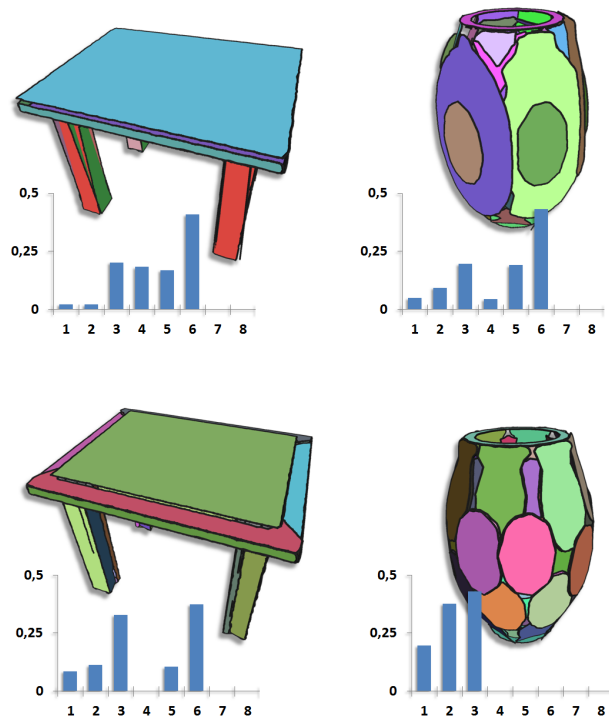
Figure 2: **Area fragmentation under multiple scales.** Top: Planar shapes detected from two point clouds with a large fitting tolerance. The area fragmentation distribution exhibits a high contribution of large shapes to the total shape area. Bottom: Using a small fitting tolerance for shape detection strongly changes the shape composition and hence distribution of the vase, while the distribution for the table exhibits little changes.

4.2 Pairwise Orientation

Assuming the pose of an object is known, the orientation of the parts is judged very discriminant by the random forest algorithm. When the pose is unknown however, the pose must be normalized to ensure bin-to-bin comparability, as the machine learning method sees each element in the feature vector on its own.

In the SIFT operator (Lowe, 1999) rotation-invariance is achieved by aligning the distribution with the reference direction derived from the largest signal peak in the neighborhood of a keypoint.

We compute instead the distribution of angles between all pairs of planar parts, as this does not require any reference direction. More specifically, we consider the range of angles $\left[0, \frac{\pi}{2}\right]$ as the normals are unoriented, and split this range evenly among the bins of the distribution. We then accumulate in each bin the prod-

Figure 3: **Pairwise Orientation.** The distribution of pairwise orientation helps distinguishing different curved objects. The cylindrical shape of the mug is translated into a mostly uniform distribution with a peak owing to the bottom. The orientation distribution for the vase (middle right) reflects the bulgy body by a broader range of angles compared to the lamp (far right).
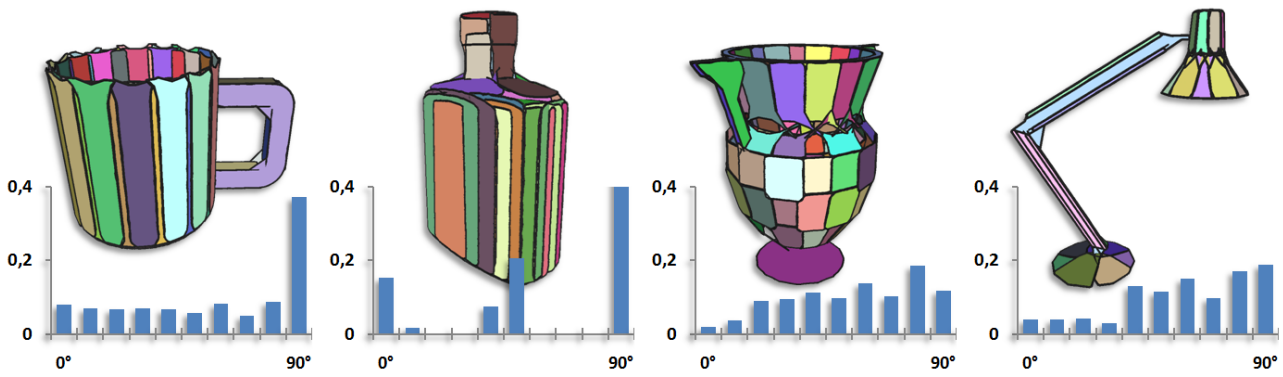
uct of areas of the corresponding pair of planar shapes. The distribution is normalized such that the all bins sum up to 1.

