Learning and inference with a compositional hierarchical representation of 3D object shape

Dominik Belter, Marek Kopicki, Ales Leonardis and Jeremy L. Wyatt

Abstract— **Modelling of object shape is an essential precursor to any method for object manipulation. In this paper we present a method for learning compositional hierarchical representations of 2.5D and 3D object shape. These are representations that consist of a hierarchy of parts. We present novel algorithms for learning and inference. We empirically demonstrate the ability to learn these representations from real depth data, and to fill in missing information caused by self-occlusions. We also demonstrate the ability of the hierarchy to model variable shape, and to share parts across different objects and object categories.**

I. Introduction

Grasping and manipulation actions that are generalisable across objects require representations of object shape that are expressive of and robust to variations in shape. These representations must have several properties. Features, for example, must be shared across objects and object categories, as well as being robust to variability. In addition, global object shape must be inferrable from partial views, and missing parts must be inferred reliably. Compositional hierarchies are a particular class of representations that have proven themselves with respect to these properties in the domain of object categorisation from intensity images. In these representations an object or image is represented as a hierarchy of parts learned in a statistical manner. In this paper we develop a learning and inference framework for compositional hierarchical models of 3D shape. This takes view based depth images as input, rather than either intensity images, or complete shape data. We show empirically that it has properties of shareability, robustness to variability, and the ability to approximately infer missing shape information. We offer the framework as a promising representation to support future work in generalisable grasping and manipulation.

Our approach to objects represantation is based on the models which explain visual processing in human cortex [3]. According to these models the visual processing is hierarchical and all layers are learned. First layers of the hierarchy are position and scale invariant. The higher layers of the hierarchy build invariance to the viewpoint [5]. Also the receptive fields increase with the layer of the hierarchy. The HMAX model [4] assumes that the inference process is feedforward to obtain fast objects recognition.

Our approach differs to biological counterparts. We build view invariance of parts from the very beginning. Our choice is motivated by the application. Our goal is to learn representation of objects from minimal number of examples. For view-variant representations the objects used for training should be observed from many viewpoints to build sufficient vocabulary. The number of parts which are required to represent the object is acceptable for 2D models [2]. Unfortunatelly, it explodes if the vocabulary of 3D parts is built. In our approach we have to run optimization which aligns parts in 3D whenever similarity between parts has to be computed but we show that this can be performed efficiently using Umeyama method. The additional advantage of our approach is the small number of parts in vocabulary which does not require GPU implementation.

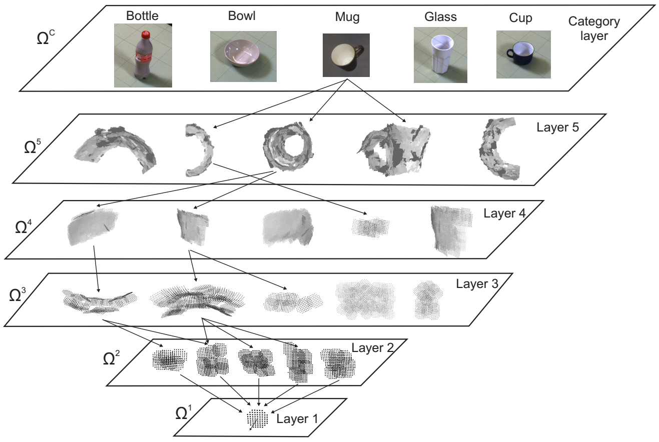

Fig. 1. Illustration of hierarchical representation of 3D objects. At i-th layer vocabulary Ω^i each word is represented by 3D shape. Arrows represent compositional relations between parts.

II. Related work

III. Learning

Our overall procedure for learning a compositional shape hierarchy is as follows. In this paper we build two linked hierarchies, one where parts are learned from single views, and one where parts are learned from many views. The view independent layers are built on top of the second view dependent layer. The hierarchy is learned layer by layer, with each layer consisting of a part vocabulary. Each part in a vocabulary is also called a word. Instantiations of parts caused by data are called realisations.

