A view-invariant compositional hierarchical representation of 3D shape

Draft for anonymous ECCV submission

Paper ID ***

Abstract. We propose a novel view-invariant compositional hierarchical representation of 3D shapes.

Keywords: Compositional Hierarchies

1 Introduction

Several compositional hierarchical representations for 2D contour shapes have been proposed in [1–3]. Some important properties of this type of representations have been reported in literature, such as scalability, sub-linear growth of representations with respect to the number of object categories, fast inference and matching, reusability of shape parts within and across the categories which leads to compactness of the representations and enables transfer of knowledge between objects and categories.

Relatively little work has been done in the domain of the 3D compositional part-based models. Classical work of Biederman was done in 80s [4]. He has shown that people are able to recognize objects by dividing them into geons, which are the main components objects that form objects. In his approach. geons are based on a set of standard geometric primitives (e.g. cylinders, cones), which are assembled to form object and category models.

There exist several more recent methods in this domain. Wessel and Klein [5] presented a feature selection technique, which decomposes 3D objects into sections that can be represented by planes, spheres, cylinders, cones and tori. They introduced a probabilistic framework modeling the spatial arrangements between these shape primitives.

Early work of Biederman [4], Wessel and Klein [5] both use a set of predefined shapes as simple primitives with very shallow hierarchical structures, which makes it difficult to generalise over complex shapes. In contrast, our methodology is entirely data-driven, and is able to learn structures of arbitrary size and complexity.

More recently, neural networks have been used in creating 3D representations [6, 7]. Wu et al. [6] proposed a 3D hierarchical compositional framework called 3D-ShapeNet. They present a 3D model as a probability distribution of binary variables on a 3D voxel grid, using a convolutional Deep Belief Network (DBN). They demonstrated ability of their approach to recognize object categories from range images and complete full 3D shape of objects. Similarly, Su et al. [7] have

applied convolution neural networks to 3D shape recognition problem. They first perform recognition of objects presented from a single view, and then present a novel CNN architecture that combines information from multiple views of each object into a single compact descriptor.

Hu and Zhu [8] presented a framework for learning of 3D templates from 3D models of cars using AND-OR trees. They used the learned templates for simultaneous object detection, localization and pose estimation. This method exploits the well-defined 3D structure of cars to learn 3D compositions, which would otherwise be a combinatorial problem. It requires manual annotation of object parts, hence addition of a new category is difficult. Our approach does not require annotation of object parts, and is able to find recurring patterns in data automatically. Finally, methods based on deep learning [6, 7] perform well on object categorization tasks. The challenge lies in designing the network to different tasks, since it is not possible to visualise the learned features in order to check if they are visually meaningful.

2 Representation

2.1 Definitions

Our representation of 3D surface shapes is a **compositional hierarchical shape vocabulary**. The hierarchy contains several layers $L_n, n \geqslant 1$, each of which comprises a set of elements called **parts** representing various 3D surfaces. The first layer L_1 contains only one part representing the small disk-shaped planar surface, while parts of the higher layers describe larger and more complex surfaces.

A part P_i^n of the layer L_n, $\forall n > 1$ is a composition of **subparts**, i.e. parts of the previous layer L_{n-1}. It describes a distribution of possible relative positions and relative orientations of these constituent subparts, thus exerting a curtain degree of shape variability. Each vocabulary part contains its own frame of reference in which relative positions and orientations of subparts are defined.

Part realization R is an instance of a part detected in the data. Part realizations for all layers $n > 1$ have their own frames of reference which are aligned with the Darboux frame of the underlying surface, i.e. the surface normal N and principal directions T_1, T_2. The frame of references of vocabulary parts and part realizations can be converted from one to another in SE3.

2.2 Properties of the representation

One of the properties of our the compositional hierarchical representation of 3D shape is **view and rotational invariance** of parts. Part index encodes shape properties only, and does not change with change of object's pose or with change of the viewing angle. This property is achieved by describing all spatial relations of subparts relatively to other parts, while frames of reference for part realizations are attached to the Darboux frames of surfaces. Any rigid

body transformation of objects does not change the relative spatial relation of objects' parts.

Next property of our representation is **scalability**, i.e. a sub-linear growth of the vocabulary size with a number of object categories. This property is achieved because of re-usability of parts across different categories.