4.3 Adjacent Pairwise Orientation

In addition to the global pairwise orientation we compute the distribution of relative orientations of planar parts that are adjacent, as they reflect the sharpness of creases. Two planar shapes are considered adjacent if their respective alpha-shapes are closer than a user-specified distance, normalized by the longest bounding box diagonal. We first compute the bounding box of each shape and insert them in a hierarchical data structure (AABB tree) to accelerate the distance computations.

4.4 Orientation

The absolute orientation of planar parts plays an important discriminant role to determine the class of an object. Absolute orientation herein refers to a reference upward direction, which is unknown. We thus estimate a reference direction for each object by fitting an object-oriented bounding box. To infer a reference direction we proceed as follows. If the axis of the box with largest extent is unique we chose it as reference direction, if not (the two major axes have comparable extend) we switch to the direction of minor axis. We then compare for each planar shape its projected area with respect to the reference direction, and accumulate these areas in a distribution, with a range of angles $\left[0, \frac{\pi}{2}\right]$. In addition to the orientation distribution, we add to the feature vector the aspect ratio of the oriented bounding box computed as the length of the major axis divided by the length of the longest diagonal.

4.5 Transversality

Transversality is a notion that describes how shapes intersect. In our context transversality also reflects the structure of an object. A compact object, like a drawer or a bottle, exhibits a low transversality while a bookshelf exhibits a high transversality. We compute the transversality of planar shapes by quantifying the relative positioning of all pairs of shapes that are adjacent. Two adjacent shapes that do not meet at their boundary are considered transverse. Given two adjacent planar shapes A and B, we compute the transversality $T(A, B)$ as the (smallest) ratio of areas of A on both sides of the supporting plane of B. For each pair of shapes (A, B) we compute the maximum transversality between $T(A, B)$ and $T(B, A)$. We then compute a transversality distribution with range $\left[0, \frac{1}{2}\right]$, and accumulate in the bins the normalized products of areas for all pairs of adjacent shape. We opt for a small number of bins to avoid confusing low transversality and detection inaccuracies.

5. RANDOM FOREST

Classification via supervised machine learning is performed in two phases. In the training phase a set of feature vectors with associated class labels is used to train a classifier. We choose random forests as machine learning approach, as it is general and effective on many classification problems. It is fast in training as well as in classification and can be parallelized. We use the implementation provided by OpenCV(Bradski, 2000).

Random forests operate by constructing a multitude of decision trees. Decision trees are built by choosing the most discriminative feature, i.e., the element in the feature vector, as a node to separate the training data according to their known class labels. Decision trees are known to overfit, i.e., to adapt to small variations and noise in the training data. Random forests overcome this issue by creating a large number of decision trees. For each decision tree a random subset of the training data is chosen and on each node only a random subset of the features are used. Additionally, the maximum depth of the trees can be limited.

The classification is performed as a voting. The feature vector of an unknown object is evaluated on each tree and the predicted label corresponds to the most voted label. Random forests aim at providing the highest prediction performance for the training data set. Choosing an imbalanced training set, where the number of training samples for each object class varies, can lead to a poor prediction performance for the underrepresented classes. The classifier can afford or sometimes even exhibit a higher prediction performance by neglecting the minority classes. There are different ways to improve the performance. A common and effective way is to downsample overrepresented classes instead of upsampling the minority classes as this may increase noise (Chen et al., 2004).