First, starting with a layer consisting of a single seed part, we learn two additional layers of parts from depth images obtained from particular viewpoints. The receptive fields for these parts are areas of the image plane. Then, to learn the three view-independent layers above these, the hierarchy brings together different views of the same parts so as to create view independent 3D shape parts. The receptive fields for these parts are 3D volumes. It should be noted that the first three layers are view dependent in that they are each

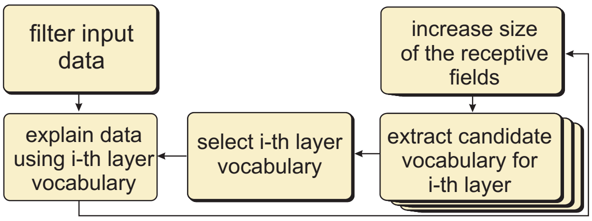
filter input data
increase size of the receptive fields
explain data using i-th layer vocabulary
select i-th layer vocabulary
extract candidate vocabulary for i-th layer

Fig. 2. General procedure of learning the hierarchical representation of objects

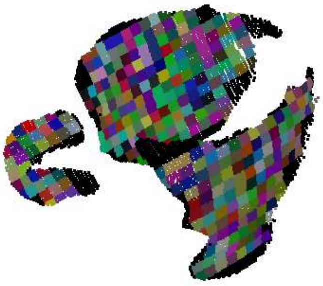

Fig. 3. A point cloud obtained from single camera view divided into first layer receptive fields. Each colour represents a receptive field. Black points are not used to compute planar patches

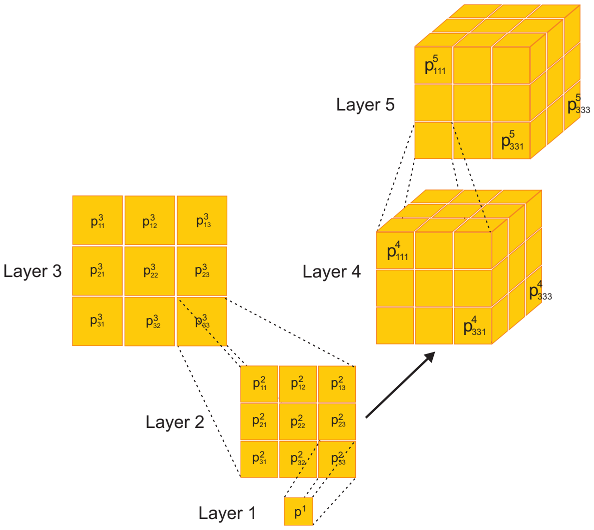

Fig. 4. Representation of receptive fields in the hierarchy. Each receptive field at the lowest layer is 5×5 pixels. Arrows represent the surface normals for those planar patches. First three layers of the hierarchy are defined on the image plane. The next two layers are volumetric.

learned from data from a single viewpoint. For maximum generalisation, however, the part representations themselves are view-invariant, achieved by attaching a frame of reference to the part itself, rather than to the camera frame. The subsequent view-invariant layers are created from multiple depth images. We thus switch from receptive fields defined as intervals on the image plane for the view dependent layers, to volumetric receptive fields defined as intervals on the 3D workspace. At each layer, each dimension of the receptive field for a part trebles. This means that parts in a layer i are compositions of 9 or 27 parts from layer $i-1$. Each layers is learned from the layer below. The procedure is repeated until the desired number of layers is obtained. In general the receptive field of the last layer should cover the whole object. A visualisation of a portion of a learned hierarchy is shown in Figure XX.

A. *Learning the view-variant layers*

The general procedure for learning is presented in Fig. 2. To pre-process the images we perform background subtraction and denoising with a 7×7 median filter. Then we compute the normal vector for each point in the image. To compute this normal vector we use Principal Component Analysis (PCA). We compute normal vectors using a 7×7 window. Because PCA does not work properly on edges we detect two surfaces inside the sliding window. Then, we remove all points which don't belong to the dominant surface inside the sliding window.

Layer 1 is simply a single atomic planar patch. To extract planar patches from a given depth image, we divide the image into a regular grid. Each cell (5×5 pixel patch) corresponds to a receptive field for the first layer. The surface of the object as divided into receptive fields is presented in Fig. 3. Given the normals for the points we now find the best orientation of the planar patch for each receptive field. This estimation process creates a piecewise planar interpretation of the image. During learning, this is performed on every image in the training set, so as to create a large set of realisations.

To create layer 2, these planar part realisations are first grouped into receptive fields consisting of 9 planar patches (Fig. 4).[1] These are the candidate parts for the 2^{nd} layer, of which a subset are selected to form the actual parts of layer 2. This compression underpins the generalization properties of the hierarchy, and is achieved by clustering. To represent each cluster the part at the centre of the cluster is chosen.

