

FP7-IST-60918

1 March 2013 (36months)

DR 2.1: Multi-modal compositional hierarchies

M. Popa, S. Rezapour-Lakani, A. Rietzler, F. Spinelli, C. Rosales, J. Piater

Institute of Computer Science, University of Innsbruck
(Mirela.Popa@uibk.ac.at)

Due date of deliverable: 2015-02-28
Actual submission date: 2015-02-28
Lead partner: UIBK
Revision: final
Dissemination level: PU

This deliverable describes the intermediate results of Tasks 2.1 and 2.2 as specified in the DoW. In this report, the representations of shape and the inference methods developed are described along with the way in which grasps are attached to the object representations.

1	Tasks, objectives, results	4
1.1	Planned work	4
1.2	Actual work performed	4
1.2.1	Task 2.3 Structuring 3D compositional models according to grasp affordances	4
1.2.2	Task 2.1 Integrating haptics into 3D compositional models	6
1.2.3	Task 2.2 Integrating graspability into 3D compositional models	9
1.2.4	Task 2.4 3D compositional models of the work space	10
1.3	Relation to the state-of-the-art	11
A	Annexes	13
A.1	Article: Learning 3D Object Parts through Grasping	13
A.2	Article: Learning Part-Based 3D Compositional Object Representations	21
A.3	Technical Report: Multi-Modal Compositional Object Representation	30
A.4	Technical Report: 3D Compositional Model associated with Grasp Templates and Haptic Data	39

A.5	Technical Report: Object classification using tactile sensor information . . .	47
A.6	Technical Report: UNIPi Grasp Dataset performed by humans using the Pisa/IIT SoftHand	58
A.7	Technical Report: A New Approach to 3D Object Recognition and Pose Estimation	67

Executive Summary

This report presents work carried out in WP2 on multi-modal compositional hierarchies of object categories, using both visual and non-visual features. The work addresses the intermediate results of Tasks 2.1, 2.2 and 2.3. We describe an approach for integrating non-visual features into the compositional hierarchy, as defined in Task 2.1.

Next, we present the association of graspability to object parts formed using a hierarchical representation, according to Task 2.2. In addition, we propose a method for structuring compositional object models into semantically meaningful parts, in the sense that the meaning is guided by their graspability.

Role of the Multi-modal compositional hierarchies in PaCMan

In WP2, we focus on integrating non-visual features into the multi-modal compositional hierarchy. Furthermore, we propose forming semantically meaningful parts based on their graspability. Both aspects contribute to a better understanding of the environment and will be used to refine object hypotheses, decompose an object into graspable parts, as well as reasoning under incomplete observations in the visual and non-visual domain.

Contribution to the PaCMan scenario

The proposed algorithms will be used to process visual and non-visual information and to support reasoning under uncertainty in the dishwasher-scenarios addressed in WP5. Furthermore, by using the algorithms described in this report, a scene can be analyzed in a refined manner, by extracting information not only about object classes, but also about the semantically meaningful object parts and their graspability property. Moreover, the analysis of the non-visual information can be used to analyze the success of an object grasp.

1 Tasks, objectives, results

The objective of WP2 is the integration of visual and non-visual modalities into a single representation. The obtained unified, rich, compositional representation that incorporates complementary modalities, namely, visual features, as well as non-visual features will allow inference of unobserved from observed modalities, joint consideration of distinct modalities for increased robustness, and generalization to novel objects via shared parts.

1.1 Planned work

DR 2.1 addresses multi-modal compositional hierarchies for object representation using both visual and non-visual information. Planned work for each Task in WP2 is as follows:

Task 2.1 consists of integration of non-visual features into the compositional model, by defining a suitable parametrization of haptic features. Next, it addresses the development of methods for incremental acquisition of haptic information and its incorporation into the multi-modal representation.

Task 2.2 refers to the integration of graspability into the compositional 3D model, by associating grasps to object parts learned using a compositional approach.

Task 2.3 develops algorithms for structuring the 3D compositional models based on graspability are designed and implemented.

Task 2.4 addresses the detailed description of a scene, using all types of information provided by the multi-modal representation, such as object models described by their pose, as well as any visual, haptic and graspability properties associated with the objects.

1.2 Actual work performed

In this section, the main achievements related to the topic of this deliverable are briefly described. We choose to introduce first Task 2.3, as it describes the developed compositional model which is used in Tasks 2.1, 2.2 and 2.4, facilitating a better understanding of the performed work.

1.2.1 Task 2.3 Structuring 3D compositional models according to grasp affordances

We designed and implemented a method for structuring the 3D compositional models according to object part graspability. In order to associate graspability to an object part and to integrate the tactile information, this

work uses, for the time being, a hierarchical model that forms just three levels at distinct levels of expressiveness (patches, parts, and object), rather than the many-layered approach of WP1. We are now also investigating the attachment of grasps to 3D parts developed using the approach of WP1.

Objects are represented in a compositional hierarchy where their parts are described by the relationship between sub-parts, which are subsequently represented based on the relationships between small adjacent patches. We learn our model based on view-based RGB-D point cloud data. We have developed a bottom-up compositional model which starts from the object points and reaches the object category level. Initially, the points form low-level patches. These patches construct the lowest level of our compositional model. In order to capture the different patch types in our dataset, we extract features from them, which quantize the distribution of normals inside the patch and perform clustering to form a patch codebook.

The next level in our hierarchical model is represented by object parts, which are composed of low-level patches. We describe each object part as a histogram of the patch types which compose it. This histogram representation constructs our visual features. Furthermore, we perform clustering on our object parts to obtain different part types and form a part codebook.

Finally, the object level is composed of a set of parts in a specific configuration. Our objective is to form parts that are meaningful in robotic interaction. Thus, parts are formed from object regions observed to be involved in robotic grasps, which is a particular novelty of our approach. An object is grasped in different ways, depending on its intended functionality. Information gathered during grasping experiments is exploited in our framework to build semantically meaningful parts.

Furthermore, from the conceptual view of our hierarchical framework, our objective is to collect two types of information. First, we are interested in computing the probabilistic occurrence of a certain part in an object class. Secondly, we aim at learning the relations between the parts.

For collecting the first type of information, we segment an object into its constituent parts, and assign each part to an entry in the part codebook. From the obtained part types, we collect statistics of occurrence of each part type in a certain object class.

To obtain the second type of information, regarding adjacency relations between parts, we are rather interested in how parts are connected. Therefore, we analyze the patches adjacent to the border between two adjacent parts, instead of all the patches inside a part. Our assumption relies on the fact that object instances which belong to one class have the same connection between the parts. As an example, mugs can have different handles and different bodies but the bodies and handles are always connected with a hyperbolic surface.

Next, we recognize these boundary patches given our patch codebook, and we collect statistics of their co-occurrence for a certain object class.

An illustration of the described approach is depicted in Figure 1, 2, 3 .

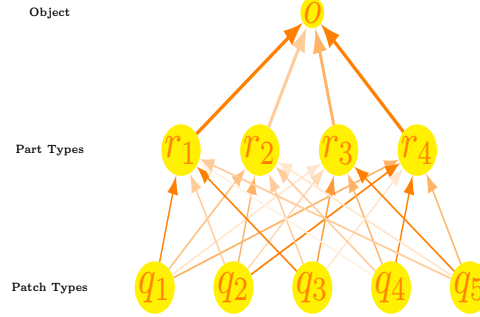


Figure 1: A schematic diagram for a learned model for object class o . We have learned patch types $Q = \{q_1, \dots, q_{N_Q}\}$ and learned part types $R = \{r_1, \dots, r_4\}$. A part is composed of patches $\mathbf{Q} = \{\mathbf{q}_1, \dots, \mathbf{q}_4\}$. Those patches are matched to the patch types. Hence a part is represented as a histogram of patch types. As shown in the figure, the lighter the color the less frequent the patch type and vice versa. Next, an object is composed of parts $\mathbf{R} = \{\mathbf{r}_1, \mathbf{r}_2, \mathbf{r}_3\}$. The parts are matched to the parts clusters. We then compute the probability of occurrence of the object parts types which belong to the object class o , $p(r_1, \dots, r_{N_R}|o)$. For object recognition we learn the $p(r_1, \dots, r_{N_R}|o)$ as well as part connectivity based on adjacent boundary patches $p(q_1, q_2|o)$.

The novelty of our approach is two-fold. First, the proposed algorithm forms semantically meaningful object parts in a hierarchical manner, by exploiting the object functionality, and more specifically in our case the object graspability. Next, it provides a generalization mechanism to recognize novel object instances by exploiting the relations between adjacent parts, where the composition at each level of the hierarchy is learned from grasping experience. We evaluated our method on our collected IKEA Object dataset [1] and on the RGB-D Washington dataset [2]. Our approach achieved promising results, outperforming competing methods [3]. More details can be found in Sec. A.1 and A.2.

1.2.2 Task 2.1 Integrating haptics into 3D compositional models

This task consists of integrating non-visual features into the 3D compositional model. To achieve the objectives of Task 2.1, we have taken several steps. First, we implemented a framework for acquiring tactile information simultaneously with visual data, using our robotic framework.

Next, we recorded a dataset of IKEA objects, consisting of RGB and depth data, grasping information and tactile readings corresponding to the grasped patches of the object. In these experiments we used a parametriza-

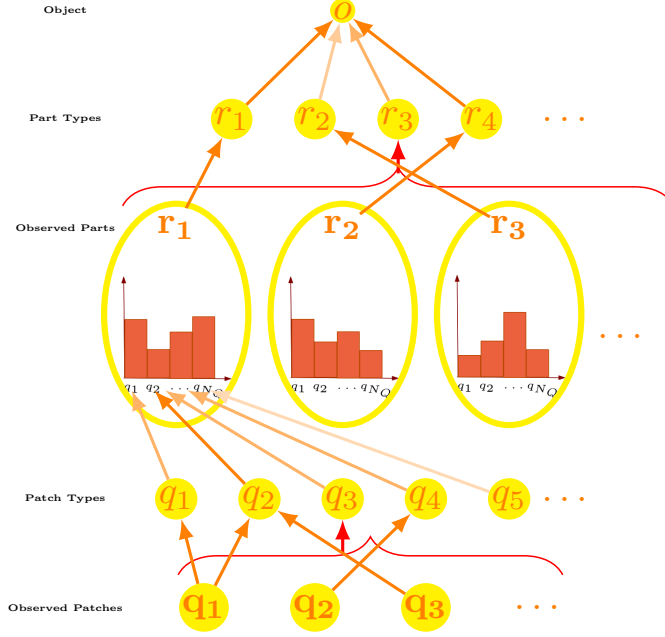


Figure 2: A schematic diagram for data flow in learning the compositional model. We segment each object to its patches $\mathbf{Q} = \{\mathbf{q}_1, \dots, \mathbf{q}_4\}$. Next, we obtain patch types by clustering the patches from multiple objects as shown by the red brace $Q = \{q_1, \dots, q_{NQ}\}$. The obtained patch types are used for part segmentation. The observed object parts are denoted by $\mathbf{R} = \{\mathbf{r}_1, \mathbf{r}_2, \mathbf{r}_3\}$. We have patches of each object part, then we match them to the learned patch types, as shown by the orange arrows. One patch can be matched to multiple patch clusters but it belongs to only one part. Next, we represent a part as the histogram of its constituent patch types. We then perform clustering on the part histograms and obtain part types $R = \{r_1, r_2, r_3\}$. For the objects which belongs to the object class o , we compute the likelihood of their constituent parts to the learned part types. The arrows from part types to the object class node show the frequency of occurrence of each of part types for an object class. The lighter the color the less frequent the part type for the object class o .

tion suitable for our SCHUNK dexterous hand. In order to find a good parametrization of tactile features, we have done a pre-analysis of the acquired data, by extracting a set of features such as object diameter, compressibility, standard deviation of intensity values and translation and rotation invariant Chebyshev moments and using a multi-class SVM classifier. The details of the developed method are included in Sec. A.5. The results of the analysis showed that our approach was useful for object discrimination using only non-visual features. Therefore, we will use a similar tactile fea-

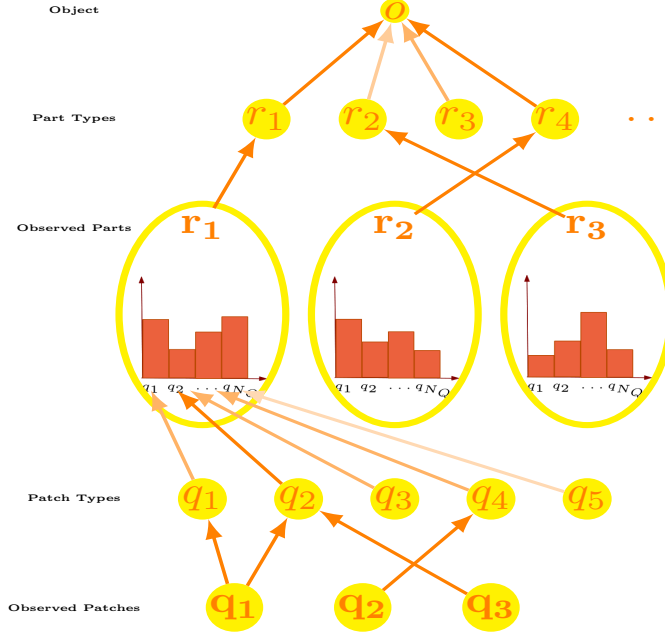


Figure 3: A schematic diagram for data flow in an instantiated object model. Initially, an object is segmented to its constituent patches which are considered as observed patches $\mathbf{Q} = \{q_1, \dots, q_4\}$. These patches are matched to the learned patch type $Q = \{q_1, \dots, q_{N_Q}\}$ and assigned to only one object part as denoted by $\mathbf{R} = \{r_1, r_2, r_3\}$. A part is represented by a histogram of patch types based on its constituent patch types. These parts are then matched to the part types as denoted by $R = \{r_1, r_2, r_3\}$. Finally, an object is classified to object class o based on the probability of occurrence of their part types as well as their adjacent boundary patches for the object class o .

ture parametrization and associate it with our developed 3D compositional model for obtaining a multi-modal representation.

Furthermore, using our developed compositional framework presented in Task 2.3 and described in detail in Sec. A.2, we decompose an object into its constituent parts and associate the corresponding tactile features. The integration of visual and non-visual features is done in a probabilistic manner, as described in Sec. A.3. One application of the proposed representation is object class recognition in the case of partial or missing sensory data, either visual or non-visual.

The actual steps required for extracting tactile features and fusing them with visual features are:

1. Extract features from the tactile readings.
2. Associate the extracted features to the part level of the hierarchy.

3. Formulate in a probabilistic manner the binding of haptic and visual features.
4. Propagate the visual and non-visual information in the hierarchy, in a bottom-up manner, to allow inference of one type of information in the absence of the other.

Particularly, point 1 will use the output of Task 3.1 in WP3 to properly feature extraction, which has already been completed and proved to be useful in fusing visual and non-visual information. This way, we can contrast the proposed approach to competitive approaches under the same framework and conditions.

1.2.3 Task 2.2 Integrating graspability into 3D compositional models

In this task, we associated grasps to object parts, obtained using our proposed compositional framework. This approach facilitates grasping unknown objects, which are composed of parts similar to those learned by our hierarchical model. To facilitate this task, we have designed and implemented a compositional framework which is described in detail in Sec. A.2. The description of the grasp representation and of the experimental setup is presented in Sec. A.4.

The steps performed to associate grasps to object parts are:

1. Record a dataset of grasped object parts, by capturing object relative wrist pose of the gripper, tactile signature and kinesthetic signature (gripper joints).
2. Detect and localize object parts using our proposed compositional model introduced in Task 2.3.
3. Form a set of 3D shape templates according to different grasp types (pinch, spherical, and parallel).
4. Given a scene, align the set of 3D shape templates to the cropped point-cloud sections, provided by the part recognition module and choose the best one according to an optimization procedure.
5. Evaluate the association of grasps to object parts, by performing robotic experiments on a test set containing objects partially similar to the ones in the training set.

Concerning point 1, it is worth noting that from the beginning of the project, we agreed that the use of the Columbia Grasp Database was to be replaced by the use of a new database showing grasps on objects. Thus, as part of this task, we also recorded a dataset of grasps performed by a

human operator with the Pisa/IIT SoftHand. Details of this dataset can be found in Sec. A.6. This Grasp Database uses the Pisa/IIT SoftHand operated through a handle to allow subjects to position and close the hand arbitrarily. Sensor data and parameters were recorded including a point cloud of the scene to validate the proposed approach.

The grasp template alignment procedure is based on our pose estimation algorithm [4], where the 6D pose dependent cross correlation between the shape distribution of the scene and a template is computed.

Next, in order to find a single best grasp for each template, we have developed a grasp inference algorithm which searches over a large number of hypothesis, while performing collision checking with the environment and checking inverse kinematics reachability of the gripper wrist pose hypothesis.

The performed experiments show that the obtained grasp association score is around 70%, while the prediction of successful from unsuccessful grasps based on haptic features achieves an average accuracy of 80%. The discussion of the experimental results is included in Sec A.4.

1.2.4 Task 2.4 3D compositional models of the work space

Reasoning about a scene is useful for action planning or grasp planning in the presence of clutter. Visual information plays the most important role, while additional information such as object graspability and non-visual features such as haptics, are useful for improving the representation of the work space.

Therefore, we have analyzed first the performance of our 3D compositional model mainly with respect to two aspects: part recognition for novel objects and object classification and localization. A detailed description of the method and the achieved results can be found in Task 2.3, while we present next a brief discussion of our main contributions.

Due to the nature of our object representation in which object parts are described by the relationship between sub-parts, which are subsequently represented based on the relationships between small adjacent patches, we were able to obtain a high average recognition rate of 85% on novel object instances, composed of parts different from the ones in the training set. The generalization property of our model is due to the fact that we learn the relations between adjacent parts in an object and therefore our method is not fully dependent on the structure of a part. Furthermore, our approach outperformed state-of-the-art methods [5], which achieved poor results for novel object instances. In addition we have performed an initial study on models of instances of object shape. This work uses SHOT features to recognise and estimate the pose of objects within a scene. We have evaluated the performance of these methods on the IKEA dataset that we generated in the first year of the project. Finally, we enriched the representation of a scene, by associating grasp templates to detected object parts, as described

in Task 2.2.

As future work we plan to analyze the stability of a grasp using haptic information which can indicate to what extent the test grasp configuration is similar to training data.

1.3 Relation to the state-of-the-art

The main contribution of the results achieved for Task 2.3 lies in a successful design and implementation of a learnt compositional hierarchy. The main difference from other works [6, 7] introducing hierarchical compositional 3D object models, consists in the generalization and scale invariant properties of our representation. Furthermore, by exploiting the relations between object parts, we were able to recognize novel object instances having a similar structure, while the local appearance of a part was different from the learned examples. Our method achieved an accuracy of about 85% for the recognition of novel object instances, outperforming state-of-the-art methods [5].

Moreover, our proposed method performed better at decomposing an object into parts, compared to [3], where the notion of object convexity and concavity was exploited for segmenting an object into parts.

In the state-of-the-art methods for part-based object recognition [8, 9, 10], parts are manually labeled in the training examples, while the novelty of our approach consists in extracting the region corresponding to a part, from grasping experiments. An object is grasped in different ways, depending on its intended functionality, information which is exploited in our framework for forming semantically meaningful parts.

References

- [1] PaCMan Object Datasets. [Online]. Available: <http://www.pacman-project.eu/datasets/>
- [2] K. Lai, L. Bo, X. Ren, and D. Fox, “A large-scale hierarchical multi-view rgb-d object dataset,” in *IEEE Int. Conf. on Robotics and Automation*, 2011.
- [3] S. Christoph Stein, M. Schoeler, J. Papon, and F. Worgotter, “Object partitioning using local convexity,” in *IEEE Conf. on Computer Vision and Pattern Recognition*, June 2014.
- [4] R. Detry and J. Piater, “Continuous surface-point distributions for 3d object pose estimation and recognition,” in *Asian Conference on Computer Vision*, vol. 6494, 2010, pp. 572–585.
- [5] R. B. Rusu, G. Bradski, R. Thibaux, and J. Hsu, “Fast 3D recognition and pose using the viewpoint feature histogram,” in *IEEE/RSJ Int. Conf., Intelligent Robots and Systems*, 2010.
- [6] W. Hu, “Learning 3D object templates by hierarchical quantization of geometry and appearance spaces,” in *IEEE Conf. on Computer Vision and Pattern Recognition*, 2012, pp. 2336–2343.
- [7] L. Zhu and A. L. Yuille, “A hierarchical compositional system for rapid object detection,” in *NIPS*, 2005.
- [8] P. F. Felzenszwalb and D. P. Huttenlocher, “Pictorial structures for object recognition,” *Int. J. Comput. Vision*, vol. 61, no. 1, pp. 55–79, Jan. 2005.
- [9] M. Burl, T. K. Leung, and P. Perona, “Face localization via shape statistics,” 1995.
- [10] M. C. Burl, M. Weber, and P. Perona, “A probabilistic approach to object recognition using local photometry and global geometry,” 1998.

A Annexes

A.1 Article: Learning 3D Object Parts through Grasping

Authors Safoura Rezapour Lakani, Mirela Popa, Antonio J. Rodriguez-Sanchez and Justus Piater

Info In preparation to be submitted to the 10th International Conference on Computer Vision Systems (ICVS), 2015

Abstract Most objects are composed of parts which have a semantic meaning. A handle can have many different shapes and be present in quite different objects, but there is only one semantic meaning to a handle, which is “a part that is designed especially to be grasped by the hand”. We introduce here a novel 3D algorithm for the decomposition of objects into their semantically meaningful parts. These meaningful parts are learned from experiments where a robot grasps different objects. Object are represented in a compositional graph hierarchy where their parts are represented as the relationship between sub-parts, which are subsequently represented based on the relationships between small adjacent regions. Unlike other 2D compositional approaches, the importance of our 3D oriented method relies on the relationship between adjacent regions and sub-parts and where the composition at each level of the hierarchy is learned from grasping experience. We evaluate the validity of our compositional approach in three different scenarios, where we test the importance of the extracted relationships, its robustness to scale and more importantly, the recognition of parts from previously unseen objects

Relation with the deliverable The paper addresses Task 2.3 by introducing a method for learning objects parts in a bottom-up way, from robotic experiments of grasping objects

Attachment (following pages until next annex)

Learning Objects Parts through Grasping

Safoura Rezapour Lakani, Mirela Popa, Antonio J. Rodríguez-Sánchez, Justus Piater

Abstract— This paper presents a novel method for decomposing objects into semantically meaningful parts based on robotic grasping experiments. We propose a bottom-up compositional method for forming meaningful object parts based on their sub-parts. The composition is guided by grasping experience. The compositional nature of the method results in finding novel object parts in unseen objects where the detected sub-parts are similar to those of the previously-encountered objects. We obtained promising results on part decomposition of novel objects.

I. INTRODUCTION

Part-based object recognition has been studied in the computer vision domain for decades [1], [2], [3]. Representing an object by a configuration of its parts is the key concept in these methods. In addition, representing a part itself is also critical. Parts should be represented distinctively in an object, and they should be semantically meaningful. Hence, decomposing an object into meaningful parts has an important impact on object recognition and classification performance.

In the state-of-the-art methods for part-based object recognition, parts are manually labeled in training examples, and conventional feature descriptors are extracted from them. For recognition, a sliding-window method at different scales matches the area of interest to the trained models. In other work [4], [5], [6], parts are compositions of sub-parts, where the sub-parts are small patches of specific sizes and at different resolutions. The parts are also labeled manually for obtaining a better parametrization of the object structure. Recognition is performed mostly in two ways. Recognition is performed mostly in two ways. First, a bag-of-words method on the patches based on different parts is employed [7]. Then an SVM-based classification is performed on the obtained histograms. Many methods make use of a graphical model based on the patches that compose a labeled part during training [8]. In the work discussed in [9], object parts are formed in a compositional representation of sub-parts that constitute them. The sub-parts are learned from their co-occurrence statistics in training data.

There are also methods that make use of geometrical information for object part segmentation. In the work discussed in [10], geometrical structure is estimated from depth cues and surface normals. Then a learning algorithm classifies major object and scene surfaces and use them for a recognition task. Looking into convexity or concavity of object patches for finding an object part is discussed in [11]. Initially, an object is segmented to supervoxels. The part segmentation works on the supervoxels. The object is segmented into parts if the degree of convexity between adjacent supervoxels is

less than a predefined threshold. In this work the semantic meaning of a part is assumed to be based on the local convexity or concavity.

Considering the aforementioned approaches, two main questions arise. First, what are the meaningful parts? Second, how distinctive must the part representation be? We start with the second question. As it can be noted in the aforementioned methods, parts are just a composition of same-size patches. These patches by themselves are not distinctive enough. In other words, these patches can be found in multiple parts in an object. Because we consider a specific scale for the patches, we might not be able to recognize parts distinctively across the object, as locally the patches are similar. Thus, a scale invariant representation of patches needs to be addressed. In addition, the relation between the patches that compose one part should be also distinctive enough for capturing intra-parts variability. Considering 3D objects, this relation is based on surfaces and curves that compose a specific object part. For the first question, the semantic meaning of a part needs to be defined. Most often, parts are defined based on performing a certain task such as manipulation or recognition. Also, part decomposition can be based on grasping experiments. In other words, segments of objects that can be grasped can form a semantically meaningful part. One source of inspiration to learn parts from grasping is humans, where infants learn to grasp objects before those objects are visually familiar [12].

Motivated by the open problems mentioned above, we propose an object segmentation method on RGB-D data. We address the problem of object decomposition into semantically meaningful parts. We consider a compositional bottom-up approach for forming object parts through a configuration of patches. We focus on a scale-invariant and distinctive patch representation. As noted above, the composition is guided by grasping experience. It should be noted that this compositional learning of meaningful parts for grasping would benefit both recognition and grasping. More precisely, from the learned object segments that form a parts, we can segment objects that seem visually different but share the same parts. The part generalization and compositional perspective of the proposed framework would then facilitate the generalization of the grasping paradigm to novel objects. Though, grasping generalization is not the focus of this work and it is regarded as future work.