6. EXPERIMENTS

We implemented our approach in C++ using the CGAL Library (CGAL, 2012), OpenCV and the efficient RANSAC approach implemented by Schnabel (Schnabel et al., 2007). The size of the feature vectors are as follows: 8 bins for the area fragmentation distribution, 10 bins for the pairwise orientation and pairwise adjacent orientation distributions and 5 bins for the orientation and transversality distributions. We achieved the best results for three different scales. This sums up to a feature vector size of dimension 115, including the oriented bounding box ratio.

Object Databases. We perform the evaluation of our classifier on a subset of the Princeton Shape Benchmark (The Princeton

(a)	Bottle	Chair	Couch	Lamp	Mug	Shelf	Table	Vase
Bottle	2	0	0	0	1	0	0	2
Chair	0	4	1	0	0	0	0	0
Couch	0	0	5	0	0	0	0	0
Lamp	0	1	0	3	1	0	0	0
Mug	0	0	0	0	4	0	0	1
Shelf	0	0	0	0	0	5	0	0
Table	0	0	0	0	0	0	5	0
Vase	0	0	0	0	0	0	0	5

(b)	Bottle	Chair	Couch	Lamp	Mug	Shelf	Table	Vase
Bottle	4	0	0	0	0	0	0	1
Chair	0	4	0	0	0	0	1	0
Couch	0	0	3	0	0	0	2	0
Lamp	0	0	0	5	0	0	0	0
Mug	0	0	1	0	4	0	0	0
Shelf	0	0	1	0	0	4	0	0
Table	0	0	0	0	1	2	2	0
Vase	0	1	0	0	0	0	0	4

(c)	Bottle	Chair	Couch	Lamp	Mug	Shelf	Table	Vase
Bottle	5	0	0	0	0	0	0	0
Chair	0	3	1	0	0	0	1	0
Couch	0	0	4	0	0	1	0	0
Lamp	0	0	0	4	1	0	0	0
Mug	0	0	0	0	5	0	0	0
Shelf	0	0	1	0	0	4	0	0
Table	0	1	1	1	0	0	2	0
Vase	1	1	0	0	1	0	0	2

(d)	Bottle	Chair	Couch	Lamp	Mug	Shelf	Table	Vase
Bottle	2	0	0	0	1	0	0	
Chair	1	4	0	0	0	0	0	0
Couch	0	1	3	0	0	0	1	0
Lamp	0	0	0	4	1	0	0	0
Mug	0	0	0	0	4	0	0	1
Shelf	0	0	0	0	0	5	0	0
Table	0	0	0	0	0	0	5	0
Vase	1	0	0	0	0	0	0	4

(e)	Bottle	Chair	Couch	Lamp	Mug	Shelf	Table	Vase
Bottle	5	0	0	0	0	0	0	0
Chair	0	4	0	0	0	0	1	0
Couch	1	0	4	0	0	0	0	0
Lamp	0	0	0	5	0	0	0	0
Mug	1	0	0	0	3	0	0	1
Shelf	1	0	1	0	0	2	1	0
Table	2	1	0	0	0	0	2	0
Vase	0	1	0	0	2	0	0	2

(f)	Bottle	Chair	Couch	Lamp	Mug	Shelf	Table	Vase
Bottle	4	0	1	0	0	0	0	0
Chair	0	2	0	1	0	1	1	0
Couch	0	0	4	0	0	0	1	0
Lamp	2	1	0	1	0	0	0	1
Mug	0	2	0	1	1	0	0	1
Shelf	0	0	0	1	0	4	0	0
Table	0	0	3	0	0	0	2	0
Vase	0	0	0	0	1	0	0	4

