Learning hierarchical representation of 3D objects

Dominik Belter, Marek Kopicki, Jeremy Wyatt

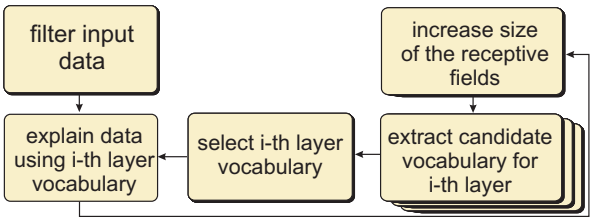

Fig. 1. General procedure of learning the hierarchical representation of objects

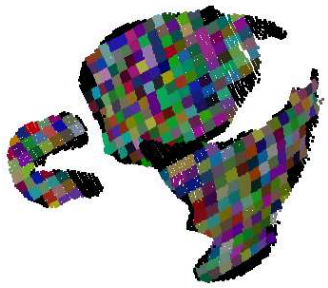

Fig. 2. Point cloud obtained from single camera view divided into first layer receptive fields. Each color represent receptive field. Black points are not used to compute planar patches

***Abstract*—**

I. Related work

II. Learning

The procedure of learning hierarchical representation of objects is presented in Fig. 1. At the beginning, the input data are filtered to remove the noise introduced by the sensor. In the next step the first layer of the hierarchy is defined. Words from the first layer vocabulary are defined over the space of image features. For the grayscale images the Gabor filters might be used [2]. Each Gabor filter correspond to the edge of the object on the 2D image. Because we create hierarchy from range data the input feature corresponds to the surface of the object and the word in the first layer is represented by the planar patch.

In the next step of the learning procedure the first layer words (planar patches) are detected on the depth image. Input data is explained using first layer vocabulary. To create next layer vocabulary the size of the receptive field is increased. The words of the $i-th$ layer are created from spatial combination of words of the previous layer. Then, the learning procedure selects representative words for the next layer vocabulary. It allows to reduce the number of words in the vocabulary and to obtain generative properties of the hierarchy. Similar parts are grouped in the same cluster. The whole cluster is represented on the higher level of the hierarchy by the representative part. The representative part might be the part in the center of the cluster. The procedure is repeated the desired number of layers is obtained. In general the size of the receptive field

A. Implementation

We learn the hierarchy from range images obtained from Kinect-like sensors. We create two types of vocabularies. First layers of the hierarchy, which are view-dependent, are created using 2.5D depth image. This allows to build representation of the object which is visible from single viewpoint. To build full 3D hierarchical model of the object we build volumetric words in 3D from many depth camera images.

To create first layer of the hierarchy we remove background from images. Then, we filter the image to remove noise introduced by the sensor. To this end, we applied median filter in 7×7 window. In the hierarchy the first layer is represented by planar patch. According to the general procedure of the hierarchy learning (Fig. 1) the next step the input data is explained using first layer vocabulary. Thus we extract planar patches from the depth image. First we compute the normal vector for each point on the image. To compute normal vector we use Principal Component Analysis (PCA). We compute normal using 7×7 window. Because PCA does not work properly on the edges we detect two surfaces inside the sliding window. Then, the points which don't lie on the surface corresponding to the considered point are removed from the patch.

To extract planar patches from the depth image we divide image space into regular grid. Each cell (5×5 patch) correspond to the receptive field. The surface of the object divided into receptive fields is presented in Fig. 2. Inside each receptive field the point which represent planar patch is found. To remove outliers the points are grouped according to the normal vector. For the most numerous group we compute mean normal vector and central position of the patch.

The next layer vocabulary is formed on the image plane. To create words of the $l+1$ layer the size of the receptive field is increased. The procedure is presented in Fig. 3. The receptive field of the $l+1$ layer covers nine receptive fields of the l-th layer. The second layer's word contains maximally nine planar patches. If the receptive fields does not cover the surface of the object the sub-part is represented by the background. The minimal number of 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.

