Pose Estimation and Shape Retrieval with Hough Voting in a Continuous Voting Space

Viktor Seib, Norman Link and Dietrich Paulus

Active Vision Group (AGAS), University of Koblenz-Landau,
Universitaetsstr. 1, 56070 Koblenz, Germany
{vseib, nlink, paulus}@uni-koblenz.de, http://agas.uni-koblenz.de

Abstract. In this paper we present a method for 3D shape classification and pose estimation. Our approach is related to the recently popular adaptations of Implicit Shape Models to 3D data, but differs in some key aspects. We propose to omit the quantization of feature descriptors in favor of a better descriptiveness of training data. Additionally, a continuous voting space, in contrast to discrete Hough spaces in state of the art approaches, allows for more stable classification results under parameter variations. We evaluate and compare the performance of our approach with recently presented methods. The proposed algorithm achieves best results on three challenging datasets for 3D shape retrieval.

1 Introduction

Traditionally, 2D images form the basis for the developed algorithms in object recognition and image classification tasks. However, with the development of low-cost consumer RGB-D cameras and 3D printers 3D data can be generated and processed by anyone. It is likely that 3D shape databases will emerge in the near future, where models need to be classified or retrieved. New approaches need to take this trend into account and handle data from these new imaging modalities.

A viable way seems to be the adaption of successful approaches from 2D data. One of the most successful and widely used methods for visual categorization is the bag-of-words or bag-of-keypoints approach [6]. Several extensions of this approach to handle 3D data were proposed [16, 18, 24]. Algorithms based on the bag-of-words approach usually do not use the spatial relations between features. However, studies show that taking into account spatial relation between features improves results [22, 3].

Apart from improving classification results, a great benefit of considering spatial relations of features lies in the ability to estimate the pose and localize objects in cluttered scenes. This is exploited by Leibe et al. in the Implicit Shape Model (ISM) formulation [14, 15]. Recently, extensions of the ISM approach to 3D data have been proposed [13, 19, 27, 20]. Inspired by these work we present a novel approach for 3D shape retrieval and classification using Hough voting that is closely related to Implicit Shape Models.

In this paper we make the following contributions: Unlike approaches in related work, we do not construct a dictionary of codewords, but rather use the features as they are to achieve higher discriminativity. We compare the obtained results with codebooks of different sizes to support our approach. Since in the proposed method no codebook is created, generalization from learned shapes is achieved by a k-NN activation strategy during classification (instead of training as in many approaches). Further, contrary to many ISM approaches we uniformly sample key points on input data instead of using a key point detector of salient points. The benefits of this strategy have been shown to improve classification rates compared to salient points [9]. Finally, we evaluate different vote weighting strategies and additionally show that weighted votes are sufficient to accurately estimate the pose of detected objects.

In the following Section we review related work on previous ISM extensions to 3D data. In Sections 3 and 4 we present our approach in detail. An extensive evaluation on various datasets and comparison with state of the art approaches is given in Section 5 and a discussion in Section 6. Finally, Section 7 concludes the paper and gives an outlook to our ongoing and future work.

2 Related Work

Leibe et al. first introduced the concept of Implicit Shape Models (ISM) in [14]. They group key points into visually similar clusters, the so called *codewords*. Each codeword is associated with vectors from positions of the clustered features to the object's center. These vectors are referred to as *activation vectors*, while the set of codewords is called *codebook*. For recognition, the ISM is employed in a probabilistic framework based on a Generalized Hough Transform [1]. Each codeword that is matched with image descriptors casts a number of votes for a possible object location into a voting space. Finally, object locations are acquired by analyzing the voting space for maxima using Mean-shift mode estimation [4].

While the general scheme is the same for 3D data, feature descriptors representing the geometry of the local key point neighborhood are applied [26, 13]. Knopp et al. [13] use 3D-SURF as descriptor which allows for a scale invariant feature representation. As a heuristic, the number of clusters is set to 10 % of the number of input features. To account for feature-specific variations, Knopp et al. introduced a weighted voting scheme. Votes are cast into a discrete 5D voting space (3D object position, scale and class). In a subsequent work, Knopp et al. [12] discuss approaches to implement rotation invariant object recognition.

