2013 IEEE/RSJ International Conference on
Intelligent Robots and Systems (IROS)
November 3-7, 2013. Tokyo, Japan

Object Recognition in RGBD Images of Cluttered Environments Using Graph-Based Categorization with Unsupervised Learning of Shape Parts

Christian A. Mueller, Kaustubh Pathak and Andreas Birk

Abstract— **We present an approach for object class learning using a part-based shape categorization in RGB-augmented 3D point clouds captured from cluttered indoor scenes with a Kinect-like sensor. A graph representation is used to detect and categorize object instances based on part-constellations found in scenes. No assumptions like objects being placed on planar surfaces or constraints on their poses are required. Our approach consists of the following steps: 1) a Mean-Shift-based over-segmentation of a point cloud into atomic patches; 2) use of topological and geometric features to merge surface-homogeneous atomic patches into super patches; 3) an unsupervised classification of these parts that allows to symbolically label distinctively unknown object parts by their surface-structural appearance; and finally, 4) a graph generation procedure that reflects the constellation of the detected parts from object instances of certain shape categories. Furthermore, an inference procedure is presented that processes extracted part constellations of a scene to detect and categorize object instances. Experiments with challenging, cluttered scenes show that the segmentation procedure provides salient parts of objects which lead to a good categorization performance using the graph-based constellation model concept.**

I. Introduction

Object recognition is a core research topic of AI and robotics that has experienced significant progress over the at least five decades in which it is investigated [1]. In the work presented here, we are interested in a very challenging application scenario, namely the unloading of containers in the context of the EU-project "Cognitive Robot for Automation of Logistic Processes (Roblog)". The objects hence tend to occur in cluttered scenes with many partial occlusions. Also, objects can be deformable, e.g., sacks, or of different sizes and shapes, e.g., various types of parcels and boxes, barrels, etc.

Our method detects surface-homogeneous parts from an over-segmented scene which are later utilized to find parts of objects and finally probable object instances. Neither the appearance of objects nor the variation in the appearance of the application environment is constrained by making assumptions like expecting object instances placed on planar surfaces - also known as table top assumption - or considering only specific object poses. In our application scenario, objects may have been piled up, stacked up or be presented in arbitrary poses.

Instead of learning shape categories from 3D object models which cover the entire or partial object surfaces from certain view points, the part-based approach used here offers a rich representation to model shape categories in the form of a part-constellation graph. A part-based approach offers several advantages for a classification task, especially robustness in form of object rotation invariance and with respect to partial observations especially in situations involving occlusion. Also, this provides a basis for handling deformable objects as discussed later. Parts of detected query instances are used to infer the corresponding shape category using graph models which encode sets of relations between parts of certain shape categories that have been observed during training. Relationships can describe shape appearance or alignment of detected parts, e.g. in case of a *barrel*, the upper planar-surface and the lower cylindric body surface would be considered as two separate parts that are approximately perpendicularly aligned.

The authors are with the Dept. of Electrical Engineering & Computer Science, Jacobs University Bremen, Germany.
[chr.mueller,k.pathak,a.birk]@jacobs-university.de

II. Related Work

The first step in the perception pipeline is typically a detection of candidate regions, i.e., segmentation. A plethora of approaches have been proposed relying on different modalities like 2D RGB cameras or 3D points from time-of-flight cameras. Popular approaches are often based on techniques which aim at detecting probable object boundaries [2], windowing [3], or even saliency detectors [4]. In recent years, with the appearance of low-cost 3D cameras, the focus has switched more and more to approaches which combine or fuse different modalities to gain a more accurate set of object instance candidates rather than e.g. rough 2D regions. In this work, we focus on a probabilistic approach to detect object-related segments in RGBD images which we later merge into object parts and instances.

The Mean-Shift algorithm [5] is a probabilistic approach for segmentation, clustering or object-tracking problems, which has been successfully used on RGB data [6]. Here, Mean-Shift is also used as an initial step but using both RGB and depth information to generate surface-homogeneous patches that over-segment the scene. Then, a surface reconstruction and description method introduced by one of the authors [7] is used to describe these segments. Based on this, we introduce a method that merges those segments, which leads to properly sized and shaped 3D parts. Finally, a method is introduced to filter these 3D parts for being part of an object instance by considering their spatial alignment.

The concept of implicit shape models or constellation models [8] has shown immense success in object recognition or categorization tasks where not only the appearance but also the spatial relationship between local features are taken into account through probabilistic analyses. Also, partial

absence of features due to occlusions or even deformations can be dealt with more efficiently than considering objects as a global and single entity. In our work we follow a similar concept, however, we use 3D object parts rather than texture-dependent 2D features of objects. These 3D parts are classified by unsupervised machine learning for a symbolic distinction by their surface-structural differences. Based on the symbolic part representation a part constellation model for each shape category is created using a graph representation. The model does not only consider part appearances, in addition we also augmented the graph with a variety of features that describe relations between parts. Such a graph representation has the inherent advantage of facilitating access to topology and structure of the containing 3D object parts of certain categories. Graph theory-related algorithms can then be easily exploited and beneficially used for searching cliques and subgraphs or finding connected components. Moreover naïve inference methods such as *Markov Networks* [9], [10] can be applied to each graph model for finding the most probable shape category for a query object instance.

The approach by Anand et al. [11] is close to ours. However, Anand et al. present a graph model for scene part classification rather than for specific object instance classification. Moreover, our graph model is based on a symbolic representation compared to the numeric feature vector representation in their work.

III. Model Formulation

A. Part-Graph Model Generation

In this section, we formulate the part-graph model that is created to represent appearing shape parts $\mathcal{P}$ of objects instances $\mathcal{I}$ and their relations $\mathcal{R}$ for n shape categories $\mathcal{C}$. $\mathcal{R}$ can represent e.g. unary relations (U), i.e. statistics regarding a single shape part, or pairwise relations (P), i.e. statistics between two shape parts. Also a relation can be boolean-like, responding $\{0, 1\}$ or continuous with range $[0, 1]$.