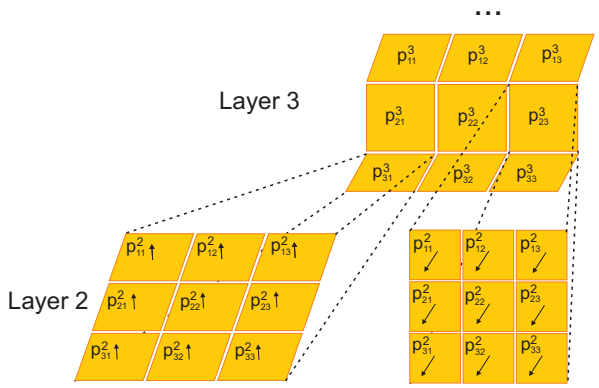

Fig. 3. Representation of words in the view-variant layers

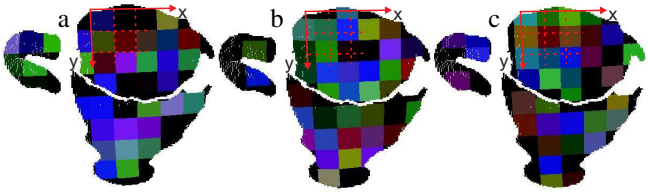

Fig. 4. Overlapping receptive fields: receptive field shifted by 0% (a), 33% (b) and 66% (c)

The words of the second layer are extracted from all images used for training the hierarchy. The variability of the words used for learning is increased by introduction overlapping receptive fields. Overlapping receptive fields allows also to deal with the problem related to the division of the image into regular grid. The position of the camera in relation to the object strongly influence the position of the receptive fields on the objects surface. The problem exists on the edges of the object. If the receptive field is shifted the part computed for the surface may overlap the object, the background or both. The position of the receptive field influences the part detected on the image. By applying overlapping receptive field the number of hypothesis about the point on the image plane is increased.

The overlapping receptive fields are presented in Fig. 4. The overlapping receptive field is moved by the width of the $l-1$ part (33% and 66% of the $l-th$ layer part). It means that each point on the image might be explained by three different parts.

In the next step of the learning procedure the candidate vocabulary for i-th layer is extracted. Words obtained from all depth images and for all overlapping receptive fields are selected. Then, we select the next layer vocabulary by clusterization of parts (words) in the vocabulary. To this end, the hierarchical agglomerative clustering is used [4]. We merge clusters according to the similarity between parts. We stop the procedure if the distance (similarity) between parts d^c is bigger than threshold. Moreover, after each step which merges two clusters we compute maximal distance $d^c_{\max}$ between parts inside the cluster. If the distance inside the cluster is bigger than threshold the cluster is splited into two separate clusters. By using this procedure we accept maximal error between parts inside cluster. The number of obtained parts in the vocabulary depends on two parameters: d^c and $d_{\max}$.

To compute similarity between two view dependent parts p_A and p_B we defined the distance metric d_{VD}:

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

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

Despite of the fact that first layers of the hierarchy are view-dependent we store in the hierarchy parts which are view-invariant. It means that that we can distinguish between planar, concave, convex and other part independently from the angle of observation. It allows to reduce significantly the number of parts which are stored in the hierarchy. In contrast to hierarchy presented for 2D images [2] the discretization of normal vectors is not required in hierarchy of 3D parts. We store continues values of the normal vector. We can also learn hierarchy from few examples. Single camera image provides multiple realizations of parts. If parts are view-variant the number of learning examples should be significant to provide sufficient statistics for learning. In this case objects should be observed from multiply viewpoints.

To obtain view invariance of parts we solve optimization problem:

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

where $\mathbf{T}$ is special Euclidean group 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 Umeyama method [3]. Umeyama methods finds 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 part in 3D space. We modify correspondences between elements only. After this step the parts are aligned by Umeyama method and the similarity distance $d_{\mathrm{VD}}(\mathbf{T})$ between parts is computed.

To align parts from the the l-th layer the hierarchical structure of parts has to be used. Parts of higher layers are created from parts of the lower layers. To obtain point cloud, find correspondences and compute optimal alignment the subparts have to be recursively represented by points with normal vectors. Thus, the part of the first layer are

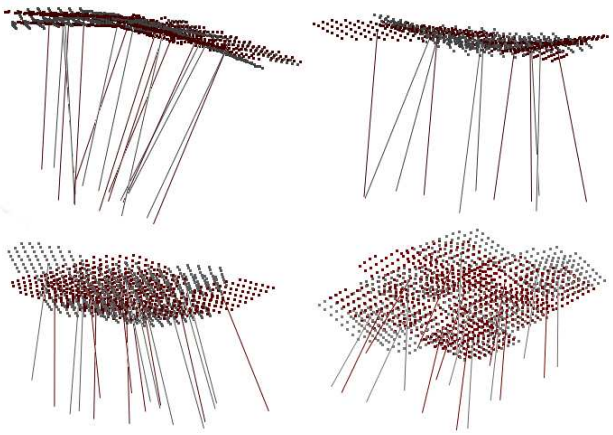

Fig. 5. Alignment of parts from the second layer. Gray – part A (p_A), red – part B (p_B)