The parts in the second layer are then again fitted to from all images used for training the hierarchy. The variability of the parts used for learning is increased by introduction overlapping receptive fields. Overlapping receptive fields also allow us to deal with problems arising from the partition of the image into a regular grid. Because of this partition the camera pose relative to the object strongly influences the position of the receptive fields on the object's surface. This causes data loss on the edges of the object. If the receptive field is shifted the part computed for the surface may cover only the object, the background or both. The position of the receptive field influences the part detected on the image. By using overlapping receptive field the number of hypothesis about each point on the image plane increases. These overlapping receptive fields are presented in Fig. 5. The stride at layer i is simply the width of a part at layer $i-1$. This means that each pixel might be explained by three

[1] If any receptive field does lie entirely within the surface of the object the sub-part is represented by the background. The minimal number of non-background sub-parts in the second layer word (part) is set to four. This is the minimal number of points required to compute similarity between parts.

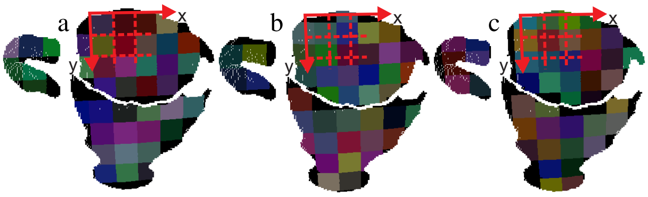

Fig. 5. Overlapping receptive fields: the receptive field shifts by 0% (a), 33% (b) and 66% (c)

different parts.

This procedure—of fitting parts, enumerating realisations, and clustering—is repeated. At each stage the vocabulary for the i^{th} layer is used to find all the realisations in the depth data for all images. Then candidate parts for the $i+1^{th}$ layer are formed from receptive fields that group these realisations. Finally, we select the $i+1^{th}$ layer vocabulary by clusterization of parts (words) in the vocabulary.

For this purpose Hierarchical Agglomerative Clustering is used [7]. We merge clusters according to the similarity between parts. We stop the procedure if the distance (similarity) between parts d^c is bigger than a threshold. Moreover, after each step which merges two clusters we compute maximal distance $d^c_{\max}$ between parts inside each cluster. If the distance inside the cluster is bigger than threshold the cluster is split in two, thus limiting the maximum intra-cluster distance. The number of obtained parts in the vocabulary depends on two parameters: d^c and $d_{\max}$. Similarity between two view dependent parts p_A and p_B is defined by the distance metric d_{VD}:

$$d_{\mathrm{VD}} = \sum_{i,j}^{N} (c_1 d(p_A^{\mathrm{ij}}, p_B^{\mathrm{ij}}) + c_2 acos(\mathbf{N}_A^{\mathrm{ij}} \cdot \mathbf{N}_B^{\mathrm{ij}})), \qquad (1)$$

where $d(p_A^{\mathrm{ij}}, p_B^{\mathrm{ij}})$ is the Euclidean distance between the centres of corresponding patches, N_A^{ij} and N_A^{ij} are the corresponding normal vectors in parts A and B, and c_1 and c_2 are constant values which scale the Euclidean distance against the angles between normal vectors. The distance d_{VD} is computed taking into account corresponding sub-parts. If both sub-parts are background the distance value is not increased. If one of the corresponding sub-part is background and the second corresponds to planar patch the distance d_{VD} is increased by constant value c_3.

As stated, the parts are created from view specific data, but the frame of reference for each part is attached to the part. This means that that we can distinguish between planar, concave, convex, and other local shapes, independent of the angle of observation. This in turn significantly reduces the number of parts in the hierarchy. In contrast to an equivalent hierarchy for 2D intensity images [2], the discretization of normal vectors is not required in hierarchy of 3D parts. We store continuous values of the normal vector. This permits learning of the hierarchy from small numbers of exemplars. To obtain this frame of reference for each part we solve the following optimization problem:

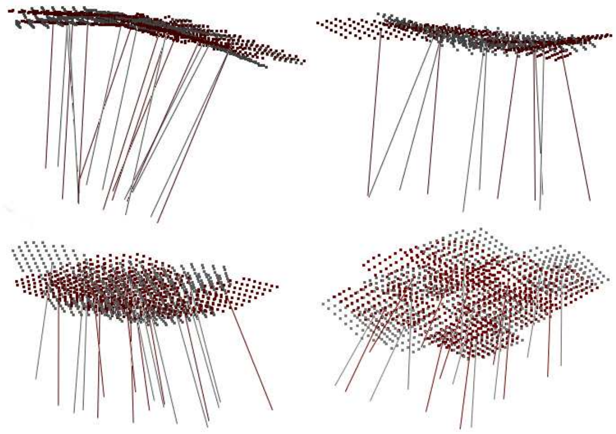

Fig. 6. Alignment of parts from the second layer. Gray – part A (p_A), red – part B (p_B)

$$\arg\min_{\mathbf{T}} d_{\mathrm{VD}}(\mathbf{T}), \qquad (2)$$

where $\mathbf{T}$ is an SE(3) rigid body transformation, $d_{\mathrm{VD}}(\mathbf{T})$ is the distance metric (1) computed for part p_A and part p_B transformed by $\mathbf{T}$. To find the rigid body transformation $\mathbf{T}$ which aligns part A and part B we use the Umeyama method [6]. The Umeyama method finds the optimal transformation $\mathbf{T}$ (root mean squared error is minimized) between points with known correspondence. To find correspondences between patches in parts we perform exhaustive search. We take into account 8 possible rotations of parts around central element. Note that we don't rotate the part in 3D space. We modify correspondences between elements only. After this step the parts are aligned by the Umeyama method and the similarity distance $d_{\mathrm{VD}}(\mathbf{T})$ between parts is computed.

To align parts from the the $i+1^{th}$ layer the hierarchical structure of parts has to be used. Parts in higher layers are created using parts from lower layers. Thus, the parts of the first layer are represented by a single point, parts of the second layer are represented by 9 points, third layer parts by 81 points, and in general the size $s_{\mathrm{i+1}}$ of the $i+1$ layer part is $9s_i$. To obtain the point cloud and to find correspondences and compute the optimal alignment the subparts have to be recursively represented by points with normal vectors.

An example alignment for parts from the second layer is presented in Fig. 6. We represent each planar patch as a point cloud. The data points for each part do not match perfectly. Small differences in the positions and orientations of sub-parts and even missing sub-parts occur, but can be dealt with by the alignment procedure. This property is what enables the hierarchy to generalize learned models and classify parts properly despite sensor noise.

The variability of sub-parts is encoded in the clusters. These clusters are created without supervision, and the algorithm generates natural groups of planar, concave and convex parts. Other groups contain various elements which represent edges in the depth image. These parts contain sub-parts created from planar patches and background elements.

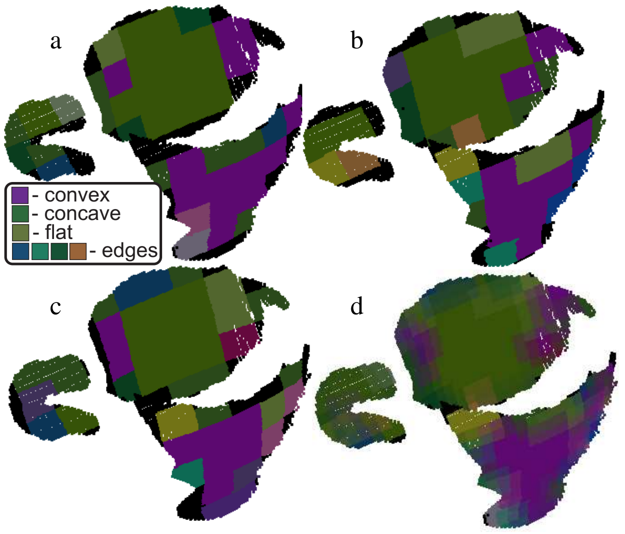

Fig. 7. Clusterization results for the second layer. Parts from the same group have the same colour on the object visualization: overlap 0% (a), overlap 33% (b), overlap 66% (a), combination of hypotheses (d)