First, we define a dictionary $\mathcal{D}$ which is a repertoire of m words $\{w_1, ..., w_m\}$ to symbolically describe shape parts $\mathcal{P}$ i.e. a word $w \in \mathcal{D}$ symbolically represents a part $p \in \mathcal{P}$ of an instance $I \in \mathcal{I}$. The corresponding word w for p is inferred by $w = f_{\mathcal{D}}(p)$. The mapping function $f_{\mathcal{D}}(\cdot)$ will be discussed in Section IV-D.

Second, we create a set of part-graph models represented by undirected graphs $\mathcal{G}$. For each shape category $c \in \mathcal{C}$ an undirected graph $g_c \in \mathcal{G}$ is created. The undirected graph g_c for category c consists of $\mathcal{V}$ vertices and $\mathcal{E}$ edges: $g_c = \{\mathcal{V}_c, \mathcal{E}_c\}$. Let us assume that a given set of training object instances $\mathcal{I}_c$ of shape category c exists where each instance $I_c \in \mathcal{I}_c$ consists of a set of shape parts $\mathcal{P}_{I_c}$. Any g_c consists of a set of words of the dictionary $\mathcal{D}$ that are represented by vertices $\mathcal{V}_c$ in g_c. A word w exists uniquely in g_c only if w has been inferred by $w = f_{\mathcal{D}}(p_{I_c})$ where $p_{I_c} \in \mathcal{P}_{I_c}$ of a training instance I_c. An edge $e \in \mathcal{E}_c$ is a relation between two words w_i and w_j where $w_i, w_j \in (\mathcal{D}, g_c)$. We define a pairwise relation (P) between two words w_i and w_j by $f^P_{\mathcal{R}}(w_i, w_j) \in \mathcal{R}$. An edge $e \in \mathcal{E}_c$ can be augmented with multiple pairwise relations in $\mathcal{R}$. Also a single word w which is represented by a vertex $v \in \mathcal{V}_c$ can be augmented with multiple unary relations (U), defined by $f^U_{\mathcal{R}}(w) \in \mathcal{R}$. Moreover, higher-order relations can exist which represent cliques of related words by $f^H_{\mathcal{R}}(W) \in \mathcal{R}$ where $W \subseteq (\mathcal{D}, g_c)$. These relations $\mathcal{R}$ ($= f^U_{\mathcal{R}}(\cdot) \cup f^P_{\mathcal{R}}(\cdot) \cup f^H_{\mathcal{R}}(\cdot)$) will allow us to infer the corresponding shape category c of a query object instance I_q by a given set of models $\mathcal{G} = \{g_1, g_2, ..., g_n\}$ for n categories where for each category a model g is created. An illustration is depicted in Fig. 1. It illustrates a set of part-graph models $\mathcal{G}$. Each model $g \in \mathcal{G}$ consists of set of words containing unary, pairwise and higher-order relations. In the next Section III-B we will discuss the concept how relations are used to infer the corresponding shape category of a given query object instance I_q.

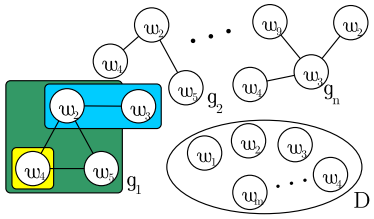


Fig. 1. A set of (simplified) models $\mathcal{G}$ for n shape categories. Each vertex and edge can be augmented with several relations $\mathcal{R}$. A vertex is represented through a word $w \in \mathcal{D}$. For illustration purposes three exemplary relations are selected and shown for model g_1 (unary (yellow), pairwise (blue) and higher-order (green), i.e. order of three).

B. Part-Graph Model Inference

Let us assume that besides the set of part-graph models $\mathcal{G}$, a query object instance I_q is given. Moreover, let us assume that I_q consists of a set of parts $\mathcal{P}_{I_q}$. For each part $p \in \mathcal{P}_{I_q}$ and a given dictionary $\mathcal{D}$, we can infer a corresponding word w ($w = f_{\mathcal{D}}(p)$). From the resulting set of words we can create a graph model $g_{I_q} = \{\mathcal{V}_{I_q}, \mathcal{E}_{I_q}\}$ where for each word a vertex $v \in \mathcal{V}_{I_q}$ is created. An edge $e \in \mathcal{E}_{I_q}$ is created between two words w_i and w_j if the corresponding parts of $\mathcal{P}_{I_q}$ are physical neighbors. These edges represent a simple pairwise relation $f^P_{\mathcal{R}_{\text{neighbor}}}(\cdot)$, i.e. the relation exists if $p_i \in \mathcal{P}_{I_q}$ and $p_j \in \mathcal{P}_{I_q}$ are physical neighbors in object instance I_q that is expressed by $f^P_{\mathcal{R}_{\text{neighbor}}}(f_{\mathcal{D}}(p_i), f_{\mathcal{D}}(p_j))$. An illustration is depicted in Fig. 2.