Contrary to Knopp et al., Salti et al. [19] claim that scale invariance does not need to be taken care of, since 3D sensors provide metric data. In their approach Salti et al. use the SHOT descriptor [26] and investigate which combinations of clustering and codebook creation methods are best for 3D object classification. Salti et al. report best results when no clustering is used and all features are stored. Further, Salti et al. propose to omit vote weighting as it does not show significant benefits in their experiments. Considering this results, Tombari and Di Stefano [25] continue their work without clustering and vote weighting. In

their proposed method Hough voting achieves promising results for 3D object recognition with occlusion in cluttered scenes.

A more recent approach presented by Wittrowski et al. [27] uses ray voting in Hough space. Like in other ISM adaptations to 3D a discrete voting space is used. However, in this approach bins are represented by spheres which form directional histograms towards the object's center. This voting scheme proves very efficient with an increasing number of training data.

Our previous work [20] where we use a continuous voting space confirms the results of Salti et al. that omitting feature quantization in 3D leads to better classification results. However, in [20] the quantization experiments were performed only on a small dataset. Thus, in this work we provide further analysis on a bigger dataset and employ a k-NN matching strategy which was not used in [20].

3 Learning Object Representations

The main difference between the algorithm proposed here and the Implicit Shape Model formulation from related work such as [13, 20] and also bag-of-features approaches [24] is the lack of feature clustering, to allow for a more precise object representation.

In a first step, consistently oriented normals are computed on the models with the method proposed by Hoppe et al. [11]. Subsequently, key points are densely sampled and a SHOT descriptor is calculated around each key point in the determined local reference frame. The local reference frame for feature f is stored as rotation matrix R^f.

For each feature, a vector pointing from the feature to the object's centroid is stored, in the following referred to as *center vector*. First, the key point positions have to be transferred from global coordinates into an object centered coordinate frame. For this purpose a minimum volume bounding box (MVBB) of the object is calculated as described by Har-Peled [10] and Barequet and Har-Peled [2]. The estimated bounding box is determined by the direction between the two most distant points of the object, $\boldsymbol{p}_i$ and $\boldsymbol{p}_j$, and the minimum box enclosing the point set. The resulting MVBB B is given by the size $\boldsymbol{s}^B = \boldsymbol{p}_i - \boldsymbol{p}_j$ and the center position $\boldsymbol{p}^B = \boldsymbol{p}_i + \frac{\boldsymbol{s}^B}{2}$. The bounding box is stored with the training data and is used later to estimate the pose of the detected object. The object's position is now given by $\boldsymbol{p}^B$ as the center of B. The relative feature position $\boldsymbol{v}^f_{\text{rel}}$ is then given in relation to the object position $\boldsymbol{p}^B$ by

$$\boldsymbol{v}^f_{\text{rel}} = \boldsymbol{p}^B - \boldsymbol{p}^f \tag{1}$$

and represents the vector pointing from $\boldsymbol{p}^f$, the location on the object where the feature was detected, to the object's center. In order to provide rotation invariance, each feature was associated with a unique and repeatable reference frame given by a rotation matrix R^f. Transforming the vector $\boldsymbol{v}^f_{\text{rel}}$ from the global into the local reference frame is then achieved by

$$\boldsymbol{v}^f = R^f \cdot \boldsymbol{v}^f_{\text{rel}}. \tag{2}$$

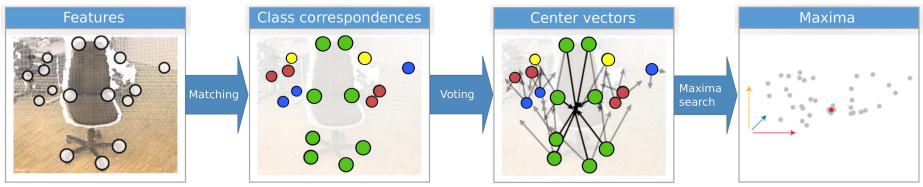

Fig. 1. Features are matched with the k closest learned features. Center vectors form hypotheses for object locations. Clusters in the voting space are detected by searching for maximum density regions.