We give first an overview of our bottom-up compositional method in Section II. In Section III, we show how grasping experience can be used in collecting statistics for forming meaningful object parts. Section IV explains the probabilistic method for transferring the learned statistics from grasping to

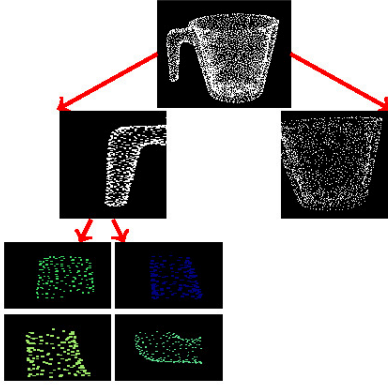


Fig. 1. Object parts and patches

form parts in a novel object. We report the evaluation results in Section V, and finally we close the discussion with a brief conclusion and an overview on future work in Section VI.

II. BOTTOM-UP OBJECT COMPOSITION

In this section, we explain our bottom-up compositional model for forming parts in an object. We consider robotics grasping experience for forming the parts.

The input to the system is an RGB-D point cloud where we only make use of the depth data. We want to form object parts from the input object points. As shown in Figure 1, an object is considered a configuration of parts such as a *cylindrical body* and a *handle* for the presented *mug*. Unlike representing a part by a combination of voxels at a specific scale, we represent it as a configuration of sub-parts that are more distinctive than voxels. As can be seen in Figure 1, a object part consists of some regions which have different surface shapes such as concave, convex, flat, hyperbolic. In each specific region, the surface normal changes smoothly. Hence, a region consists of a patches that seem locally flat but considering pairs of adjacent patches, substantial change in normal vectors can be observed.

Hence, we want to have a compositional representation of object in terms of its *parts*, *regions* that compose a part and *patches* that compose a region. We want to have distinctive parts in an object. Hence, we enforce that to the regions and patches that compose a part. We did not considered composition of object points to patches in our compositional representation because the points that form a patch are not distinctive by themselves. The patches are in the lowest level of compositional model. The patches by themselves are not distinctive enough but considering pairs of them, we can compose a *region* that can be found distinctively in a part. It should be noted that, in this work, we did not consider surface estimation in terms of concavity, convexity or flatness in a specific region, and we only considered the relation between patches to compose a region and inherently a part.

A. From Points to Patches

As mentioned in Section II, in the lowest level of our object compositional representation, we have locally flat

patches. So, our first goal is to compose object input points into locally flat patches such that the surface shape considering each two adjacent patches should be non-flat. An obvious way to form these patches is to merge points that have parallel normal vectors. However, estimation of normals based on small neighborhoods around points is very noisy in typical depth data. To overcome this problem, we should consider larger regions. To this end, we form supervoxels on the input RGB-D point cloud. We used the supervoxel algorithm [13] (available in the Point Cloud Library¹). This method uses K-Means, starting with evenly-distributed cluster seeds over an object. Then we merge the supervoxels if their mean normal vectors are close to parallel. In Figure 2, the supervoxels and merged supervoxels for some objects are shown. We then use the obtained flat patches considering their relation between adjacent patches and grasp experience to form object parts.

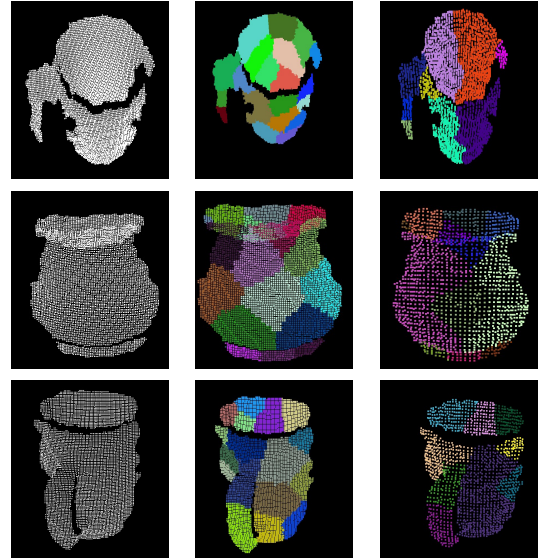


Fig. 2. From left to right, original RGB-D pointcloud, Supervoxels and flat patches

III. TRAINING A PART MODEL

Given the flat patches described in Section II, we want to form parts based on grasping experience. To this end, we decompose an object into patches. We then perform kinesthetic grasp teaching on the object as shown in Figure 3. From the grasping contact points, we can identify the patches that are grasped. In Figure 4 an example of extracting patches based on kinesthetic grasp teaching is shown. This gives us information about the patches that can be combined into a part. Therefore, we collect statistics about co-occurrence of two adjacent patches, and we use these statistics for finding parts in a novel object.

A. Patch Representation

We need to have a descriptor to characterize a patch. As mentioned in Section II, patches are assumed to form

¹<http://pointclouds.org/>

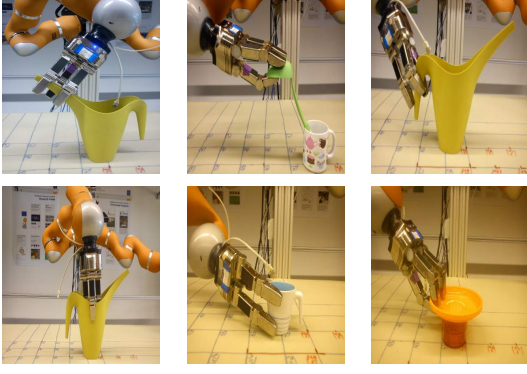


Fig. 3. Kinesthetic grasp teaching for collecting patches that form a region and hence a part.

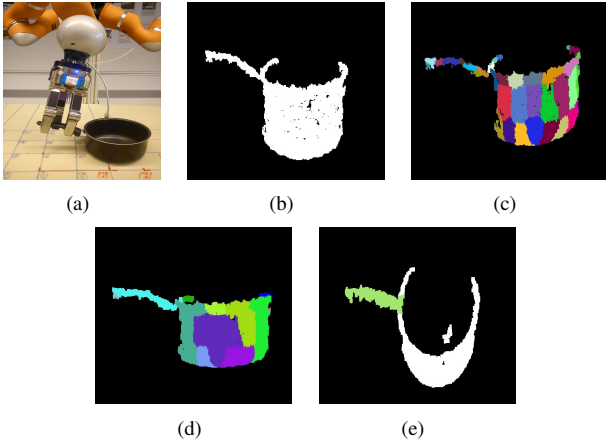


Fig. 4. Figure 4(a) shows the kinesthetic grasp teaching on a *pot*. Object decomposition into patches is shown in Figure 4(b). The patches contacted by grasping are shown in Figure 4(c).

regions based on the relation with neighbors. We exploit this assumption to construct a descriptor for a patch. In this way, we encode the relation between each patch and either of its neighbors by the angular relation between their normal vectors. Since the patches are locally flat, we only consider their mean normal vector as the representative normal vector. We then quantize these angular relations into a histogram, which becomes our patch descriptor.

B. Clustering Patches

For part identification, we want to find the co-occurrences of the adjacent patches from a dataset of objects. For efficiency and compactness, we form a codebook from the extracted patches by clustering the patches based on their aforementioned descriptors. We use hierarchical agglomerative clustering for this purpose. In this procedure, each patch initially constitutes its own cluster; we then merge clusters incrementally if the measured distance between them is lower than a certain threshold.

There are two choices to be made here, which are the distance measure and the merging threshold. As we represent the patches by a histogram, we use the *Chi-Square* distance



Fig. 5. One sample cluster in IKEA kitchen objects. From left to right, three different instances of the cluster, shown in the context of the whole object. For better visualization, the images are shown.

as the distance measure. For the distance merging threshold, we make use of the training data as follows. We decomposed training objects into patches and observed which patches form a part together. We consider all pairs of adjacent parts in an object. For each pair, we then measure the distance between the patches along the boundary between the two adjacent parts. Finally, we pick the minimum distance among all the objects' parts as the merging threshold. We then form a codebook of the obtained clusters based on their centers.

Figure 5 shows one cluster of our own collected set of IKEA objects. The patches inside each cluster are projected to their corresponding objects. For better visualization, the corresponding patch and its respective object are shown. As it can be seen, the clusters are formed based on the surface shape relation with local neighborhood.

C. Co-occurrence Statistics

We want to collect statistics of co-occurrences of adjacent patches that form the same part. We have clusters of patches from an object database described in Section III-B. Based on these clusters, we can estimate the co-occurrences of adjacent patches. Hence, for each two adjacent patches that belong to the same part (x_1, x_2) , we match them to the codebook and find all possible matching clusters for either of them, (h_1, h_2) , where h_1 contains all the matched clusters for patch x_1 and likewise h_2 for x_2 . We then increment the table entry for each two pairs in h_1 and h_2 . After this, we normalize the table entries to get the probabilities. Therefore, we have a co-occurrence table based on clusters where each cluster represents a patch type and we use that for forming parts in novel objects.

IV. PART INFERENCE IN NOVEL OBJECTS

We want to form parts in novel objects based on learned co-occurrence statistics. As a first step, we decompose the novel object into patches. Starting from one patch, we estimate the co-occurrence probabilities between each patch and its neighbors. We then decide to merge the patch with its neighbor of maximum probability. We perform this procedure in multiple iterations. At the first iteration, patches merge together and form regions. Next, regions merge together. We stop when no more merges are possible. At the end, parts based on learned statistics have been identified.

A. From Patches to Regions

As mentioned earlier, we want to estimate the co-occurrence between each patch and its neighbors based on

the learned codebook. Then, we grow the region incrementally to form a part.

Let $Y(x_1, x_2)$ denote the predicate asserting that patches x_1 and x_2 belong to the same region. Then, $p(Y(x_1, x_2)|x_1, x_2)$, or simply $p(Y|x_1, x_2)$ for short, denotes the probability that x_1 and x_2 belong to the same region.

Given the object patches, we start from a random patch x_1 and merge it with the patch x_l from its neighborhood $N(x_1)$ that is most probable to form a region with x_1 , based on the learned codebook:

$$x_l = \operatorname{argmax}_{x \in N(x_1)} p(Y|x_1, x) \quad (1)$$

Assuming an uninformative prior $p(Y)$, we instead maximize $p(x_1, x|Y) \propto p(Y|x_1, x)$. We make use of our codebook C to compute the probability of x_1 and x forming a region:

$$p(x_1, x|Y) = \sum_{c \in C} p(x, x_1|c, Y) p(c|Y) \quad (2)$$

$$= \sum_{c \in C} p(x, x_1|c) p(c|Y) \quad (3)$$

Equation 3 follows from the conditional independence of neighboring patches forming a region given their common codebook vector, since we encoded the co-occurrences of neighboring patches that form a region in the codebook. Inside the sum, the first factor factorizes as

$$p(x, x_1|c) = p(x|x_1, c) p(x_1|c). \quad (4)$$

The second factor of Eqn. 4 indicates whether a patch x_1 can be matched to cluster c , while the first factor indicates whether a patch x can then be merged with x_1 . Based on the learned co-occurrence table, we extract a set of patches that can co-occur with cluster c , and check whether patch x can be matched to them, that is,

$$p(x|x_1, c) = \sum_{h \in H} p(x|x_1, c, h) p(h|x_1, c) \quad (5)$$

$$= \sum_{h \in H} p(x|h) p(h|c) \quad (6)$$

where H is a set of clusters that can co-occur with cluster c . Equation 6 follows from the facts that x is conditionally independent of c and x_1 given h , and that h is conditionally independent of x_1 given c .

We compute $p(h|c) = \frac{p(h, c)}{p(c)}$ by taking $p(h, c)$ from our co-occurrence table and assuming $p(c)$ is uniformly distributed. Finally, we assume that the conditional probability of a cluster c given Y is uniform which substitutes the last term in Eqn. 3.

After calculating potential matches x for all the neighboring patches to x_1 in this fashion, we merge those that maximize the probability $p(Y|x_1, x)$ of forming a part (1).

B. From Regions to Parts

From the above procedure, we obtain a collection of regions. To merge regions incrementally to compose parts, we follow a similar procedure as above, starting from a random region and merging neighboring regions. We start with a region r_1 as depicted in Figure 6. We want to merge

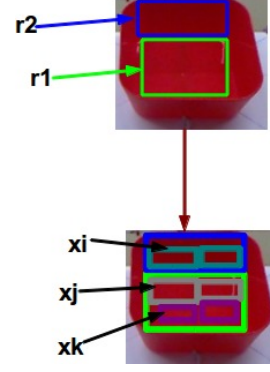


Fig. 6. Two adjacent regions in the *Red Container* object. Starting from region r_1 , x_j denotes its boundary patches. Patches adjacent to x_j inside r_1 are denoted x_k , and x_i inside r_2 . For visualization purposes a region has rectangular shape, on general a region can have any shape

region r_1 to a region which is most probable to form a part with r_1 , i.e.,

$$r_l = \operatorname{argmax}_{r \in N(r_1)} p(Y|r_1, r), \quad (7)$$

where $p(Y|r_1, r)$ denotes the probability that r_1 and r belong to the same part. Assuming an uninformative prior $p(Y)$, we instead maximize $p(r_1, r|Y) \propto p(Y|r_1, r)$.

Any two adjacent regions contain adjacent component patches along their common boundary. We marginalize over these boundary patches to calculate $p(r_1, r|Y)$. As depicted in Figure 6, region r_1 has patches x_j that are adjacent to region r . We would like to form a contiguous region by enforcing the co-occurrence of boundary patches with their neighbors in r_1 and r :

$$p(r_1, r|Y) = \sum_{x_j} p(r_1, r|x_j, Y) p(x_j|Y) \quad (8)$$

The first term in the summation is assumed to be independent of Y given x_j

$$p(r_1, r|Y) = \sum_{x_j} p(r_1, r|x_j) p(x_j|Y) \quad (9)$$

We then factorize the first term in Eqn. 9 as follows,

$$p(r_1, r|x_j) = \sum_{x_j} p(r|r_1, x_j) p(r_1|x_j) p(x_j|Y) \quad (10)$$

We consider $p(r_1|x_j)$, the conditional probability of a region given a boundary patch x_j , to be the conditional probability of the individual patches in that region that are adjacent to x_j :

$$p(r_1|x_j) = \prod_{\{x|x \in r_1 \wedge x \in N(x_j)\}} p(x|x_j) \quad (11)$$

where $N(x_j)$ stands for the neighbors of patch x_j . To estimate the conditional probability of adjacent patches, we marginalize over codebook clusters c , similarly to Eqn. 2:

$$p(x|x_j) = \sum_{c \in C} p(x|x_j, c) p(c|x_j) \quad (12)$$

The conditional probability of a cluster c given patch x_j is not given, hence we calculate it as,

$$p(c|x_j) = \frac{p(x_j|c)p(c)}{p(x_j)} \quad (13)$$

where $p(x_j)$ is considered uniform based on all the patches in an object, and $p(c)$ is also uniform based on the number of clusters. Therefore, $p(x_j) = \frac{1}{N_p}$ where N_p stands for the number of the patches and $p(c) = \frac{1}{N_c}$ where N_c stands for the number of clusters. In the same way, we calculate the conditional probability for region r based on those patches in r that are adjacent to patch x_j :

$$p(r|r_1, x_j) = \prod_{\{x|x \in r \wedge x \in N(x_j)\}} p(x|r_1, x_j) \quad (14)$$

We consider that $p(x|r_1, x_j)$, the conditional probability of a patch x in region r given region r_1 and its neighboring patch in r_1 , x_j to be independent of r_1 ,

$$p(r|r_1, x_j) = \prod_{\{x|x \in r \wedge x \in N(x_j)\}} p(x|x_j) \quad (15)$$

To calculate the conditional probability of adjacent patches, we marginalize over the codebook clusters c ,

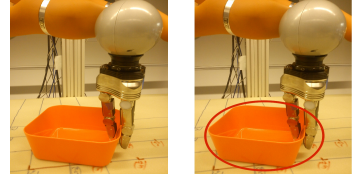
$$p(x|x_j) = \sum_{c \in C} p(x|x_j, c)p(c|x_j) \quad (16)$$

We then substitute Equations 11 and 16 into Equation 10. Furthermore, we assume that the conditional probability of a patch x_j given Y is uniform, eliminating the last factor inside the sum of Equation 10. From this, we obtain the potential of merging two adjacent regions r_1 , r to form a part. We merge those that maximize the probability $p(Y|r_1, r)$ of forming a part.

V. EXPERIMENTS

We evaluated our part compositional method on our own collected IKEA kitchen objects as well as the RGB-D Washington object database [14]. For the IKEA dataset, we performed kinesthetic grasp teaching to obtain the parts. We considered semantically meaningful grasps that are associated with one and only one part of an object. Also, we tried to cover the entire area of a specific object part. For example, a single rim grasp as shown in Figure 7 cannot cover the entire area of the grasped part. Hence, we did multiple rim grasps around the object rim to cover the part completely. Our robot is a two KUKA 7-DoF Light-Weight Robot 4+ with servo-electric 3-Finger Schunk SDH-2 dexterous hands. There is a Kinect mounted in front of the robot for capturing the RGB-D data.

To show the applicability of the proposed part segmentation method, we evaluated it on the Washington RGB-D dataset. However this dataset is very rich for RGB-D data, but the objects in this dataset do not have a complex structure. Therefore, we considered the objects which have more complex structure. For that dataset, we labeled the graspable parts for three categories of mugs. We sampled 30% of the data from each instance category for training and 10% of them as the models for collecting statistics.



(a) Grasp teaching (b) Semantic part related to the grasp

Fig. 7. Example of kinesthetic grasp teaching 7(a). As shown, the grasp covers only one portion of the semantic object part 7(b).

Category	Proposed method	LCCP
IKEA kitchen objects	83%	52%
Washington RGB-D mugs	76%	56%

TABLE I
OVERLAP ACCURACY COMPARISON

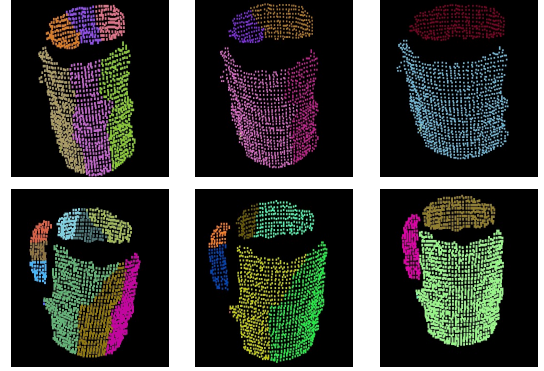


Fig. 8. Example of detected object parts in multiple iterations. From left to right, patches to regions to parts. Rows show different objects from Washington RGB-D mugs.

We divided the dataset into training and test sets. We used a random sample from the training set as the models to collect statistics for co-occurrences of patches, and the rest for clustering. For testing, we have the ground truth of object parts. We use the proposed compositional method to obtain the parts. Then, we calculate the overlap between the parts obtained by the method and the ground truth. We estimated the overlap between each ground-truth part and each estimated part, and we took the average overlap for each object. The overlap is calculated as

$$\text{overlap}(x, y) = \frac{x \cap y}{x \cup y}, \quad (17)$$

where x is a part obtained by the method, and y is the ground truth part. Table I gives overlap accuracy comparison for our proposed method and for LCCP [11] on the collected IKEA objects, as well as for the three categories of the Washington RGB-D dataset. Figure 8 shows some of the detected object parts in multiple iterations of the algorithm. Also some examples of extracted parts in the IKEA dataset are shown in figure 9.

We also evaluated our method on novel objects that

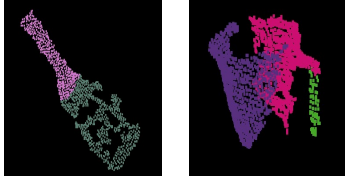


Fig. 9. Example of detected parts in IKEA kitchen dataset.

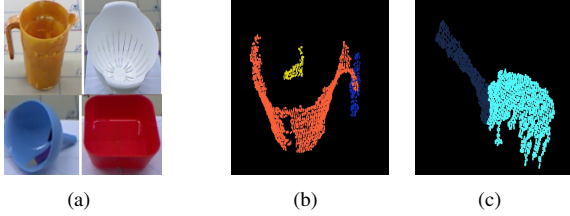


Fig. 10. Example of part segmentation on two object in *IKEA* objects (Figures 10(b), 10(c)) and the corresponding training models Figure 10(a). For better visualization the corresponding images for the models are shown.

share only some patches and we obtained promising results. Figure 10 shows two examples of the identified parts in the objects and the corresponding training models. We tested it with two completely different subset of objects in our own collected *IKEA* kitchen objects. The generalization is due to the compositionality nature of our approach that considers a part as a composition of regions and hence patches. It can be observed that in spite of the noise in RGB-D data, the method gives us promising results. We obtained overlapping accuracy of 72% in our *IKEA* objects. We also evaluated it on Washington dataset where we trained the models of three types of mugs and we evaluated it on another type of mug. The result for this experiment is also shown in Figure 11. We obtained 79% overlapping in this experiment. The evaluation on two different sets of mugs in Washington RGB-D database is noted in Table II. In the table, we can observe the overlapping accuracy of training on one set of mugs verse the other as well as evaluation on the same type of mugs.

As can be seen in Figures 12(a) and 12(b), the detected part for the *handle* is not precise. The main source of error lies in the poorly-estimated supervoxels, which propagate to imprecisely estimated patches. An example of this mis-

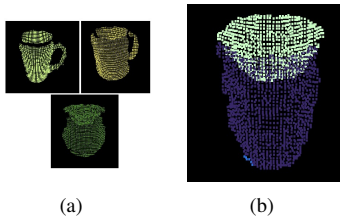


Fig. 11. Example of part segmentation on a mug in Washington RGB-D dataset (Figure 11(b)). The corresponding training models are shown in Figure 11(a).

Object Category		
	72%	67%
	72%	69%

TABLE II

OVERLAP ACCURACY COMPARISON BETWEEN TWO SETS OF MUGS. HORIZONTAL AND VERTICAL AXIS STAND FOR TRAINING AND TEST CATEGORIES.

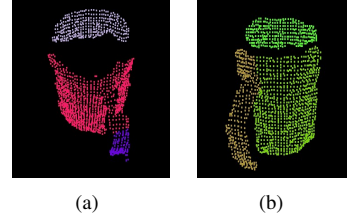


Fig. 12. Examples of poorly-estimated parts.

estimation is illustrated in Figure 13. Also, since we are using statistical learning, for parts which are not often seen, we cannot obtain a high probability and hence they will not be merged together.

VI. CONCLUSIONS

We presented a novel object-part segmentation approach. Our objective is to produce semantically-meaningful parts, focusing in this work on graspable parts. From grasping experiments, we collect statistics for forming parts based on grasped segments of an object. We utilize these segments as patches to compose object parts. We proposed a compositional method for forming parts based on scale-invariant though locally distinctive object patches. The evaluation results show promising results on novel objects.

As the patch descriptor, we considered semi-local relations between adjacent patches. This descriptor proved to fulfill our requirement for the moment. However, as future work, we will considering a stronger descriptor that jointly encodes local, relational information and surface shape.

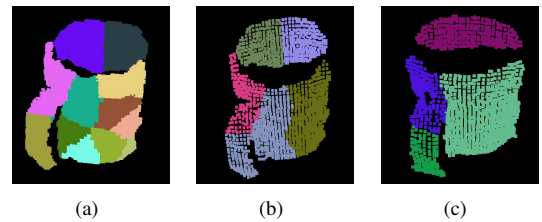


Fig. 13. Example of a poorly-estimated patch due to inaccurate supervoxel segmentation.

APPENDIX

ACKNOWLEDGMENT

The research leading to these results has received funding from the European Community's Seventh Framework Programme FP7/2013-2016 (Specific Programme Cooperation, Theme 3, Information and Communication Technologies) under grant agreement no. 600918, PaCMan.

REFERENCES

- [1] P. F. Felzenszwalb and D. P. Huttenlocher, "Pictorial structures for object recognition," *Int. J. Comput. Vision*, vol. 61, no. 1, pp. 55–79, Jan. 2005.
- [2] M. C. Burl, M. Weber, and P. Perona, "A probabilistic approach to object recognition using local photometry and global geometry," 1998.
- [3] M. Burl, T. K. Leung, and P. Perona, "Face localization via shape statistics," 1995.
- [4] B. Pepik, M. Stark, P. Gehler, and B. Schiele, "Teaching 3d geometry to deformable part models," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2012.
- [5] P. F. Felzenszwalb, R. B. Girshick, D. A. McAllester, and D. Ramanan, "Object detection with discriminatively trained part-based models," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 9, pp. 1627–1645, 2010.
- [6] X. Ren and J. Malik, "Learning a classification model for segmentation," in *Proceedings of the Ninth IEEE International Conference on Computer Vision - Volume 2*, ser. ICCV '03. IEEE Computer Society, 2003, pp. 10–.
- [7] R. Toldo, U. Castellani, and A. Fusiello, "A bag of words approach for 3d object categorization," in *Proceedings of the 4th International Conference on Computer Vision/Computer Graphics CollaborationTechniques*, ser. MIRAGE '09. Springer-Verlag, 2009, pp. 116–127.
- [8] S. M. A. Eslami and C. Williams, "A generative model for parts-based object segmentation," in *NIPS*, 2012, pp. 100–107.
- [9] S. Fidler and A. Leonardis, "Towards scalable representations of object categories: Learning a hierarchy of parts," in *CVPR*, 2007.
- [10] P. K. Nathan Silberman, Derek Hoiem and R. Fergus, "Indoor segmentation and support inference from rgb-d images," in *ECCV*, 2012.
- [11] S. Christoph Stein, M. Schoeler, J. Papon, and F. Worgotter, "Object partitioning using local convexity," June 2014.
- [12] R. K. Clifton, D. W. Muir, D. H. Ashmead, and M. G. Clarkson, "Is visually guided reaching in early infancy a myth?" *Child Development*, vol. 64, pp. 1099–1110, 1993.
- [13] J. Papon, A. Abramov, M. Schoeler, and F. Wrgtter, "Voxel cloud connectivity segmentation - supervoxels for point clouds," in *IEEE Conference on Computer Vision and Pattern Recognition CVPR*, June 2013.
- [14] K. Lai, L. Bo, X. Ren, and D. Fox, "A large-scale hierarchical multi-view rgb-d object dataset," in *IEEE International Conference on on Robotics and Automation*, 2011.