The target of the model inference procedure is to find the most probable corresponding model $g_c \in \mathcal{G}$ for a given model g_{I_q} of object instance I_q. The defined set of models $\mathcal{G}$ allows to compute a score for the given appearing words and their relations in g_{I_q}. *Markov Networks* applied in computer vision problems are one paradigm that is exploited in our work to determine such scores which allow us to infer the corresponding category c of the instance I_q. If for a given instance I_q a set of shape parts $\mathcal{P}_{I_q}$ has been observed, then we can compute the probability for a shape category c given $\mathcal{P}_{I_q}$ in the form of a Gibbs distribution [10], [12] by:

$$P(c \mid \mathcal{P}_{I_q}) = \frac{1}{\mathcal{Z}(\mathcal{P}_{I_q})} e^{-E(c, \mathcal{P}_{I_q})} \quad (1)$$

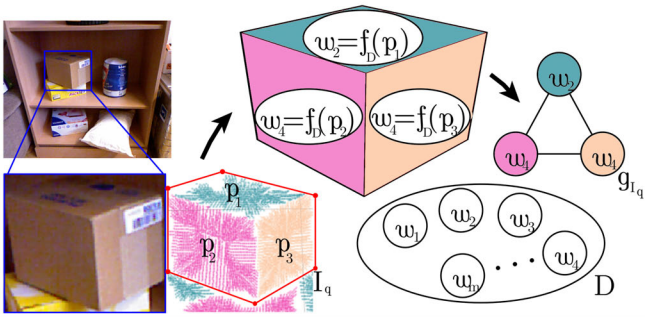


Fig. 2. An image section of a parcel (blue framed) is depicted from a shelf scene and their segmented parts (p_1, p_2, p_3) of the parcel instance I_q (red framed). The figure shows the generation of the graph model g_{I_q} based on the segmented parts and the inferred words (w_2, w_4, w_4) of the dictionary $\mathcal{D}$. The edges between the words (w_2, w_4, w_4) are created through the relation $f^P_{\mathcal{R}_{\text{neighbor}}}(\cdot)$ described in Section III-B.

where $\mathcal{Z}(\cdot)$ is the partition function,

$$\mathcal{Z}(\mathcal{P}_{I_q}) = \sum_{c \in \mathcal{C}} e^{-E(c, \mathcal{P}_{I_q})} \tag{2}$$

and $E(\cdot)$ represents the energy function,

$$E(c, \mathcal{P}_{I_q}) = \sum_{l} f_{\mathcal{R}_{c_l}}(f_{\mathcal{D}}(\mathcal{P}_{I_q})) \tag{3}$$

where $f_{\mathcal{R}_{c_l}}(\cdot)$ represents a relation in $\mathcal{R}_c$ applied on the model g_c of shape category c. In the context of *Markov Networks* we can also denote $f_{\mathcal{R}_c}(\cdot)$ as a clique potential of the instance parts $\mathcal{P}_{I_q}$, defined over the model g_c. Such potentials are described by the prior probabilities of the particular configuration of relations in g_c observed in $\mathcal{P}_{I_q}$. We can elaborate an energy function to *unary*, *pairwise* and *higher-order* clique potentials:

$$\begin{aligned} E(c, \mathcal{P}_{I_q}) = & \sum_{p_i \in \mathcal{P}_{I_q}} \sum_{l} f^U_{\mathcal{R}_{c_l}}(f_{\mathcal{D}}(p_i)) \\ + & \sum_{p_i, p_j \in \mathcal{P}_{I_q}} \sum_{m} f^P_{\mathcal{R}_{c_m}}(f_{\mathcal{D}}(p_i), f_{\mathcal{D}}(p_j)) \\ + & \sum_{P_k \subseteq \mathcal{P}_{I_q}} \sum_{n} f^H_{\mathcal{R}_{c_n}}(f_{\mathcal{D}}(P_k)) \end{aligned} \tag{4}$$

Basically we can summarize that for an observed part $p \in \mathcal{P}_{I_q}$ the potential $f^U_{\mathcal{R}_c}(\cdot)$ is computed which was observed as a part of an instance of shape category $c \in \mathcal{C}$ during model generation g_c. While for two and more observed parts in $\mathcal{P}_{I_q}$ the potentials $f^P_{\mathcal{R}_c}(\cdot)$ and $f^H_{\mathcal{R}_c}(\cdot)$ are computed which were observed as related parts of an instance of shape category $c \in \mathcal{C}$ during model generation g_c.

Finally we can infer the shape category of a set of parts $\mathcal{P}_{I_q}$ of instance I_q by solving the *maximum a posteriori* (MAP) or the *energy minimization* problem, respectively.

$$c = \arg\max_c P(c \mid \mathcal{P}_{I_q}) = \arg\min_c E(c, \mathcal{P}_{I_q}) \tag{5}$$

IV. Shape Part Detection and Description

In this section we present a hierarchical approach to find reasonable and coherent surface parts in the scene. First we propose an over-segmentation approach that partitions the scene into surface-homogeneous segments which we call atomic patches. These atomic patches provide the basis for finding surface coherent parts (super patches) in the scene by merging the atomic patches in an appropriate way. Later on these parts are used for detecting object instances and finally for inferring the shape category of such instances.

(a) RGB Scene (b) Segmentation result

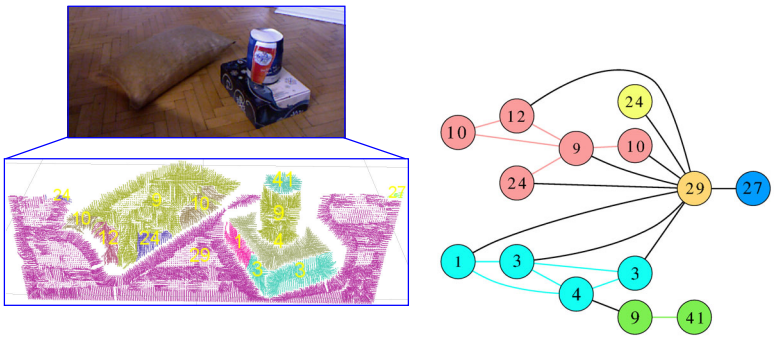
Fig. 3. Mean-Shift segments (atomic patches) colored randomly.