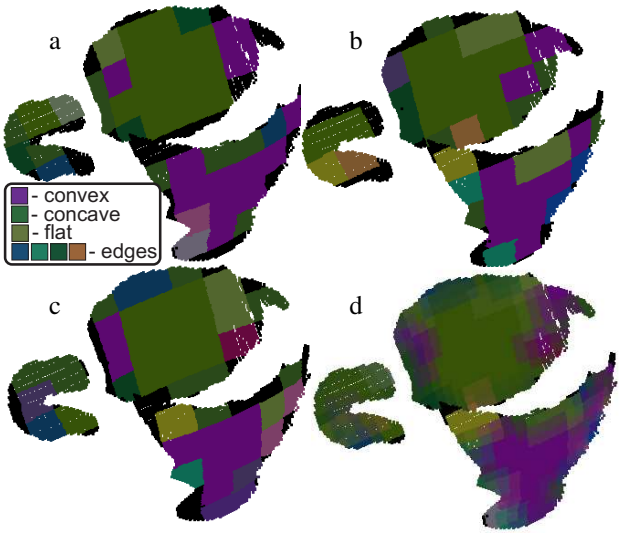

Fig. 6. Clusterization results for the second layer. Parts from the same group have the same color on the object visualization: overlap 0% (a), overlap 33% (b), overlap 66% (a), combination of hypotheses (d)

represented by single point, part of the second layer is represented by 9 points, third layer part by 81 points, size s_{l+1} of the $l+1$ layer part is $9s_l$.

The example alignment for parts from second layer is presented in Fig. 5. We represent each planar patch as a point cloud. The parts does not match perfectly. Small differences in position of subparts and their orientation and even missing sub-part are acceptable. This property enables the capability of the hierarchy to generalize learned models and classify parts properly despite of the sensor noise.

The variability of subparts is encoded in the clusters. Clusters are created without supervision. The algorithm can properly generate group of planar, concave and convex parts. Other groups contain various elements which represent edges on the depth image. These parts contain sub-parts created from planar patches and background. The example clusterization results for overlapping receptive fields and single camera view are presented in Fig. 6. The hierarchy can distinguish

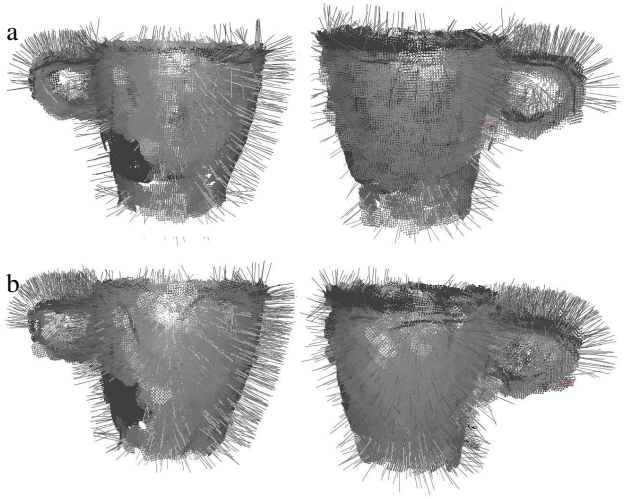

Fig. 7. Explanation of data using vocabulary from second layer without (659 parts) (a) and with compression (38 parts) (b)

between various parts and group them if the local shape is similar. Some surface are misclassified. The size of the receptive field in the second layers is small and the noise of the sensor plays important role here. However, the multiple hypothesis about the parts help to deal with this problem (Fig. 6).

B. data explanation using l-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. Thud, 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. 7. In Fig. 7a the object created from 659 parts is presented. In this case the compression rate is set to 0%. In Fig. 7b 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 is comes out from the clusterization. The variation of the shape inside clusters which caused also 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. 8. The obtained clusters are presented in Fig. 9. For the third layer the compression rate is 25.2 %. 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 layer (Fig. 3). At this size of the receptive field the difference

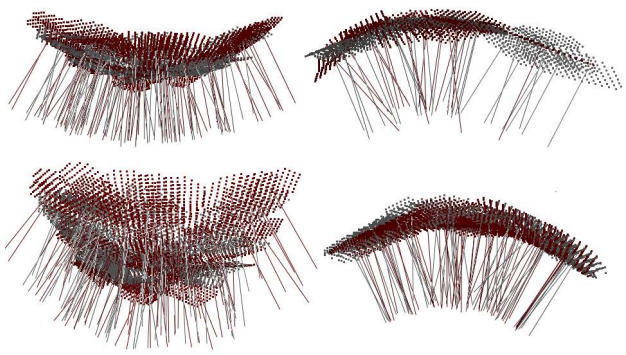

Fig. 8. Alignment of parts from the third layer. Gray – part A (p_A), red – part B (p_B)