A.2 Article: Learning Part-Based 3D Compositional Object Representations

Authors Safoura Rezapour Lakani, Antonio J. Rodriguez-Sanchez and Justus Piater

Info Submitted to IEEE Computer Vision and Pattern Recognition (CVPR), 2015.

Abstract This paper presents a novel approach to parts-based object representation from depth images. We propose a bottom-up compositional model for representing object classes. Our model uses a probabilistic approach to build object representations starting from small patches that are successively built up into regions, and finally meaningful (possibly graspable) parts. Parts represented this way are the main representatives of the identity of an object class. We have evaluated our method at parts recognition and object categorization, outperforming competing methods

Relation with the deliverable The paper presents a probabilistic model for learning object representations in a hierarchical manner. This representation is useful for all the tasks, as it enables forming objects by graspability (Task 2.3), it allows integration of non-visual features (Task 2.1), and it can be used for a refined scene description (Task 2.4)

Attachment (following pages until next annex)

Learning Part-Based 3D Compositional Object Representations

Anonymous CVPR submission

Paper ID 2416

Abstract

This paper presents a novel approach to parts-based object representation from depth images. We propose a bottom-up compositional model for representing object classes. Our model uses a probabilistic approach to build object representations starting from small patches that are successively built up into regions, and finally meaningful (possibly graspable) parts. Parts represented this way are the main representatives of the identity of an object class. We have evaluated our method at parts recognition and object categorization, outperforming competing methods.

1. Introduction

Object recognition is of interest for applications in robotics, scene understanding and tracking, just to name a few. In these applications, the generalization of recognition to novel previously unseen objects at the category or instance levels are of the great importance. In the same level of importance for these applications is robustness in the presence of clutter and occlusion. Parts-based object representation methods can help overcome some problems object recognition has to face [5]. In these methods, objects are represented by a configuration of parts where certain parts are to be shared between different objects or even object classes. This shareability allows for generalization of recognition for certain classes of objects. Representing objects by the configuration of its parts allow us to propose a probabilistic object representation which can be useful for overcoming occlusion scenarios

The most important challenge parts-based methods have to face is characterizing object parts in a learning stage and establishing the relation between those parts. Parts are either labeled from training examples [14, 7, 18] or are obtained in an unsupervised manner by making the assumption that certain specific geometrical constraints - such as specific amounts of local convexity or concavity [3] - hold. The parts are mostly characterized by the sub-parts that compose them in terms of either fixed size patches [8], supervoxels [3] or a distribution of visual features [17]. For

classification, these methods use either Graphical models, such as belief propagation (imposing a relationship on the parts and their constituting sub-parts [15]) or bag-of-words approaches applied on the extracted parts [11, 17].

Compositional representations methods have shown to be of interest on 2D contours and edges based systems [6, 20, 12]. We propose that compositional representations can also be useful when dealing with 3D object representations. We will present here a novel 3D parts-based representation that 1) follows a compositional representation rather than a feature-based representation and 2) uses parts as the main distinctive characteristics that differentiates objects. For the former, this compositionality gives us the generalization capabilities of recognizing parts and their components. Our compositional model allows us not only to build a model of an object in a bottom-up fashion starting from patches, but also - since the representations are there - provides the possibility of extracting more abstract properties of the objects than its features, such as its component parts. For the latter, parts are composed of patches, which by themselves have little discriminative power, but they will be combined into a representation that constitutes a discriminative object part. Furthermore, we propose a scale-invariant representation of patches that allows us to form parts with different geometrical shapes and scales.

In Section 2 we give an overview of our compositional bottom-up system. Then, in Section 3 we explain our learning pipeline. Section 4 describes the probabilistic inference based on the compositional method to recognize object parts and categories. We report on the experiments in Section 5. Finally, we summarize the discussion with a brief conclusion and an overview of possible future work in Section 6.

2. Compositional Representation of Objects

We introduce here a bottom-up compositional model to represent objects. The input data is an RGB-D point cloud where we only use depth information (thus, we do not consider color information). Objects in our representation are composed of parts which subsequently contain regions that are obtained from the input data points through their compo-

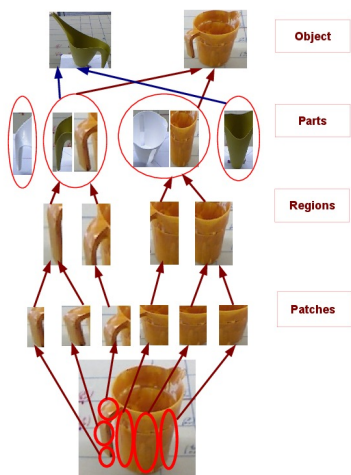


Figure 1. A general overview of the compositional method.

nent patches. The reason why we are considering a bottom-up approach is because it provides us with a more robust estimation in the presence of noise and occlusion rather than, otherwise considering a top-down approach.

As mentioned earlier, our objects are represented based on their parts configuration. We then need a representation for the parts. Our parts configuration is a configuration of the regions that constitute them. In order to differentiate between object classes, we must use parts information since at the region level, regions are not discriminative enough. Regions can be grouped into concave, convex, hyperbolic and flat. We define the regions as being smoothed surfaces where the surface normals change smoothly. Regions are further composed of patches which are defined as small local flat areas, where the changes of surface normals is negligible. On the other hand, there exists a variation of surface normals between neighboring patches.

Considering this compositional model, we can then infer that an object is composed of patches at the lowest level, which subsequently form regions, which are the components of parts. Figure 2 shows an overview of the architecture of our approach. How each of these representational levels are constructed is explained in detail in the following sections.

3. Learning the Compositional Object Model

We explain here the training procedure used for learning our compositional object models. Based on our bottom-up composition, we have three levels in the model which form *patches*, *regions*, *parts* and *objects* (Figure 2). Details of this procedure are given in section 4. We summarize next how we perform the training and representation at each of these levels.

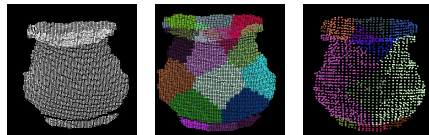


Figure 2. Decomposition of an object into patches. From left to right, original object pointcloud, its supervoxels and patches.

3.1. Training Patch Models

As shown in Figure 2, patches form the lowest level of our compositional model. Patches are combined into regions that in turn form parts. We define patches as local surfaces that can be approximated by a plane, but have a relationship to neighboring patches where there is a substantial change in surface normals. When considered individually, patches carry barely any information. The importance of patches lies here in the relationships to neighboring patches. We now explain how patches are obtained from depth images and how they are represented such that they build up to discriminative parts.

From Points to Patches We first transform the 3D point clouds into local planar surfaces. We do this by merging adjacent points whose normal vectors are parallel. Since depth data is quite noisy, just performing this operation yields quite unreliable approximations. To overcome this problem, we require larger regions that allow robust surface normals to be estimated. We use RGB-D supervoxels, computed with the supervoxel algorithm [13] (available in the Point Cloud Library¹). This method uses K-Means, starting with evenly-distributed cluster seeds over an object pointcloud. We then consider only supervoxels for forming patches. We merge adjacent supervoxels if their mean normal vectors are close to parallel. Figure 2 shows decomposition of an object into patches starting from its supervoxels. It should be noted that even though the supervoxels compose patches, they are not considered in our compositional model.

Patch Representation We represent a patch in terms of its relation to its neighbors. For the representation we take two criteria into account, the spatial relation between adjacent patches and the surface curvature. For the first criterion, we consider a reference coordinate system on the patch of interest and represent the spatial relation to its neighbors with respect to that coordinate system. The reference system is the axis of symmetry corresponding to the patch. It should be noted that, since we only consider a smooth object region, we have robust estimation of the symmetrical axis. To calculate this axis, we compute the inertia matrix [19]. We then place each patch's neighbor in a 3D histogram indexed by polar coordinates based on the calculated axis. For the second criterion, we consider the curvature on

¹<http://pointclouds.org/>

the separation between neighboring patches and set it into the relative polar location calculated as before.

3.2. Training Part Models

In our compositional object representation, object parts are composed of regions of adjacent patches. In order to form parts, the objective is to find out which patch types co-occur together to form an object part. With this idea in mind, we collect statistical co-occurrences of certain patch types that compose a part. The input to the system is an object dataset, where only about 10% of them are labeled. The idea is to use the unlabeled ones to train the different patch types and utilize the labeled objects to collect the statistical data about the co-occurrence of certain patch types in their labeled parts.

For finding the different patch types, we decompose our training data into patches which are then grouped by hierarchical agglomerative clustering [4]. First, each patch is considered a cluster by itself. Clusters are merged if their average distance is less than a certain threshold. This threshold is obtained from the parts corresponding to the labeled data. We obtain the minimum distance between the patches which are adjacent as well as belonging to different parts.

Then, we create a codebook from the clusters and use that for collecting the patch co-occurrence statistics. In order to do that, we first decompose our labeled data into patches. We then consider each two adjacent patches that compose a part and match each of them to the codebook. This gives us the hypotheses regarding which clusters match each of the patches. After that, we increment the frequency of co-occurrence of each of the two matching clusters. Hence our statistical data is based on the co-occurrence of the clusters which are implicitly related to object patches. Next, we make a codebook of parts based on the obtained clusters. Finally, we follow the probabilistic method described in Section 4.1 to compose parts based on the learned patch codebooks.

3.3. Training Object Models

Objects are at the highest level of our compositional model. They are composed of a configuration of parts whose starting point were locally planar surfaces, or patches. At this point, we need to represent which object parts co-occur together to form a certain class of objects. Thus, we will decompose objects into parts and recognize part types. We then follow the idea presented in section 3.2 to collect statistical data, but this time regarding adjacent parts.

The common way to recognize different part types is to cluster them. Since, parts are composed of patches, we represent them as a histogram of patches. As described in section 3.2, we obtained a codebook of different patch types. Hence, the parts are encoded as a histogram consisting of

how frequent they appear on certain codebook entries. We then do a clustering on the aforementioned histogram to obtain specific part types. A codebook is then build from the clusters of parts. This second codebook gives us only information about the occurrence of a certain part in an object. In addition to this information, we will also need the statistics on the co-occurrence of parts. To achieve this, we will consider the composing patches that connects two adjacent parts. Just considering these selected part patches provides a higher degree of flexibility that allows us to achieve a better generalization capabilities (than otherwise considering all patches inside both related parts). We will collect on the statistical co-occurrences of these selected patches corresponding to neighboring parts as described in detail in section 4.4.

4. Probabilistic Part and Object Recognition

In this section, we describe details regarding the inference in our compositional object model that we briefly explained in Section 3.

4.1. Part Recognition

As described in Section 3.2, object parts are composed of patches. We collected the statistics on adjacent patches that compose a part. In order to form a part in a novel object, we consider adjacent patches and we merge them based on the learned patch co-occurrence statistics. This is an iterative procedure, in which we first merge adjacent patches to obtain regions. Then, these regions can be further merged to form an object part. We explain next the probabilistic formulation for composing regions from patches and composing parts from regions.

4.2. From Patches to Regions

In this section, we provide the mathematical formulation behind how a part region is represented from patches that are adjacent. Let z_1 and z_2 indicate two adjacent patches then $p(Y|z_1, z_2)$ is the probability that asserts whether the two patches can form a part region.

Given the object patches, we start from a random patch z_1 and we merge it with its neighbor z_l that is more likely to form a part region.

$$z_l = \operatorname{argmax}_{z \in N(z_1)} p(Y|z_1, z) \quad (1)$$

Assuming an uninformative prior $p(Y)$, we instead maximize $p(z_1, z|Y) \propto p(Y|z_1, z)$. We make use of our patch codebook C to compute the probability of z_1 and z forming

a region:

$$p(z_1, z|Y) = \sum_{c_1 \in C} p(z, z_1|c_1, Y)p(c_1|Y) \quad (2)$$

$$= \sum_{c_1 \in C} p(z, z_1|c_1)p(c_1|Y) \quad (3)$$

Equation 3 follows from the conditional independence of neighboring patches forming a region given their common codebook vector c_1 , since we previously encoded the co-occurrences of neighboring patches that form a region in the codebook. Inside the sum, the first factor factorizes as

$$p(z, z_1|c_1) = p(z|z_1, c_1)p(z_1|c_1). \quad (4)$$

The second factor of Eqn. 4 indicates whether a patch z_1 can be matched to cluster c_1 , while the first factor indicates whether a patch z can then be merged with z_1 . Based on the learned co-occurrence table, we extract a set of patches that can co-occur with cluster c_1 , and check whether patch z can be matched to them, that is,

$$p(z|z_1, c_1) = \sum_{c_2 \in C_2} p(z|z_1, c_1, c_2)p(c_2|z_1, c_1) \quad (5)$$

$$= \sum_{c_2 \in C_2} p(z|c_2)p(c_2|c_1) \quad (6)$$

where C_2 is a set of clusters that can co-occur with cluster c_1 . Equation 6 follows from the facts that x is conditionally independent of c_1 and z_1 given c_2 , and that c_2 is conditionally independent of z_1 given c_1 .

We compute $p(c_2|c_1) = \frac{p(c_2, c_1)}{p(c_1)}$ by taking $p(c_2, c_1)$ from our co-occurrence table and assuming $p(c_1)$ is uniformly distributed. Finally, we assume that the conditional probability of a cluster c_1 given Y is uniform which substitutes the last term in Eqn. 3.

After calculating potential matches z for all the neighboring patches to z_1 in this fashion, we merge those that maximize the probability $p(Y|z_1, z)$ of forming a part (1).

4.3. From Regions to Parts

We have now a collection of regions. To merge regions incrementally to compose parts, we follow a similar procedure as above, starting from a random region and merging neighboring regions. We start with a random region r_1 and we merge it to an adjacent region r which is most probable to form a part, i.e.,

$$r_l = \operatorname{argmax}_{r \in N(r_1)} p(Y|r_1, r), \quad (7)$$

where $p(Y|r_1, r)$ denotes the probability that r_1 and r belong to the same part. Assuming an uninformative prior $p(Y)$, we instead maximize $p(r_1, r|Y) \propto p(Y|r_1, r)$.

Any two adjacent regions contain adjacent component patches along their common boundary. We marginalize over these boundary patches to calculate $p(r_1, r|Y)$. Region r_1 has patches z_j that are adjacent to region r . We would like to form a contiguous region by enforcing the co-occurrence of boundary patches with their neighbors in r_1 and r :

$$p(r_1, r|Y) = \sum_{z_j} p(r_1, r|z_j, Y)p(z_j|Y) \quad (8)$$

The first term in the summation is assumed to be independent of Y given z_j

$$p(r_1, r|Y) = \sum_{z_j} p(r_1, r|z_j)p(z_j|Y) \quad (9)$$

We then factorize the first term in Eqn. 9 as follows,

$$p(r_1, r|z_j) = \sum_{r_1} p(r|r_1, z_j)p(r_1|z_j)p(z_j|Y) \quad (10)$$

We consider $p(r_1|z_j)$, the conditional probability of a region given a boundary patch z_j , to be the conditional probability of the individual patches in that region that are adjacent to z_j :

$$p(r_1|z_j) = \prod_{\{z|z \in r_1 \wedge z \in N(z_j)\}} p(z|z_j) \quad (11)$$

where $N(z_j)$ stands for the neighbors of patch z_j . To estimate the conditional probability of adjacent patches, we marginalize over the codebook clusters c , similarly to Eqn. 2:

$$p(z|z_j) = \sum_{c \in C} p(z|z_j, c)p(c|z_j) \quad (12)$$

The conditional probability of a cluster c given patch z_j is not given, hence we calculate it as,

$$p(c|z_j) = \frac{p(z_j|c)p(c)}{p(z_j)} \quad (13)$$

where $p(z_j)$ is considered uniform based on all the patches in an object, and $p(c)$ is also uniform based on the number of clusters. Therefore, $p(z_j) = \frac{1}{N_P}$ where N_P stands for the number of the patches and $p(c) = \frac{1}{N_C}$ where N_C stands for the number of clusters. In the same way, we calculate the conditional probability for region r based on those patches in r that are adjacent to patch z_j :

$$p(r|r_1, z_j) = \prod_{\{z|z \in r \wedge z \in N(z_j)\}} p(z|r_1, z_j) \quad (14)$$

We consider that $p(z|r_1, z_j)$, the conditional probability of a patch z in region r given region r_1 and its neighboring patch in r_1 , z_j to be independent of r_1 ,

$$p(r|r_1, z_j) = \prod_{\{z|z \in r \wedge z \in N(z_j)\}} p(z|z_j) \quad (15)$$

To calculate the conditional probability of adjacent patches, we marginalize over the codebook clusters c ,

$$p(z|z_j) = \sum_{c \in C} p(z|z_j, c)p(c|z_j) \quad (16)$$

We then substitute Equations 11 and 14 into Equation 10. Furthermore, we assume that the conditional probability of a patch z_j given Y is uniform, eliminating the last factor inside the sum of Equation 10. From this, we obtain the potential of merging two adjacent regions r_1, r to form a part. We merge those that maximize the probability $p(Y|r_1, r)$ of forming a part.

4.4. Object Recognition

Obtaining object parts as described in Section 4.1, we can further recognize its category based on our compositional model. To this aim, we make use of the patch codebook C and the parts codebook H . For the recognition task we are given a set of parts X and their configuration S in terms of the adjacency matrix. We want to recognize the category of the novel object t_X ,

$$t_X = \operatorname{argmax}_t p(t|X, S) \quad (17)$$

Due to considering a uniform prior for each object class t , we maximize then the following equation,

$$t_X = \operatorname{argmax}_t p(X, S|t) \quad (18)$$

We assume conditional independence for each object part and its configuration. Hence, we can re-write the conditional probability in Eqn 18 as following,

$$p(X, S|t) = \prod_{s_i \in X, s_i \in S} p(x_i, s_i|t) \quad (19)$$

Further we can marginalize over all parts that vector s_i gives us. In fact, those are the adjacent parts to x_i ,

$$p(x_i, s_i|t) = \sum_{x \in s_i} p(x_i, s_i|t, x)p(x|t) \quad (20)$$

We further factorize the first term in the summation,

$$p(x_i, s_i|t, x) = \frac{p(x_i, x|t)p(t|s_i)p(s_i)}{p(x|t)p(t)} \quad (21)$$

We assume that a configuration s_i is uniformly distributed for all the object classes, hence $p(t|s_i) = \frac{1}{N_T}$. Also, as we presumed earlier, the prior probability of each object class is also uniform. We further assume a uniform distribution for a specific configuration s_i over all the configuration N_S . These assumptions leave us with the following formula for Eqn 20

$$p(x_i, s_i|t) = \sum_{x \in s_i} \frac{p(x_i, x|t)}{N_S} \quad (22)$$

In order to compute the conditional probability of the adjacent patches x_i and x , we make use of their border patches. Patches z_j connects x_i to x , hence we marginalize over these patches,

$$p(x_i, x|t) = \sum_{z_j \in x_i, z_j \in N_x} p(x_i, x|t, z_j)p(z_j|t) \quad (23)$$

We can then factorize the first term in the summation of Eqn 23,

$$p(x_i, x|t) = \sum_{z_j \in x_i, z_j \in N_x} p(x|t, z_j, x_i)p(x_i|t, z_j)p(z_j|t) \quad (24)$$

We also consider that the conditional probability of the part x_i is independent of its own patch z_j given that object class t . This assumption is made because we compute this conditional probability based on the composed patches of x_i . Therefore, we obtain the following formulation for computing the conditional probability of the two adjacent part,

$$p(x_i, x|t) = \sum_{z_j \in x_i, z_j \in N_x} p(x|t, z_j, x_i)p(x_i|t)p(z_j|t) \quad (25)$$

To compute the conditional probability of the part x which is adjacent to part x_i , we consider its patches that are neighbor to z_j and we compute them independently,

$$p(x|t, z_j, x_i) = \sum_{z_k \in x, z_k \in N(z_j)} p(x|t, z_j, x_i, z_k)p(z_k|t, z_j, x_i) \quad (26)$$

where z_k denotes the adjacent patches to z_j in x . We consider that the conditional probability of the patch z_k is independent of the part x_i when its adjacent patch is given. Furthermore, we also consider that when the conditional probability of patch z_k which belongs to x is given, conditional probability of part x is only dependent to the object class t ,

$$p(x|t, z_j, x_i, z_k) = p(x|t) \quad (27)$$

We factorize the conditional probability of z_k in Eqn 26 as followings,

$$p(z_k|t, z_j) = \frac{p(z_k, z_j|t)}{p(z_j|t)} \quad (28)$$

To compute the conditional probability of two adjacent patches in Eqn 28, we marginalize over our patch codebook C ,

$$\begin{aligned} p(z_k, z_j|t) &= \sum_{c_1 \in C} p(z_k, z_j|t, c_1)p(c_1|t) \\ &= \sum_{c_1 \in C} p(z_k|z_j, t, c_1)p(z_j|t, c_1)p(c_1|t) \\ &= \sum_{c_1 \in C} p(z_k|z_j, t, c_1)p(z_j|c_1)p(c_1|t) \end{aligned} \quad (29)$$

Further, let's consider that patch z_j matches cluster c_1 , z_k must then be matched to the clusters that co-occur with c_1 . Hence, we marginalize over the clusters that co-occur with c_1

$$p(z_k|z_j, t, c_1) = \sum_{c_2 \in C_2} p(z_k|z_j, t, c_1, c_2)p(c_2|z_j, t, c_1)$$

where C_2 stands for the set of clusters that co-occur with c_1 . For the first term, we assume that when the co-occurrence codebook c_2 is given, conditional probability of z_k is only dependent to c_2 , which yields us,

$$p(z_k|z_j, t, c_1) = \sum_{c_2 \in C_2} p(z_k|c_2)p(c_2|z_j, t, c_1) \quad (30)$$

where the conditional probability of z_k given cluster c_2 indicates if it can be matched to c_2 . The second term can be factorized as followings,

$$p(c_2|z_j, t, c_1) = \frac{p(c_2, c_1|z_j, t)}{p(c_1|z_j, t)} \quad (31)$$

Since, we collected statistics on the co-occurrences of adjacent patches that belong to different part based on a certain class of objects, we consider that the nominator in Eqn 31 is independent of the patch z_j when the object class is given. That also holds for the denominator,

$$p(c_2|z_j, t, c_1) = \frac{p(c_2, c_1|t)}{p(c_1|t)} \quad (32)$$

Finally, we substitute the obtained formulas into Eqn 22,

$$p(x_i, s_i|t) = \frac{1}{N_S} p(x_i|t) \sum_x p(x|t) \sum_{z_j \in x_i, z_j \in N(x)} \sum_{z_k \in x, z_k \in N(z_j)} \sum_{c_1 \in C} \sum_{c_2 \in C_2} p(z_k|c_2)p(c_2, c_1|t)p(z_j|c_1)$$

To compute the conditional probability of a part given the object class, we marginalize over our part codebook H ,

$$p(x_i|t) = \sum_{h \in H} p(x_i|h)p(h|t) \quad (33)$$

where the first term indicates whether part x_i can be matched to cluster h . We compute the second term based on our collected statistics of occurrence of a certain part codebook in the object classes.

In practice, we considered two matters concerns about clustering and computing the conditional probabilities. Because we used agglomerative clustering to group patches and parts, we have clusters which contain very few number of samples or even one sample. That is due to the nature of this type of clustering. For those clusters, we allowed for

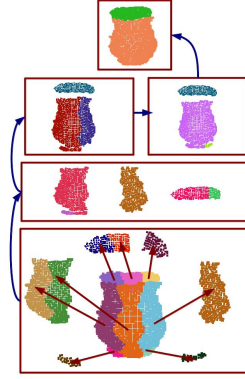


Figure 3. Compositional representation of forming object parts.

a small variation ϵ when making the codebook. For computing the conditional probability, since we made independent assumption between object parts and patches, we face the problem with the parts/patches that have zero probability. That propagates to the overall computation, and gives overall zero probability. To overcome this problem, we also considered a very small probability ϵ for those parts. However, this problem can also be solved by allowing missing parts/patches in the probabilistic model and using dynamic programming to solve it, we consider that for our future work.

5. Experimental Results

We evaluated our compositional model mainly on two aspects: part recognition on novel objects, and object classification. We evaluated our method in the RGB-D Washington dataset [10].

5.1. Part recognition in novel objects

Parts Evaluation is computed based on the overlap of the parts recognized by our method in comparison with the ground truth. The overlap score is computed as [9, 2],

$$\text{overlap}(x, y) = \frac{|x \cap y|}{|x \cup y|}, \quad (34)$$

where x is a part obtained using our method, and y is the ground truth part.

We divide the data into training data, test data and training models. The training data are unlabeled and used for learning the codebook of patches and parts as explained in Section 3, whereas the training models correspond to the labeled data. For the test data, we labeled the parts as our ground truth for evaluation.

Figure 3 shows the compositional capability of our method in a real case scenario.

In order to show the applicability of our part recognition method on the Washington dataset, we selected the objects which have more than one part. From two categories of



Figure 4. Example of a poorly-estimated part due to inaccurate supervoxel segmentation.

objects, we sampled a set of them for evaluation. We compared our method in this dataset with Locally Convex Connected Patches (*LCCP*) from [3]. The overlapping results as given in Table 1 which proves that our method outperforms the *LCCP* method.

Category	Proposed method	LCCP [3]
mugs	76%	58%
caps	75%	40%
staplers	88%	61%

Table 1. Overlap accuracy comparison on the Washington RGB-D dataset

In addition to the evaluating our method using the aforementioned overlap measure, we are also interested in showing the generalization capabilities of our method. With this goal in mind, we performed an experiment where our algorithm had to find parts that were already present in our part-based model, but in objects not seen previously. Table 2 shows the overlapping accuracy for novel part recognition in some objects of the RGB-D Washington dataset. Each row shows in its first column the object that was used to store the part representation in our compositional model. Then, we computed the part recognition through evaluating the overlapping accuracy for the objects in the rest of the columns.

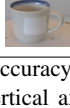
Object Category		
	72%	67%
	72%	69%

Table 2. Overlap accuracy comparison between two sets of mugs. Horizontal and vertical axis stand for testing and training categories.