(g)	Bottle	Chair	Couch	Lamp	Mug	Shelf	Table	Vase
Bottle	4	0	0	0	0	0	0	1
Chair	0	2	1	1	0	1	0	0
Couch	0	1	4	0	0	0	1	0
Lamp	0	0	1	3	0	0	0	0
Mug	2	0	0	0	1	0	0	2
Shelf	0	0	0	0	0	5	0	0
Table	0	0	0	0	0	0	5	0
Vase	0	0	0	1	0	0	0	4

(h)	Bottle	Chair	Couch	Lamp	Mug	Shelf	Table	Vase
Bottle	5	0	0	0	0	0	0	0
Chair	0	3	0	0	0	0	2	0
Couch	0	0	5	0	0	0	0	0
Lamp	0	0	0	4	0	0	0	1
Mug	0	0	0	0	1	0	0	4
Shelf	0	0	0	0	1	4	0	0
Table	0	0	2	0	0	2	0	1
Vase	1	0	0	0	1	0	0	3

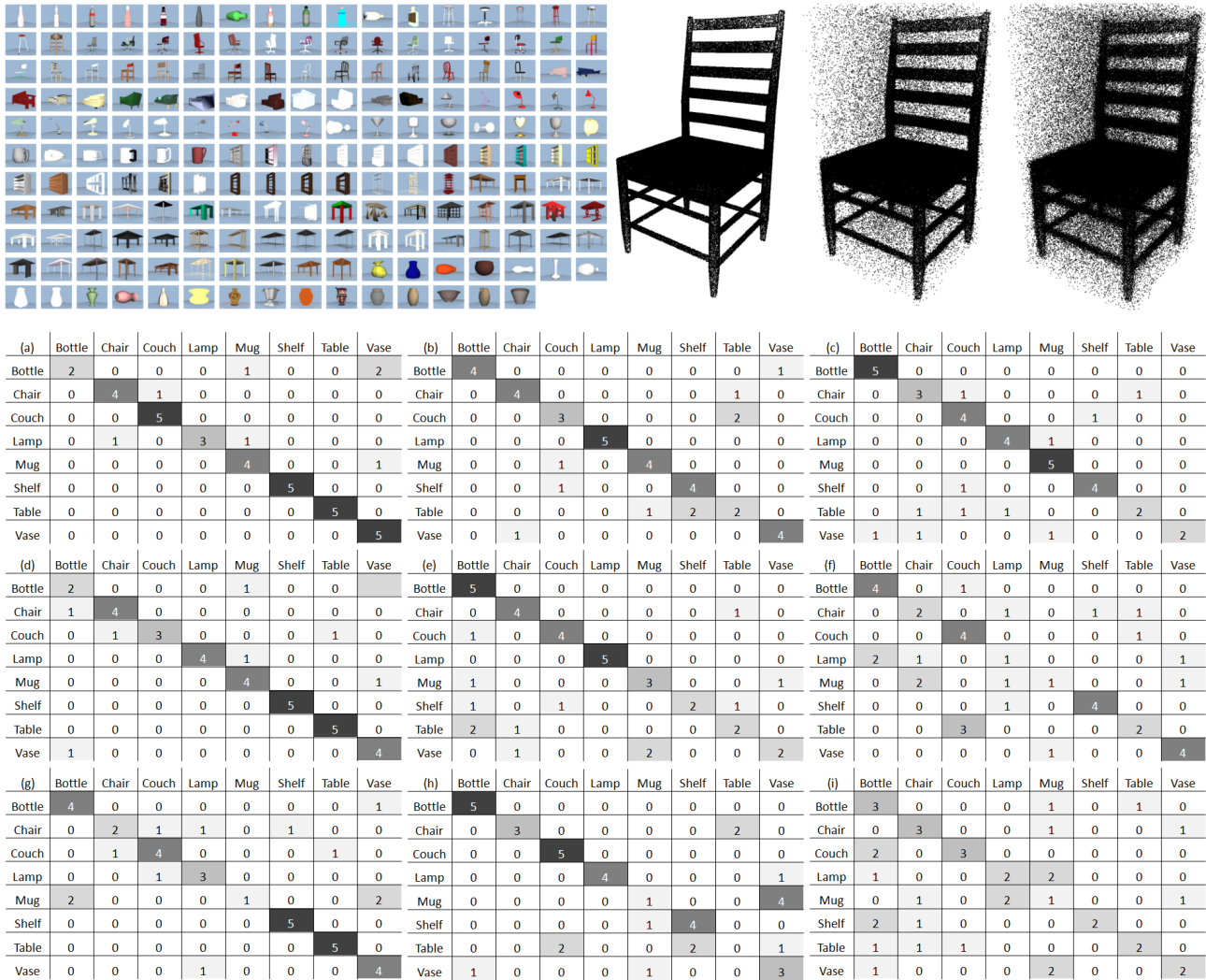| (i)	Bottle	Chair	Couch	Lamp	Mug	Shelf	Table	Vase
Bottle	3	0	0	0	1	0	1	0
Chair	0	3	0	0	1	0	0	1
Couch	2	0	3	0	0	0	0	0
Lamp	1	0	0	2	2	0	0	1
Mug	0	1	0	2	1	0	0	1
Shelf	2	1	0	0	0	2	0	0
Table	1	1	1	0	0	0	2	0
Vase	1	0	0	0	2	0	0	2

