Learning hierarchical representation of 3D objects

Dominik Belter, Marek Kopicki, Jeremy Wyatt

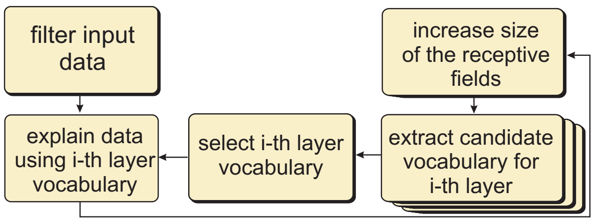

Fig. 1. General procedure of learning the hierarchical representation of objects

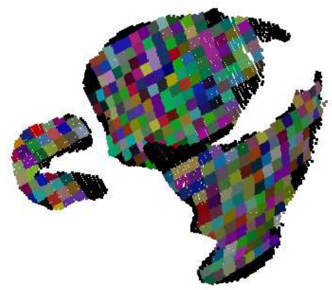

Fig. 2. Point cloud obtained from single camera view divided into first layer receptive fields. Each color represent receptive field. Black points are not used to compute planar patches

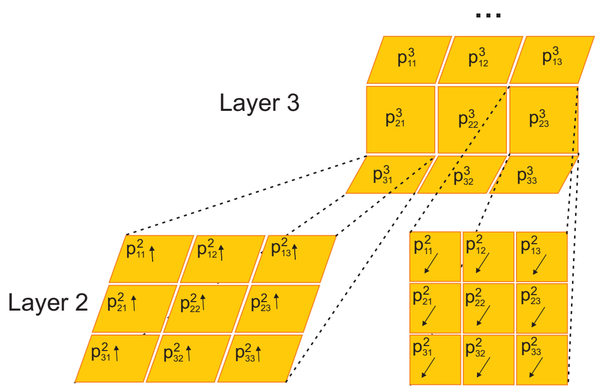

Fig. 3. Representation of words in the view-variant layers

***Abstract*—**

I. 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 [1]. 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. First, 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

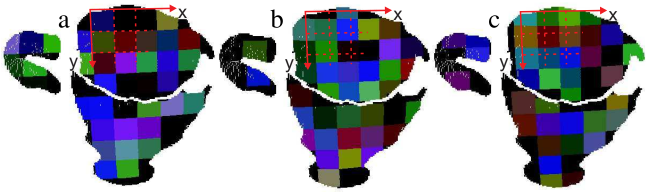

Fig. 4. Overlapping receptive fields: receptive field shifted by 0% (a), 33% (b) and 66% (c)

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.

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 [3]. 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}:

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

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 [1] 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:

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

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

The example alignment for parts from second layer is presented in Fig. 5. We represent each planar patch as a point

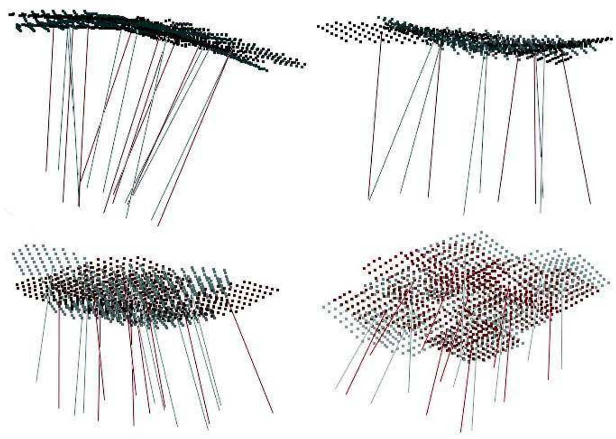

Fig. 5. Alignment of parts from the second layer. Gray – part A (p_A), red – part B (p_B)

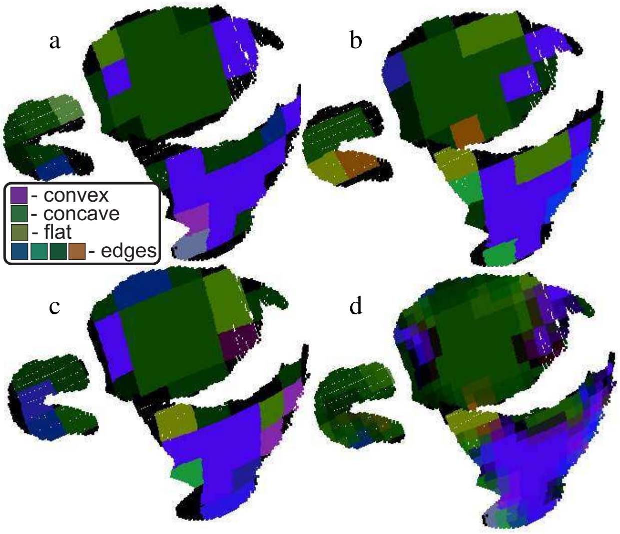

Fig. 6. Clusterization results for the second layer. Parts from the same group have the same color on the object visualization: overlap 0% (a), overlap 33% (b), overlap 66% (a), combination of hypotheses (d)