A. Atomic Patch Detection based on Mean-Shift

We over-segment an RGBD image using Mean-Shift to generate small atomic patches (Fig. 3) that can be used as fundamental elements for further detection and classification purposes. Mean-Shift [5] aims to iteratively converge into local maxima of multivariate probability distribution/density function (*pdf*) that are defined over a feature space. The feature-space can be defined e.g. by RGB pixels, ranges, 3D points, local normals, curvature, etc. The Mean-Shift iterations perform an implicit gradient ascent on the multivariate probability distribution in the feature-space without having to explicitly compute the probabilities themselves. The implicit gradient-ascent iterations are started at uniformly placed points on the RGBD image grid. The iterations converge to local maxima of the *pdf*; the basin of convergence of each local maximum defines a segment. The parameter selection for Mean-Shift requires mainly the determination of the bandwidths in each feature space partition. Several features spaces from RGBD information were experimented with to achieve a detection of surface-homogeneous patches in cluttered environments. We finally selected the following features: the L component from the LUV color space, x, y, z coordinates and local surface normals from the 3D point cloud space.

B. Super Patch Detection by Merging of Atomic Patches

Due to the small region of the scene which is covered by each atomic patch, often neighboring patches reflect similar surface structures and hence the surface diversity is minor. These fine-granular atomic patches provides a baseline for merging atomic patches into a group of similar neighboring patches that we denote as a super patch. In Algorithm 1 the super patch detection procedure is shown. Super patches typically already cover reasonable coherent parts of objects with similar geometric attributes as illustrated in Fig. 4(a) and 7(a). Hence neighboring super patches are preferred to be surface-diverse. For instance, a *barrel* is typically extracted into two parts, namely the upper top planar surface combined



(a) (b)

Fig. 4. A simplified scene $\mathcal{O}$ and the detected super patches are depicted in 4(a). The super patches are labeled by the corresponding visual words ($|\mathcal{D}| = 45$) – see Section IV-D. Subsequently a graph g_O of scene $\mathcal{O}$ (4(b)) is accordingly created – see Section V. By applying Algorithm 2 a set of subgraphs (randomly colored) is extracted which correspond to detected object instances whereas black edges indicate removed relations.

Algorithm 1 Super Patch Detection

Input: RGBD point cloud of an observed scene $\mathcal{O}$

1: Detect atomic patches $\mathcal{A}_O$ of scene $\mathcal{O}$ (Fig. 3)
2: Create a graph by adding a vertex for each atomic patch in $\mathcal{A}_O$
3: Create an edge between two vertices if the corresponding patches satisfies the similarity that is conditioned by:
   1) patches are neighbors,
   2) patches are aligned by an angle less than β ($= 75°$),
   3) description of patches (see Section IV-C) are similar (measured with *Jensen-Shannon Divergence* [13]) by a threshold λ ($= 0.5$).
4: Find connected components in the graph
5: Merge point clouds of atomic patches which correspond to vertices that are part of a connected component. Each merged point cloud represent a super patch $s \in \mathcal{S}_O$.

Output: The set of super patches $\mathcal{S}_O$ (Fig. 4(a))

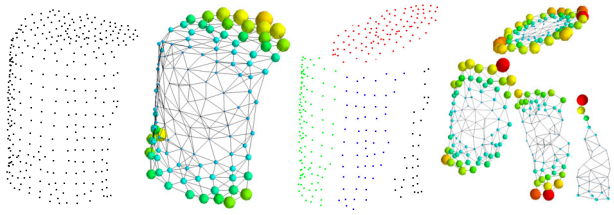
with the lower cylindric part. Therefore, we can consider super patches in practice also as actual object parts.

C. Global Patch Description

Both atomic patches $\mathcal{A}$ and super patches $\mathcal{S}$ of an observed scene $\mathcal{O}$ have to be represented for comparison and inference purposes. We aim to express a patch which is represented as an unorganized point cloud t by a single description vector $\boldsymbol{\delta}$ where $\boldsymbol{\delta} = f_{\mathcal{P}}(t)$. In our previous work [7] we proposed a description method that not only considers topology but also curvature characteristics of point clouds. Here the method is applied on patches rather than on objects. It is divided into two steps. First, a surface mesh m of point cloud t is reconstructed by an unsupervised topology learner [14] with $m = L_{GNG}(t)$. Second, so called shape functions [15] are applied on m to extract the topological ($f_{\mathcal{S}_1}(m)$) and curvature ($f_{\mathcal{S}_2}(m)$) characteristics. Based on the concatenated responses of the two shape functions we computed $\boldsymbol{\delta}$ as

$$\boldsymbol{\delta} = f_{\mathcal{P}}(t) = \Big\|_{i=1} f_{\mathcal{S}_i}(L_{GNG}(t)) \tag{6}$$

1) Surface Reconstruction – $L_{GNG}(\cdot)$: The surface reconstruction step plays a particularly important role since a surface – in the form of a mesh – encodes geometric properties of a patch which can be used for description purposes. Standard triangulation approaches create connections between points in the point cloud in a way such that a Delaunay-like graph is formed. Therein the aim is to reconstruct a surface to a mesh which is coherent with the actual patch surface. However such reconstruction is influenced by noise in the patch point cloud which is captured by the camera from the real-world. In contrast, our surface reconstruction approach treats the patch point cloud as a distribution of points in 3D space. The goal is to learn the distribution of these points rather than to directly project the point cloud into a mesh as this is the case in the first mentioned approach. An unsupervised learning method, a modified Growing Neural Gas (GNG) [14] is applied to learn the distribution of the points in a Hebbian learning manner. This leads to a mesh-like representation that reflects the topology of the point cloud. In principle, random points are iteratively selected from the cloud and fired into a mesh. The mesh gradually adapts to the point distribution and to the topology of the point cloud – samples of reconstructed surfaces are shown in Fig. 5. Note that a node in the mesh does not represent a point in the point cloud but a distinctive location in the distribution of points of the cloud. In order to adapt this basic distribution learner (GNG) to the surface reconstruction problem several modifications were introduced in our



(a) Object (b) GNG Object (c) Patches (d) GNG Patches

Fig. 5. A GNG generated mesh (5(b)) of a *barrel* point cloud (5(a)). Samples of segmented object parts are shown in 5(c) and 5(d). Note that a partial point cloud observation is given as input. Nodes are sized and colored by the mean *All-Pair-Shortest-Path* (ASPS) distances (see Sec. IV-C.2).

previous work [7], like repeatedly retraining the point cloud to gain a more consistent mesh regarding the actual patch surface, the improvement of the triangulation through adding mild noise to each point which is fired into the mesh or the removal of diverted nodes from the mesh which are not coherent with the actual patch surface. Several beneficial properties of these methods have been observed such as a smoothing, de-noising and a reorganizing effect on the distribution of nodes in the mesh. This leads to similarly reconstructed surfaces from structurally similar ground truth object patches despite possible difference in appearance due to for example different noise levels.