Figure 4: **Benchmark.** We compared the performance of our method with the performance of the *D2* by Osada *et al.*(Osada et al., 2002) and *ESF* by Wohlkinger *et al.*(Wohlkinger and Vincze, 2011) shape descriptors on a subset of the Princeton Shape Benchmark (The Princeton Shape Benchmark, 2004) *(top left)* with different added amounts of noise and outliers *(top right)*. The results of each method are shown as confusion matrices in the columns: ours *(left)*, D2 *(mid)*, ESF *(right)* under added defects in the rows: no added defects *(top)*, 0.5% noise and 10% outliers *(mid)*, 1% noise and 20% outliers *(bottom)*. The classification results are displayed in the columns versus the reference classes in the rows. The precision of our method is (a) $82,5\%$ without added defects, (d) $77,5\%$ with some defects and (g) 70% with more defects. The *D2* shape descriptor by Osada *et al.*(Osada et al., 2002) performs with (b) 75%, (e) $67,5\%$ and (h) $62,5\%$ respectively. The *ESF* shape descriptor reveals more sensitive to noise and outliers as the precision drops quickly with increasing amounts of defects: (c) $72,5\%$, (f) 55% and (i) 45%.

Shape Benchmark, 2004), see Fig. 4. A subset of the full dataset is used as many objects do not belong to the indoor environment. We select 100 objects from 8 different object classes that are common to indoor scenes: Bottle, Chair, Couch, Lamp, Mug, Shelf, Table and Vase.Each model in the object database is sampled into a point cloud by ray shooting, and oriented into a random direction to evaluate invariance to orientation. The calculated set of features is split into two sets: 60% for training and 40% for evaluation. To avoid a bias towards overrepresented classes, we remove samples until every class is represented evenly. On the benchmark we achieve a precision of $82,5\%$. The confusion matrix records which are predicted for the objects of one class. Misclassifications occur more often among the objects with curved surfaces. However, the classification of furniture is precise.

Our method is also evaluated from scanned indoor objects, see Fig. 5. Contrary to the previous experiment, the input point clouds are incomplete and suffer from anisotropy, noise and outliers due to acquisition constraints. 20 objects from two different classes, i.e., chair and non-chair, are considered. The training was performed on the scanned indoor objects from randomly chosen 60%, i.e., 12 samples. The classification of the remaining 8 objects predicted correct labels for all chairs and misclassified one non-chair object. The overall precision is $87,5\%$.