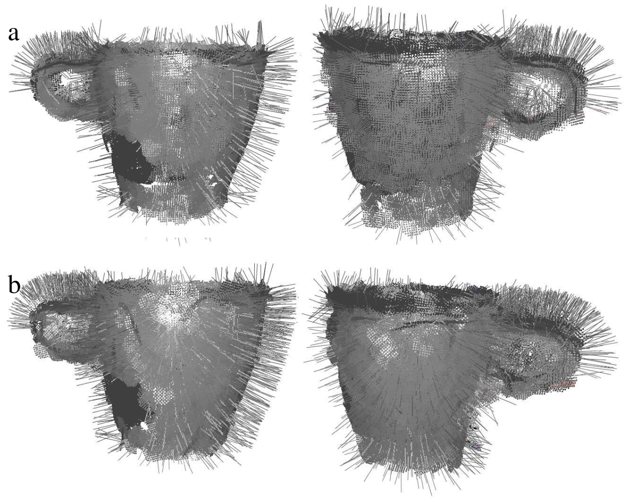

Fig. 8. Explanation of data using vocabulary from second layer without (659 parts) (a) and with compression (38 parts) (b)

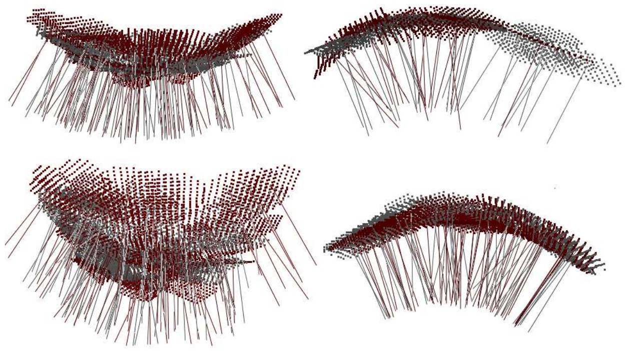

Fig. 9. Alignment of parts from the third layer. Gray – part A (p_A), red – part B (p_B)

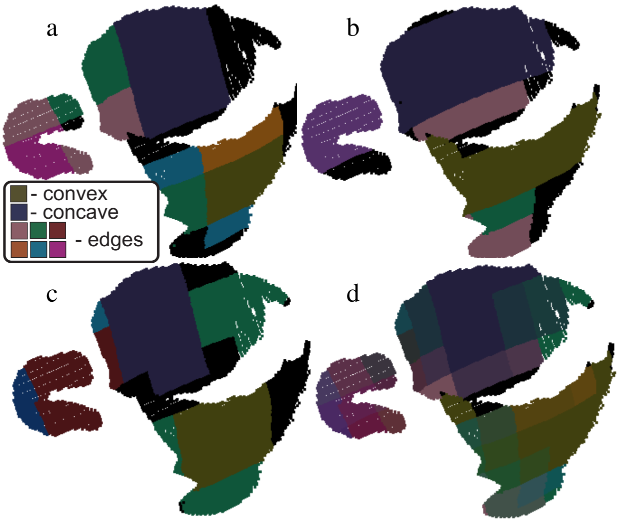

Fig. 10. Clusterization results for the second layer. Parts from the same group have the same color on the object visualization: overlap 0% (a), overlap 33% (b), overlap 66% (a), combination of hypotheses (d)

The example clusterization results— given overlapping receptive fields and a single view-point—are presented in Fig. 7. The hierarchy distinguishes between various parts and groups them if the local shape is similar. Some surfaces are misclassified. The size of the receptive field in the second layers is small and the noise of the sensor plays an important role here. However, the multiple hypotheses about the parts help to deal with this problem (Fig. 7).

B. *Data explanation using i-th layer vocabulary*

In the next step the input data is explained using the obtained l-th layer vocabulary. We explain the image using representative parts only. Thus, some areas of the image are replaced by l-th layer representative parts. This enables the compression of the input data as well as generalization. Using information about camera pose the parts can be moved into 3D space and reconstruct the object. Then, we can check visually the compression and clusterization results. The obtained representation of the object created from second layer vocabulary is presented in Fig. 8. In Fig. 8a the object created from 659 parts is presented. In this case the compression rate is set to 0%. In Fig. 8b the objects is created from 38 parts only. The compression rate for this layer is 5.8%. Despite of the high reduction of parts the explanation of the input data is still precise. Moreover, the normal angles are more smooth for the object created from compressed vocabulary. This property comes out from the clusterization. The variation of the shapes inside clusters which is also caused by the sensor noise is represented by the “average” part in the center of the cluster.