The main source of error lies in poorly-estimated supervoxels as shown in Figure 4, which then build up into imprecisely estimated patches and parts.

5.2. Object Categorization

For our second experiment, we evaluated our compositional framework on an object classification task. Again, we want to prove how our method generalizes to novel or unseen objects. In this case, we went even farther and tested

our method on an object recognition task. Due to the compositionality nature of our object representation, we are able to obtain a very high recognition rate as shown in Table 3 and 4. As before, each row shows in its first column the object that was used to store the part representation in our compositional model. Then, we computed the part recognition through evaluating the overlapping accuracy for the objects in the rest of the columns. Since, the staplers and caps have more intra-class overlapping, and due to space limitation, we only show a few examples. The accuracy is computed as,

$$\text{accuracy} = \frac{(\text{TP} + \text{TN})}{P + N}$$

where TP is the number of true positives and TN the number of true negatives. P and N stand for the number of positive and negative examples.

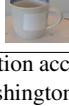
Object Instances		
	98%	97%
	96%	80%

Table 3. Classification accuracy on novel object instances of mugs in the RGB-D Washington dataset.

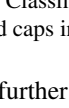
Object Instances				
	93%	97%	-	-
	91%	95%	-	-
	-	-	92%	85%
	-	-	84%	85%

Table 4. Classification accuracy on novel object instances of staplers and caps in the RGB-D Washington dataset.

We further evaluated the classification performance of the proposed approach using three classes of objects from the RGB-D Washington dataset. We performed binary classification between pairs of them. The precision (or positive predictive value *PPV*) and recall (or true positive rate *TRP*) for two pair of classes are shown in Table 5 and 6. We compared our results with viewpoint feature histogram (*VFH*) [16] as well as Oriented, Unique and Repeatable Clustered *VFH* (*OUR-CVFH*) [1] classified with K-nearest neighbor. In addition, we also learned the distance threshold for the K-nearest neighbor. Since, the most important aspect of our system is the generalization to novel object instances, we also evaluated our binary classifier on that. Hence, we trained our model with only one object instance of each

class and we tested it on the other object instances. As denoted in the table, our system outperformed other method in this aspect which proves the applicability of our system.

The source of error in classification which can be observed in the stapler instances in Table 5 is again related to the poorly estimated supervoxels in the training models. Therefore, the statistics that we collect from those wrong models would be also incorrect. A common way to solve this problem is to consider an error for the training models and integrate it into our probabilistic model which we considered that as our future work.

Method				
	PPV	TPR	PPV	TPR
Our Method Novel Instances	26%	85%	94%	36%
VFH [16]	92%	98%	100%	45%
VFH [16] Novel Instances	80%	9%	76%	22%
OUR-CVFH [1]	100%	57%	100%	36%
OUR-CVFH [1] Novel Instances	-	-	100%	50%

Table 5. Precision and Recall for binary classification for object classes staplers and mugs.

Method				
	PPV	TPR	PPV	TPR
Our Method Novel Instances	63%	91%	92%	93%
VFH [16]	92%	98%	100%	45%
VFH [16] Novel Instances	50%	12%	25%	1%
OUR-CVFH [1]	100%	81%	100%	61%
OUR-CVFH [1] Novel Instances	-	-	-	-

Table 6. Precision and Recall for binary classification for object classes caps and mugs.

6. Conclusion

We have presented a novel probabilistic approach for representing 3D objects and applied it on an object recognition task. Our goal was to present a scalable and generalizable representation where object parts play a main role. We have presented a novel 3D compositional object representation where the object is composed of a configuration of parts which represent a composition of regions built from planar patches. This compositional probabilistic approach allows us to recognize certain object parts in novel, previously unseen objects as well as correctly recognizing them by using the discriminative power of parts our compositional model provides.

References

[1] A. Aldoma, F. Tombari, R. B. Rusu, and M. Vincze. Our-cvfh - oriented, unique and repeatable clustered viewpoint

feature histogram for object recognition and 6dof pose estimation. In *DAGM/OAGM Symposium*, volume 7476, pages 113–122. Springer, 2012. 7, 8

[2] J. Carreira and C. Sminchisescu. Cpmc: Automatic object segmentation using constrained parametric min-cuts. *PAMI*, 34(7):1312–1328, 2012. 6

[3] S. Christoph Stein, M. Schoeler, J. Papon, and F. Worgötter. Object partitioning using local convexity. In *CVPR*, June 2014. 1, 7

[4] R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern Classification*. Wiley-Interscience, 2000. 3

[5] P. F. Felzenszwalb and D. P. Huttenlocher. Pictorial structures for object recognition. *IJCV*, 61(1):55–79, Jan. 2005. 1

[6] S. Fidler and A. Leonardis. Towards scalable representations of object categories: Learning a hierarchy of parts. In *CVPR*, 2007. 1

[7] W. Hu. Learning 3d object templates by hierarchical quantization of geometry and appearance spaces. In *CVPR*, pages 2336–2343. IEEE, 2012. 1

[8] M. Kiefel and P. Gehler. Human pose estimation with fields of parts. In *ECCV*, volume 8693, pages 331–346. Springer International Publishing, Sept. 2014. 1

[9] P. Krhenbhl and V. Koltun. Geodesic object proposals. In *ECCV 2014*, volume 8693, pages 725–739. Springer International Publishing, 2014. 6

[10] K. Lai, L. Bo, X. Ren, and D. Fox. A large-scale hierarchical multi-view rgb-d object dataset. In *ICRA*, 2011. 6

[11] T. Leung and J. Malik. Representing and recognizing the visual appearance of materials using three-dimensional textures. *IJCV*, 43(1):29–44, June 2001. 1

[12] B. Ommer and J. Buhmann. Learning the Compositional Nature of Visual Object Categories for Recognition. *PAMI*, (99):1, Jan. 2009. 1

[13] J. Papon, A. Abramov, M. Schoeler, and F. Wörgötter. Voxel cloud connectivity segmentation - supervoxels for point clouds. In *CVPR*, June 2013. 2

[14] B. Pepik, M. Stark, P. Gehler, and B. Schiele. Teaching 3d geometry to deformable part models. In *CVPR*, June 2012. 1

[15] L. Pishchulin, M. Andriluka, P. Gehler, and B. Schiele. Poselet conditioned pictorial structures. In *CVPR*, Portland, OR, June 2013. 1

[16] R. B. Rusu, G. Bradski, R. Thibaux, and J. Hsu. Fast 3d recognition and pose using the viewpoint feature histogram. In *IROS*, 10/2010 2010. 7, 8

[17] J. Sivic, B. C. Russell, A. A. Efros, A. Zisserman, and W. T. Freeman. Discovering object categories in image collections. In *ICCV*, 2005. 1

[18] M. Stark, M. Goesele, and B. Schiele. Back to the future: Learning shape models from 3d cad data. In *BMVC*, September 2010. 1

[19] C. Sun and J. Sherrah. 3-d symmetry detection using the extended gaussian image. *PAMI*, 19:164–168, 1997. 2

[20] L. Zhu and A. L. Yuille. A hierarchical compositional system for rapid object detection. In *NIPS*, 2005. 1

A.3 Technical Report: Multi-Modal Compositional Object Representation

Authors Safoura Rezapour Lakani and Justus Piater

Info Internal technical report

Abstract The objective of this technical report is to introduce an object representation based on both visual and non-visual features such as haptics features. The multi-modal representation will allow us to infer visual information based on haptics information and vice-versa. The non-visual features are useful at object recognition in case of missing or incomplete visual observations. We have developed a 3D compositional part-based object representation, where the non-visual features are represented by tactile information after grasping an object

Relation with the deliverable The technical report introduces a method for associating grasp templates to object parts obtained using the 3D compositional model presented in Task 2.3

Attachment (following pages until next annex)

A Multi-Modal Compositional Object Representation

Safoura Rezapour Lakani, Justus Piater

February 28, 2015

1 Introduction

The objective of this report is to represent an object with visual and non-visual features such as haptics features. The multi-modal representation will allow us to infer visual information based on haptics information and vice-versa. In this context, we considered multi-modal object recognition problem. The non-visual features then help in object recognition in the lack of visual observations. We have developed a 3D compositional part-based object representation and as the non-visual features we make use of tactile information after grasping an object.

2 3D Compositional Object Model

In this section, we briefly explain our part-based compositional object model. We learn our model based on view-based RGB-D pointclouds. We have a bottom-up compositional model which starts from the object points and reaches to the object category level. Our model consists of three levels: patches, parts and objects as shown in Figure 1. There is two main concepts in each level, composition and representation. More precisely, each level is composed of the elements of the lower levels, but its representation might be different than its individual composed low level elements. For representation, we extract features in each level based on its composed elements.

From Points to Patches Initially, the points form low-level patches. As can be seen in Figure 1, they are denoted by $\mathbf{q}$. These patches construct the lowest level of our compositional model. We extract scale-invariant features based on surface curvatures from the patches, and represent each patch with the extracted feature. So, a patch is represented by the extracted feature but is composed of the input points. We then perform clustering based on the extracted feature and we obtain a set of patch clusters $Q = \{q_1, q_2, \dots, q_n\}$ which gives us the patch types in our dataset. Then $p(\mathbf{q}|q)$ denotes the observation probability for patch $\mathbf{q}$ based on patch type q .

From Patches to Parts Object parts, as shown in Figure 2 with $\mathbf{r}$, are composed of the low-level patches. We represent each object part as a histogram of the patch types which compose it. That is we compute the observation likelihood for all the observed patches which belong to a part by computing $p(\mathbf{q}|\mathbf{q})$. A patch can be assigned to multiple patch types as shown in Figure 2. We then make a normalized histogram based on them. We then perform clustering based on our object parts histograms to obtain different part types $R = \{r_1, r_2, \dots, r_n\}$.

From Parts to Object An object is composed of a set of parts in a specific configuration. Therefore, our objective is to collect two information: 1) occurrence of a certain part for an object class, 2) connection between the parts.

For the first information, we segment an object to its constitutes parts $\mathbf{R} = \{\mathbf{r}_1, \dots, \mathbf{r}_n\}$ and we recognize the parts based on the part types. That is we compute $\forall \mathbf{r} \in \mathbf{R}, r \in R p(\mathbf{r}|r)$ and a part can be matched to more than one type. From the obtained part types, we collect statistics of occurrence of the part types for a certain object class $p(r_1, \dots, r_{N_R}|o)$.

The second information cannot be obtained at the part level. The reason is that we represented a part as a histogram of patch types. This representation is not local enough to give us the connectivity information. In order to overcome this problem, we make use of the compositional nature of our model, and we use the patches which consists the parts. However, we do not use all the constitutes patches, but only the adjacent bounding patches between the pair of adjacent parts. We then recognize these adjacent bounding patches based on our learned patch types and we collect statistics of their co-occurrence for a certain object class. That is we compute $p(q_1, q_2|o)$ for all the patch pairs which are adjacent but belong to the different parts.

3 Multi-Modal Representation

The objective in multi-modal representation is to represent an object with tactile and visual information. We considered multi-modal object recognition problem. In Section 2, we noted that an object is represented by its parts and parts connectivities. For recognition purpose, we gathered information based on the probability of occurrence of certain *part types* for an object class and the adjacent bounding *patch types*. In multi-modal representation for object recognition, the tactile and visual information are joint together. Since, we perform recognition in the part level, we add tactile information in the part level as well. Hence, the tactile information is added at the part level of our compositional model.

The visual information comes from our compositional model and tactile information from grasping. We perform multiple grasps on object parts (from our *IKEA* kitchen object dataset) and we obtain tactile features from that. The first step is to quantize the tactile features. To this aim, we perform an agglomerative clustering based on all the extracted tactile features from all the objects parts. This form our tactile feature types $T = \{t_1, \dots, t_m\}$

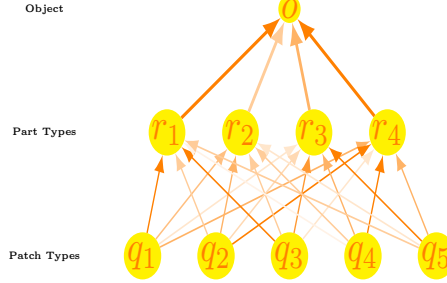


Figure 1: A schematic diagram for a learned model for object class o . We have learned patch types $Q = \{q_1, \dots, q_{N_Q}\}$ and learned part types $R = \{r_1, \dots, r_4\}$. A part is composed of patches $\mathbf{Q} = \{\mathbf{q}_1, \dots, \mathbf{q}_4\}$. Those patches are matched to the patch types. Hence a part is represented as a histogram of patch types. As shown in the figure, the lighter the color the less frequent the patch type and vice versa. Next, an object is composed of parts $\mathbf{R} = \{\mathbf{r}_1, \mathbf{r}_2, \mathbf{r}_3\}$. The parts are matched to the parts clusters. We then compute the probability of occurrence of the object parts types which belong to the object class o , $p(r_1, \dots, r_{N_R}|o)$. For object recognition we learn the $p(r_1, \dots, r_{N_R}|o)$ as well as part connectivity based on adjacent boundary patches $p(q_1, q_2|o)$.

The second step is to associate tactile features to object parts. From our robotic part-based grasping experience, we have three information. 1) We know the object label t . 2) Thanks to our compositional representation, the grasped part is matched to the respective part type(s) $\{r_1, \dots, r_m\}$. 3) The obtained tactile feature is matched to its corresponding tactile feature type(s) $\{t_1, \dots, t_n\}$. We then collect the joint tactile-part co-occurrence for object class o , that is $p(r, t|o)$. We use this information instead of a part only information $p(r|o)$ in object recognition task.

4 Object Recognition Using Visual-Tactile Features

In this section, we explain the probabilistic object recognition formulation based on tactile and visual features. We extract tactile features from the tactile readings and the visual features from the object parts. As described in Section 2, the feature for each part is a histogram of its constitutes patch types. It should be noted that the same formulation can be applied for instance-level object recognition.

As mentioned in Section 2, for object recognition we are interested in the parts which compose an object and the connectivity between them. Therefore the input data to our object recognition system is the part segmented object. Since, the parts are composed of patches and the patches from points, we also

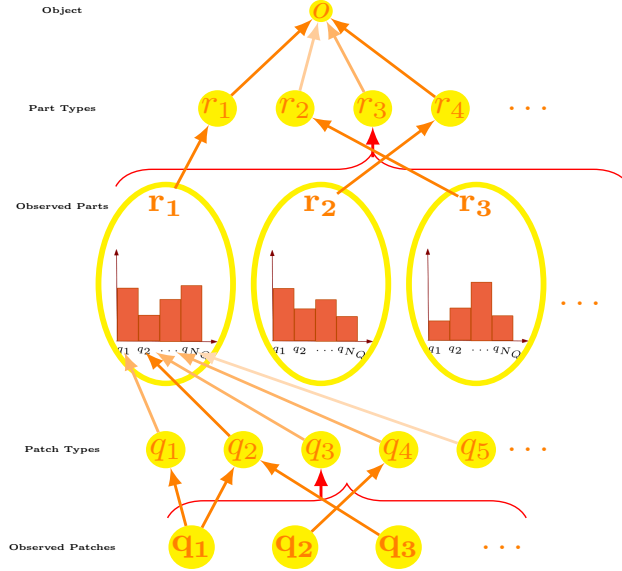


Figure 2: A schematic diagram for data flow in learning the compositional model. We segment each object to its patches $\mathbf{Q} = \{\mathbf{q}_1, \dots, \mathbf{q}_4\}$. Next, we obtain patch types by clustering the patches from multiple objects as shown by the red brace $Q = \{q_1, \dots, q_{N_Q}\}$. The obtained patch types are used for part segmentation. The observed object parts are denoted by $\mathbf{R} = \{\mathbf{r}_1, \mathbf{r}_2, \mathbf{r}_3\}$. We have patches of each object part, then we match them to the learned patch types, as shown by the orange arrows. One patch can be matched to multiple patch clusters but it belongs to only one part. Next, we represent a part as the histogram of its constituent patch types. We then perform clustering on the part histograms and obtain part types $R = \{r_1, r_2, r_3\}$. For the objects which belongs to the object class o , we compute the likelihood of their constituent parts to the learned part types. The arrows from part types to the object class node show the frequency of occurrence of each of part types for an object class. The lighter the color the less frequent the part type for the object class o .

have adjacency information between the object parts. We then use the parts and their adjacency information for object classification task.

We are interested in recognizing class o for the observed parts $\mathbf{R}$ in the configuration $\mathbf{S}$. $\mathbf{S}$ is a binary adjacency matrix in terms of the neighborhood between each two part. More precisely, each row in $\mathbf{S}$ indicates the adjacent parts for a certain object part. In addition, we perform multiple grasps on each object part and extract tactile feature from them. We denote these tactile features with $\mathbf{T}$. In the following formulation, we indicate each individual part cluster with r which belongs to a set R of all the parts clusters. Likewise, each individual patch cluster is shown with q which belongs to a set of patch clusters

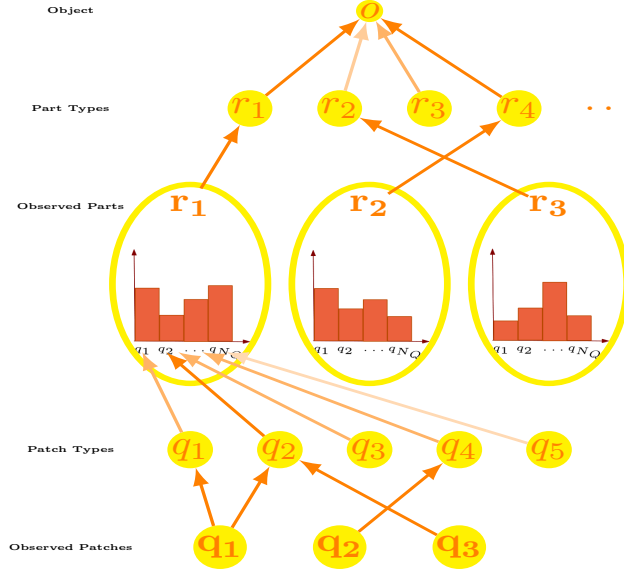


Figure 3: A schematic diagram for data flow in an instantiated object model. Initially, an object is segmented to its constituent patches which are considered as observed patches $\mathbf{Q} = \{\mathbf{q}_1, \dots, \mathbf{q}_4\}$. These patches are matched to the learned patch type $Q = \{q_1, \dots, q_{N_Q}\}$ and assigned to only one object part as denoted by $\mathbf{R} = \{\mathbf{r}_1, \mathbf{r}_2, \mathbf{r}_3\}$. A part is represented by a histogram of patch types based on its constituent patch types. These parts are then matched to the part types as denoted by $R = \{r_1, r_2, r_3\}$. Finally, an object is classified to object class o based on the probability of occurrence of their part types as well as their adjacent boundary patches for the object class o .

Q . Finally, each tactile cluster is written as t which belongs to the set T of all the tactile clusters.

$$o_{\mathbf{R}, \mathbf{S}, \mathbf{T}} = \underset{o}{\operatorname{argmax}} p(\mathbf{R}, \mathbf{S}, \mathbf{T} | o) \quad (1)$$

having an independent assumption between the object parts, the above conditional probability can be re-written as follows,

$$p(\mathbf{R}, \mathbf{S}, \mathbf{T} | o) = \prod_{\mathbf{i}} p(\mathbf{r}_{\mathbf{i}}, \mathbf{t}_{\mathbf{i}} | \mathbf{s}_{\mathbf{i}}, o) p(\mathbf{s}_{\mathbf{i}} | o) \quad (2)$$

We consider a uniform probability distribution for $p(\mathbf{s}_{\mathbf{i}} | o) = \frac{1}{N_{\mathbf{S}}}$, where $N_{\mathbf{S}}$ is

the number of all the configurations.

$$p(\mathbf{R}, \mathbf{S}, \mathbf{T}|o) = \prod_{\mathbf{i}} \frac{p(\mathbf{r}_{\mathbf{i}}, \mathbf{t}_{\mathbf{i}}|\mathbf{s}_{\mathbf{i}}, o)}{N_{\mathbf{S}}} \quad (3)$$

$\mathbf{s}_{\mathbf{i}}$ is the configuration or adjacency vector for the part $\mathbf{r}_{\mathbf{i}}$. Hence, it contains all the adjacent parts to $\mathbf{r}_{\mathbf{i}}$. Therefore, we marginalize over all the parts that vector $\mathbf{s}_{\mathbf{i}}$ gives us,

$$p(\mathbf{r}_{\mathbf{i}}, \mathbf{t}_{\mathbf{i}}|\mathbf{s}_{\mathbf{i}}, o) = \sum_{\mathbf{r} \in \mathbf{s}_{\mathbf{i}}} p(\mathbf{r}_{\mathbf{i}}, \mathbf{r}, \mathbf{t}_{\mathbf{i}}|\mathbf{s}_{\mathbf{i}}, o) \quad (4)$$

We compute the conditional probability of the pair parts $p(\mathbf{r}_{\mathbf{i}}, \mathbf{r}, \mathbf{t}_{\mathbf{i}}|\mathbf{s}_{\mathbf{i}}, o)$ independent of the given configuration $\mathbf{s}_{\mathbf{i}}$. Hence, we drop it from Eqn 4,

$$p(\mathbf{r}_{\mathbf{i}}, \mathbf{t}_{\mathbf{i}}|\mathbf{s}_{\mathbf{i}}, o) = \sum_{\mathbf{r} \in \mathbf{s}_{\mathbf{i}}} p(\mathbf{r}_{\mathbf{i}}, \mathbf{r}, \mathbf{t}_{\mathbf{i}}|o) \quad (5)$$

In order to compute the conditional probability of the adjacent patches $\mathbf{r}_{\mathbf{i}}$ and $\mathbf{r}$, we make use of their adjacent boundary patches. Patch $\mathbf{q}_{\mathbf{j}}$ belong to the part $\mathbf{r}_{\mathbf{i}}$ and is the neighbor of the part $\mathbf{r}_{\mathbf{j}}$ as denoted by $N_{\mathbf{r}}$ in the following equation. Therefore, we marginalize over these patches,

$$p(\mathbf{r}_{\mathbf{i}}, \mathbf{r}, \mathbf{t}_{\mathbf{i}}|o) = \sum_{\mathbf{q}_{\mathbf{j}} \in \mathbf{r}_{\mathbf{i}}, \mathbf{q}_{\mathbf{j}} \in N_{\mathbf{r}}} p(\mathbf{r}_{\mathbf{i}}, \mathbf{r}, \mathbf{t}_{\mathbf{i}}|o, \mathbf{q}_{\mathbf{j}})p(\mathbf{q}_{\mathbf{j}}|o) \quad (6)$$

We can then factorize the first term in the summation of Eqn 6,

$$p(\mathbf{r}_{\mathbf{i}}, \mathbf{r}, \mathbf{t}_{\mathbf{i}}|o) = \sum_{\mathbf{q}_{\mathbf{j}} \in \mathbf{r}_{\mathbf{i}}, \mathbf{q}_{\mathbf{j}} \in N_{\mathbf{r}}} p(\mathbf{r}|o, \mathbf{q}_{\mathbf{j}}, \mathbf{r}_{\mathbf{i}}, \mathbf{t}_{\mathbf{i}})p(\mathbf{r}_{\mathbf{i}}, \mathbf{t}_{\mathbf{i}}|o, \mathbf{q}_{\mathbf{j}})p(\mathbf{q}_{\mathbf{j}}|o) \quad (7)$$

We consider that the conditional probability of $\mathbf{r}_{\mathbf{i}}, \mathbf{t}_{\mathbf{i}}$ is independent of $\mathbf{q}_{\mathbf{j}}$ when the object class is given,

$$p(\mathbf{r}_{\mathbf{i}}, \mathbf{t}_{\mathbf{i}}|o, \mathbf{q}_{\mathbf{j}}) = \frac{p(\mathbf{r}_{\mathbf{i}}, \mathbf{t}_{\mathbf{i}}, o, \mathbf{q}_{\mathbf{j}})}{p(o, \mathbf{q}_{\mathbf{j}})} \quad (8)$$

$$= \frac{p(\mathbf{r}_{\mathbf{i}}, \mathbf{t}_{\mathbf{i}}, \mathbf{q}_{\mathbf{j}}|o)p(o)}{p(o, \mathbf{q}_{\mathbf{j}})} \quad (9)$$

$$= \frac{p(\mathbf{r}_{\mathbf{i}}, \mathbf{t}_{\mathbf{i}}|o)p(\mathbf{q}_{\mathbf{j}}|o)p(o)}{p(o, \mathbf{q}_{\mathbf{j}})} \quad (10)$$

$$= p(\mathbf{r}_{\mathbf{i}}, \mathbf{t}_{\mathbf{i}}|o) \quad (11)$$

Furthermore, we consider that the conditional probability of a part $\mathbf{r}$ is independent of the tactile features of $\mathbf{r}_{\mathbf{i}}$. Therefore, we obtain the following formulation for computing the conditional probability of the two adjacent part,

$$p(\mathbf{r}_{\mathbf{i}}, \mathbf{r}, \mathbf{t}_{\mathbf{i}}|o) = \sum_{\mathbf{q}_{\mathbf{j}} \in \mathbf{r}_{\mathbf{i}}, \mathbf{q}_{\mathbf{j}} \in N_{\mathbf{r}}} p(\mathbf{r}|o, \mathbf{q}_{\mathbf{j}}, \mathbf{r}_{\mathbf{i}})p(\mathbf{r}_{\mathbf{i}}, \mathbf{t}_{\mathbf{i}}|o)p(\mathbf{q}_{\mathbf{j}}|o) \quad (12)$$

The only difference with our probabilistic vision based formulation is the computation of the joint probability of a part $\mathbf{r}_i$ and its tactile feature $\mathbf{t}_i$. So, we only derive that here. In order to compute this probability we marginalize over our parts clusters r ,

$$p(\mathbf{r}_i, \mathbf{t}_i|o) = \sum_r p(\mathbf{r}_i, \mathbf{t}_i|o, r)p(r|o) \quad (13)$$