2) Shape Functions – $f_{\mathcal{S}_i}(\cdot)$: We focus on two characteristic properties of a mesh, topology and curvature which we denote as shape functions. To express these characteristics in an expressive numerical and computationally efficient manner a distribution-based descriptor [15] for each shape function is used. The topological characteristic of m is reflected by $f_{\mathcal{S}_1}(\cdot)$ through a density estimation over the *All-Pair-Shortest-Path* (APSP) distances using Johnson's algorithm. Therein the shortest path for each node to all other nodes in m is computed. By using this method the distances are computed in a *geodesic* manner. This means that the connectivity of the nodes that reflects the surface characteristics of the mesh is considered as the distance measure between nodes rather than using a naïve direct node-to-node euclidean distance measure. The estimated density of the distances is projected to a discrete probability distribution that is a normalized histogram computed over the distance distribution. For identifying a descriptive bin size and width of the histogram with a computationally low cost, Scott's Rule [16] is used. On the other hand we applied the same procedure to reflect the curvature characteristic of m by $f_{\mathcal{S}_2}(\cdot)$. In this case we applied a density estimation over the surface normals of the nodes in m. Finally these shape function responses with respect to m are concatenated into a single vector $\boldsymbol{\delta}$ which *globally* describes the *topological* and *curvature* characteristics of m.

3) Rotation and Scale Invariance: A major issue of a global surface description is its invariance to rotation and scale. The first part in the histogram of the surface description vector $\boldsymbol{\delta}$, namely the geodesic part (APSP) is rotation invariant since it is based on the distances between the nodes in the mesh. In contrast the second part which describes the distribution of the surface normals is not rotation invariant: it will change with the rotation of the patch. Hence, a pose normalization approach is applied in advance which is inspired by Sfikas et al. [17] and is based on 3D reflective symmetry. It shows robustness to conditions such as noisy and partial observations.

Due to the proposed combination of a distribution-based descriptor applied on the GNG-generated mesh, a beneficial effect on scale invariance is observed – see further details in [7]. Nevertheless, in context of shape categorization, additionally in a preprocessing step patch point clouds are normalized in an unit sphere-like manner. That means the distance between the centroid and the farthest point of a point cloud is 1 and all remaining points are adapted proportionally. This step is applied to unify changing appearances of similar patches due to scale variations.

4) Atomic and Super Patch Descriptions: For an expressive surface description $\boldsymbol{\delta}$ of atomic patches, the bin size and width is individually determined by atomic patches that are collected from random scenes. Those patches are neither labeled nor deliberately selected. The same procedure is also applied for the super patch description since the properties according to the patch size and actual surface information differs from the description space defined for atomic patches.

D. Symbolic Parts through Unsupervised Classification

In this section we describe the unsupervised classification of the super patches. We have previously referred to it as a mapping function $f_{\mathcal{D}}(\cdot)$ which symbolically describes a part p by the corresponding word w ($w = f_{\mathcal{D}}(p)$) where $\mathcal{D}$ is a dictionary – see Section III-A.

A dictionary is often used in *Bag-of-Words* [18] approaches to generate a common feature space whose dimensionality reduces the original space and provides a first unsupervised classification e.g. of image features. Here, we also exploit this concept of an unsupervised classification to generalize the appearance variety of super patches to a set of symbolic labels which we call (visual) words. Such a dictionary $\mathcal{D}$ consists of a set of words which are generated through clustering the descriptions (see Section IV-C) of super patches. A fast *k*-means clustering algorithm is applied to group similar surface patches. In succession of the k-means clustering of the super patches, the center of each group represents a word. Hence an unknown super patch can be labeled by the nearest center or word, respectively. A major parameter is the number of visual words (*k*) of the dictionary used for a discriminative classification – *k* can be seen as a resolution parameter for the dictionary quantization of the super patches. An appropriate *k* (dictionary size $|\mathcal{D}|$) can be determined by the *Dunn Validity Index* [19]. Such a procedure was introduced in our previous work [20]. Thereby the dictionary size is varied in sequence from 2 to n_{max} words. An indication of a dictionary size that leads to a reasonable classification is found by the identification of local maxima of the sequence of validity values. That one local maximum is empirically selected whose magnitude is most pronounced.

For training of the dictionary, a set of 500 unlabeled super patches was collected from random scenes. Note that in order to classify similar patches to similar words even in cases where patches differ by scale or rotation, the pose normalization procedure described in Section IV-C.3 is applied. An exemplary classification result of super patches is shown in Fig. 4(a) and 8(b). Already in this stage correspondences can be inferred between the responded words and the surface-structural appearance of the super patches.