We obtain $\boldsymbol{v}^f$, the translation vector from the feature location to the object center in relation to the feature-specific local reference frame. Thus, $\boldsymbol{v}^f$ provides a position and rotation independent representation of a feature f.

The final data pool after training contains all features that were computed on the training models. Along with each feature, the center vector, a bounding box B and the class c of the trained object is stored.

4 Object Classification and Pose Estimation

To classify objects, features are detected on the input point cloud in the same manner as in the training stage. Matching detected features with the previously trained data pool yields a list of feature correspondences. The distance between learned feature descriptor f_l and detected feature descriptor f_d is determined by the distance function $\mathrm{d}(f_\mathrm{l}, f_\mathrm{d}) = \|f_\mathrm{l} - f_\mathrm{d}\|_2$. The center vectors of the created correspondences are used to create hypotheses on object center locations in a continuous voting space (Figure 1).

Please note that we omitted the vector quantization step during training to retain a higher number of features and reduce training time, since no high-dimensional clustering needs to be performed. In the classification step, each of the detected features is associated with the k best matching features in the learned data pool. Thus, we effectively move the feature generalization step from training to detection. This procedure has the advantage of a generalized feature matching while having a broad data pool for each object class. The degree of generalization is controlled by the parameter k and can be changed without retraining. While this parameter is also applied in approaches from related work, matching is performed with a clustered codebook which has a lower descriptiveness as was shown in [19, 20].

During training, the center vector $\boldsymbol{v}^f_\mathrm{rel}$ of feature f has been rotated into the feature's local reference frame given by the rotation matrix R^f as shown in Eq. (2). Now the rotation is reversed by the inverse rotation matrix $R^{f-1} = R^{fT}$ computed from feature f on the scene, resulting in the back rotated vector $\hat{\boldsymbol{v}}^f_\mathrm{rel}$. This vector is used to create an object hypothesis at position $\boldsymbol{x}$ relative to the

position $\boldsymbol{p}^f$ of the detected feature f:

$$\boldsymbol{x} = \boldsymbol{p}^f + \hat{\boldsymbol{v}}^f_{\text{rel}}. \tag{3}$$

To reduce the dimensionality of the voting space the object's rotation is ignored in this step. Further, a separate voting space for each class reduces the voting space dimensionality to three, namely the 3D position of the hypotheses.

Optionally, each point in the voting space can be assigned a weight. However, there is on open debate of whether or not vote weighting should be used [19] and different strategies are applied (e.g. two weights in [13] and three weights in [20]). In Section 5 we report our results on two different weighting strategies. We compare uniform weighting (i.e. no weighting) and weighting votes by their likelihood

$$\omega = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{\mathrm{d}(\mathrm{f_l}, \mathrm{f_d})^2}{2\sigma^2}\right). \tag{4}$$

Here, f_l is the learned feature descriptor and f_d the detected feature descriptor. The value σ^2 is class specific and is determined during training by the sample covariance. Given F_c, the set of features detected on all training models for a class c, the sample mean of distances is computed by

$$\mu_c = \frac{1}{M^2} \sum_{i=1}^{M} \sum_{j=1}^{M} \mathrm{d}(f_i, f_j) \tag{5}$$

over all features $f \in F_c$, where $M = \|F_c\|$ is the number of training features for class c. The final value of σ_c^2 is then computed as the sample covariance

$$\sigma_c^2 = \frac{1}{M^2 - 1} \sum_{i=1}^{M} \sum_{j=1}^{M} (\mathrm{d}(f_i, f_j) - \mu_c)^2. \tag{6}$$

4.1 Maxima Extraction

To avoid issues arising from discrete Hough spaces we implemented a continuous voting space for each object class. Each voting space can be seen as a sparse representation of a probability density function. Maxima in the probability density function are detected using the Mean-shift algorithm described by Fukunaga and Hostetler [8]. We use the Mean-shift formulation by Comaniciu and Meer [5] and account for weighted votes as proposed by Cheng [4].