In order to compute $p(\mathbf{r}_i, \mathbf{t}_i|o, r)$, we need to make use of the object-part-tactile co-occurrence table that we collected in the training phase. That is $p(r, t|o)$ where t is the tactile feature type and r is the part type which $\mathbf{t}_i$ and $\mathbf{r}_i$ can be matched to them. To this aim, we marginalize over all the tactile clusters t ,

$$\begin{aligned} p(\mathbf{r}_i, \mathbf{t}_i|r, o) &= \sum_t p(\mathbf{r}_i, \mathbf{t}_i, t|r, o) \\ &= \sum_t \frac{p(\mathbf{r}_i|\mathbf{t}_i, r, t, o)p(\mathbf{t}_i|r, t, o)p(r, t|o)}{p(r|o)} \end{aligned} \quad (14)$$

We compute the conditional probability of a part $\mathbf{r}_i$ given a part type r independent of its tactile information (tactile feature and tactile cluster). In the same way, we compute the conditional probability of a tactile feature $\mathbf{t}_i$ given a tactile type t independent of the visual information. Therefore, we get, We integrate Eqn 14 into Eqn 13 which yields,

$$p(\mathbf{r}_i, \mathbf{t}_i|o) = \sum_r \sum_t p(\mathbf{r}_i|r)p(\mathbf{t}_i|t)p(r, t|o) \quad (15)$$

The conditional probability of $\mathbf{r}_i$ based on a part cluster r denotes simply the observation likelihood for clusters r . This probability is zero if $\mathbf{r}_i$ cannot be matched to r otherwise this is the inverse of the number of clusters that $\mathbf{r}_i$ can be matched to them $NC_{\mathbf{r}_i}$. For each cluster we compute its center μ_r and we keep a distance threshold. Then, matching to a cluster means if the distance between the cluster center μ_r and the part $\mathbf{r}_i$ is less than the determined threshold. This is our simplified assumption at the moment, in the future we will use Gaussian Mixture Models for assigning the observation likelihood,

$$p(\mathbf{r}_i|r) = \begin{cases} 0, & \text{if } dist(\mathbf{r}_i, \mu_r) > t \\ \frac{1}{NC_{\mathbf{r}_i}}, & \text{otherwise} \end{cases} \quad (16)$$

The conditional probability of $\mathbf{t}_i$ given tactile cluster t indicates if the tactile feature $\mathbf{t}_i$ can be matched to cluster t . We compute $p(\mathbf{t}_i|t)$ as we computed in Eqn 16. $p(r, t|o)$ is computed from the object-part-tactile co-occurrence table. We now, integrate this joint multi-modal probability into our object class conditional probability. Skipping the detail of the formulation, we finally get Eqn 4 and we integrate this joint probability into this final formulation,

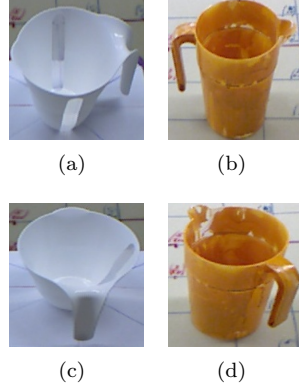


Figure 4: Figures 4(a), 4(b) show an example of object recognition problem with the fixed view-point and scale-invariant features. The difference between two objects can only be perceived when the tactile features of their parts are considered.

$$p(\mathbf{R}, \mathbf{S}, \mathbf{T}|o) = \prod_{\mathbf{i}} \frac{1}{N_{\mathbf{S}}} p(\mathbf{r}_{\mathbf{i}}, \mathbf{t}_{\mathbf{i}}|o) \sum_{\mathbf{r}} p(\mathbf{r}|o) \sum_{\mathbf{q}_{\mathbf{j}} \in \mathbf{r}_{\mathbf{i}}, \mathbf{q}_{\mathbf{j}} \in N(\mathbf{r})} \sum_{\mathbf{q}_{\mathbf{k}} \in \mathbf{r}, \mathbf{q}_{\mathbf{k}} \in N(\mathbf{q}_{\mathbf{j}})} \sum_{q_1 \in Q} \sum_{q_2 \in Q_2} p(\mathbf{q}_{\mathbf{k}}|q_2) p(q_2, q_1|o) p(\mathbf{q}_{\mathbf{j}}|q_1)$$

This multi-modal representation help us when the visual features are not enough to distinguish object classes or object instances. That is due to the fact that we have a view-dependent acquisition. And some object parts or patches are not clearly visible from one view-point. An example is given in Figure 4. The two different objects in Figure 4(a), 4(b) seem visually similar, especially the handles look similar from one view-point. As shown in Figure 4(c), 4(d), their difference in the shape of the handles can be seen from another view-point. There is the use case of the tactile features which can discriminate between these two objects without changing the view-point. In addition to that, our visual features are only dependent to the curvature of object surface and are invariant to scale. These shortcomings can be handled by integrating tactile features.

A.4 Technical Report: 3D Compositional Model associated with Grasp Templates and Haptic Data

Authors Alexander Rietzler

Info Internal technical report

Abstract The objective of this technical report is to associate graspability and haptic characteristics to object parts produced by a 3D compositional model. The advantage of the proposed method resides in the ability to perform robust grasping of novel object instances, composed of parts similar to the ones existing in the training set

Relation with the deliverable The technical report introduces a method for associating grasp templates to object parts obtained using the 3D compositional model presented in Task 2.3

Attachment (following pages)

3D-Compositional Model associated with Grasp Templates and Haptic Data

Alexander Rietzler

*Institute of Computer Science
University of Innsbruck
Innsbruck, Austria*

ALEXANDER.RIETZLER@UIBK.AC.AT

1. Problem Statement

The problem dealt with here is to associate graspability and haptic characteristics to parts produced by a 3D compositional models in order to perform robust grasping of even unknown objects exhibiting known parts. The work is strongly tied to requirements of Task 2.1 and 2.2 of the PaCMan project Workpackage 2.

2. Experimental Setup

As a vision sensor, the depth sensor Microsoft Kinect is used. The 3D pointcloud data captured by the sensor is represented as a continuous surface distribution in $SE(3)$ (Detry and Piater, 2010).

In order to perform grasp experiments and to gather tactile data our setup is equipped with a Kuka Lightweight Robot and a Schunk Dextrous Hand 2 with 6 tactile arrays at its 6 finger links.

First a set of successful grasps are shown on a training set, capturing object-relative wrist pose of the gripper, tactile signature and kinesthetic signature (gripper joints), see Figure 1.



Figure 1: IKEA objects used for training and test set.

After recognizing meaningful parts of the training objects with the 3D-compositional model (Rezapour-Lakani et al., 2014) the grasps including haptic characteristics are associated to the part the grasp is performed on.

To evaluate the system, a grasp experiment is performed measuring grasp success rates.

3. Method

Part and Grasp Representation

The method chosen builds on part recognition software developed by Rezapour-Lakani et al. (2014). Given a novel scene, the software segments unseen objects containing similar parts into their known parts. Each part belongs to a previously learned part-cluster marked with a cluster id. To restrict grasp template alignment to the associated part, its oriented bounding box (OBB) is computed and passed to a grasp inference algorithm.

To each part-cluster a set of grasps is attached. In order to match grasps on an unseen but familiar part, the shape-relevant information of graspability is captured by a 3D template $G(x)$, where each point x is represented by a 6D Kernel resulting in a continuous shape distribution. The template is generated by cropping the shape distribution with a cube-shaped region of interest (ROI) close to the gripper contact points during a trained grasp. Grasps are further represented by a pre-grasp pose and a retreat-pose, and by three different grasp types – pinch, spherical and parallel. Figure 2 shows a set of selected templates.

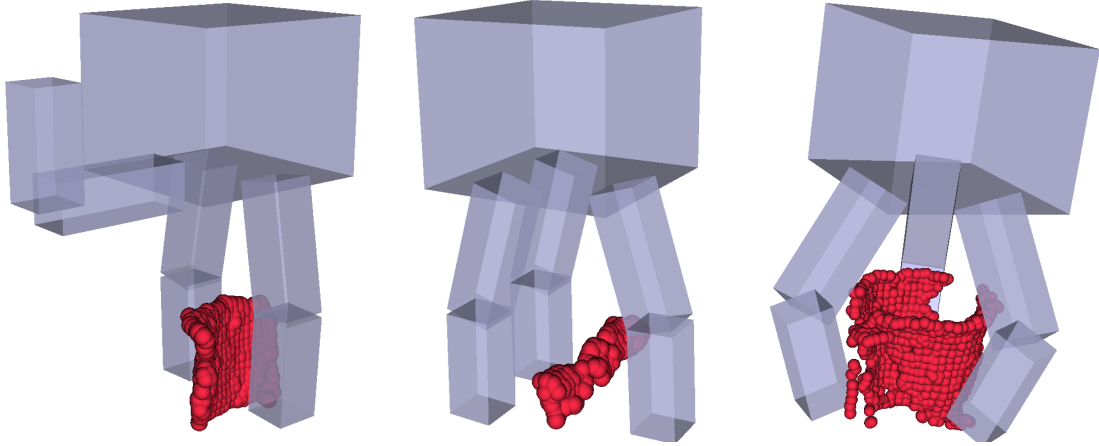


Figure 2: Three selected templates for visualization. From left to right: body-pinch (container), handle-parallel (pot), body-spherical (mug).

Tactile data is gathered during a training grasp. Each of the six phalanges of the gripper is equipped with a tactile array of 6×14 taxels for proximal and 6×13 for distal phalanges, and a sensor value ranging from 0 to 5000, which is proportional to taxel pressure, here treated as continuous values. For our training grasps only the distal arrays are active, which yields $3 \times (13 \times 6)$ continuous values.

Grasp Inference

The grasp template alignment procedure is based on a pose estimation algorithm,¹ where the 6D pose-dependent cross-correlation between the shape distribution of the scene $Q(x)$ and the template $G(x)$ is computed as:

$$S(G, Q) = \int G(x)Q(x) dx \quad (1)$$

This integral is computed by Monte Carlo approximation. In addition, to force the alignment close to the selected part only the OBB of the grasp-template (G) has to intersect with the OBB of the selected part (P):

$$C(G, P) = \begin{cases} 1 & OBB(G) \cap OBB(P) \neq \{\} \\ 0 & OBB(G) \cap OBB(P) = \{\} \end{cases} \quad (2)$$

As the output of our grasp inference algorithm computes a single best grasp per template, we check for collisions with the environment as well as for reachability of the gripper wrist pose hypothesis (r, t) via inverse kinematics. The reachability term reads as

$$R_{r,t} = \begin{cases} 1 & (r, t) \text{ reachable} \\ 0 & (r, t) \text{ not reachable} \end{cases} \quad (3)$$

Combing equations (1-3), our optimization over wrist poses yields

$$S_{r^*, t^*}^* = \operatorname{argmax}_{r,t} \{S(T_{r,t}(G), Q) \cdot C(T_{r,t}(G), P) \cdot R_{r,t}\} \quad (4)$$

where $T_{r,t}(\cdot)$ denotes a rigid body transformation.

Success Classification

To measure the additional value of using haptic data for grasping, we calculate success prediction performance based on haptics. We represent grasp success as categories $\{1 = \text{success}, -1 = \text{failure}\}$. Grasps that are unstable in the sense that the object could be lifted but moved loosely in hand after pickup are marked as a failure.

As raw data, we measure:

- Tactile: 3 fingers $\times$ 13 $\times$ 6 sensor matrix: 234 sensor values
- Kinesthetic: 7 joints: 7 sensor values
- Total: 241 sensor values

To classify the performance, we use the following features as motivated in Section 4.

- Centroid of the active sensor matrix: $S = \left(\frac{M_{10}}{M_{00}}, \frac{M_{01}}{M_{00}} \right)$, where $M_{ij} = \sum_x \sum_y x^i y^j I(x, y)$ are the raw image moments

1. <http://nuklei.sourceforge.net>

- Average pressure per active taxel: $\frac{M_{00}}{N}$
- Sum of absolute difference of measured gripper joint positions and target joint positions $\sum_i |q_{mes}(i) - q_{learned}(i)|$
- Total number of features: 10

To capture the assumed nonlinearities in classification we choose Gaussian Process Classification with a diagonal square-exponential covariance function, which is proven to be robust with respect to a large number of feature dimensions (large compared to our dataset size).

As our grasp inference algorithm computes an alignment score S^* , we compare grasp success prediction performance based on haptic features and S^* .

4. Experiment and Results

For our experiment we chose 3 training objects from the PaCMan IKEA dataset, where we performed 11 training grasps in total to generate the grasp templates, see Figure 1(a). We associated each of the training grasps with the corresponding part-clusters, which were recognized on the training objects.

The two meaningful part-clusters that were generated contained mostly body-parts and handle-parts respectively, associated with 5 and 4 templates each. For the reason of fitting a template in a stable manner, we chose to perform our grasp experiments only with templates from the part-cluster containing body-parts.

Our experiment was performed in scenes containing a single object. For each object we used the 3D compositional model to compute the parts, fitted all the 5 templates on the parts, and executed their grasp. Figure 3 shows a scene with a selected part and corresponding aligned grasp template with highest alignment score.

In total, we planned 50 grasps on 12 parts of 9 different objects. To measure our grasp success rate we make a distinction between test and training objects. Not all grasps could be executed due to planning failures. For training objects we executed 19 grasps, where 15 of them were successful. This gives a success rate of 79%. For novel objects we executed 21 grasps, where 11 of them were successful, which a success rate of 52%. Note that in a real application scenario success rates would be higher since we would choose the grasp with highest alignment score for an object, whereas here we execute all grasps associated with a given part. The goal of this experiment is to obtain a balanced training set for success classification. Still the experiment shows that, especially for highly similar objects like mug-blue and mug-brown, the grasp transfer works very well. A summary of the experiment can be found in Table 1.

Success Prediction

We use Gaussian Process classification to predict grasp success rates as mentioned in Section 3. Our dataset is slightly imbalanced with 26 positive compared to 14 negative samples.

To evaluate the performance we use leave-one-out cross-validation and compute accuracy, precision, recall and F1 measure. In addition we compute the same performance

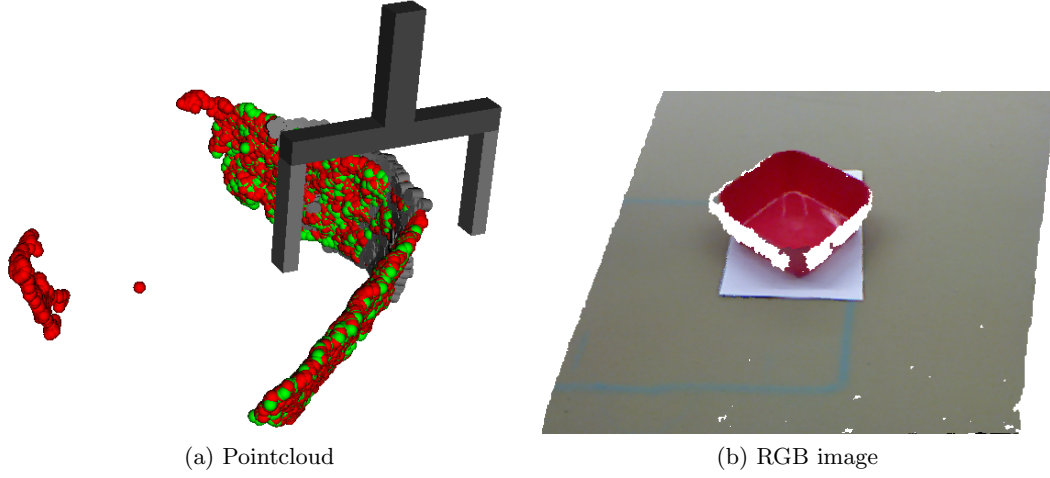


Figure 3: Instantiated scene containing the red box. Points in green show the selected part the grasp template is aligned to. Points in gray show the aligned grasp template. The gripper indicates wrist pose and grasp type. Here a pinch grasp has the highest alignment score.

Executed Grasps	Grasps/Part	Parts	Objects	Success Train	Success Novel
40/50	5	12	9	15/19	11/21

Table 1: Overview of the grasp experiment. Training objects with success rates: mug-blue(5/7), container-1 (7/8), pot(3/4). Test objects with success rates: mug-brown (3/4), box-red(3/4), funnel(2/3), frying-pan(0/2), watering-can(2/5), plate(1/3).

measures with respect to only classifying the data with respect to the alignment score S^* . Our results are shown in Table 2.

Features	Accuracy	Precision	Recall	F1	TP	TN	FP	FN
Haptic&Alignment Score	0.78	0.79	0.88	0.84	23	8	6	3
Haptics	0.85	0.83	0.96	0.89	25	9	5	1
Alignment Score	0.70	0.73	0.85	0.79	22	6	8	4

Table 2: Success classification performance for haptics features and or alignment score. TP, TN, FP, FN are the number of true positives, true negatives, false positives and false negatives.

First of all, our performance measures show that we can quite well predict grasp success from haptic features. Interestingly using only haptic features instead of all gives us the best result. Of importance is especially the precision value, as this expresses the confidence that a grasp will work after measuring its haptic features at contact. In our case that means

that no matter what we do, based on our precision value using haptic features, 83% of successfully predicted grasps will subsequently succeed when lifting up the object. From our intuition as humans, this rate could be higher. One reason for that can be that the when a pinch-grasp doesn't close around the object at all, touch sensors may be activated due to self-collision. This is why we created the kinesthetic feature (see Sec. 3) that should detect when the fingers are fully closed without object. Still if the the grasp closes around a thin rim, the signatures can be very similar for successful and failed grasps.

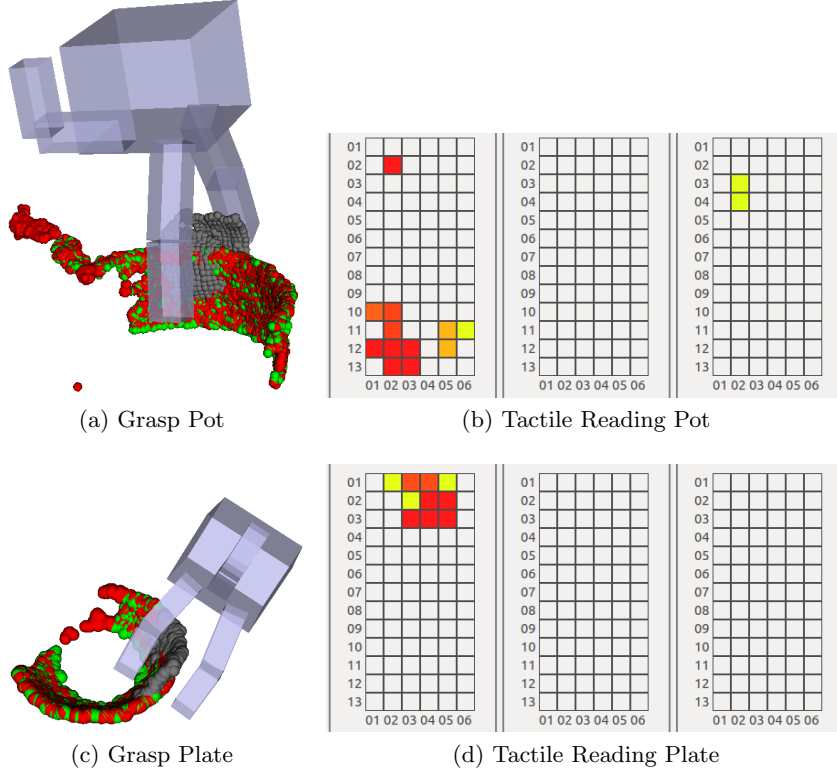


Figure 4: (a,b) Grasp and tactile reading of a successful grasp on the pot despite a non-matching template ($S^*=0.18$). Also the grasp is correctly predicted by the classifier. One can see a stable tactile reading only on one of the pads due to convexity of the object. (c,d) Grasp and tactile reading of an unsuccessful grasp on the plate. The grasp is classified wrongly, probably due to the strong tactile reading. Also the template matches well but is too far from the real contact point. The tactile reading is very close to the edge of the sensor pad and the object is heavier than most other objects in the test set.

Also for some successful grasps the tactile sensor doesn't show any value because the fingers touch the object at an oblique angle. The Schunk Hand is not compliant, and the sensors are not sensitive enough to detect such contacts.

The fact that classification performance of haptics is significantly better than prediction based on alignment score indicates the importance of haptics for grasping in unstructured environments with novel objects. Also there exist cases where the alignment score and the template alignment are bad, but still the resulting grasps are stable, see Figure 4(a). This increases the number of possible positive grasps in a scenario where grasps are rejected based on an alignment score threshold.

5. Discussion

We performed a grasping experiment where we successfully showed that our method produces successful grasps on partly similar but unseen objects. Most importantly it also shows that using haptic features are useful to predict grasp success. This allows overall grasp success rate to be increased by triggering regrasps based on tactile readings.

Another important point of the method is that due to the pre-selection of parts with our 3D-compositional model, we are able to reduce computation time compared to a scenario where we try to fit templates to the whole object or cluttered scenes, which is especially important in the PacMan project.

The methods also has its weaknesses. The templates need a certain size to be matched in a stable way. In the future we plan to make templates more local and adjust their sizes based on the part in use.

References

- Renaud Detry and Justus Piater. Continuous surface-point distributions for 3D object pose estimation and recognition. In Ron Kimmel, Reinhard Klette, and Akihiro Sugimoto, editors, *Asian Conference on Computer Vision*, volume 6494 of *LNCS*, pages 572–585, Heidelberg, 2010. Springer. doi: 10.1007/978-3-642-19318-7_45. URL <https://iis.uibk.ac.at/public/papers/Detry-2010-ACCV.pdf>.
- Safoura Rezapour-Lakani, Mirela Popa, Antonio J. Rodríguez-Sánchez, and Justus Piater. Scale-Invariant, Unsupervised Part Decomposition of 3D Objects. In *Parts and Attributes*, 9 2014. URL <https://iis.uibk.ac.at/public/papers/Rezapour-2014-PA.pdf>. Workshop at ECCV.

A.5 Technical Report: Object classification using tactile sensor information

Authors Peter Kiechle

Info Internal technical report

Abstract The objective of this technical report is to describe the tactile data acquisition procedure using our robotic framework. Next, it presents a parametrization of haptic information, by extracting a set of features such as object diameter, compressibility, standard deviation of intensity values and translation and rotation invariant Chebyshev moments. Finally, it presents an analysis of the obtained results, showing that the proposed method is useful for object discrimination

Relation with the deliverable TThe technical report introduces a method for parametrizing haptic features and analyzes its performance for object discrimination. This analysis is useful for Task 2.1, in which a similar tactile feature parametrization will be associated with our developed 3D compositional model for obtaining a multi-modal representation

Attachment (following pages)

Summary: Object classification using tactile sensor information

Peter Kiechle

February 24, 2015

1 Overview

The acquisition of accurate 3D sensor data is a crucial point for robot grasping applications. But all state-of-the-art 3D scanners have to deal with certain limitations such as difficulties with reflective and transparent objects or cluttered scenes. Tactile sensors provide an opportunity to improve the perception of an already touched object which can be used to locally optimize the grasp or perform a regrasp if necessary.

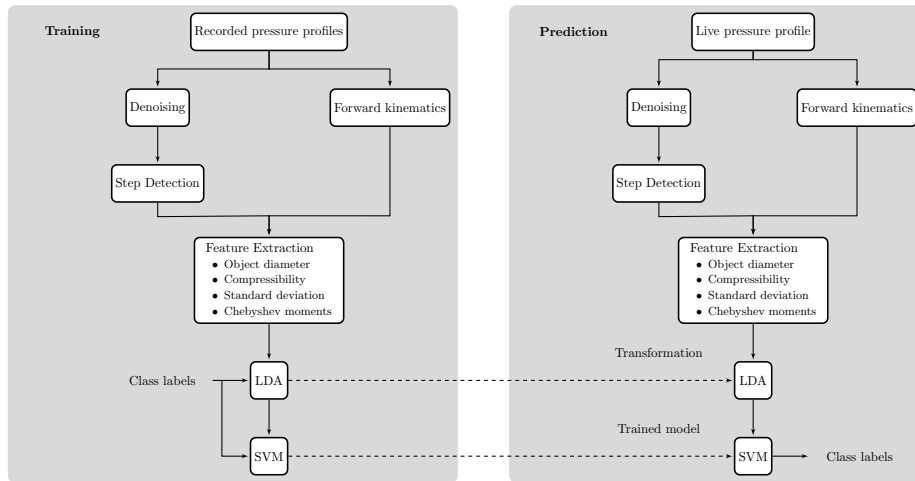


Figure 1: Classification workflow

Figure 1 gives an overview of a working object-grasp classification procedure using SCHUNK's Dexterous Hand 2.0 with tactile sensors by Weiss Robotics. During the training phase various similar grasps of the same object are performed and recorded. The position controller of the SDH-2 delivers joint angle readings at an average data rate of about 110 Hz. The corresponding step size is smaller than 0.02° . The tactile sensor controller transmits data at a rate of roughly 30 frames per second depending on the number of active sensor cells. Due to the internal run-length encoding this value varies significantly over time. By applying a Denavit-Hartenberg-Transformation, the 3D position of each active sensor cell is then computed from the recorded joint angles. Since the tactile

readings are noisy, especially while capturing in high sensitivity mode, a spatial digital filter is applied in order to smooth the signal. A median filter that replaces each value with the median of neighboring entries showed to perform the best in experiments with a variety of denoising algorithms. In the following step, the actual grasps have to be identified and separated from each other. Begin and end frames are therefore determined using a simple step detection algorithm.

The concatenated feature vector consists 54 elements of four different types of features which are further discussed in the following section. Experiments revealed, that the individual features are not equally important. But since they are also not independent from each other, a classical feature selection approach does not work well on the available data. Instead a Linear Discriminant Analysis (LDA), a dimensionality reduction techniques related to Principal Component Analysis (PCA) that maximizes the separation between multiple classes is performed as an intermediate step. A grid-search suggested that the best results are achieved by reducing the 54 dimensions down to 14. A one-vs-one multiclass approach using Support Vector Machines (SVM) is used to execute the final classification task. The trained classifier, that is the support vectors as well as the performed LDA transformation are retained for the prediction phase.