The procedure of learning l-th layer vocabulary is repeated for view-dependent layers until the desired number of layers is obtained. The obtained alignment of parts in the third layer is presented in Fig. 9. The obtained clusters are presented in Fig. 10. For the third layer the compression rate is 60.1 %. For the third layer the obtained compression is significantly smaller than for the second layer. In the third layer the receptive fields are three times bigger than in the second

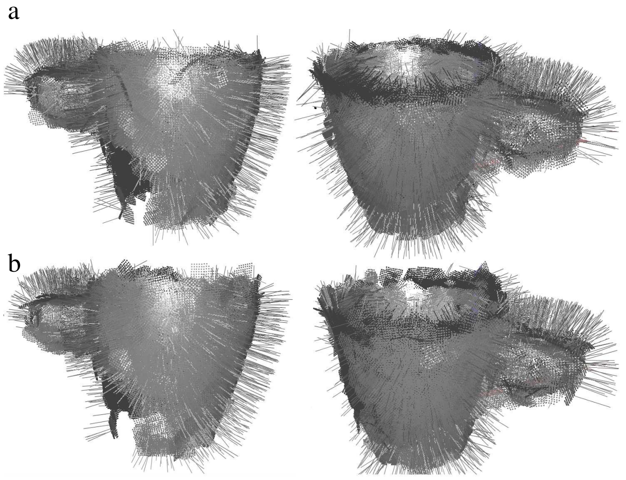

Fig. 11. Explanation of data using vocabulary from third layer without (143 parts) (a) and with compression (86 parts) (b)

layer (Fig. 4). At this size of the receptive field the difference between concave, convex and flat parts is bigger. On the other hand, parts overlapped by receptive fields are more unique and clusterization of parts produces higher number of groups. The higher compression is possible but the higher reconstruction error has to be accepted in this case. The explanation of the input data using third layer vocabulary is presented in Fig. 11.

C. View-invariant representation of parts

Parts obtained from single camera view do not provide full information about the shape of the object. It's difficult to conclude about parts which are located on the edges of the object. The system which learns from single view does not have information that some edges visible from a single view correspond to continuous surface of the object. The full information about these parts is important when the robot grasp the object. The grasps that can be attached to the real 3D edge of the object and the edge visible from single camera view are different. Without the knowledge that some edges on 2D image might represent continuous surface the grasping method might fails. To deal with the proper representation of objects in 3D we build the hierarchy which contains view-invariant layers defined in the volumetric space.

The general learning principle is the same for view-variant and view-invariant layers (Fig. 2). For view-invariant volumetric layers the receptive field is represented by the box. The relation between words from l and $l+1$ layers is presented in Fig. 4. The word of the $l+1$ layer consists 27 words (parts) from the l-th layer. The receptive fields for the volumetric layers grows in three dimensions.

To create first layer of the volumetric vocabulary we transform each part from the view-dependent layer into 3D using known position of the camera for each viewpoint. In current implementation we start to build volumetric layers from parts of the second view-dependent layer. Then, we hierarchically convert each part into point cloud. We store all points related to each object in separate octree. To reduce number of points and to combine information about surface observed from many viewpoints we run voxel grid filtering [1]. In contrast to [1] we preserve normal vectors for each voxel. During computation of mean point for each voxel the point are clustered according to the normal vector. The maximal distance (angle) between normal vectors inside single cluster is set to 45°. For each obtained cluster the mean position and normal vector is computed. This approach allows to preserve inner and outer surfaces of the objects.

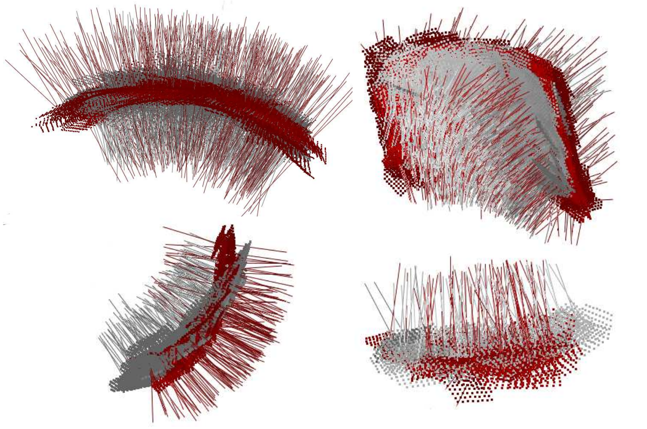

Fig. 12. Alignment of parts from the fourth layer. Gray – part A (p_A), red – part B (p_B)