Another property of our framework in **redundancy** of the explanations provided by the part realizations. We start inference of vocabulary parts on each data point (e.g. at each pixel) on the object surface. This usually results in lots of part realizations with overlapping receptive fields.

Next property is **flexibility** of compositions parts. In each vocabulary part positions and orientations of the sub-parts are parameterized by the Gaussian distributions encoding a certain degree of shape variability.

3 Learning of the vocabulary

The vocabulary learning procedure comprises several steps. Assume we are learning the vocabulary of the layer L_n.

3.1 Collecting of co-occurrence statistics

The first step is the collection of co-occurrence statistics of parts of the layer L_{n-1} in the training data. At this stage, we learn statistical maps for each pair of parts (P_i^{n-1}, P_j^{n-1}) of the layer L_{n-1}. Statistical maps are functions $f_{i,j}(x, y, z, q_1, q_2, q_3, q_4)$ which give number of observations of realizations of the part P_j^{n-1} at the relative position $p = [x, y, z]$ and relative orientation described by the quaternion $q = [q_1, q_2, q_3, q_4]$ w.r.t. realizations of the part P_i^{n-1}. Statistical maps describe co-occurrences of parts only within a local geodesic neighborhood (called **receptive field**) of each part. The receptive fields grow with each subsequent layer.

Positions and orientations are described in the frame of reference of the part P_i^{n-1}, which is referred as a *reference sub-part* in this context. To process statistical maps faster, we make them discrete, i.e. we define steps for p and q, thus partitioning the receptive field into bins. Each co-occurrence observed in the training data increases the count for the relevant bin.

It is important to mention, that the term *relative orientation* has a different meaning for different layers of the hierarchy. Assume we are describing the relative orientation of the part P_j^{n-1} w.r.t the central sub-part part P_i^{n-1}. For the layers $n < 5$ the quaternion q describes the rotation that transforms the surface normal of part P_i^{n-1} into the surface normal of the part P_j^{n-1}. For the higher layers the quaternion describes the transformation that is required to rotate the full reference frame of the part P_i^{n-1} to the frame of the part P_j^{n-1}, i.e. in the higher layers the quaternions are computed from directional cosine matrices.

We make this difference because: a) lower layer parts usually represent relatively small surface patches that resemble planar surfaces, b) at the lower layers

quality of the input data may have a significant influence due to noise, discretization and meshing properties; that is why robust estimation of the principal directions of curvatures may be problematic. We find it is enough to describe the relative orientations of normals for the lower layers and then switch to description of relative orientations of reference frames for higher layers.

3.2 Clustering of the statistical maps

Clustering of the statistical maps is performed to obtain a set of pairs (doublets). It is performed in four steps:

- Assume, that we have a list of size m of non-empty bins. Each bin B_i is characterized by the average relative position $p_i = [x, y, z]$ and the average quaternion $q_i = [q_1, q_2, q_3, q_4]$ representing the relative orientation.
- Compute the distance matrix D of size $m \times m$, which defines distances between each pair of bins. The distance $D_{hk} = d_{ER}(B_h, B_k)$ between two bins B_h and B_k is defined as follows:

$$d_{ER}(B_h, B_k) = d_E(p_h, p_k) + \alpha\, d_R(q_h, q_k), \tag{1}$$

  where $d_E(\cdot, \cdot)$ is the Euclidean distance between bins' positions and $d_R(\cdot, \cdot)$ is a rotational distance between quaternions q_h and q_k. The rotational distance is defined as

$$d_r(q_h, q_k) = \sqrt{1 - q_h\, q_k}. \tag{2}$$

  All quaternions must be normalized before computing the distance matrix. Similar distance measures are discussed in [9].
- Apply hierarchical agglomerative clustering using the distance matrix D. To find the best number of clusters we use a modified Davies – Bouldin index. In the original work the Davies – Bouldin index is defined as follows [10]:

$$DB = \frac{1}{n} \sum_{i=1}^{n} \max_{i \neq j} \left(\frac{\sigma_i + \sigma_j}{d(c_i, c_j)} \right), \tag{3}$$