Feature importance. Random forest can record the importance of each feature after training. The importance describes the relevance of the feature for separating the class labels during the training process. Table 1 shows the feature importance for evaluation with the Princeton Shape Benchmark. The most relevant feature is the pairwise orientation histogram. The least meaningful feature for the Princeton Shape Benchmark is the transversality, yet it improves the precision. The importance for each scale shows, that the multiscale approach provides a significant advan-

	Area fragmentation	Pairwise orientation	Adjacent pairwise orientation	Orientation	Transversality	Total
Coarse scale	3.3%	9.6%	5%	1.9%	0.3%	**20.1**%
Medium scale	5.8%	14.3%	6.6%	2.4%	2.1%	**31.2**%
Fine scale	6.4%	15.1%	6.8%	4.7%	4.1%	**37**%
All scales	**15.5**%	**39**%	**18.4**%	**9**%	**6.5**%	**88.4**%

Table 1: Feature importance in the classifier by using the Princeton Shape Benchmark. In addition to the histogram features per scale there is the oriented bounding box ratio as a single scalar feature with importance 11.6%.

tage for classification. The shape detection on the fine scale, i.e. with a small fitting tolerance, typically results in the highest number of shapes, but contributes the most information for classification. However, every scale contributes to the classification performance, increasingly from coarse to fine. The transversality at coarse scale provides no significant contribution. Using a high fitting tolerance for non-simple objects leads to overlapping and intersection of detected shapes and induces meaningless transversality.
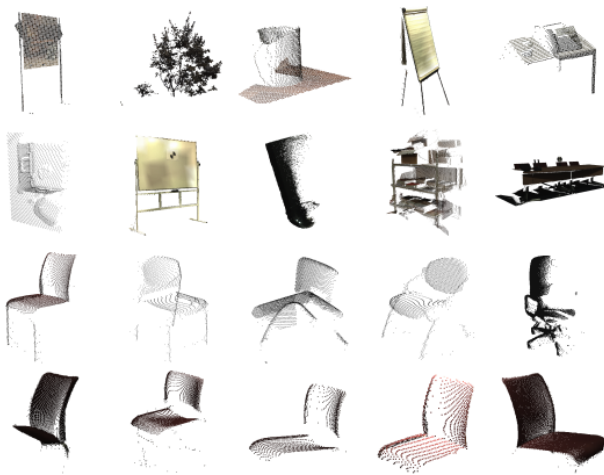

Figure 5: **Indoor objects.** We acquired 20 indoor objects with a Leica Scanstation P20 laser scanner. The sampling of the objects is heterogeneous and partly suffers from anisotropy. The lower 10 objects are labeled as chairs whereas the upper ten objects are labeled as non chairs.

Robustness. To evaluate the robustness of our method, we use the Princeton Shape Benchmark as before, but add noise and outliers before performing the multiscale shape detection. The performance under addition of strong noise is shown as confusion matrices in Fig 4. We performed two experiments and added 10% (20%) outliers and 0.5% (1%) noise w.r.t. bounding box diagonal, see upper right images in Fig. 4. The precision of our classifier is $77, 5\%$ and 70% respectively.

Comparison with existing work. We tested our algorithm against two other global point-based shape descriptors (Osada et al., 2002, Wohlkinger and Vincze, 2011). We implemented the *D2* shape descriptor introduced by Osada *et al.*(Osada et al., 2002) using a 64-bin histogram and 20k samples. Instead of performing pairwise comparison of shape descriptors for classification as proposed by Osada *et al.*, we use a random forest as for our approach. We also compare with the *ESF* shape descriptor using the implementation provided by the *Point Cloud Library* (Aldoma et al., 2012). We performed the same experiments with varying noise and outliers, see Fig. 4. The classifier using *D2* shape descriptor yields a precision of 75% on a noise and outlier free sampling, compared to $72, 5\%$ by the *ESF* shape descriptor

and $82, 5\%$ by our method. Under addition of 0.5% noise and 10% outliers the performance of our methods slightly drops to $77, 5\%$ whereas the point-based descriptors exhibits a stronger loss of precision: $67, 5\%$ by *D2* and 55% by *EFS*. A further addition of noise challenges especially the *ESF* shape descriptor whose performance falls to 45%, while *D2* reaches $62, 5\%$ and our method still provides the strongest performance with 70% precision. Our method shows an advantage of our feature set over the other shape descriptors by not just showing a higher precision but also through higher robustness against defect-laden data. The abstraction of point data by planar shapes provides robustness towards noise and outliers. Point-based shape descriptors provide some robustness against noise as shown by Osada *et al.*., however, they are prone to outliers as our experiments records.