To compute similarity between view-invariant parts we find the SE(3) transformation T which aligns two parts. To find the optimal transformation T we minimize the error:

$$\arg\min_{\mathbf{T}} (n_A - n_B)^2 \sum_i ||\mathbf{N}_B^i \cdot (\mathbf{T} \cdot \mathbf{p}_A^{\mathrm{i}} - \mathbf{p}_B^{\mathrm{i}})||^2, \qquad (3)$$

where $\mathbf{p}_A^{\mathrm{i}}$ and $\mathbf{p}_B^{\mathrm{i}}$ are corresponding points from part A and part B, n_A and n_B are number of points in parts, $\mathbf{N}_B^i$ is the surface normal at $\mathbf{p}_B^{\mathrm{i}}$. The element $(n_A - n_B)^2$ in (3) is added to prevent matching parts with significantly different size. To find correspondence between points and to compute optimal transformation T we use point-to-plane variant of Iterative Closest Point [8]. To prevent sub-optimal solutions we run ICP multiple times. Each optimization starts from randomly initialized initial guess. Because positions of points are defined in the center of the part we only initialize randomly the orientation of the parts.

The example alignment of parts from the fourth layer is presented in Fig. 12. In contrast to parts from view-dependent layers, which represent single surface, the parts from volumetric layers contain two surfaces. View invariant parts from volumetric layers represent inner and outer surfaces of the objects (cf. Fig. 9 and Fig. 6).

The procedure parts selection is the same as for view-dependent layer. When the next layer vocabulary is selected the receptive fields are increased and the part extraction procedure is repeated (Fig. 2). To minimize the influence of the object pose in relation to the global coordinate system the view-invariant receptive fields are also shifted by 33%.

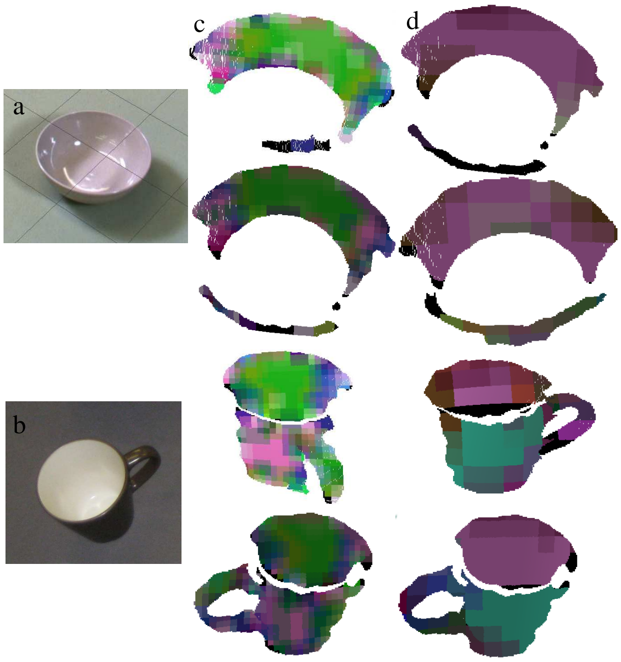

Fig. 13. Shareablity of parts from view-dependent layers between two categories of objects (bowl (a) and mug (b)) for: second (c) and third (d) layer parts of the hierarchy

This strategy also provides redundant information about parts covered by receptive fields. Thus, the identification of parts for grasping is more robust.

IV. Results

The learning of the hierarchy allows to find parts which explain the input data. The most important property oh the hierarchy is shareability. The parts are shared between object thus the grasp can be also transfered from the training object to another object. The shareability of parts from view-dependent layers is presented in Fig. 13. The shareability is presented for two types of objects: mug and bowl. In Fig. 13c parts from second layers are presented. The parts which represent concave surface (green) can be found on the surface mug which is visible from two various viewpoints as well as on the inner surface of the bowl. Convex parts (pink) are mainly visible on the outer surface of the mug. Some realizations of convex parts can be also found on the bowl's surface but they only represent local shape of the surface.

Shareability of parts is also visible on the 3rd layer of the hierarchy (Fig. 13c). Because the size of the receptive field is bigger than the size of the first layer's receptive field also the difference between geometric properties of the surface (convex and concave) are better visible. The same parts realizations can be found on the surface of the mug and on the surface of the bowl (concave parts are represented by purple color).