Given a point $\boldsymbol{x} \in \mathbb{R}^3$, the Mean-shift algorithms applies a Gaussian kernel K to all neighboring points $\boldsymbol{x}_i$ within the kernel bandwidth. Since we search for maxima in the voting space, the data points $\boldsymbol{x}_i$ are the individual votes. To find the maximum density regions, the gradient $\boldsymbol{m}_{h,g}$ of the probability density function needs to be estimated. The step size is computed adaptively. Maxima are obtained by iteratively following the direction of $\boldsymbol{m}_{h,g}$.

To create seed points for the Mean-shift algorithm a regular grid is superimposed on the data. Each cell containing at least a minimum number of data

points creates a seed point. A pruning step performs non-maximum suppression to eliminate duplicate maxima. The final probability for the detected maximum at $\boldsymbol{x}_{max}$ is given by the kernel density estimation at the maximum position in the voting space.

4.2 Pose Estimation

When casting votes into the Hough space the associated bounding boxes are transferred back into the global coordinate system using the corresponding local reference frame for the current feature. After maxima detection yields the most likely object hypotheses all votes that contributed to a hypothesis and lie around the maximum location within the kernel bandwidth are collected.

This results in a list of bounding box hypotheses weighted with the corresponding vote weight. Estimation of the bounding box is performed by creating an average bounding box based on the collected votes. While the size can be averaged, computing an average weighted rotation is more complex. The rotation matrix is converted into a quaternion representation. Averaging quaternions is achieved by computing the 4×4 scatter matrix

$$\boldsymbol{M} = \sum_{i=1}^{N} \omega_i \boldsymbol{q}_i \boldsymbol{q}_i^T \tag{7}$$

over all quaternions $\boldsymbol{q}_i$ and their associated weights ω_i. After computing the eigenvalues and eigenvectors of $\boldsymbol{M}$, the eigenvector with the highest eigenvalue corresponds to the weighted average quaternion [17]. Together with the position in the voting space, this quaternion defines the 6 DOF pose of the object.

5 Experiments and Results

In this paper we use the following datasets for evaluation (example objects from each dataset are shown in Figure 2):

1. Aim@Shape-Watertight (ASW): This dataset consists of 400 shapes in 20 different categories. The first 10 objects of each category are used for training, the remaining 10 for testing. In [27] evaluation was performed on a partial dataset (here denoted as ASWp) that consisted of 19 different categories.
2. Princeton Shape Benchmark (PSB) [21]: This dataset consists of 1814 shapes and different levels of class granularity. For better comparison with other approaches we use the class granularity named *coarse 2* (7 classes), with half of the shapes assigned for training and the other half for testing.
3. SHREC'09 (SH) [7]: This dataset from the Partial Shape Retrieval Contest has 720 objects divided into 40 classes and used for training. Classification is performed on 20 partial query shapes.
4. Stanford 3D Scanning Repository (SSR) [23]: 6 models from this dataset were used to build up 45 scenes of 3 to 5 models in [26]. The models were randomly rotated and translated, ground truth is provided. We use this dataset to evaluate the accuracy of pose estimation of our approach.