V. Object Instance Detection

A major challenge is to detect object instances from a scene by determining a subset of neighboring parts, which

(a) RGB scene (b) Convex part alignment

Fig. 6. Convex aligned neighboring parts (super patches) are connected with a green edge, non-convex neighboring parts with a red edge.

when combined, plausibly represents an object instance. Based on the detected parts in Section IV, we create a part-graph model g_O of an observed scene $\mathcal{O}$. The model g_O consists of an extracted set of parts $\mathcal{P}_O$ of $\mathcal{O}$. As in Section III-B described we apply the $f^P_{\mathcal{R}_{\text{neighbor}}}(\cdot)$ relation to create edges between neighboring parts of the scene. As a result g_O represents a graph of the scene of which object instances are also a part. Such a graph is illustrated in Fig. 4(b). In the next step, we propose a method to detect potential instances in the scene based on the model g_O. Therein we aim to detect subgraphs in g_O which probably represent object instances. Due to the camera *view point* and the *partial* object observation we can exploit these facts that neighboring parts generally appear in a *convex* or *planar* alignment as it is observable for instances in Fig. 4(a) or 8(b). An illustration can be found in Fig. 6. In Algorithm 2 we summarize the object instance detection procedure. Finally the algorithm

Algorithm 2 Object Instance Detection

Input: RGBD point cloud of an observed scene $\mathcal{O}$

1: Apply Algorithm 1 on $\mathcal{O}$ to detect super patches $\mathcal{S}_O$ so that a set of parts $\mathcal{P}_O$ is detected (Fig. 4(a))
2: Apply dictionary $\mathcal{D}$ on $\mathcal{P}_O$ by $f_{\mathcal{D}}(\mathcal{P}_O)$ (Fig. 4(a))
3: Create a graph g_O by applying the relation $f^P_{\mathcal{R}_{\text{neighbor}}}(\mathcal{P}_O)$ (Fig. 4(b))
4: Remove relations in g_O where parts are non-convex aligned
5: Extract connected components in graph g_O. Each connected component represents a subgraph in g_O. Each subgraph is a potential object instance $I \in \mathcal{I}_O$.

Output: A set of instances $\mathcal{I}_O$ (Fig. 4(b))

returns a set of subgraphs in which each subgraph is a potential object instance, see Fig. 4(b). In the next Section VI, it is explained how these object instances are processed to identify the corresponding shape category.

VI. Object Shape Category Recognition

The previous steps of detecting parts and labeling these parts with a dictionary $\mathcal{D}$, allow us to generate models $\mathcal{G}$ of appearances of such labeled parts for object instances of different shape categories as described in Section III-A. In Section III-B we introduced the model inference procedure as an *energy minimization* problem by solving the energy effort of the observed parts $\mathcal{P}_{I_q}$ of an instance I_q for a given model g_c in $\mathcal{G}$ of shape category c. The energy effort for $\mathcal{P}_{I_q}$ is computed by the set of unary, pairwise and higher-order relations defined in Table I for each g_c model.

TABLE I

Summary of unary, pairwise and higher-order relations

Type/ Range	Description	Relation
U, $[0,1]$	Word appearance in I_q	$f^U_{\mathcal{R}_{c_1}}(f_{\mathcal{D}}(p))$, $p \in \mathcal{P}_{I_q}$
U $[0,1]$	Number of parts in I_q	$f^U_{\mathcal{R}_{c_2}}(\lvert\mathcal{P}_{I_q}\rvert)$
U, $[0,1]$	Proportional size of parts in I_q	$f^U_{\mathcal{R}_{c_3}}(f_{\mathcal{D}}(p))$, $p \in \mathcal{P}_{I_q}$
P, $\{0,1\}$	Words in I_q being physical neighbors	$f^P_{\mathcal{R}_{c_1}}(f_{\mathcal{D}}(p_i), f_{\mathcal{D}}(p_j))$,$p_i, p_j \in \mathcal{P}_{I_q}$
P, $\{0,1\}$	Words in I_q being convex aligned	$f^P_{\mathcal{R}_{c_2}}(f_{\mathcal{D}}(p_i), f_{\mathcal{D}}(p_j))$, $p_i, p_j \in \mathcal{P}_{I_q}$
P, $\{0,1\}$	Words in I_q being planar aligned	$f^P_{\mathcal{R}_{c_3}}(f_{\mathcal{D}}(p_i), f_{\mathcal{D}}(p_j))$, $p_i, p_j \in \mathcal{P}_{I_q}$
P, $[0,1]$	Angular alignment between words in I_q	$f^P_{\mathcal{R}_{c_4}}(f_{\mathcal{D}}(p_i), f_{\mathcal{D}}(p_j))$, $p_i, p_j \in \mathcal{P}_{I_q}$
H, $[0,1]$	Clique of word appearances in I_q	$f^H_{\mathcal{R}_{c_1}}(f_{\mathcal{D}}(P))$, $P \subseteq \mathcal{P}_{I_q}$, where P forms a subgraph in g_c of > 2 vertices

A set of unary (U), pairwise (P) and higher-order (H) relations that are used to infer the energy of the parts $\mathcal{P}_{I_q}$ of instance I_q for a model $g_c \in \mathcal{G}$. The relations represent statistics, e.g. $f^U_{\mathcal{R}_{c_1}}(\cdot)$ can be interpreted as the confidence that $w = f_{\mathcal{D}}(p))$ has appeared in category c.

Therein each relation $f_{\mathcal{R}_{c_i}}$ returns a response according to the given constellation of words and relations between words of category c in $g_c \in \mathcal{G}$. The responses of the relations are accumulated as defined in Eq. 3 for each category. Solving Eq. 5 we receive the most probable shape category c for I_q.

Algorithm 3 Object Shape Category Recognition

Input: RGBD point cloud of an observed scene $\mathcal{O}$, the given set of models $\mathcal{G}$

1: Apply Algorithm 2 on $\mathcal{O}$ to detect a set of object instances $\mathcal{I}_O$
2: For each instance $I \in \mathcal{I}_O$ compute the energy effort (Eq. 4) considering the relations in Table I for each model g_c in $\mathcal{G}$
3: By solving Eq. 5 infer the MAP label for an instance $I \in \mathcal{I}_O$.
4: Remove a labeled instance in $I \in \mathcal{I}_O$ whose MAP is lower than a threshold $\tau (= 50\%)$

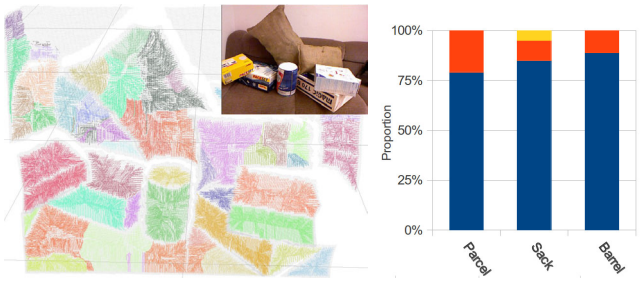
Output: A set of labeled instances $\mathcal{I}_L \subseteq \mathcal{I}_O$

The proposed *part-based* approach analyzes object instances in a local fashion, i.e. parts are detected and inferred by their constellation to be related to a certain category. Hence even the appearance of e.g. a parcel with a single planar facing part is most likely to be classified as a *parcel* – however with a low confidence. Therefore a threshold τ is introduced to remove less confident instances. In Algorithm 3 we summarize the recognition procedure.