  where σ_x is s the average distance of all bins in cluster x to centroid c_x, and $d(c_i, c_j)$ is the distance between centroids c_i and c_j. In our framework two degenerate cases may appear where this index becomes ill-defined: a) when there is only one bin in a cluster (σ_x can not be defined), b) when the statistical map can be fit with one cluster ($d(c_i, c_j)$ can not be defined). To avoid these problems we introduce an additional constraint specifying maximal size of clusters. The results of clustering on the statistical maps are shown in Figure 1.
- Parameterize each cluster with two Gaussians. One of them (3 dimensional) represents variance of the relative positions, and another one (4 dimensional) represents variance of the relative orientations. That means for each cluster

k we compute μ_{E_k}, Σ_{E_k}, μ_{q_k} and Σ_{q_k}. Note, that since the statistical maps are defined for each pair of parts (P_i^{n-1}, P_j^{n-1}) of the layer L_{n-1}, a set of clusters described by the parameters μ_E, Σ_E, μ_q and Σ_q is different for each pair (P_i^{n-1}, P_j^{n-1}).

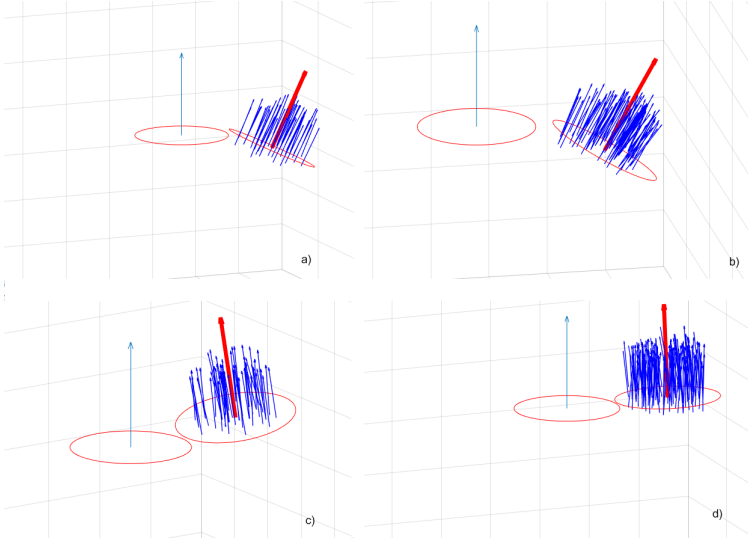

Fig. 1. The statistical map is clustered into four clusters. Circle in the middle shows the reference part. Blue arrows show positions and orientations represented by the non-empty bins of the statistical map, that were assigned to one of the clusters. Circles of the right define a sub-part which is located in the mean position and mean orientation of the cluster. Bold red arrow defines normal of this part.

These steps result in a set of doublets which are then used to build triples which form the vocabulary of the layer L_n, $\forall n > 1$.

3.3 Form a set of candidate parts

Assume we have a set of doublets after clustering of the statistical maps. It often happens that there are two or more doublets that exist simultaneously within the same receptive field. We measure the probabilities of joint observation of two doublets fully within a receptive field. Those combinations of pairs which have probabilities above a pre-defined threshold value are included in the set of candidate parts $\mathcal{C}_n$ for the layer L_n.

3.4 Compression by OR-nodes

Some of the candidate parts $\mathcal{C}_n$ for the layer L_n may represent similar surface types, although represented as compositions of different subparts. This problem

is illustrated in Figure 2, where different spatial arrangements of different subparts result in the same surface types. In order to reduce the number of parts and facilitate generalization we perform compression by introducing OR-nodes. Each of these nodes accommodates several candidate parts from $\mathcal{C}_n$ which represent similar surface types.

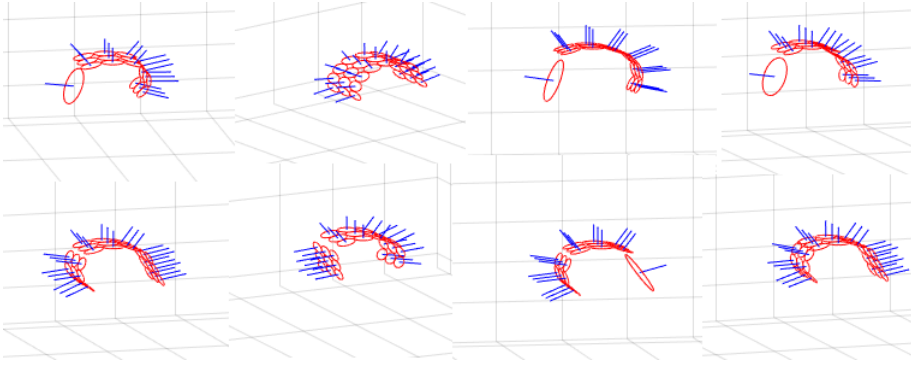

Fig. 2. Similar surface types are achieved as compositions of different subparts, or different spatial arrangements of the same sub-parts.