Performance. As recorded by Tab. 2, feature computation is the most compute-intensive operation of our approach. The timing is however reasonable as only a few minutes are necessary to compute all the features of the hundred objects of the Princeton dataset, which represent a total of 25M input points. The timings for learning and testing phases are negligible.

	Feature computation	Learning	Testing
Princeton (8 classes)	232.4	0.11	$< 10^{-3}$
Indoor (2 classes)	1.82	0.02	$< 10^{-3}$

Table 2: Running times (in seconds).

Limitations. Our method assumes that the input objects have been preliminarily extracted from its environment. Although this problem has been explored in depth in the literature, there is still no general solution that separates objects from scanned scenes with a 100% correctness. In terms of robustness, our method is less resilient to missing data than to noise, outliers and heterogeneous sampling. The robustness is gained from abstracting the input point data into planar shapes. Our method might not perform satisfactory on objects that cannot be approximated well by planar shapes, e.g., vegetation.

7. CONCLUSIONS

We introduced a novel method for classifying objects from sampled point data. Departing from previous approaches, our method exploits a planar abstraction to discriminate the different classes of interest. Planar shapes are easy to detect and and manipulate, and allow for a compact object representation, typically a few dozen planar shapes instead of hundred thousands of points. This approach offers several added values in terms of (i) robustness, (ii) orientation and scale invariance, and (iii) low computational complexity.

As future work we plan to explore additional geometric features with improved robustness to missing data. We also wish to extend

our abstraction to richer geometric primitives such as quadrics in order to better classify free-form objects. This new direction may however require designing an altogether different set of geometric features to discriminate the classes of interest.

REFERENCES

Aldoma, A., Marton, Z.-C., Tombari, F., Wohlkinger, W., Potthast, C., Zeisl, B., Rusu, R., Gedikli, S. and Vincze, M., 2012. Tutorial: Point cloud library: Three-dimensional object recognition and 6 dof pose estimation. IEEE Robotics Automation Magazine 19(3), pp. 80–91.

Alexandre, L., 2012. 3D descriptors for object and category recognition: a comparative evaluation. In: Workshop on Color-Depth Camera Fusion in Robotics.

Boulch, A., de La Gorce, M. and Marlet, R., 2014. Piecewise-planar 3d reconstruction with edge and corner regularization. Computer Graphics Forum 33(5), pp. 55–64.

Bradski, G., 2000. The OpenCV Library. Dr. Dobb's Journal of Software Tools.

Breiman, L., 2001. Random forests. Machine Learning.

CGAL, 2012. Computational Geometry Algorithms Library. http://www.cgal.org.

Chehata, N., Guo, L. and Mallet, C., 2009. Airborne lidar feature selection for urban classification using random forests. International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences 39(Part 3/W8), pp. 1117–1120.

Chen, C., Liaw, A. and Breiman, L., 2004. Using Random Forest to Learn Imbalanced Data. Technical report, University of Berkeley.

Davies, E. R., 2005. Computer and Machine Vision: Theory, Algorithms, Practicalities. Morgan Kaufmann Publishers Inc.

Golovinskiy, A., Kim, V. G. and Funkhouser, T., 2009. Shape-based recognition of 3D point clouds in urban environments. In: ICCV.

Holz, D. and Behnke, S., 2012. Fast Range Image Segmentation and Smoothing Using Approximate Surface Reconstruction and Region Growing.