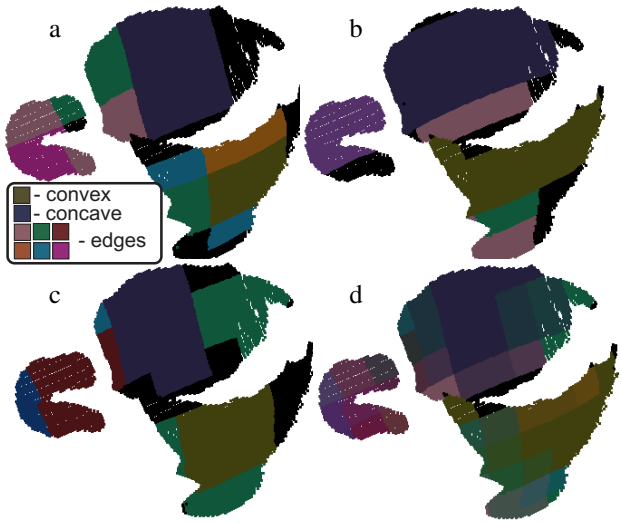

Fig. 9. Clusterization results for the second layer. Parts from the same group have the same color on the object visualization: overlap 0% (a), overlap 33% (b), overlap 66% (a), combination of hypotheses (d)

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. The explanation of the input data using third layer vocabulary is presented in Fig. 10.

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 discrepancy between 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.

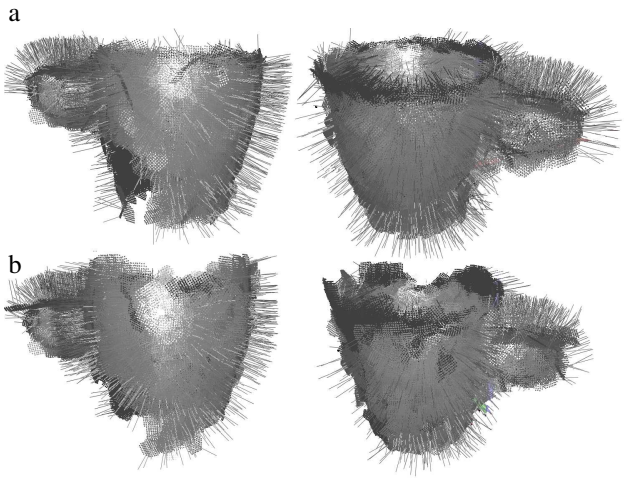

Fig. 10. Explanation of data using vocabulary from third layer without (143 parts) (a) and with compression (36 parts) (b)

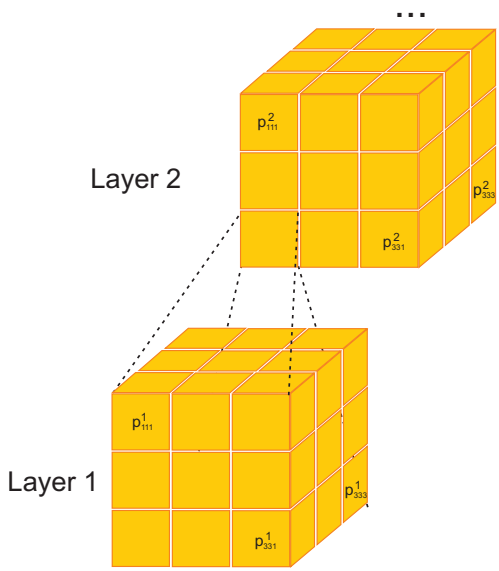

Fig. 11. Representation of words in the view-invariant layers

To deal with the proper representation of objects in 3D we build the hierarchy which contains view view-variant and view invariant layers. View-variants layers contain view-invariant parts (the id of the part depend on the local surface of the object and the id should be independent from camera viewpoint) obtained from single camera frame. The parts are created using 2.5D depth image space. The view-invariant layers are created from multiple camera frames in the volumetric space.