2 Feature Extraction

As mentioned before, the actual feature vector is a concatenation of four different types of features. For the sake of simplicity, all grasps are performed with a pinch grip using two fingers only which restricts the tactile feedback to the sensor matrices at the finger tip.

2.1 Object diameter

The size of a grasped object is intuitively an important property in the context of classification. Training a linear Support Vector Machine before applying a dimensionality reduction shows, that the corresponding size feature has the largest assigned weight to it - a clear indication of its discriminatory power. Based on the hand's dimensions the actual forward kinematics can be computed according to the Denavit-Hartenberg-Transformation (DHT). This results in a 3D point cloud of all active sensor cells. The minimal bounding sphere or enclosing ball has proven to be a versatile metric for the point cloud's extent. The actual implementation relies on an iterative algorithm by Gärtner (1999).

2.2 Compressibility

The SDH-2 is strong enough to actually compress soft objects. It is therefore possible to measure the object's rigidity by comparing the objects diameter at first contact and after the grasping movement has come to rest. It turns out in practice, that the object's initial size cannot be measured accurately enough if the grasp applies torque. In that case, the contact surface between object and both sensor matrices is tilted leading to an overestimation of the real diameter. This asymmetry disappears when all forces reach an equilibrium state. In order to get more robust object size features, the pressure profile's centroids of the end

position frame is used to recalculate the diameter during each step of the grasp. Robust minimal and maximal diameter values are then chosen to estimate the difference.

2.3 Standard deviation of intensity values

Pressure profiles of rigid bodies differ significantly from those of soft objects. The impression on the tactile sensor frame appears smoothed out and lacks sharp edges. The standard deviation of all active sensor cell's intensity values represents an easy way to capture this effect. Note that this feature differs from the second central two-dimensional image moment.

$$\sigma = \sqrt{\frac{1}{N} \sum_{i=1}^N (x_i - \mu)^2}, \quad \text{where } \mu = \frac{1}{N} \sum_{i=1}^N x_i$$

2.4 Translation and rotation invariant Chebyshev moments

Since the spatial resolution of the tactile sensor matrices in use is only 6 by 14 taxels, the number of available feature extraction methods is limited. In addition, certain invariance properties are also crucial for the classification success. For example, the extracted features should be invariant to the actual position of the contact area. Rotating the object inside the hand (or vice versa) should not influence the outcome. But similar impressions of different sizes do indicate different objects. To sum it up, the required set of features has to be translation- and rotation invariant but not scale invariant.

Since the original publication of Hu (1962) image moments are a well known concept in computer vision. They are widely used in pattern-recognition and shape description applications as they capture characteristic properties of a segmented object. Originating in mechanics where they describe the mass distribution of physical bodies, moments found their way into statistics to characterize the shape of probability density functions. Features like mean, variance, and skewness for instance are commonly referred to as first-, second central- and third standardized moment. Interpreting an image as a spatial random variable gives rise to the notion of images moments. Flusser et al. (2009) provides extensive information on the topic.

For the two-dimensional tactile sensor matrix $T_s(x, y)$ the *geometric moments* of order $(p + q)$ are defined as:

$$m_{pq} = \sum_{y=0}^{M-1} \sum_{x=0}^{N-1} x^p y^q T_s(x, y)$$

Geometric image moments contain redundant information since the basis set is not orthogonal. Replacing them with Chebyshev polynomials gives rise to Chebyshev moments:

$$T_{pq} = \frac{1}{\rho(p, N)\rho(q, N)} \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} t_p(x)t_q(y)f(x, y)$$

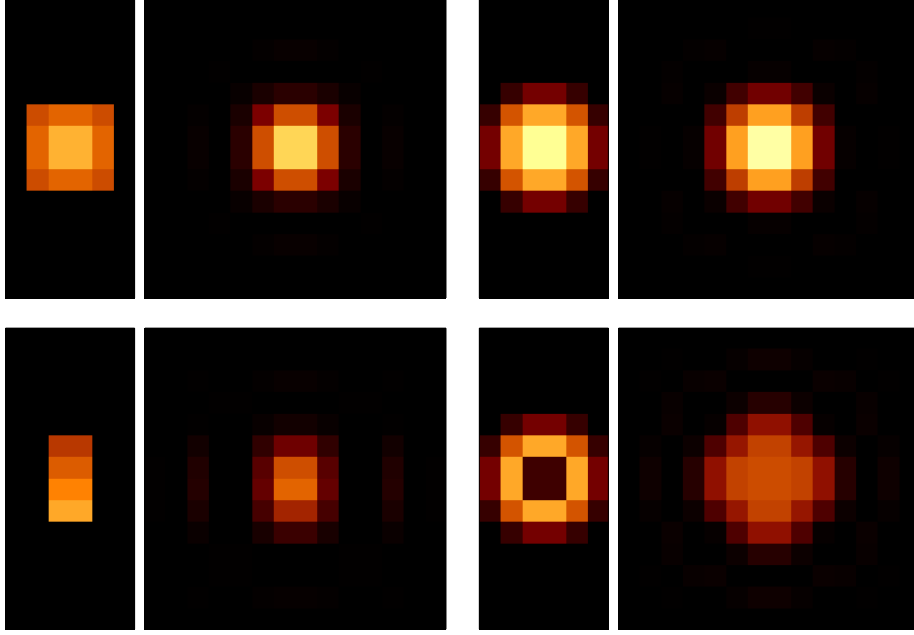


Figure 2: Reconstruction from Chebyshev moments: The left image of each pair represents the original matrix while the right frame is the reconstruction. Maximal moment orders $p, q = 5$

where the Chebyshev polynomials can be defined as:

$$t_n(x) = n! \sum_{k=0}^n (-1)^{n-k} \binom{N-1-k}{n-k} \binom{n+k}{n} \binom{x}{k}$$

The orthogonality property is achieved with:

$$\rho(p, N) = (2n)! \binom{N+n}{2n+1}, n = 0, 1, \dots, N-1$$

Translation and rotation invariance can then be constructed by replacing $t_p(x)$ and $t_p(y)$ with $t_p(x - x_0)$ and $t_p(y - y_0)$ followed by a mapping to the complex number domain. For a fast computation based on a lookup-table approach see Shu et al. (2010).

3 Results and Discussion

Scikit-learn by Pedregosa et al. (2011) offers an easy to use environment to carry out various experiments. In order to estimate the performance of the multiclass SVM approach and to avoid overfitting, 10-fold cross-validation has shown to be the method of choice for most applications. The entire dataset is therefore randomly partitioned into training- and test sets with a ratio of 80/20.

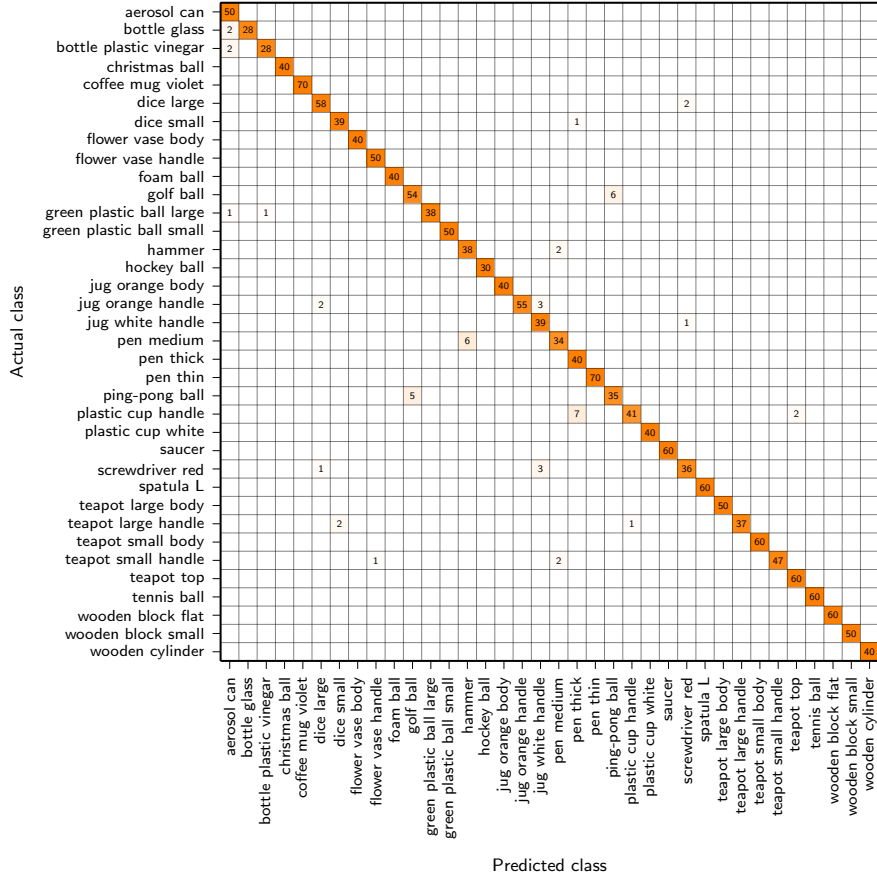


Figure 3: Cumulative confusion matrix of a randomized and stratified 10-fold cross-validation.

In addition, each fold approximately contains the same percentage of samples for each class as the entire set.

The confusion matrix of figure 3 depicts the cumulative result of the performed cross-validation. Symmetric misclassifications occur between the ping-pong- and the golf ball as well as between the medium sized pen and the hammer shaft. The distances to the separating hyperplanes in the one-vs-one classifier scheme do not have to be equal between the same pair of classifiers. A noticeable instance of such a misclassification can be observed in the case of the actual plastic cup handle and the predicted thick pen but not vice versa.

A common metric to compare the performance of classifiers is the average accuracy, i.e. the overall correctness of the model which is calculated as the sum of correct classifications divided by the total number of classifications. Precision also known as reliability describes the classifier's ability not to predict the false class and is computed from the confusion matrix. Recall also known as accuracy intuitively measures if the classifier is able to find all the positive instances of a class. Since there is a trade-off between precision and recall, the F1-score as

a weighted harmonic mean might be more suitable. Table 1 summarizes the results.

A natural extension of the presented concept includes the active exploration of grasped objects. Combining the resulting grasp-object configurations to a single object model can then be seen as an instance of *haptic perception*.

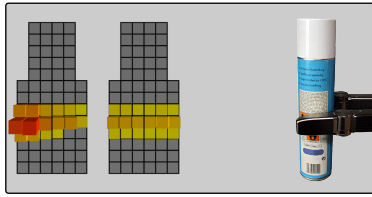
Table 1: Classification performance metrics using a randomized and stratified 10-fold cross-validation

Class	Precision (Reliability)	Recall (Accuracy)	F1 score	Support
<i>aerosol can</i>	0.909	1.000	0.952	50
<i>bottle glass</i>	1.000	0.933	0.966	30
<i>bottle plastic vinegar</i>	0.966	0.933	0.949	30
<i>christmas ball</i>	1.000	1.000	1.000	40
<i>coffee mug violet</i>	1.000	1.000	1.000	70
<i>dice large</i>	0.951	0.967	0.959	60
<i>dice small</i>	0.951	0.975	0.963	40
<i>flower vase body</i>	1.000	1.000	1.000	40
<i>flower vase handle</i>	0.980	1.000	0.990	50
<i>foam ball</i>	1.000	1.000	1.000	40
<i>golf ball</i>	0.915	0.900	0.908	60
<i>green plastic ball large</i>	1.000	0.950	0.974	40
<i>green plastic ball small</i>	1.000	1.000	1.000	50
<i>hammer</i>	0.864	0.950	0.905	40
<i>hockey ball</i>	1.000	1.000	1.000	30
<i>jug orange body</i>	1.000	1.000	1.000	40
<i>jug orange handle</i>	1.000	0.917	0.957	60
<i>jug white handle</i>	0.867	0.975	0.918	40
<i>pen medium</i>	0.895	0.850	0.872	40
<i>pen thick</i>	0.833	1.000	0.909	40
<i>pen thin</i>	1.000	1.000	1.000	70
<i>ping-pong ball</i>	0.854	0.875	0.864	40
<i>plastic cup handle</i>	0.976	0.820	0.891	50
<i>plastic cup white</i>	1.000	1.000	1.000	40
<i>saucer</i>	1.000	1.000	1.000	60
<i>screwdriver red</i>	0.923	0.900	0.911	40
<i>spatula</i>	1.000	1.000	1.000	60
<i>teapot large body</i>	1.000	1.000	1.000	50
<i>teapot large handle</i>	1.000	0.925	0.961	40
<i>teapot small body</i>	1.000	1.000	1.000	60
<i>teapot small handle</i>	1.000	0.940	0.969	50
<i>teapot top</i>	0.968	1.000	0.984	60
<i>tennis ball</i>	1.000	1.000	1.000	60
<i>wooden block flat</i>	1.000	1.000	1.000	60
<i>wooden block small</i>	1.000	1.000	1.000	50
<i>wooden cylinder</i>	1.000	1.000	1.000	40
average total	0.971	0.969	0.969	1720

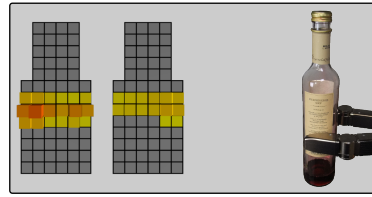
References

- Jan Flusser, Barbara Zitova, and Tomas Suk. *Moments and Moment Invariants in Pattern Recognition*. Wiley Publishing, 2009. ISBN 0470699876, 9780470699874.
- Bernd Gärtner. Fast and robust smallest enclosing balls. In *Proceedings of the 7th Annual European Symposium on Algorithms, ESA '99*, pages 325–338, London, UK, UK, 1999. Springer-Verlag. ISBN 3-540-66251-0. URL <http://dl.acm.org/citation.cfm?id=647909.740295>.
- Ming-Kuei Hu. Visual pattern recognition by moment invariants. *Information Theory, IRE Transactions on*, 8(2):179–187, February 1962. ISSN 0096-1000. doi: 10.1109/TIT.1962.1057692.
- F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12: 2825–2830, 2011.
- Huazhong Shu, Hui Zhang, Beijing Chen, Pascal Haigron, and Limin Luo. Fast computation of tchebichef moments for binary and grayscale images. *IEEE Transactions on Image Processing*, 19(12):3171–3180, 2010. URL <http://dblp.uni-trier.de/db/journals/tip/tip19.htmlShuZCHL10>.

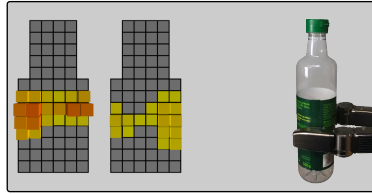
A Appendix



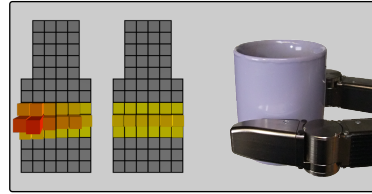
(4.1) Aerosol can



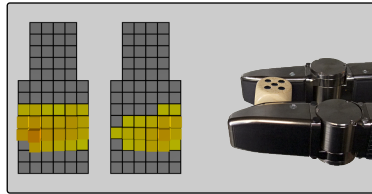
(4.2) Bottle glass



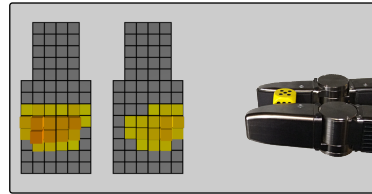
(4.3) Bottle plastic vinegar



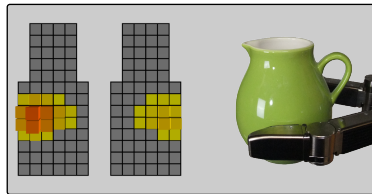
(4.4) Coffee mug violet



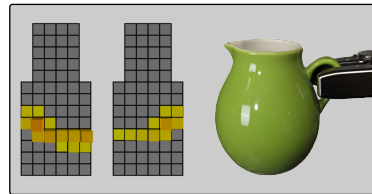
(4.5) Dice large



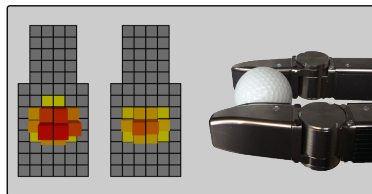
(4.6) Dice small



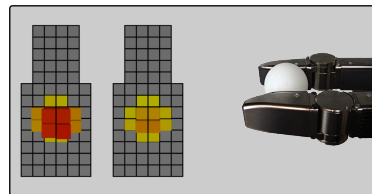
(4.7) Flower vase body



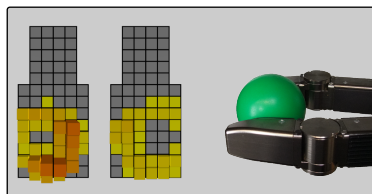
(4.8) Flower vase handle



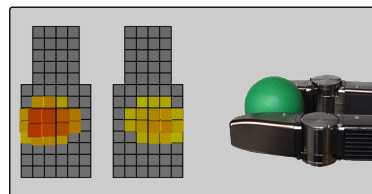
(4.9) Golf ball



(4.10) Ping-pong ball

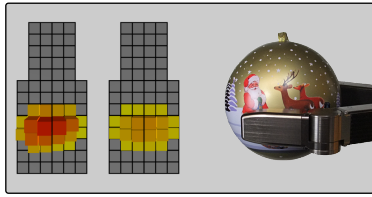


(4.11) Green plastic ball large

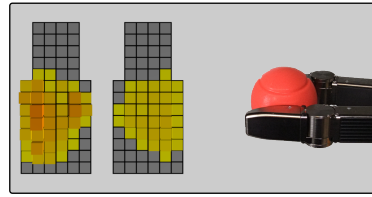


(4.12) Green plastic ball small

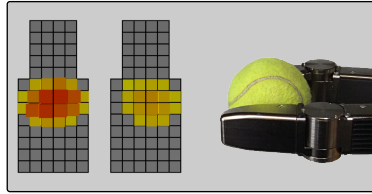
Figure 4: Learned grasp-object configurations



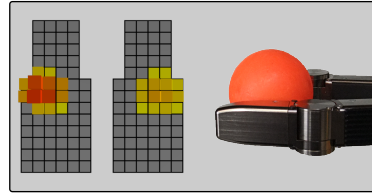
(4.13) Christmas ball



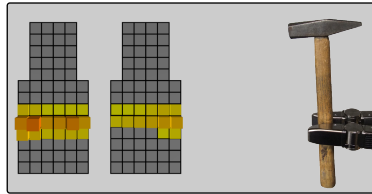
(4.14) Foam ball



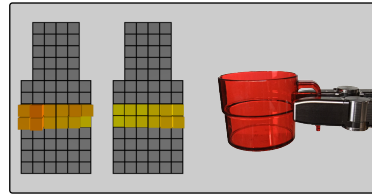
(4.15) Tennis ball



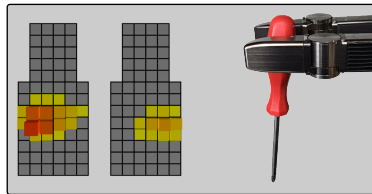
(4.16) Hockey ball



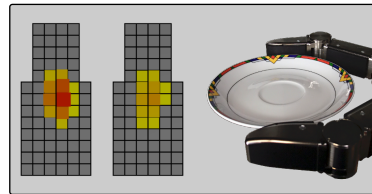
(4.17) Hammer



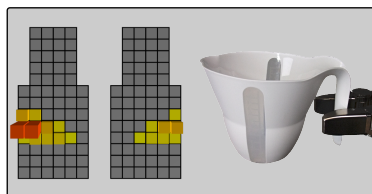
(4.18) Plastic cup handle



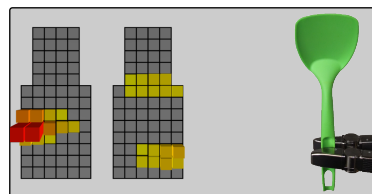
(4.19) Screwdriver



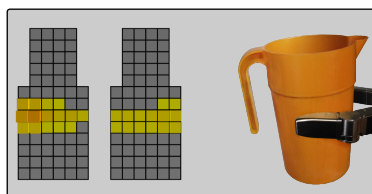
(4.20) Saucer



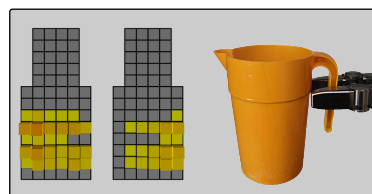
(4.21) Jug white handle



(4.22) Spatula (R)

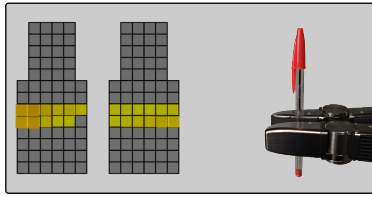


(4.23) Jug orange body

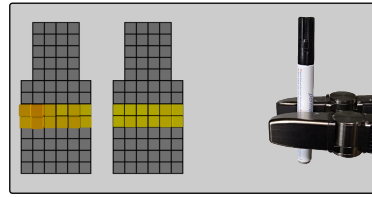


(4.24) Jug orange handle

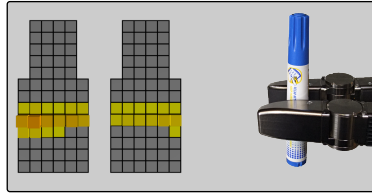
Figure 4: Learned grasp-object configurations (continued)



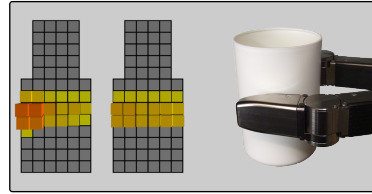
(4.25) Pen thin



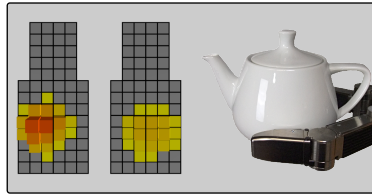
(4.26) Pen medium



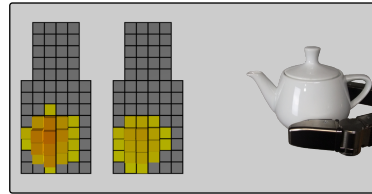
(4.27) Pen thick



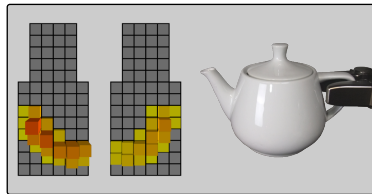
(4.28) Plastic cup white



(4.29) Teapot large body



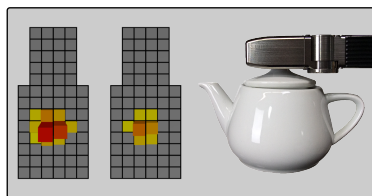
(4.30) Teapot small body



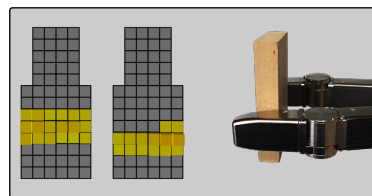
(4.31) Teapot large handle



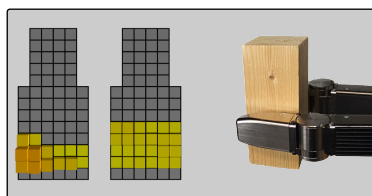
(4.32) Teapot small handle



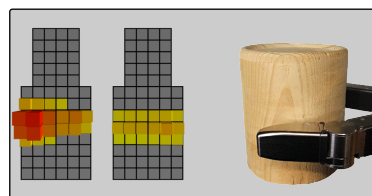
(4.33) Teapot top



(4.34) Wooden block flat



(4.35) Wooden block small



(4.36) Wooden cylinder

Figure 4: Learned grasp-object configurations (continued)

A.6 Technical Report: UNIPi Grasp Dataset performed by humans using the Pisa/IIT SoftHand

Authors Federico Spinelli, Carlos J. Rosales, Edoardo Farnioli

Info Internal technical report

Abstract This technical report describes a dataset of grasps, performed by a human operator using the Pisa/IIT SoftHand. Data was recorded using various hardware at the University of Pisa and it is composed by several grasps per object. The envisioned use of this dataset is to use the relative pose of the SoftHand, with respect to the object to be grasped, so that it is possible to associate grasps with object models.

Relation with the deliverable Dataset is useful for Task 2.2, because it provides to object models an association with grasps performed, so that grasp hypothesis can be built.

Attachment (following pages)

Grasp Dataset performed by humans using the Pisa/IIT SoftHand

Federico Spinelli - *fspinelli@gmail.com*
Carlos J. Rosales - *cjrosales@gmail.com*
Edoardo Farnioli - *edoardo.farnioli@gmail.com*

Research Center E.Piaggio, University of Pisa

INTRODUCTION

In this document we present an object grasping dataset, collected at the University of Pisa, to be used within the PaCMan project. Dataset contains a series of grasps performed on kitchen environment objects by a human operator using the Pisa/IIT SoftHand. The goal of the dataset is to provide a grasp database for each object so that the relative pose between object and SoftHand is always known during the grasp. This information can be used to attach grasps on objects and reproduce the grasp autonomously by a robot.

The dataset is composed by sub-datasets, each one containing a single grasp recording performed on a kitchen environment object. The grasps are performed by a human operator, wearing a special handle on his right arm, which supports and operates the Pisa/IIT SoftHand, the operator drives the SoftHand to grasp the object put on a table, while the system records all sensor data. Figure. 1 shows photographs taken during the acquisition of a sub-dataset. Each sub-dataset contains a single grasp type, the number of grasps associated with each object depends on its



FIGURE 1. Pictures taken during the recording of the jug sub-dataset. The operator drives the SoftHand towards the object, then grasps and lifts it, while sensor data is being recorded.

shape and grasp feasibility. A goal we kept in mind was to mimic human behaviour in grasping everyday objects, for instance a cup was grasped by the handle, by the top without putting fingers inside it or by the sides and so on. . .

Each grasp recorded is composed by a pre-grasp phase and an actual grasp phase. In the first, the operator moves the SoftHand towards the object and starts closing over the designated spot and in the latter the SoftHand closes on the object and lifts it up for a few seconds, then the object is put back on the table and the record stops.