We use the **volume between two surfaces** to measure the similarities of parts P_i^n and P_j^n from $\mathcal{C}_n$. The volume is measured when the frames of references of these parts are aligned, i.e. the centres of parts are in the same location, and the axes are pointing in the same direction. We call this distance between parts $d_v(P_i^n, P_j^n)$.

4 Inference

Once learning from a training data set has occurred we wish to infer the existence of parts on a new example objects. The goal of inference is to find activations (realizations) of vocabulary parts in the input data. Inference can be done from depth images, point clouds, or triangulated mesh models. The general inference pipeline remains the same for all data types, however, some implementation details differ depending on the type of input data.

Our algorithm starts with inference from the first layer, and then subsequently infers parts for the higher layers (bottom-up inference) growing the receptive field size and matching combinations of part realizations in each receptive field against vocabulary parts of the next layer. We start inference at each data point, (e.g. at each pixel in the case of depth images), thereby obtaining many part realizations with overlapping receptive fields.

Inference of the first layer realizations is performed on three different scales. As the first layer part is a disk-shaped planar patch, we perform inference of

planar patches of three different radii. For instance, a relatively large planar surface patch can be expressed either as the first layer realization of larger radius, or as a composition of the first layer realizations of a smaller radii. We have chosen the second way as it gives a more compact representation and helps to reduce inference time.

In the following we first describe how to perform inference of the first layer on a single scale, and then describe multi-scale inference.

4.1 Inference of the first layer from a single scale

Assume we infer the first layer realizations on the scale S_i. Scale defines the radii of surface patches to be tested for planarity, let's assume that scale S_i corresponds to the radius rad_i. The algorithm for inference looks as follows:

1. Define a geodesic receptive field of a radius rad_i around each data point ρ_i.
2. Extract all other data points that reside in this receptive field.
3. Perform a planarity test. This test differs depending on the data type. For point clouds or range images we perform least square fitting for these points; for mesh models where surface normals are available we measure the spread of normal directions around the direction of the normal at the point ρ_i. Based on the error value (or spread value) we make a decision about the planarity of the surface patch. If the planarity of the surface is not confirmed, we stop inference at the point ρ_i, otherwise we go to step 4.
4. Estimate a surface normal N_i for this patch. For depth images and point clouds it is estimated analytically using plane parameters obtained from the least squares fitting. For mesh models, when surface normals at each data point are given, we take the average of the normals of all data point which belong to the patch.
5. If the planarity of a surface patch is confirmed and the surface normal is estimated, we say we have found a part realization of layer L_1. The position of this part is defined by the position of ρ_i, and the orientation is N_i.

4.2 Inference of the first layer from multiple scales

Inference of the first layer parts is done at three different scales, i.e. we try to find planar patches of three different radii $rad_1 > rad_2 > rad_3$ in the data. We have a general rule: If the data around a point ρ_i can be fit with a larger planar patch, then we don't try to fit the smaller patches around this point. In this manner we automatically search for the most compact representation of the data across different scales. Assume we have a set of data points Φ, the multi-scale inference procedure can be described as follows:

1. Perform inference (as described in Section 4.1) of part realizations of the largest radius rad_1. We perform inference for each data point ρ_i from the set Φ. Assume that inference was successful for the set of points $\Psi \subset \Phi$, while it failed for the set $\Upsilon \subset \Phi$. (Note that $\Phi = \Psi \bigcup \Upsilon$ and $\Psi \bigcap \Upsilon = \emptyset$).

2. Perform inference of part realizations of the radius rad_2 for each data point from the set Υ. Assume inference was successful for some points, however it failed for the set of points $\Xi \subset \Upsilon$.
3. Perform inference of part realizations of the radius rad_3 for each data point from the set Ξ.