VII. Experimental Evaluation

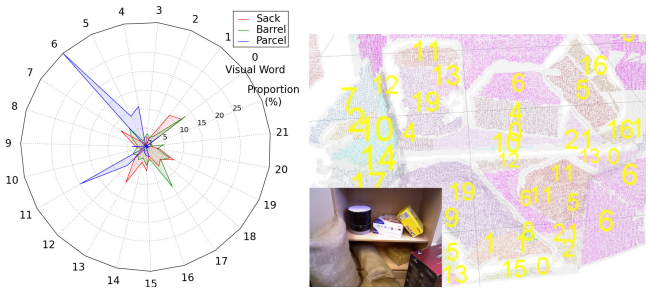
A. Super Patches and Unsupervised Part Classification

The super patch detection procedure merges the atomic patches into a set of patches which significantly cover entire object parts. Typical evaluation measures are over- and under-segmentation. In our experiment 50 random surfaces



(a) Super patch detection (b) Segmentation Results

Fig. 7. Super patch detection results of a cluttered scene (7(a)). 7(b) shows the segmentation result of super patches regarding *parcel*s, *sack*s and *barrel* (over (red), under (yellow) and good (blue) segmentation).



(a) Visual word distribution (b) Visual word assignment

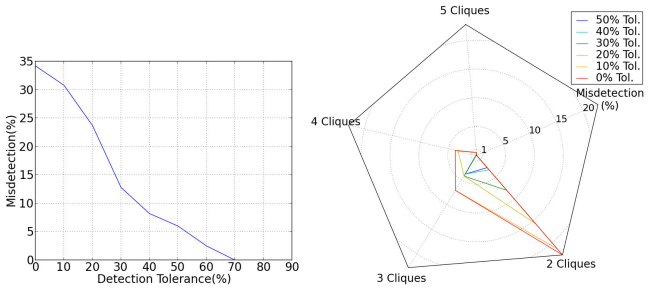
Fig. 8. Visual word distribution (8(a)) based on frequencies of labeled object instances. $|\mathcal{D}| = 21$ visual words. Visual words assignments of super patches based on $\mathcal{D}$ are shown in 8(b).

are captured from the scene and evaluated by these measures. As a result, 87.1% of the super patches are appropriately detected. Compared to this result, 5.5% are over-segmented and 7.4% under-segmented. A similar segmentation behavior is shown in Fig. 7(b) for *parcel*, *barrel* and *sack*-like surfaces.

The unsupervised classification of super patches and their actual correspondences to the surface characteristics that the super patches describe are illustrated in Fig. 8. The distribution of visual word assignments (Fig. 8(a)) of super patches detected from *parcels*, *sacks* and *barrels* are also reflected in the scene Fig. 8(b) – e.g. the most prominent words for *parcels* are word 11 and 6 which also appear for the white and brown parcel (see center and bottom in Fig. 8(b)).

B. Object Instance Detection and Shape Categorization

Finding objects as a single instance is challenging in cluttered scenes. To analyze the classification behavior of an occluded or a partially observed instance, we define a detection tolerance criterion which is defined by the portion (in %) containing the actual detected instance from a testing set, i.e. in case of 50% tolerance at least 50% of the parts of an object instance are supposed to be detected as a single clique of connected parts or a graph, respectively. In Fig. 9 detection results are illustrated regarding the defined tolerance. As it can be observed, the higher the tolerance, the more (partial) object instances are detected. A detection rate of up to 65.2% is achieved with no tolerance (0%) where



(a) Misdetection rate (b) Number of cliques of misdetections

Fig. 9. Instance detection result (60 instances per category) regarding detection tolerance (c.f. definition in Section VII-B).

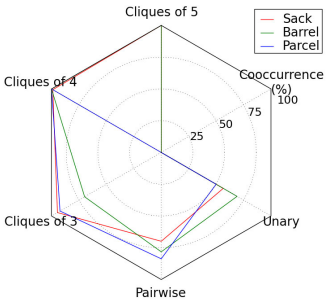


Fig. 10. Cooccurrence of part constellations (unary, pairwise, higher-order) in test instances (20 instances per category) and models of $\mathcal{G}$.

instances were mostly misdetected as two separated cliques of parts – see Fig. 9(b). However a tolerance which covers an object by at least 50% leads to a detection rate of 93.2%. High tolerances might lead to partial detections of object instances and thus to a higher chance of misclassification as we show later on.

In Fig. 10 the cooccurrence of part constellations in a test instance and the trained models of $\mathcal{G}$ is illustrated. A cooccurrence is only found if a detected constellation in the labeled test instance also appears in the model $g \in \mathcal{G}$ of the same label and is most pronounced compared to the remaining models in $\mathcal{G}$. The result verifies that the higher the clique order of an observed clique in an instance, the more distinctive is the clique for being an evidence for a certain category. For instance, *parcel* cliques of order three that appear also in a test instance have a chance of 90% of being actually a part of a *parcel*.