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 between various parts and group the 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

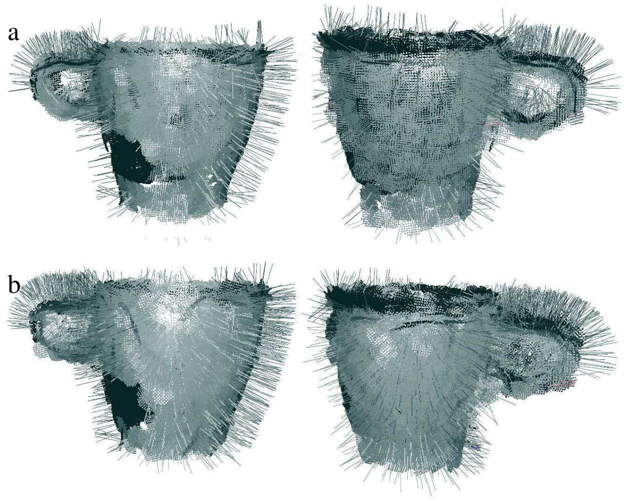

Fig. 7. Explanation of data using vocabulary from second layer without (659 parts) (a) and with compression (38 parts) (b)

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

problem (Fig. 6).

REFERENCES

[1] 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)

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

[3] C. D. Manning, P. Raghavan, H. Sch utze, Introduction to Information Retrieval, Cambridge University Press. 2008

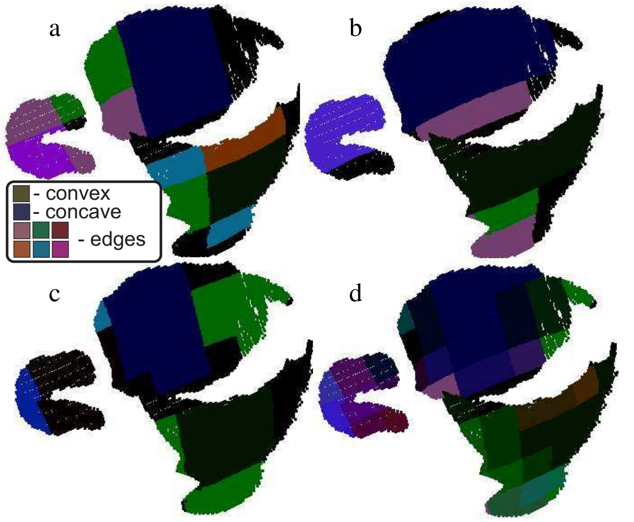

Fig. 9. Clusterization results for the second layer. Parts from the same group have the same color on the object visualization: overlap 0% (a), overlap 33% (b), overlap 66% (a), combination of hypotheses (d)

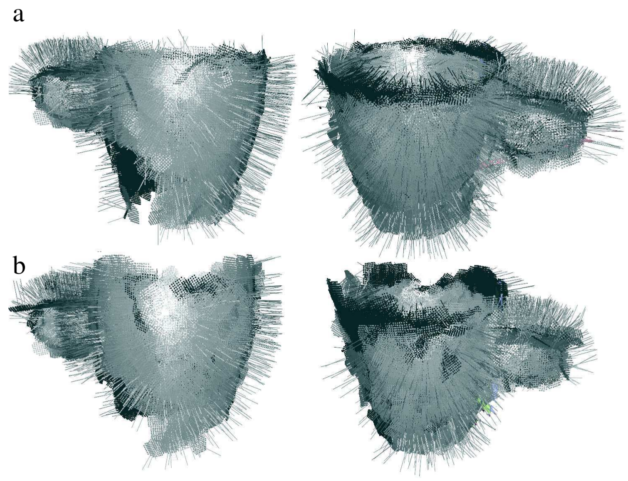

Fig. 10. Explanation of data using vocabulary from third layer without (143 parts) (a) and with compression (36 parts) (b)

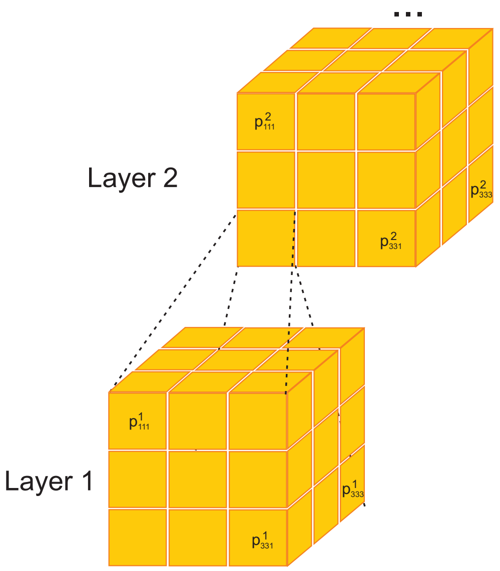

Fig. 11. Representation of words in the view-invariant layers