4.3 Inference of the subsequent layers

After inference of the layer L_1 is finished, we perform inference for each subsequent next layer. Assume, we perform inference of the layer L_n given a set Λ of realizations of parts of the previous layer L_{n-1}. We do it using the following procedure:

1. For each part realization R_i from Λ define a geodesic receptive field of radius defined by the layer n. Extract all other part realizations lying within this receptive field (call this set Ω).
2. If R_i is a realization of the layer L_1 part (at any of three scales), its frame of reference is not defined. Then we go to step 3, otherwise set the reference frame of the receptive field F_i equal to the reference frame of R_i and go to step 4.
3. Estimate the frame of reference for this receptive field. The position of R_i defines origin of this frame. As for the axes, we make them aligned with the Darboux frame, i.e. the surface normal at this point and the principal directions of curvature. We use positions of all part realizations from Ω to define the tensor of curvature [11]. Eigenvectors of this tensor define the frame of reference $F = [N, T_1, T_2]$ for this receptive field, where N is a surface normal and T_1 and T_2 are are principal directions. If the principal directions can not be estimated, for example for planar surfaces or around umbilic points, we select T_1 and T_2 as arbitrary directions orthogonal to each other and to the surface normal N.
4. Convert all positions and orientations of all part realizations belonging to the set Ω from the global reference frame to the reference frame F_i. This conversion makes positions and orientations **relative** to R_i.
5. Perform matching of duplets. Assume, that R_i is a realization of the vocabulary part P_k^{n-1}. Also assume that the set Ω contains realization R_j of the vocabulary part P_a^{n-1} at the relative position $p_j = [x, y, z]$ and the relative orientation defined by the quaternion $q_j = [q_1, q_2, q_3, q_4]$ w.r.t. R_i. Extract from the vocabulary of the layer L_n a set of clusters corresponding to the pair of parts (P_k^{n-1}, P_a^{n-1}) described by the following parameters: μ_E, Σ_E, μ_q and Σ_q. (See section 3.2 for more detailes). Then we measure the Mahalanobis distances from the points p_j and q_j to the clusters. If both distances (for translation and rotation) are smaller is than pre-defined thresholds, then we have inferred a duplet. Repetition of this procedure for all realization from the set Ω results in a set of inferred duplets in the receptive field of R_i.
6. Perform inference of triples. For each pair of inferred duplets we form a triple which is matched to the triples in the vocabulary of the layer L_n.

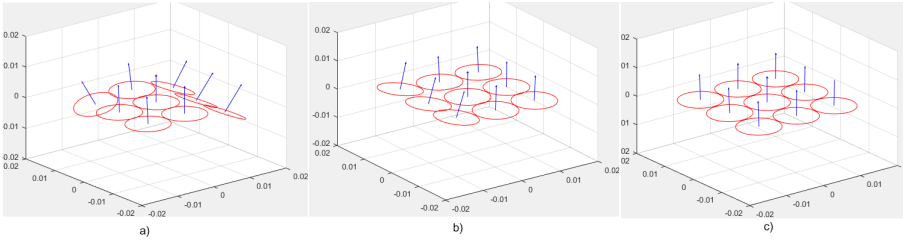

Fig. 3. Some parts from the vocabulary of the layer 4

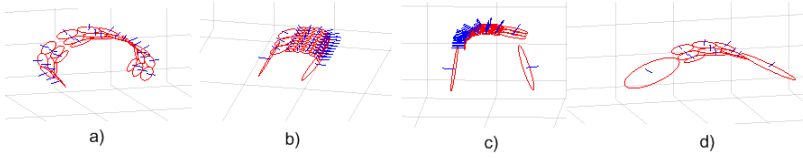

Fig. 4. Some parts from the vocabulary of the layer 7

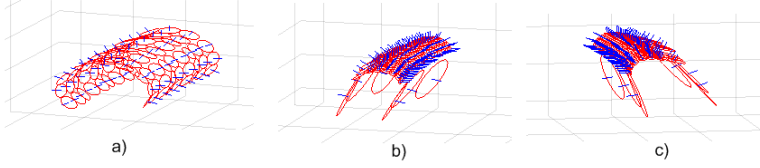

Fig. 5. Some parts from the vocabulary of the layer 8

5 Experiments

5.1 Learning of compositional hierarchical shape vocabulary from the data set of kitchen objects

We used our framework for learning a compositional hierarchical shape vocabulary form a data set of kitchen objects. The dataset contains 15 categories of objects, e.g. mugs, plates, jugs, vases, etc. Figures 3 4 5 show some of the vocabulary parts of different layers of the hierarchy.