The general learning principle is the same for view-variant and view-invariant layers (Fig. 1). 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. 11. 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

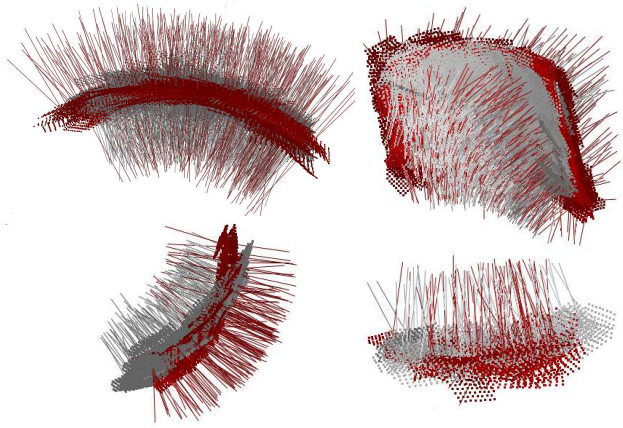

Fig. 12. Alignment of parts from the fourth layer. Gray – part A (p_A), red – part B (p_B)

using known position of the camera for each viewpoint. 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 fro each voxel. During computation of mean point for each voxel the point are clustered according to the normal vector. The maximal distance 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.

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 [5]. To prevent sub-optimal solutions we run ICP multiple times. Each optimization starts from randomly initialized initial guess. Because positions of points in parts 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 the parts from volumetric layers contain two surfaces. Parts created from single view represent single surface. View invariant parts from volumetric layers represent inner and outer surfaces of the objects (cf. Fig. 8 and Fig. 5).

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

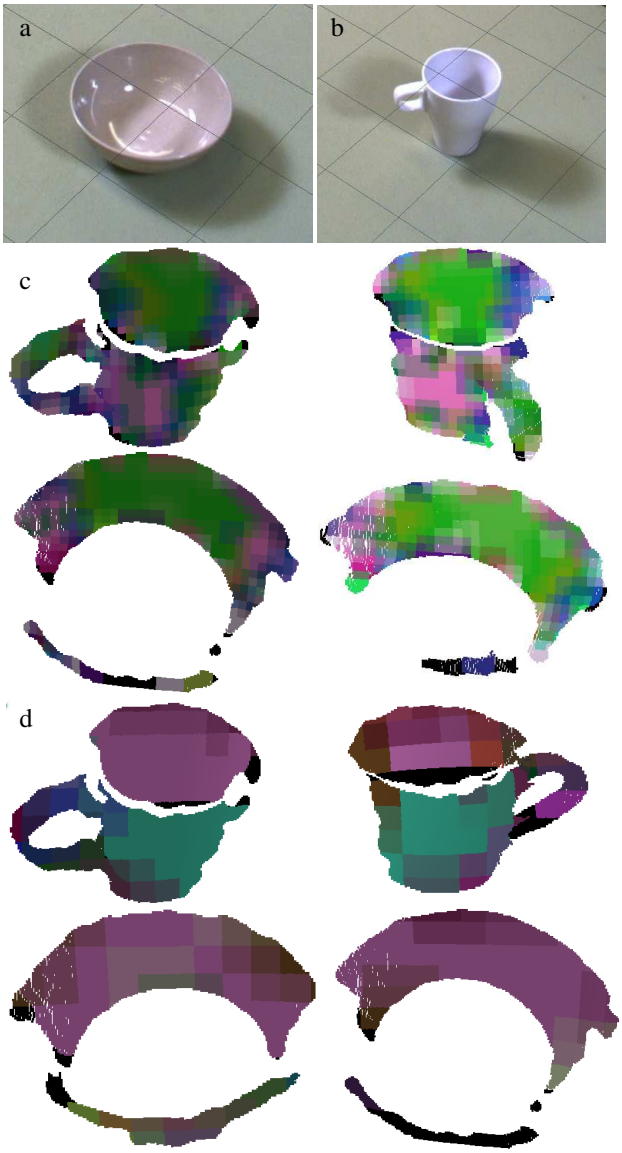

Fig. 13. Shareablity of parts from view-dependent layers between two categories of objects (bowl (a) and mug (b)) for: second (c) and third (d) layer parts of the hierarchy

procedure is repeated (Fig. 1). 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%. This strategy also provides redundant information about parts covered by receptive fields. Thus, the identification of parts for grasping is more robust.

III. Results

The learning of the hierarchy allows o 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

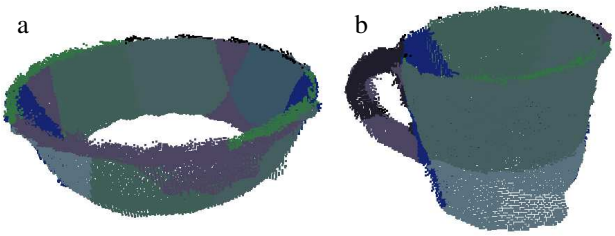

Fig. 14. Shareablity of 4th layer parts between objects

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 the difference between geometric properties of the surface (convex and concave) are clearly visible. The same parts realizations can be found on the surface of the mug and on the surface of the bowl (concave parts 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 comes from the 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.

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

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

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