The user is able to distinguish between these two phases by reading sensor data, for instance joint positions of each SoftHand finger is being recorded, so one can notice when the hand starts closing on the object or when it is just moving towards it.

The dataset is then populated by a few seconds long records, each classified in sub-datasets according to the object name and grasp type. On each record the user has constant access to relative and absolute hand posture, object pose, joint positions and point clouds of the whole scene. In fact the philosophy of the dataset was to collect as much data as possible during the grasp and then give the user the flexibility to chose which data to use for his application. The dataset is available for download from the PaCMan project website, under the “datasets” page [2].

The next sections describe which hardware was used during records and how it was configured, which software was used and a description and usage of the dataset.

HARDWARE SETUP

The whole system used to capture grasp recordings is composed by the following subsystems:

- PhaseSpace Impulse Motion Tracking System.
- Pisa/IIT SoftHand.
- Handle for SoftHand.
- Flexi Force glove for SoftHand.
- Asus Xtion Pro Live.

The PhaseSpace Impulse system captures real-time motion by using cameras and LEDs. The cameras detect the positions of the LEDs, which can be identified via an ID, and transmit the position of each LED in real-time at a frequency of 480Hz. The main use of this system is to track the position of the SoftHand and the object to be grasped, so that the user has access to these poses.

To accomplish this feature two star-like plastic prints were created to hold five LEDs each, one star was fixed to the SoftHand, the other on the object to be grasped. Once calibrated the system identifies the two stars and attach a local reference system to them, which can be used to calculate the SoftHand pose and the object’s. In Figure. 2 shows pictures of the stars attached to the object and the SoftHand.

The Pisa/IIT SoftHand was dressed with a special glove, called Flexi Force Glove, also designed at the University of Pisa. Glove readings give a measure of finger bending and consequently joint angle values are estimated to be one third of the finger bending. This is a good approximation considering the SoftHand synergies and the type of application. Figure. 3 shows the SoftHand with and without the glove.

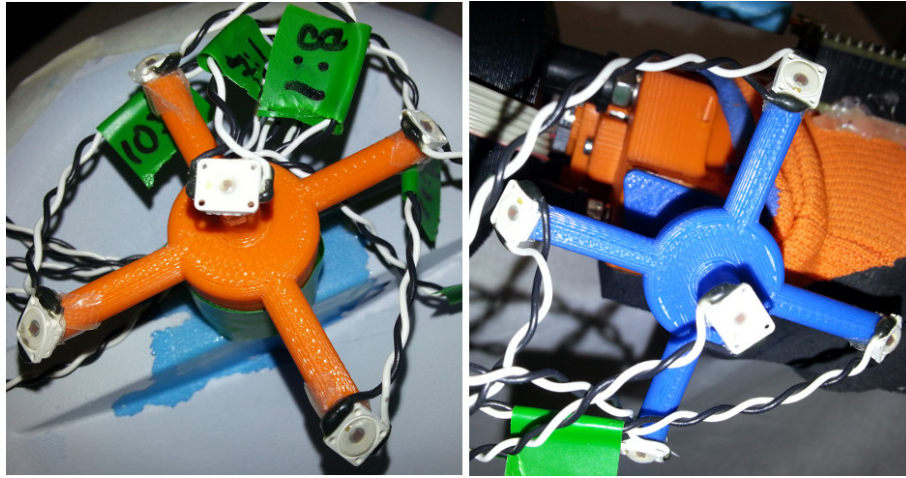


FIGURE 2. The star with five LEDs used to track the pose of objects (left). The star used to track the SoftHand (right). The five unique LED IDs and their relative position, identify a local reference system for each star, which are then tracked by the PhaseSpace system.

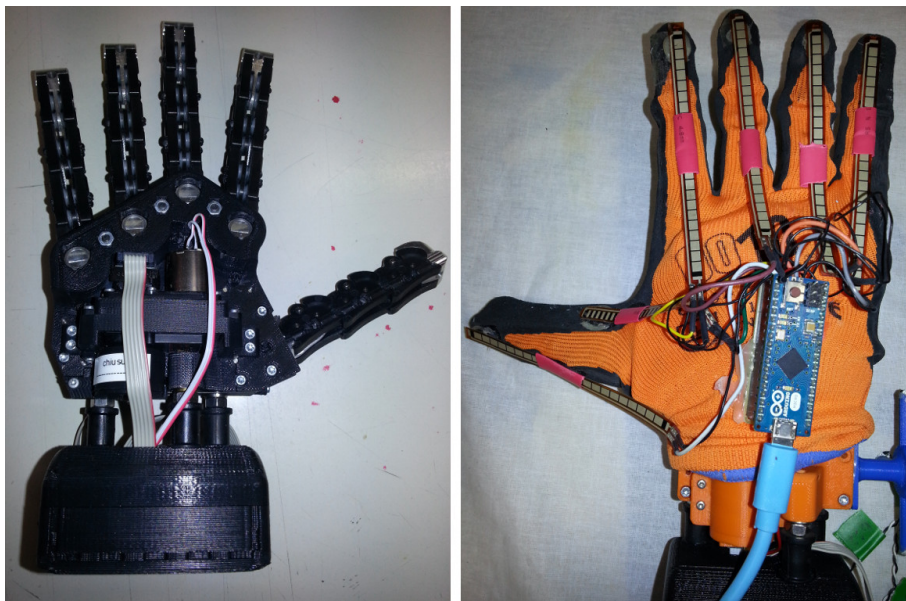


FIGURE 3. The Pisa/IIT SoftHand without the FlexiForce Glove (left) and another one with it (right). Note the sensors alongside the hand fingers which measure the angular displacements.

Additionally, to ease the operator in manoeuvring the SoftHand a special support handle was created and used. This device can be attached to the operator's arm so that he can efficiently move and operate the SoftHand, similarly as he would with his own hand. The handle has a lever to open and close the SoftHand with ease, a photograph of this device is visible in Figure. 4.



FIGURE 4. The handle for SoftHand without (left) and with (right) the operator's arm. The lever to close the SoftHand is located where the operator's hand should rest.

Finally the Asus Xtion Pro Live RGB-D sensor was used to record point clouds and images of the scene while the grasp was recorded. This not only provides the user with more visual feedback, but it also gives more data that could be post processed to analyze the grasp in a perceptual sense.

SOFTWARE SETUP

The software employed was developed under the Robot Operating System (ROS) framework [3]. ROS is an open-source set of software and tools widely used in robotics community, that provides structured communications between heterogeneous hardware designed for robotics application in general.

For our scenario we needed a common platform to communicate with all the various hardware we used, so that recorded data is meaningful. The software is available at the project website [2], along with a description of the various packages used, the operative procedure to execute the grasps and to play them back in any system without the need of the hardware we used.

As a brief summary, the software employs a communication framework like in Figure. 5, where all devices are configured and calibrated together.

To achieve this level of design, the various systems need to be calibrated between each other, in particular:

- (1) First star and SoftHand, so that PhaseSpace can track SoftHand pose.
- (2) PhaseSpace and Asus Xtion, to provide PointClouds and images consistent with the PhaseSpace.
- (3) Second star and the object to be grasped, so that PhaseSpace can track the object pose.

Calibration one was performed by knowing the SoftHand and star cad models and where the latter was fixed on the hand. So with concatenations of static geometric transformations, the two systems were calibrated with each other. Since the transformation obtained is static and the star was never removed from the hand, the first calibration was performed just once and was not reported in Figure. 5 for simplicity.

Second calibration is to be performed every time the system is rebooted and typically was performed once before the start of a recording session. This calibration

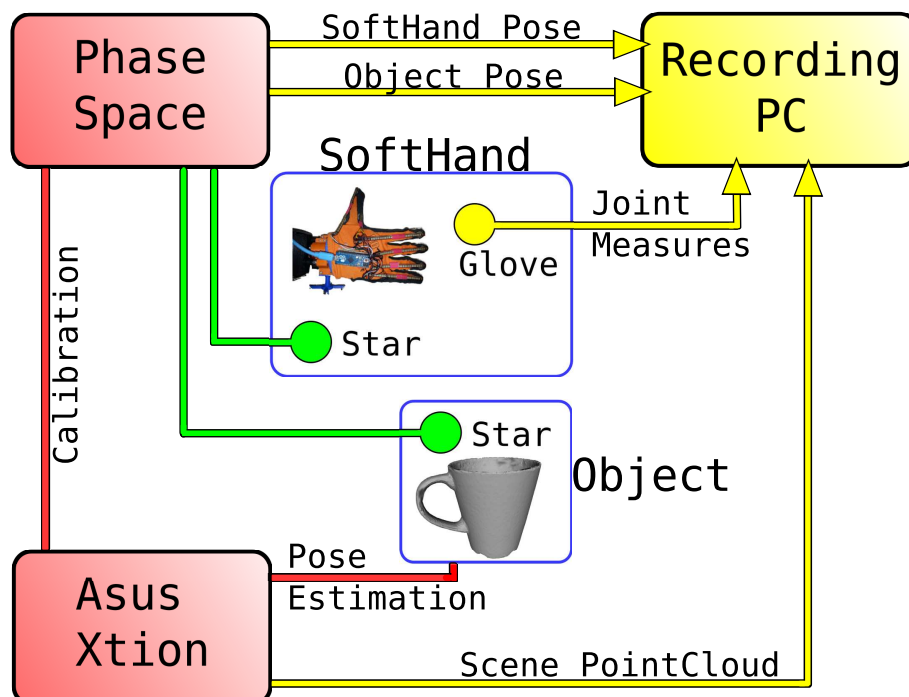


FIGURE 5. Software infrastructure during grasp recording. Green connections represent PhaseSpace tracking of the two stars, red ones the calibration steps needed in order to communicate with different systems. Finally yellow connections represent recorded data during the grasp.

procedure is available and detailed on the project website [2] under the calibration section of grasp dataset, so here, it is explained briefly. The idea is to let the two systems track the same object, that we called the calibrator, so that they both produce the same local reference system as output. If both tracking are precise the transformation between the two systems can be calculated from concatenations of known transformations.

The last calibration is more involving, since it has to be performed every time an object is selected for grasping or an old one is replaced. This is performed by knowing the pose of the second star with respect to the PhaseSpace and the pose of the object with respect to the Asus camera, now since the two systems are calibrated with each other (from previous calibration), the pose of the object with respect to the star is obtained by concatenating known transformations. The calibration remains valid until the object is replaced or the star is moved from its fixed position. Thus to obtain the last calibration we needed a procedure to obtain the object pose estimation with respect to the camera. To this end we developed a software to achieve online pose estimation of various objects with the Asus Xtion sensor and the Point Cloud Library (PCL) [1], and can be summarized with the following.

The procedure employs four global 3D descriptors to obtain multiple representations of the same object geometry and combines these descriptors together with the aim of improving the performance and robustness of object recognition and

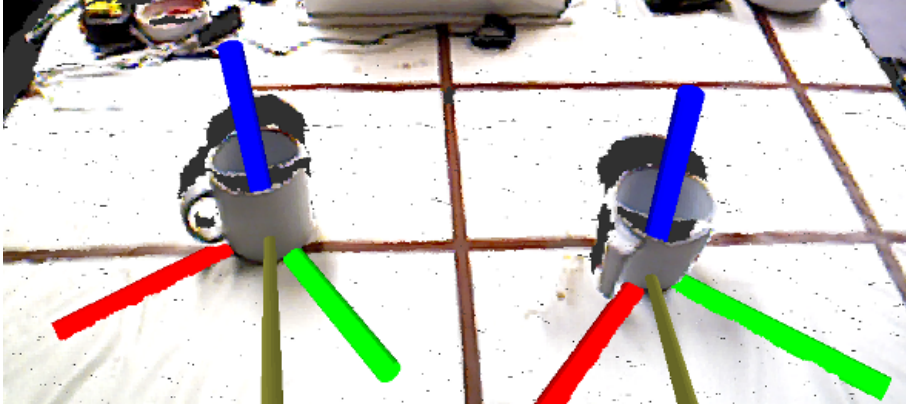


FIGURE 6. A successful test result for the pose estimation procedure. Here two cups are correctly recognized from a scene and a local reference system is attached at their bottom, giving the pose estimation of the two objects with respect to the camera. The local reference system is built with z-axis (blue) aligned with the object vertical axis and x-axis (red) aligned with the cup handle.

pose estimation. Then given an unknown object and database of known poses, for each descriptor, the procedure generates a list of “k” nearest neighbors. Then, it combines these lists for the generation of a composite list, which contains the neighbors agreed by the majority of the descriptors. Finally, it extracts the best neighbor from this list by using an iterative refinement procedure that uses Iterative Closest Point (ICP) algorithm [4]. The procedure was tested with real data acquired with the Asus Xtion Pro sensor, preliminary results show improvements in object recognition capabilities of features and consequently in pose estimation performance. The procedure was successfully applied to our grasp dataset software and is capable of providing a pose estimation in a few seconds. A test result for this procedure is visible in Figure. 6, the interested reader can find more details on the pose estimation procedure on this master thesis [5].

DATA DESCRIPTION AND USAGE

Grasp recordings that populate each sub-dataset contain all the information gathered by all packages involved in a single file. This special file, called bag file in ROS terminology, is an archive of all messages published in the software infrastructure during the grasp, a timestamp is also included to give all data a proper timeline. So from the software infrastructure point of view, playing back the recorded bag file is like performing the same grasp again. Putting things in another way, the user has access to all the information he would have if he had performed the grasp live. In particular the user can extract the following from the bag file:

- Transformations at any given time between all reference systems present: the SoftHand (palm, fingers, phalanges), object, stars, cameras and PhaseSpace.
- Readings of finger bending from the FlexiForce glove at any given time.
- Point Clouds and images recorded from the camera.

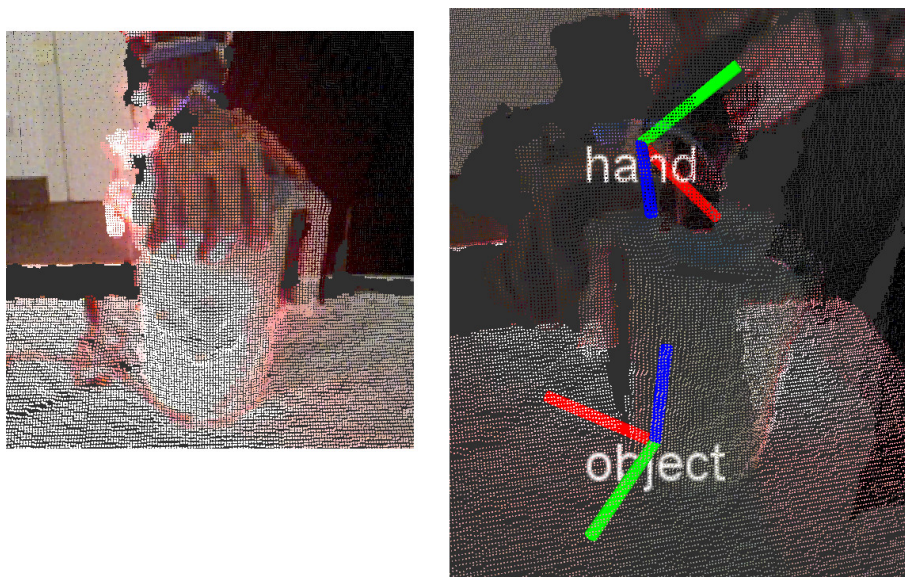


FIGURE 7. Some data extracted from a recording. A point cloud acquired during the grasp phase (left) and another point cloud taken during pre grasp with hand and object pose visible on screen (right).

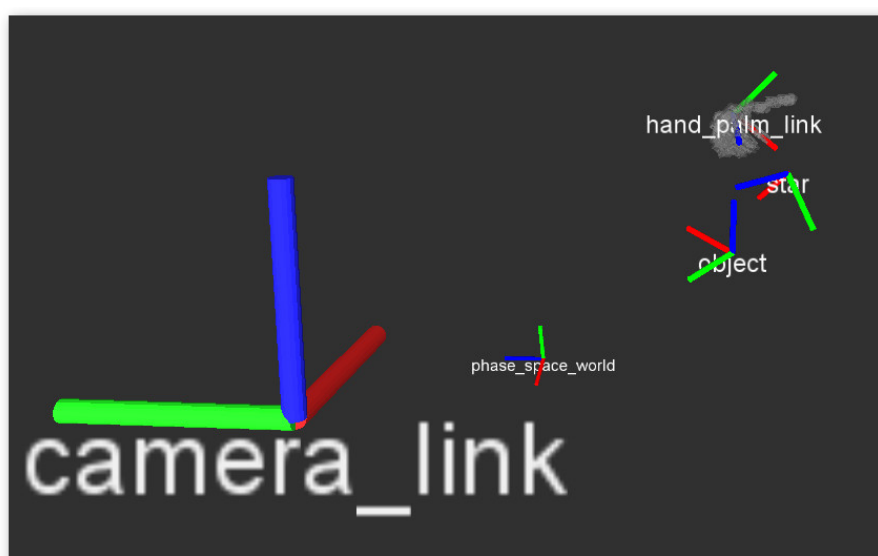


FIGURE 8. Some other data extracted from a recording. The pose of some systems during a grasp are visible on screen, they are from left to right: the Asus camera, the PhaseSpace, the object, the SoftHand palm and the star fixed on the object. A mesh model of the SoftHand is also reproduced on screen to visually identify the hand posture.

This type of approach was preferred, because it's compact and yet leaves the user with the flexibility to chose what to extract, furthermore is easily expandable

with additional hardware or improvable for future situations. Figures. 7 and 8 give some examples of what can be extracted from the recordings.

The dataset is accessible from the project website [2] under “grasp datasets” voice. At present day the dataset is still under development, but it is planned to be improved in number and quality of grasps.

REFERENCES

- [1] Point cloud library ([PCL](#)).
- [2] Probabilistic and compositional representations for object manipulation (pacman) - [Datasets page](#).
- [3] Robot operating system ([ROS](#)).
- [4] Paul J. Besl and Neil D. McKay. A method for registration of 3-d shapes. *IEEE Trans. Pattern Anal. Mach. Intell.*, 14(2):239–256, February 1992.
- [5] Federico Spinelli. [Optimal Fusion of 3D Feature Descriptors for Pose Estimation in Robust Grasping Applications](#). Master’s thesis, University of Pisa, 2014.

A.7 Technical Report: A New Approach to 3D Object Recognition and Pose Estimation

Authors Michela Antonelli, Marco Gabiccini, Francesco Marcelloni

Info Internal technical report

Abstract In this paper, we approach the problem of 3D object recognition and pose estimation by employing a 3D local descriptor, namely Unique Signatures of Histograms for Local Surface Description (SHOT). In particular, we describe each object by means of a set of SHOT descriptors, one for every keypoint identified in the point cloud representing the object itself. Since each SHOT descriptor is represented by a histogram, each object is defined by a set of histograms. Given a database of point clouds of known objects along with their set of SHOT descriptors, the classification of an unknown object is performed by, first, computing the SHOT descriptors for the object and then searching for the object in the database with the most similar set of SHOT descriptors. To compare sets of SHOT descriptors, we introduce a new measure of similarity. We first build a similarity histogram for each object in the database by comparing the sets of SHOT descriptors of this object with the ones of the unknown object. Then, we extract five features from this similarity histogram. We use these features as input to a feed forward neural network (FFNN) that evaluates the similarity between the two objects: the unknown object is recognized as the object in the database with the highest similarity. We tested our method on two datasets, one available online and the other acquired in our laboratory. On both the datasets, the results are quite promising.

Relation with the deliverable This paper is a contribution to Task 2.4 on models of the workspace.

Attachment (following pages)

A New Approach to 3D Object Recognition and Pose Estimation

Michela Antonelli

Dip. di Ingegneria dell'Informazione
University of Pisa
56122, Pisa, Italy

Email: michela.antonelli@iet.unipi.it

Marco Gabiccini

Dip. di Ingegneria Civile e Industriale,
University of Pisa,
56122, Pisa, Italy

email: marco.gabiccini@unipi.it

Francesco Marcelloni

Dip. di Ingegneria dell'Informazione,
University of Pisa,
56122, Pisa, Italy

Email: francesco.marcelloni@unipi.it

I. INTRODUCTION

In the last years, object recognition and pose estimation have become a key passage in several application scenarios, such as robot grasping and manipulation, scene recognition and understanding, large environment mapping and human posture recognition [1], [2], [3], [4]. The term pose estimation usually refers to the computation of a Special Euclidean transformation with 6 degrees of freedom, denoted SE(3) in the literature, between a model recognized from a database and its instance in the scene under analysis. This task is often paired with object recognition based on 3D descriptors [5], [6], [7], [8], [9].

In this paper, we focus on 3D images represented by point clouds acquired by an RGB-D depth camera. The geometry of a given object can be described by means of two different types of approaches: a local approach [10], [11], [12], [13] and a global approach [14], [15], [16], [17], [18]. The former determines some points of interest on the object, called *keypoints*, and calculates the descriptors on each spherical volume around these keypoints. The latter describes the object by means of features calculated on the point cloud taken as a whole (i.e., a single descriptor is calculated for each object). Local descriptors can be applied directly on cluttered scenes and are usually more accurate in describing the object. On the other hand, they are more complex in terms of both computational time and memory occupation than global descriptors.

In this paper, we approach the 3D object recognition and pose estimation by employing a 3D local descriptor, namely the Unique Signatures of Histograms for Local Surface Description (SHOT). In particular, we describe each object by means of a set of SHOT descriptors, one for every keypoint identified in the point cloud representing the object itself. Since each SHOT descriptor is represented by a histogram, each object is defined by a set of histograms.

Given a database of point clouds of known objects along with their set of SHOT descriptors, the classification of an unknown object is performed by, first, computing the SHOT descriptors for the object and then searching for the object in the database with the most similar set of SHOT descriptors.

To compare sets of SHOT descriptors, we introduce a new measure of similarity. We first build a similarity histogram for

each object in the database by comparing the sets of SHOT descriptors of this object with the ones of the unknown object. Then, we extract five features from this similarity histogram. We use these features as input to a feed forward neural network (FFNN) that evaluates the similarity between the two objects: the unknown object is recognized as the object in the database with the highest similarity.

We tested our method on two datasets, one available online and the other acquired in our laboratory. On both the datasets, the results are quite promising.

In Section II, we introduce an overview of the system. From Section III to Section VI, we describe each module of the system in detail. Section VII presents some experimental results and finally Section VIII draws some conclusion.

II. SYSTEM OVERVIEW

Fig.1 shows an overview of the system where each step of the object recognition and pose estimation process is highlighted. The input to the system is a point cloud of a scene recorded by means of an RGB-D depth camera. The point cloud is processed by the *scene pre-processing* step. This step first performs a scene segmentation for isolating the point cloud of the unknown object (O_x). Then, it removes the outliers from the point cloud of the isolated object since these outliers could affect the estimation of local point cloud characteristics. Finally, a down-sampling of the point cloud is applied for reducing the number of points that represent each object.

The point cloud of O_x , therefore, undergoes the step of *object descriptor computation* that computes the 3D local descriptors used to characterize the point cloud. In particular, we determine a set of keypoints on the object and calculate a descriptor for each of them. The output of the object descriptor extraction is a set of 3D descriptors that characterize O_x and are used in the subsequent steps.

To recognize the unknown object, we first calculate the similarity histogram between O_x and each object contained in the object database. The similarity histogram is built by comparing the set of descriptors calculated on O_x with the set of descriptors calculated offline on the labeled objects contained in the object database; similar objects will have a similarity histogram with high values around zero. Then, a

set of meaningful features are extracted from the similarity histogram: these features are used as inputs of a feed forward neural network (FFNN). The output of FFNN is the similarity degree between the two objects, which have generated the similarity histogram. Finally, the system assigns to the unknown object the label of the object that has obtained the highest similarity degree.

In the following sections, each step of the system is described in more detail.

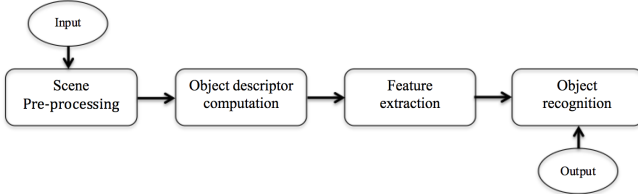


Fig. 1. System overview

III. SCENE PRE-PROCESSING

The *scene pre-processing* step is applied to a 3D image recorded by an RGB-D depth camera. The scene is set up by a table with few objects on the top. An example of an image of a scene containing a table with a cup, a funnel and a container on the top is shown in Fig. 2. The scene pre-processing step first removes the table from the scene, then applies a filter to eliminate the outliers and finally segments the scene for extracting the single object. Since this paper mainly focuses on recognition of objects and their pose estimation, we consider simple scenes consisting of only few objects on the table without any occlusion. Further, to extract the point clouds related to each object we follow the approach introduced in [19] by applying the functions defined in the PCL library [20]. In particular, we employ the following procedure:

- we find the area related to the table by looking for clusters of points with planar shapes. In our experiments, the clusters with planar shapes will be related to both the floor and the table. Between these two clusters, we identify the cluster representing the table as the one with the lower number of points;
- we calculate the average value of the height of the table (i.e., the distance between the floor and the table);
- we remove the table from the scene;
- we apply the filter to remove the outliers;
- we find all other clusters related to the other objects in the scene;
- we identify the objects as the clusters that have the same height of the table.

The process of outlier removal is based on the method proposed in [21]. An outlier is a point that is distant from other points: an outlier could arise from sensor noise or simply from a bad acquisition (bad light). To detect an outlier, we use a statistical analysis of each point with respect to its neighborhood.

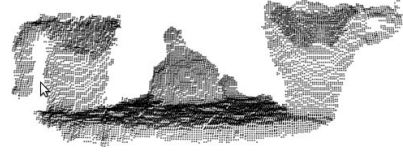


Fig. 2. An example of RGB-D image of a scene containing a table with a cup, a funnel and a container on the top (from the right to the left)

Let p , P^K and $\bar{d}_p$ be a point of the point cloud P , the set of k nearest points of p , and the mean distance between p and each point in P^K , respectively. We calculate the mean μ_p and the standard deviation σ_p of the k nearest points of p and remove from the point cloud the points that are not in the set $P^* = \{p^* \in P | (\mu_p - \alpha * \sigma_p) \leq d^* \leq (\mu_p + \alpha * \sigma_p)\}$

The parameter α is a weight that controls the width of the interval, making the filtering process more or less restrictive (we consider $\alpha = 1$ in our experiments).