5.2 Analysis of the properties of the hierarchy

In this section we analyze properties of our compositional hierarchy. In particular, we prove its view and rotational and rotational-invariant properties and share-ability of parts across categories.

In the first experiment we analyze how the size of the vocabulary grows with number of object categories. We learn the vocabulary on a set of objects of a single category. Then we add categories one after another, and perform

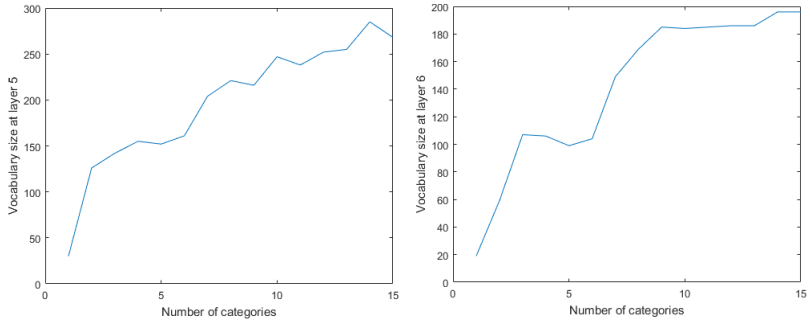

Fig. 6. Number of vocabulary parts learned from subsets comprising different number of object categories. Left: for the layer 5, Right: for the layer 6

vocabulary learning again, measuring size of the vocabulary of each layer after each iteration. All parameters and threshold values remain the same.

Figure 6 shows the result of this experiment. We can observe sub-linear growth of the vocabulary size with number of categories, which is explained by *share-ability* of the vocabulary parts across categories.

6 Conclusions

We have developed a novel view-invariant framework for learning and inference of the 3D compositional hierarchical shape vocabulary. So far we have demonstrated some properties of the framework, such as view-invariance and scalability, i.e. sub-linear growth of the shape vocabulary with number of object categories, due to share-ability of parts.

In the nearest future we aim to demonstrate some other properties of the framework, for instance ability to predict hidden (occluded) parts of the objects, and ability to recognize and localize the objects in the cluttered scenes, and approximately reconstruct their hidden parts.

References

1. Fidler, S., Boben, M., Leonardis, A.: Learning hierarchical compositional representations of object structure. In: Object Categorization: Computer and Human Vision Perspectives. Cambridge University Press (2009)
2. Aktas, U.R., Ozay, M., Leonardis, A., Wyatt, J.L.: A graph theoretic approach for object shape representation in compositional hierarchies using a hybrid generative-descriptive model. In: Computer Vision–ECCV 2014. Springer (2014) 566–581
3. Ommer, B., Buhmann, J.M.: Learning the compositional nature of visual objects. In: CVPR. (2007) 1–8
4. Biederman, I.: Recognition-by-components: a theory of human image understanding. Psychological review **94**(2) (1987) 115
5. Pratikakis, I., Spagnuolo, M., Theoharis, T., Veltkamp, R.: Learning the compositional structure of man-made objects for 3d shape retrieval. In: Eurographics Workshop on 3D Object Retrieval. (2010)
6. Wu, Z., Song, S., Khosla, A., Yu, F., Zhang, L., Tang, X., Xiao, J.: 3d shapenets: A deep representation for volumetric shapes. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2015) 1912–1920
7. Su, H., Maji, S., Kalogerakis, E., Learned-Miller, E.: Multi-view convolutional neural networks for 3d shape recognition. In: The IEEE International Conference on Computer Vision (ICCV). (December 2015)
8. Hu, W., Zhu, S.C.: Learning 3d object templates by quantizing geometry and appearance spaces. Pattern Analysis and Machine Intelligence, IEEE Transactions on **37**(6) (2015) 1190–1205
9. Kuffner, J.J.: Effective sampling and distance metrics for 3d rigid body path planning. In: Robotics and Automation, 2004. Proceedings. ICRA'04. 2004 IEEE International Conference on. Volume 4., IEEE (2004) 3993–3998
10. Davies, D.L., Bouldin, D.W.: A cluster separation measure. Pattern Analysis and Machine Intelligence, IEEE Transactions on (2) (1979) 224–227
11. Taubin, G.: Estimating the tensor of curvature of a surface from a polyhedral approximation. In: Computer Vision, 1995. Proceedings., Fifth International Conference on, IEEE (1995) 902–907