Hough, P., 1962. A method and means for recognizing complex patterns. U.S. Patent No. 3,069,654.

Johnson, A. and Hebert, M., 1999. Using spin images for efficient object recognition in cluttered 3d scenes. PAMI.

Kalogerakis, E., Chaudhuri, S., Koller, D. and Koltun, V., 2012. A probabilistic model for component-based shape synthesis. ACM Trans. on Graphics 31(4), pp. 55:1–55:11.

Kim, Y. M., Mitra, N. J., Yan, D.-M. and Guibas, L., 2012. Acquiring 3D indoor environments with variability and repetition. ACM Trans. on Graphics 31(6), pp. 138:1–138:11.

Knopp, J., Prasad, M. and Van Gool, L., 2011. Scene cut: Class-specific object detection and segmentation in 3d scenes. In: 3DIMPVT.

Knopp, J., Prasad, M., Willems, G., Timofte, R. and Van Gool, L., 2010. Hough transform and 3d surf for robust three dimensional classification. In: ECCV.

Lowe, D., 1999. Object recognition from local scale-invariant features. In: ICCV.

Mattausch, O., Panozzo, D., Mura, C., Sorkine-Hornung, O. and Pajarola, R., 2014. Object detection and classification from large-scale cluttered indoor scans. Computer Graphics Forum 33(2), pp. 11–21.

Mehra, R., Zhou, Q., Long, J., Sheffer, A., Gooch, A. and Mitra, N. J., 2009. Abstraction of man-made shapes. ACM Trans. on Graphics 28(5), pp. 137:1–137:10.

Mura, C., Mattausch, O., Villanueva, A. J., Gobbetti, E. and Pajarola, R., 2014. Automatic room detection and reconstruction in cluttered indoor environments with complex room layouts. Computers & Graphics 44(0), pp. 20–32.

Oesau, S., Lafarge, F. and Alliez, P., 2016. Planar shape detection and regularization in tandem. Computer Graphics Forum 35(1), pp. 203–215.

Osada, R., Funkhouser, T., Chazelle, B. and Dobkin, D., 2002. Shape distributions. ACM Trans. on Graphics 21(4), pp. 807–832.

The Princeton Shape Benchmark, 2004. In: Shape Modeling International conference (SMI).

Rusu, R., Blodow, N. and Beetz, M., 2009. Fast point feature histograms (fpfh) for 3d registration. In: ICRA.

Rusu, R., Marton, Z., Blodow, N. and Beetz, M., 2008. Learning informative point classes for the acquisition of object model maps. In: International Conference on Control, Automation, Robotics and Vision.

Schnabel, R., Wahl, R. and Klein, R., 2007. Efficient ransac for point-cloud shape detection. Computer Graphics Forum 26(2), pp. 214–226.

Silberman, N., Hoiem, D., Kohli, P. and Fergus, R., 2012. Indoor segmentation and support inference from rgbd images. In: ECCV.

Teran, L. and Mordohai, P., 2014. 3d interest point detection via discriminative learning. In: ECCV.

Vosselman, G., 2013. Point cloud segmentation for urban scene classification. International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences XL-7/W2(2), pp. 257–262.

Wohlkinger, W. and Vincze, M., 2011. Ensemble of shape functions for 3d object classification. In: International Conference on Robotics and Biomimetics.

Xu, S., Oude Elberink, S. and Vosselman, G., 2012. Entities and Features for Classifcation of Airborne Laser Scanning Data in Urban Area. ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences I-4, pp. 257–262.

Yumer, M. E. and Kara, L. B., 2012. Co-abstraction of shape collections. ACM Trans. on Graphics 31(6), pp. 166:1–166:11.

Zheng, Y., Cohen-Or, D., Averkiou, M. and Mitra, N. J., 2014. Recurring part arrangements in shape collections. Computer Graphics Forum 33(2), pp. 115–124.