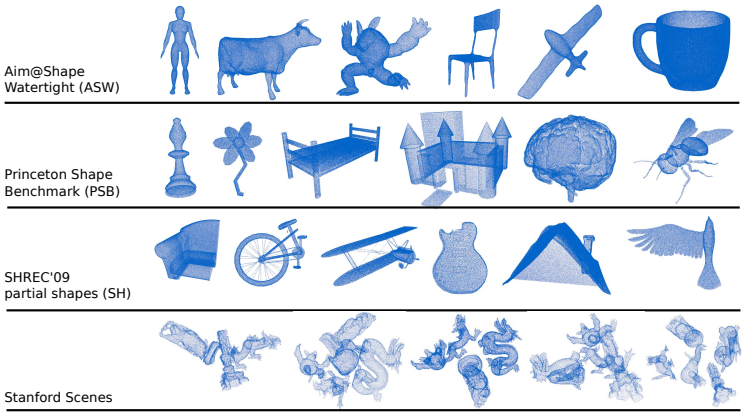

Fig. 2. Examples for the variety of different shapes in the used datasets.

Table 1. Comparison of our classification results with state of the art approaches (correct classification rate). The proposed approach outperforms previous methods on all evaluated datasets.

	Salti et al. [19]	Wittrowski et al. [27]	Seib et al. [20]	Liu et al. [16]	Toldo et al. [24]	Knopp et al. [13]	proposed approach
ASW	79%	-	80.5%	-	-	-	**85.0%**
ASWp	81%	82%	82.6%	-	-	-	**86.8%**
PSB	50.2%	-	-	55%	52%	58.3%	**61.7%**
SH	-	-	-	-	60%	40%	**70.0%**

All of these datasets are available as mesh files. We converted the meshes to point clouds and scaled each model to the unit circle for shape classification.

5.1 Shape Classification

For shape classification, each test scene consisted of a single shape without any clutter - a typical classification task for shape retrieval as might occur if shapes need to be found in a database. Evaluation was performed on previously unseen instances with a continuous and a discrete voting space. Each detected feature was matched with the $k \in \{1, \dots 5\}$ closest ones from the learned dataset to simulate different degrees of generalization. In all experiments a bandwidth of 0.5 m (half the object radius), a SHOT support radius of 0.3 m as well as the two vote weighting strategies were used (no weights and likelihood weights). The highest ranked hypothesis per object was taken as classification result.

Table 1 compares our best results with approached that use the same partitions in training and testing data as we do. The rightmost column shows that the proposed method outperforms current state of the art approaches on all tested datasets. An overview over all our results is given in Figure 3 (a)-(c). In