In Fig. 11 we compare the detection, test and a 5-fold crossvalidation error (CV) regarding the number of training samples per category and detection tolerance. An inversely-proportional tendency can be observed between the detection error and the classification error of the testing set with respect to the detection tolerance. By increasing the number of instances used for the model training, a final 6.1% testing set classification error has been achieved on a model trained with 60 samples per category with a 0% detection tolerance. A detection tolerance of 40% still lead to a 7% classification error, whereas a high detection tolerance (60%) leads to more misclassifications (testing set error = 12.8%, CV error = 2.2%) and can be analogously interpreted as partial or

(a) Scene 1, τ=50% (b) Scene 2, τ=50%

Fig. 12. Recognition result of two scenes (labels are red for *sacks*, green for *barrels*, blue for *parcels*). A label is augmented with the detected instance number and the recognition confidence.

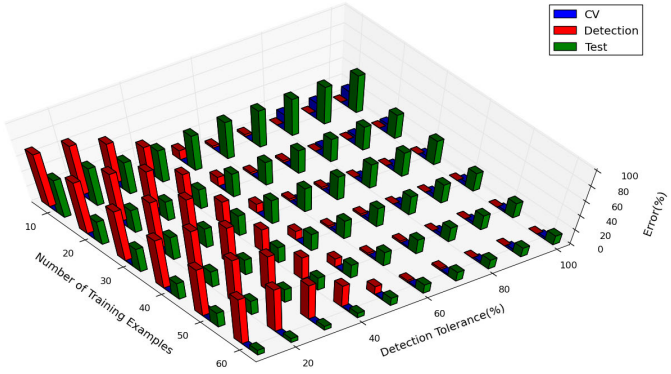


Fig. 11. CV error (blue), detection (red) and classification error (green) on a testing set (20 instances per category) with respect to detection tolerance and number of training instances.

occluded object observations in real-world scenes. Fig. 12 shows examples of such scenes with several detected and classified instances which are differently posed and partially occluded.

VIII. Conclusion

We presented a graph-based 3D object shape categorization approach – by over-segmenting RGBD images of scenes, detecting atomic patches, merging them to super patches, classifying super patches by a dictionary to visual words, and finally learning the visual words constellations and their relations for certain shape categories. The experiments have shown the detection of reasonable object parts in cluttered scenes and the correspondences between structural appearance and the symbolic visual word labeling of these parts. Also the actual reoccurrence of patterns of visual word constellations for instances of certain categories could be shown and exploited for a final shape categorization of instances.

Acknowledgment

The research leading to the results presented here has received funding from the European Community's Seventh Framework Programme (EU FP7 ICT-2) within the project "Cognitive Robot for Automation of Logistics Processes (RobLog)".

References

[1] L. G. Roberts, "Machine perception of three-dimensional solids," 1963.
[2] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pp. 888–905, 2000.
[3] Q. Nie, S. Zhan, and W. Li, "Object Recognition Based on Efficient Sub-window," *Pattern Recognition*, pp. 435–443, 2009.
[4] E. Rahtu, J. Kannala, M. Salo, and J. Heikkilä, "Segmenting Salient Objects from Images and Videos," in *ECCV (5)*, 2010, pp. 366–379.
[5] D. Comaniciu, "Mean shift: A robust approach toward feature space analysis," *Analysis and Machine Intelligence,*, vol. 24, no. 5, pp. 603–619, 2002.
[6] C. Pantofaru, G. Dorkó, C. Schmid, and M. Hebert, "Combining regions and patches for object class localization," in *Conference on Computer Vision and Pattern Recognition Workshop (Beyond Patches workshop, CVPR '06)*, 2006.
[7] C. A. Mueller, P. Ploeger, and M. S. Roscoe, "Towards Scalable 3D Object Shape Categorization," *Active Semantic Perception Workshop on Intelligent Robots and Systems(IROS)*, 2012.
[8] B. Leibe, A. Leonardis, and B. Schiele, "Combined object categorization and segmentation with an implicit shape model," in *In ECCV workshop on statistical learning in computer vision*, 2004, pp. 17–32.
[9] R. Kindermann and J. L. Snell, *Markov Random Fields and Their Applications*, 1980.
[10] S. Geman and D. Geman, "Stochastic relaxation, gibbs distributions, and the bayesian restoration of images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 6, no. 6, pp. 721–741, 1984.
[11] 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*, 2012.
[12] P. Kohli, M. P. Kumar, and P. H. S. Torr, "P3 & beyond: Move making algorithms for solving higher order functions," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 9, pp. 1645–1656, 2009.
[13] B. Fuglede and F. Topsoe, "Jensen-Shannon divergence and Hilbert space embedding," *Information Theory, 2004. ISIT 2004. Proceedings. International Symposium on*, pp. 31+, 2004.
[14] B. Fritzke, "A Growing Neural Gas Network Learns Topologies," in *NIPS*, 1994, pp. 625–632.
[15] R. Osada, T. Funkhouser, B. Chazelle, and D. Dobkin, "Matching 3D Models with Shape Distributions," *Proceedings International Conference on Shape Modeling and Applications*, pp. 154–166, 2001.
[16] D. W. Scott, "On Optimal and Data-Based Histograms," *Biometrika*, vol. 66, no. 3, pp. 605–610, 1979.
[17] K. Sfikas, T. Theoharis, and I. Pratikakis, "ROSy+: 3D Object Pose Normalization Based on PCA and Reflective Object Symmetry with Application in 3D Object Retrieval," *Int. J. Computer Vision*, 2011.
[18] F. Jurie and B. Triggs, "Creating efficient codebooks for visual recognition," in *ICCV*, 2005, pp. 604–610.
[19] J. C. Dunn, "A Fuzzy Relative of the ISODATA Process and Its Use in Detecting Compact Well-Separated Clusters," *Cybernetics and Systems*, vol. 3, pp. 32–57, 1973.
[20] C. A. Mueller, N. Hochgeschwender, and P. G. Ploeger, "Towards Robust Object Categorization for Mobile Robots with Combination of Classifiers," *RoboCup International Symposium*, vol. 15, 2011.