The shareability of parts can be also observed in volumetric layers of the hierarchy (Fig. 14). Some parts representing edges (green and purple) as well as bigger parts which correspond to inner and outer surface of the object are shared between mug and bowl. Note that some parts which are related to the edges can be also found on the objects surface (purple parts in Fig. 14a). This is the result of space division into regular grid which represent receptive fields. The position of the receptive fields in 3D strongly influence how the object is divided into parts. To mitigate this problem we implemented overlapping receptive fields. The information about spurious edges can be also ignored because the location of the edge is not supported by parts from lower layers of the hierarchy.

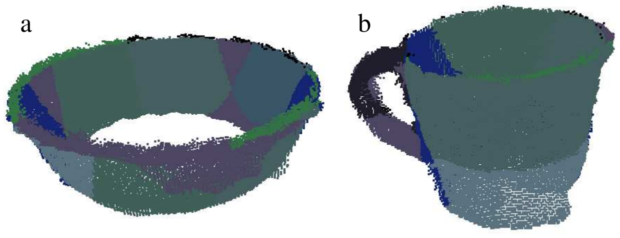

Fig. 14. Shareablity of 4th layer parts between objects

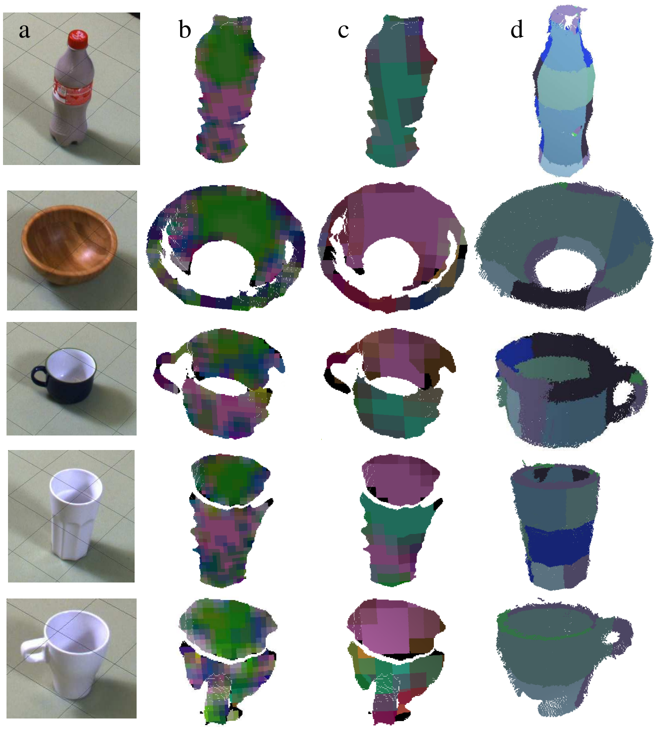

Fig. 15. Inference results for five new objects (a): parts obtained on second (b), third (c) and fourth (d) layer. Shareability of parts obtained during training the hierarchy is presented in Fig. 13 and Fig. 14

References

[1] A. Aldoma, Z.C. Marton, F. Tombari, W. Wohlkinger, C. Potthast, B. Zeisl, R.B. Rusu, S. Gedikli, M. Vincze, Point cloud library: Three-Dimensional Object Recognition and 6 DoF Pose Estimation. IEEE Robotics & Automation Magazine, pp. 80-91, 2012

[2] S. Fidler, M. Boben, A. Leonardis, Learning a Hierarchical Compositional Shape Vocabulary for Multi-class Object Representation, Computer Vision and Pattern Recognition, 2014 (under review)

[3] M. Riesenhuber, T. Poggio,Hierarchical models of object recogni-tion in cortex. Nature Neuroscience, Vol. 2, pp. 1019–1025, 1999

[4] T. Serre, G. Kreiman, M. Kouh, C. Cadieu, U. Knoblich, T. Poggio, A quantitative theory of immediate visual recognition. Progress in Brain Research, Vol. 165, pp. 33–56, 2007

[5] J. Spehr, On Hierarchical Models for Visual Recognition and Learning of Objects, Scenes, and Activities, Studies in Systems, Decision and Control, Springer, 2015

[6] S. Umeyama, "Least-squares estimation of transformation parameters between two point patterns", IEEE Trans. on Pattern Analysis & Machine Intelligence, 13(4), 1991, 376–380.

[7] C. D. Manning, P. Raghavan, H. Schütze, Introduction to Information Retrieval, Cambridge University Press. 2008

[8] A. V. Segal, D. Haehnel, and S. Thrun, Generalized-ICP, Robotics: Science and Systems, 2009