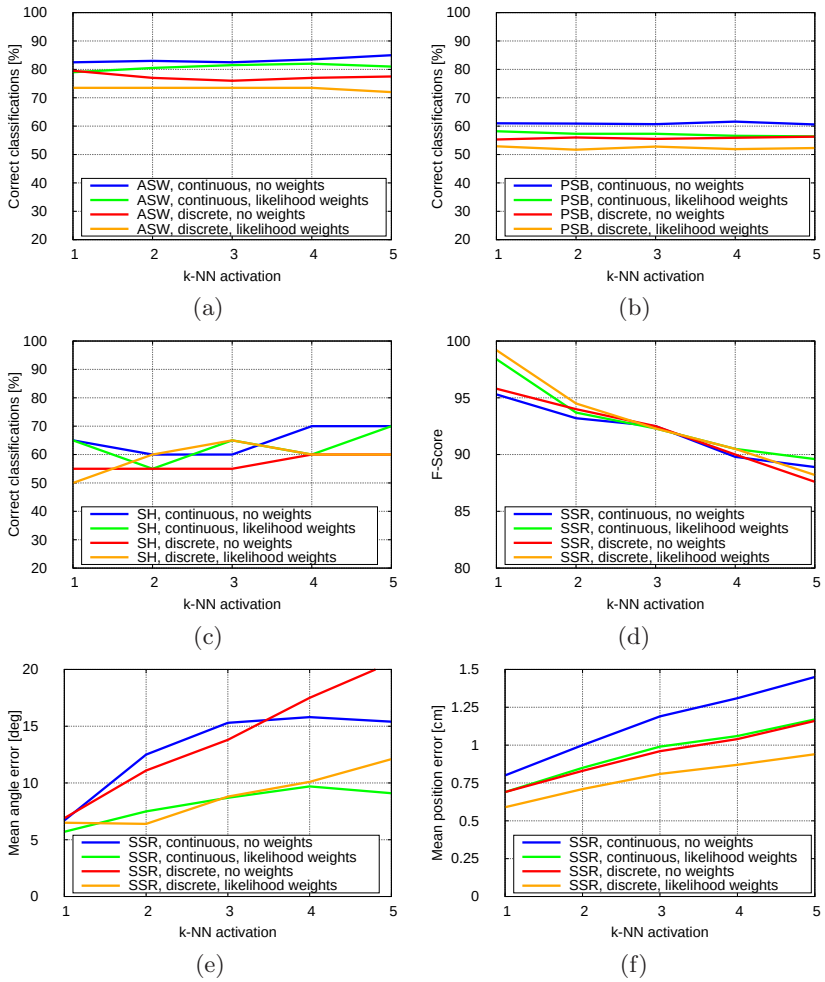

Fig. 3. Classification rates on ASW (a), PSB (b) and SH (c) datasets and the f-score (d), mean angle errors (e) and mean position errors (f) on SSR dataset are shown.

general, when classification is performed on full 3D models (ASW and PSB), the continuous voting space leads to better results. For both types of voting spaces, not using any vote weighting achieves higher classification rates. However, with partial shapes (SH) the results are not as clear as with full 3D models. Still, on average the continuous voting space performs better than the discrete one. Best classification rates were achieved with $k = 4$ or $k = 5$ on all datasets.

5.2 Pose Estimation

Pose estimation was tested on the SSR dataset. Additionally, we tested the ability of the algorithm to classify known objects in scenes. This is a particularly

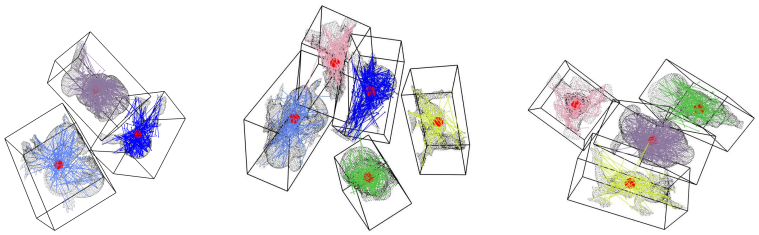

Fig. 4. Pose estimation on scenes from the SSR dataset. Object centers are shown as red dots inside the bounding box. The colored lines represent votes that contributed to the detected maximum.

different task than shape classification. In this case several maxima need to be considered and a meaningful threshold needs to be defined determining which maxima should be discarded. Further, if the bandwidth parameter is set too low, true maxima are split resulting in false positive object detections. Consequently, in these experiments we report the f-score instead of the true classification rate.

The complete scenes were scaled to the unit circle and we set the SHOT radius to 3 cm for these experiments. The bandwidth and bin size were set to 0.2 m and 0.4 m, respectively. The results are reported in Figure 3 (d)-(f). Example pose estimations are shown in Figure 4.

The resulting f-score is best for $k = 1$ activation, where both voting spaces perform better with weighted votes. For all other values for k no significant differences between the voting spaces or weighting strategies are observed.

Since vote weights are used to filter out outliers, applying likelihood weights leads to lower angular and position errors than the not weighted counterparts in the corresponding voting spaces. While the continuous voting space seems to perform worse regarding the mean position error, with higher k both voting space designs have higher errors. This is also true for the angular error, however, the errors seem to stabilize around $k = 3$ for the continuous voting space, while they continue to rise for the discrete voting space.

6 Discussion

As was shown in Section 5, the proposed approach achieves higher classification rates on the tested datasets than the state of the art. These results stem from omitting the vector quantization and instead using k-NN feature activation during classification. Additionally, a dense key point sampling leads to a better object representation than salient key point detection as in many approaches in related work.