Figure 3 shows the results of the pre-processing step applied to the scene represented in Fig. 2. In particular, Figs. 3(a), 3(b) and 3(c) show the point clouds after removing the points belonging to the table, after the application of the outlier removal filter and after the segmentation of the three detected objects, respectively. We observe that the points belonging to the point cloud of the three objects have different colors. After the segmentation, the point clouds of the objects are processed independently of each other, and undergo to object descriptor extraction.

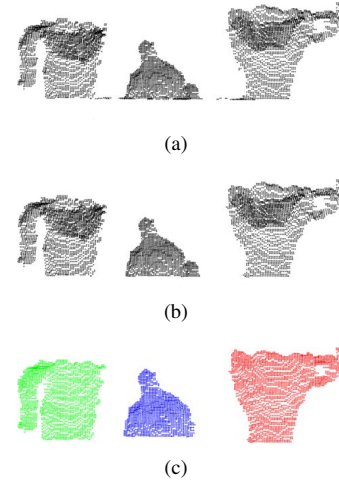


Fig. 3. (a) Point cloud of the scene without the table, (b) point cloud after the application of the filter for removing the outliers, (c) segmented objects

IV. OBJECT DESCRIPTOR COMPUTATION

Once the point cloud of the unknown object has been isolated, we compute a set of descriptors which characterize the point cloud of the object. To this aim, we use a local approach, which determines some keypoints on the object and calculate the descriptors on a spherical volume around these keypoints. In particular, we use the SIFT Keypoint Detector

algorithm [22] to identify the keypoints from the point cloud. This algorithm uses a cascade filtering approach. First, a difference-of-Gaussian function is used to identify potential points of interest that are invariant to scale and orientation. Then the points that have low contrast or are poorly localized along an edge are rejected. Figure 4 highlights in red the keypoints extracted from the point cloud of a bottle.



Fig. 4. Keypoints (in red) extracted from the point cloud of a bottle

For each keypoint, we compute the Signatures of Histograms of Orientation (SHOT) descriptor [23]. Following the categorization introduced in [23], the state-of-the-art 3D descriptors can be grouped into two main categories: Signatures-based methods and Histograms-based methods. The first methods describe the neighbourhood of a keypoint (i.e., the support of a keypoint) by encoding geometric measurements calculated with respect to the coordinates according to an invariant local reference frame (RF) defined on the support. The second methods describe the support of a keypoint by encoding counters of local geometrical or topological measurements into histograms. SHOT belongs to the intersection of these two categories. Indeed, beside encoding histograms of the normals within the support, it also adds geometrical information regarding the location of the points within the support by computing a set of local histograms on a 3D volume generated by superimposing a 3D grid on the support. In particular, let $K = \{K_1, K_2, \dots, K_n\}$, S_{K_i} and $\mathbf{n}_l$ be the set of keypoints, the set of points within a sphere of radius R centered into K_i and the normal at the l -th point of the point cloud. For each $k_i \in K$, the SHOT descriptor is calculated as follows:

- 1) superimpose a spherical grid S_{K_i} , centered in K_i and divided into 32 volumes (see Fig. 5): each volume results from 4 azimuth, 2 radial and 2 elevation divisions of the sphere;
- 2) calculate the local RF using all the points in S_{K_i} ;
- 3) for each volume V_j , with $j \in \{1, \dots, 32\}$, calculate a local histogram of 11 bins by accumulating point counts depending on the angle θ_q between the normal $\mathbf{n}_l$ at the l -th point within V_j and the normal at the keypoint K_i (see Fig. 5);
- 4) merge together all the local histograms obtaining a descriptor length of 352;
- 5) normalize the descriptor to Euclidean norm.

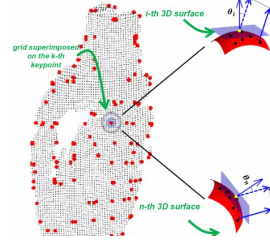


Fig. 5. Computation of the SHOT descriptors on the point cloud of a bottle

V. FEATURE EXTRACTION

After executing the object descriptor computation, each object is characterized by a set of SHOT descriptors, one for each keypoint detected in the point cloud. Let us assume that a database of known objects along with the corresponding sets of descriptors is available. Then, the descriptors of an unknown object (*query object*) are compared with the sets of descriptors of all the objects in the database: the query object is labeled with the class of the known object with the most similar sets of descriptors to the ones of the query object. Since each object is described by a set of histograms, we have to introduce a measure of similarity between sets of histograms.

Let O_d and O_x be a known object of the dataset and the query object, respectively. To define the similarity between O_d and O_x , first of all, we compute a histogram, the *similarity histogram*, as follows. For each keypoint K_i of O_x , we search for the nearest keypoint K_t of O_d and store the distance d_{it} . The distance between the keypoints is computed as the distance between the corresponding histograms. This distance is calculated by considering the histograms as vectors (each value of the bin is an element of the vector) and therefore applying the Euclidean distance between them. After computing all the distances d_{it} for each keypoint K_i , we plot these distances in a histogram. The x-axis of the histogram represents the distances from 0 to D_{MAX} : D_{MAX} is the maximum distance we consider (in the experiments, $D_{MAX} = 0.6$). If a distance is larger than D_{MAX} , this distance is inserted into the last bin. The x-axis is partitioned into M equi-width bins. If the query object O_x and the object O_d are close to each other, the histogram is skewed right since distances close to 0 are more frequent than large distances; otherwise, it should be skewed left. Figures 6(a) and 6(b) show two examples of similarity histograms calculated, respectively, on two equal and two different objects. As expected, the distribution of the distances is different in the two cases.

From each similarity histogram, we extract a set of features that are used as inputs to the FFNN. Let $D = \{d_1, \dots, d_M\}$ be the set of bins of the similarity histogram. Let $Hs = \{h_1, h_2, \dots, h_M\}$ be the set of occurrences of the distances in the corresponding bins. We extract the following features:

- i) Expected value:

$$\mu = \frac{1}{NK_{O_x}} \sum_{i=1}^N h_i \cdot d_i \quad (1)$$

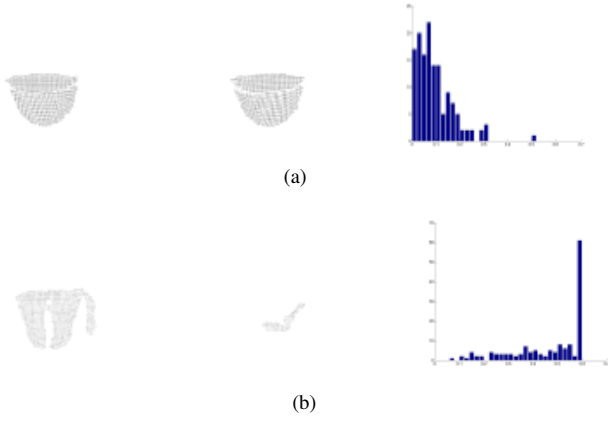


Fig. 6. Similarity histograms calculated on two point clouds (a) of the same object in the same pose and (b) of different objects

where $NK_{O_x} = \sum_{i=1}^N h_i$ is the number of SHOT descriptors considered, which coincides with the number of keypoints of the unknown object O_x .

ii) Variance:

$$\mu = \frac{1}{NK_{O_x}} \sum_{i=1}^N h_i \cdot d_i^2 - \mu^2 \quad (2)$$

iii) Standard Deviation:

$$\sigma = \sqrt{\frac{h_i}{NK_{O_x}} \cdot (d_i - \mu)^2} \quad (3)$$

iv) Symmetry:

$$\gamma = \frac{1}{NK_{O_x} \cdot \sigma^3} \sum_{i=1}^N (d_i - \mu)^3 \cdot \frac{h_i}{N_D} \quad (4)$$

v) Different number of keypoints between the two objects:

$$ND = |NK_{O_x} - NK_{O_d}| \quad (5)$$

where NK_{O_d} is the number of keypoints calculated on the reference object O_d of the database and $||$ is the absolute value operator.

This feature takes into account that similar objects should have point clouds characterized by a similar number of keypoints.

For each pair (O_x, O_d) , the five features are used as input to a feed forward neural network (FFNN). The FFNN outputs a value in $[0,1]$ which determines the similarity between O_x and O_d .

VI. OBJECT RECOGNITION

In order to calculate the similarity between two objects we exploit the FFNN of Fig. 7.

The input layer contains as many neurons as the number of features used to describe the similarity histogram calculated between object O_d and object O_x . Each input neuron is connected to all the neurons of the hidden layer, which in

their turn are connected to the unique neuron of the output layer.

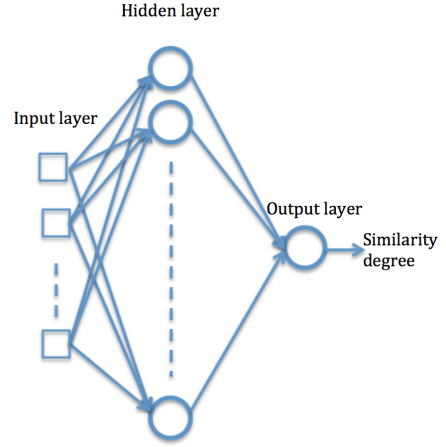


Fig. 7. Structure of the FFNN

The neurons of the hidden layer perform a weighted sum of the inputs and their outputs are calculated by applying an activation function to this weighted sum. More precisely, let x_s , with $s = 1, \dots, 5$, and $w_{s,h}$ be the inputs to the network and the weights on the connections between the neurons of the input layer and the h -th neuron of the hidden layer, respectively. The output y_h of the h -th neuron in the hidden layer is calculated by applying the activation function $f(\cdot)$ to the weighted sum of the inputs, that is,

$$y_h = f\left(\sum_{s=1}^5 x_s \cdot w_{s,h}\right) \quad (6)$$

In the experiments we have used as activation function the sigmoid function $f(x) = \frac{2}{1+e^{-2x}} - 1$.

All the H neurons of the hidden layer are connected to the single neuron n_o of the output layer. The output O of this neuron, that is the similarity degree between the unknown object O_x and one object O_d of the database, is determined by applying the sigmoid function $f(x) = \frac{1}{1+e^{-x}}$ to the weighted sum calculated as:

$$O = \sum_{h=1}^H y_h \cdot w_{h,o} \quad (7)$$

where y_h is the output of the h -th hidden neuron, and $w_{h,o}$ is the weight of the connection between the h -th hidden neuron and the output neuron. In our experiments, we adopted $H = 30$ hidden neurons.

To train the network we use a training set made up by pairs of both equal and different objects. Furthermore, as explained in the experimental part, we also consider pair of equal objects but in different pose. The output of the network is 1 either when the two objects are equal and in the same pose, or when the two objects are equal and have a symmetrical shape. We calculate the error between the output of the FFNN and the

desired output and we use the back-propagation algorithm to update the weights.

The query object O_x is recognized to be the same object and in the same pose of the object O_d with the highest degree of similarity with O_x .

VII. EXPERIMENTAL PART

The proposed approach has been tested on two datasets of household objects. The first, called the Willow Garage dataset [24], contains 31 objects. The second, called Ikea dataset, is composed of 23 objects whose point clouds have been acquired in our laboratory.

In both datasets, each object is described by 36 point clouds. Indeed, since we are interested in recognizing also the pose of the objects, each object is rotated along the vertical axis at steps of 10° and a point cloud is acquired at every step. In the following subsections, we introduce in more detail the two datasets and the obtained results.

A. The Willow garage dataset

This dataset contains 1116 point clouds (31 objects, each acquired in 36 rotations). We divide the dataset into two sets, S_{tr} , and S_{ts} , to be used for training and testing, respectively.

S_{tr} contains, for each object, the point clouds corresponding to the rotations multiple of 20° and 30° starting from 0° (0° , 20° , 30° , 40° , 60° , ..., 330° , 340°). The training set contains 713 point clouds.

The test set contains the point clouds corresponding to all the remaining rotations (10° , 50° , 70° , ..., 350°), for a total of 403 point clouds.

Since the inputs to the FFNN are pairs of objects and the output is a similarity value, the training and test sets of the FFNN have been generated as follows. As regards the training set, we made up every possible pair of point clouds contained in S_{tr} (for a total of 553,536 pairs). For each pair we use as input to the FFNN the 5 features extracted from the similarity histogram and the output of each pair is assigned using the following criterion.

We group each object into categories (cups, bottles, books, etc.) since the resolution of the RGB-D camera is not able to appreciate small differences. Further, as the final aim of this object recognition system is to infer how the object can be grasped, we do not need to distinguish between very similar objects.

For each pair of the training set we assign an output equal to 1 (equal point clouds) in the following cases:

- if they represent two objects of the same category in the same pose (same rotation along the vertical axes).
- if they represent two objects of the same category and the difference between the rotation angles is lower than $\pm 20^\circ$.
- if they represent two symmetrical objects of the same category independently of the rotation. We consider as symmetrical axes the vertical axes used to rotate the object.

Output	Target	
	0	1
0	21138	3695
1	5821	23433

TABLE I
AVERAGE CONFUSION MATRIX CALCULATED ON THE TRAINING SET

Specificity	78.4
Sensitivity	86.4
Accuracy	82.4

TABLE II
SPECIFICITY, SENSITIVITY AND ACCURACY CALCULATED ON THE TRAINING SET

Further, as examples of different objects (output of the network equal to 0), we include in the training set only the pairs made up by point clouds of objects belonging to different categories. In this way, we avoid to give wrong information to the network. Indeed, two point clouds of objects of the same category could be very similar also when the rotation is greater than $\pm 20^\circ$. For this reason, to train the network we prefer to use only pairs of objects whose equality/disequality is unambiguous.

By using the above criteria we generate a training set made up by 476,928 and 33,804 pairs whose output is equal to 0 (negative samples) and 1 (positive samples), respectively. Being this training set unbalanced, we randomly choose only 33,804 pairs among the negative samples and we remove the others from the training set. In this way we end up with a training set of 67,608 samples, 33,804 positive samples and 33,804 negative samples. This process is repeated ten times so as to avoid any bias on the random choice of the negative samples.

As regards the test set, we generate every possible pair between each point cloud of S_{ts} and the point clouds contained in S_{tr} (for a total of $403 \cdot 713 = 287,339$ pairs). We automatically assign the output using the same criteria used for the training set.

In Tab. I we show the average confusion matrix obtained on the ten trials and on the training set after 50 epochs. Table II shows the corresponding values of specificity, sensitivity, and accuracy.

In Tab. III, we show the average results calculated on the test set. To assign the output to the pairs of the test set, we consider the same criterion used for the pairs of the training set.

The results are encouraging, mainly considering that the output on some test samples could be considered wrong but, due to a particular symmetry of the object, the two point clouds could be, in fact, very similar. By improving the way of assigning the output to each pair of point clouds, the results should improve further.

Object correctly recognized	98.2%
Object correctly recognized and rotation recognized with an error of $\pm 10^\circ$	76.3%
Object recognized but wrong rotation	21.9%
Object not recognized	1.8%

TABLE III
RESULTS ON THE TEST SET

B. The Ikea dataset

We acquired the point clouds of 23 objects with the aid of a DoF turntable, with vertical axis. Objects like mugs, various containers, plates, glasses, spoons and kitchen utensils were put on the turntable center, with Xtion Pro sensor pointing at it and located at a fixed position. Figs. 8 and 9 show, respectively, the objects contained in this dataset and an example of point cloud acquired by means of the turntable.



Fig. 8. Objects used in the second experiment

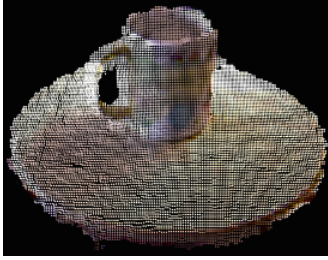


Fig. 9. Example of point cloud acquired by using the turntable

The turntable was rotated at steps of 10° : at each step the sensor acquired a point cloud, thus collecting 36 point clouds per object (828 point clouds in total). The overall process was repeated in different three days, so as to have three distinct rounds of acquisition. Since the quality of acquisition strongly depends on the ambient light, the point clouds acquired for the same object and the same orientation result to be different from one round to another.

Another factor affecting data quality is the presence of shining or semi-transparent surfaces, since these types of surface could scatter the infrared signal, causing large chunks of the point cloud to be missing or appearing very cluttered. To

Output	Target		
	0	13106	3594
	1	4755	14348

TABLE IV
AVERAGE CONFUSION MATRIX CALCULATED ON THE TRAINING SET

Specificity	73.4
Sensitivity	80.4
Accuracy	76.7

TABLE V
SPECIFICITY, SENSITIVITY AND ACCURACY CALCULATED ON THE TRAINING SET

cope with this issue, we covered sensible surfaces, like metal, plastic, and shining ceramics, by using an opaque film with the aim of reducing the signal scattering. However, intrinsic noise in the depth sensor still lowers the quality of point clouds for certain objects, as highlighted in Figure 9. We decided to keep those bad quality acquisitions in order to simulate realistic situations met by an autonomous robot. Data diversification is also ensured by the unavoidable human error in positioning the object on the table. This could lead to inaccuracies in objects acquired at the same orientation, across different rounds of acquisition. For example, it is possible that a mug, acquired at an orientation of 10° in a round, is more similar to the same mug, acquired at 20° in another round. As explained in the previous sections, before saving the point cloud we apply some pre-processing on the images in order to extract from the scene only the object.

We used, in turn, one of the three rounds of acquisition as training set, and the others as test sets (defined as R_{TR} and R_{TS} , respectively). Each round contains 828 point clouds (23 objects times 36 rotations). As explained in the previous subsection, for building the training and test sets we consider, respectively, every possible pair of the point clouds contained in R_{TR} (685,584 pairs) and every possible pair formed by the point cloud contained R_{TS} and the point clouds contained R_{TR} (1,371,168 pairs). The output of each pair has been assigned as described in the previous subsection. Tables IV and V show the average confusion matrix and the values of specificity, sensitivity and accuracy obtained on the training set.

As regards the test set, the results are shown in Tab. VI. We observe that our method obtain better results on the first dataset. Indeed, for the Ikea dataset, the recognition performance drops for both objects and rotations. This can be due to several aspects: the point clouds of the first dataset contain a lower amount of noise than the ones contained in the Ikea dataset. Further, the Ikea dataset contains a larger variety of objects and the acquisitions are performed in different conditions of light.

Object correctly recognized	76.8%
Object correctly recognized and rotation recognized with an error of $\pm 10^\circ$	58.9%
Object recognized but wrong rotation	41.1%
Object not recognized	23.2%

TABLE VI
RESULTS ON THE TEST SET

The recognition performance of the pose is quite low (58.9%) due mainly to the automatic procedure of assigning the output. Indeed, two point clouds representing the same object acquired in two different rotations, could be similar even if the difference between the two rotations is greater than 20° . Fig. 10 shows examples of pairs of point clouds (one in red and the other in green) considered erroneously equal by the system. The reader can easily realize that actually these point clouds are very similar and the error is only due to the automatic labeling procedure used to build the training and test sets.

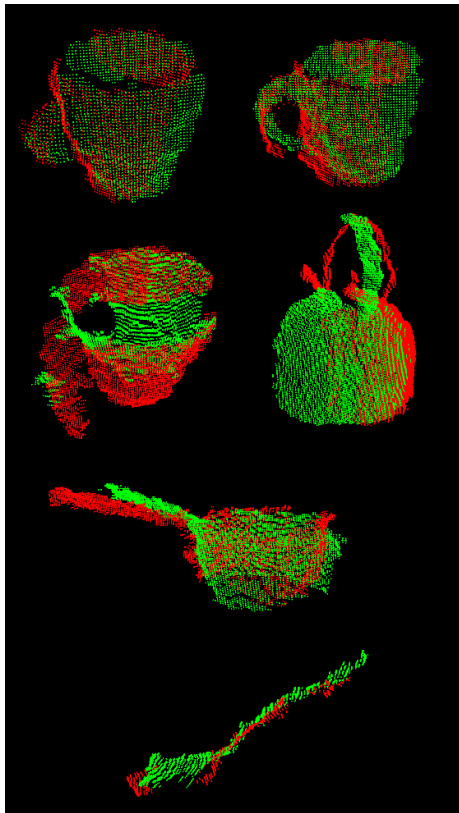


Fig. 10. Objects used in the second experiment

REFERENCES

- [1] C. English, G. Okouneva, P. Saint-Cyr, A. Choudhuri, and T. Luu, "Real-time dynamic pose estimation systems in space: Lessons learned for system design and performance evaluation," *Special issue on Quantifying the performance of intelligent systems, International Journal of Intelligent Control and Systems*, vol. 16, no. 2, pp. 79–96, 2011.
- [2] J. Bandouch, F. Engstler, and M. Beetz, "Accurate human motion capture using an ergonomics-based anthropometric human model," in *Articulated Motion and Deformable Objects*, ser. Lecture Notes in Computer Science, F. Perales and R. Fisher, Eds. Springer Berlin Heidelberg, 2008, vol. 5098, pp. 248–258.
- [3] H. Ning, W. Xu, Y. Gong, and T. Huang, "Discriminative learning of visual words for 3d human pose estimation," in *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*. IEEE, 2008, pp. 1–8.
- [4] P. Doliotis, V. Athitsos, D. Kosmopoulos, and S. Perantonis, "Hand shape and 3d pose estimation using depth data from a single cluttered frame," in *Advances in Visual Computing*. Springer, 2012, pp. 148–158.
- [5] S. May, D. Droschel, D. Holz, C. Wiesen, S. Fuchs *et al.*, "3d pose estimation and mapping with time-of-flight cameras," in *International Conference on Intelligent Robots and Systems (IROS), 3D Mapping workshop, Nice, France, 2008*.
- [6] C. Choi, Y. Taguchi, O. Tuzel, M.-Y. Liu, and S. Ramalingam, "Voting-based pose estimation for robotic assembly using a 3d sensor," in *Robotics and Automation (ICRA), 2012 IEEE International Conference on*. IEEE, 2012, pp. 1724–1731.
- [7] M. Ye, X. Wang, R. Yang, L. Ren, and M. Pollefeys, "Accurate 3d pose estimation from a single depth image," in *Computer Vision (ICCV), 2011 IEEE International Conference on*. IEEE, 2011, pp. 731–738.
- [8] H. Kollnig and H.-H. Nagel, "3d pose estimation by directly matching polyhedral models to gray value gradients," *International Journal of Computer Vision*, vol. 23, no. 3, pp. 283–302, 1997.
- [9] M. Andriluka, S. Roth, and B. Schiele, "Monocular 3d pose estimation and tracking by detection," in *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*. IEEE, 2010, pp. 623–630.
- [10] R. B. Rusu, Z. C. Marton, N. Blodow, and M. Beetz, "Persistent Point Feature Histograms for 3D Point Clouds," in *Proceedings of the 10th International Conference on Intelligent Autonomous Systems (IAS-10), Baden-Baden, Germany, 2008*.
- [11] F. Tombari, S. Salti, and L. Di Stefano, "Unique signatures of histograms for local surface description," in *Proceedings of the 11th European Conference on Computer Vision Conference on Computer Vision: Part III, ser. ECCV'10*. Berlin, Heidelberg: Springer-Verlag, 2010, pp. 356–369. [Online]. Available: <http://dl.acm.org/citation.cfm?id=1927006.1927035>
- [12] R. B. Rusu, N. Blodow, and M. Beetz, "Fast point feature histograms (fpfh) for 3d registration," in *Robotics and Automation, 2009. ICRA'09. IEEE International Conference on*. IEEE, 2009, pp. 3212–3217.
- [13] A. Johnson, "Spin-images: A representation for 3-d surface matching," Ph.D. dissertation, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, August 1997.
- [14] R. B. Rusu, A. Holzbach, M. Beetz, and G. Bradski, "Detecting and segmenting objects for mobile manipulation," in *Computer Vision Workshops (ICCV Workshops), 2009 IEEE 12th International Conference on*. IEEE, 2009, pp. 47–54.
- [15] W. Wohlkinger and M. Vincze, "Ensemble of shape functions for 3d object classification," in *Robotics and Biomimetics (ROBIO), 2011 IEEE International Conference on*, Dec 2011, pp. 2987–2992.
- [16] R. Rusu, G. Bradski, R. Thibaux, and J. Hsu, "Fast 3d recognition and pose using the viewpoint feature histogram," in *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, Oct 2010, pp. 2155–2162.
- [17] A. Aldoma, M. Vincze, N. Blodow, D. Gossow, S. Gedikli, R. Rusu, and G. Bradski, "Cad-model recognition and 6dof pose estimation using 3d cues," in *Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on*, Nov 2011, pp. 585–592.
- [18] A. Aldoma, F. Tombari, R. Rusu, and M. Vincze, "Our-cvfh oriented, unique and repeatable clustered viewpoint feature histogram for object recognition and 6dof pose estimation," in *Pattern Recognition*, ser. Lecture Notes in Computer Science, A. Pinz, T. Pock, H. Bischof, and F. Leberl, Eds. Springer Berlin Heidelberg, 2012, vol. 7476, pp. 113–122.
- [19] R. B. Rusu, *Semantic 3D Object Maps for Everyday Robot Manipulation*, ser. Springer Tracts in Advanced Robotics. Springer, 2013, vol. 85.
- [20] R. B. Rusu and S. Cousins, "3d is here: Point cloud library (pcl)," in *ICRA*. IEEE, 2011.
- [21] R. B. Rusu, Z. C. Marton, N. Blodow, M. Dolha, and M. Beetz, "Towards 3d point cloud based object maps for household environments," *Robotics and Autonomous Systems*, vol. 56, no. 11, pp. 927 – 941, 2008.

- [22] D. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.
- [23] S. Salti, F. Tombari, and L. D. Stefano, "Shot: Unique signatures of histograms for surface and texture description," *Computer Vision and Image Understanding*, vol. 125, no. 0, pp. 251 – 264, 2014.
- [24] J. Tang, S. Miller, A. Singh, and P. Abbeel, "A textured object recognition pipeline for color and depth image data." in *ICRA*. IEEE, 2012, pp. 3467–3474.