However, this huge amount of additional features comes at the price of higher runtime during object classification. We therefore investigated the influence of codebook creation on the classification performance and runtime of the algorithm. These experiments were performed on the Aim@Shape Watertight dataset. Training was performed multiple times and a different number of

Table 2. Influence of different codebooks on the classification rate and runtime on the Aim@Shape Watertight dataset

clustering factor in codebook	classification rate	classification time per object [s]	relative classification time per object
1.0 (no clustering)	85%	18.4	1
0.7	77%	14.1	0.77
0.5	70%	13.5	0.73
0.3	60%	14.4	0.78

feature clusters were created. The number of clusters was set to 100% (no clustering), 70%, 50% and 30% of the number of all extracted features. Subsequently, classification was performed with each of these codebooks using the parameters that led to best classification results in Section 5 (SHOT radius of 0.3 m and $k = 5$). The results are reported in Table 2 and clearly show that smaller codebook sizes lead to a significant loss in the ability of the algorithm to correctly classify objects. At the same time the runtime of the algorithm also decreases. However, the gain in runtime is not as high as one might expect from the reduction of the codebook size. This is due to the fact that a lot of runtime is used for feature extraction and computation, while the feature matching is performed in a very efficient way.

7 Conclusion and Outlook

The presented approach enables us to detect and classify objects in scenes as well as determine their classes. This is supported by the good results obtained in our shape classification experiments. We showed that the choice of a continuous voting space is superior to a discrete Hough space in terms of 3D object classification. The results obtained on the three challenging shape retrieval datasets Aim@Shape Watertight, the Princeton Shape Benchmark and SHREC'09 outperform current state of the art approaches. We attribute these results to the use of the continuous voting space without feature clustering, the dense key point sampling and k-NN matching during classification.

In estimating the object's pose, both voting space designs perform similar. However, when vote weighting is used to remove outliers the errors decrease compared to not weighted votes. Specifically for the angular errors the continuous voting space is superior to the discrete voting space for higher values of k in k-NN activation.

It needs to be pointed out that the scenes in our evaluation consisted only of known objects without any clutter. Our current work thus concentrates on improving the robustness of the proposed method towards clutter and partial models. We further plan to perform experiments with data from RGB-D cameras.

References

1. Ballard, D.H.: Generalizing the hough transform to detect arbitrary shapes. Pattern Recognition 13(2), 111–122 (1981)
2. Barequet, G., Har-Peled, S.: Efficiently approximating the minimum-volume bounding box of a point set in three dimensions. Journal of Algorithms 38(1), 91–109 (2001)
3. Bronstein, A.M., Bronstein, M.M., Guibas, L.J., Ovsjanikov, M.: Shape google: Geometric words and expressions for invariant shape retrieval. ACM Transactions on Graphics (TOG) 30(1), 1 (2011)
4. Cheng, Y.: Mean shift, mode seeking, and clustering. IEEE Transactions on Pattern Analysis and Machine Intelligence 17(8), 790–799 (1995)
5. Comaniciu, D., Meer, P.: Mean shift: A robust approach toward feature space analysis. Pattern Analysis and Machine Intelligence, IEEE Transactions on 24(5), 603–619 (2002)
6. Csurka, G., Dance, C., Fan, L., Willamowski, J., Bray, C.: Visual categorization with bags of keypoints. In: Workshop on statistical learning in computer vision, ECCV. vol. 1, pp. 1–2 (2004)
7. Dutagaci, H., Godil, A., Axenopoulos, A., Daras, P., Furuya, T., Ohbuchi, R.: Shrec'09 track: querying with partial models. In: Proceedings of the 2nd Eurographics conference on 3D Object Retrieval. pp. 69–76. Eurographics Association (2009)
8. Fukunaga, K., Hostetler, L.D.: The estimation of the gradient of a density function, with applications in pattern recognition. IEEE Transactions on Information Theory 21(1), 32–40 (1975)
9. Gall, J., Lempitsky, V.: Class-specific hough forests for object detection. In: Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on. pp. 1022–1029 (June 2009)
10. Har-Peled, S.: A practical approach for computing the diameter of a point set. In: Proceedings of the Seventeenth Annual Symposium on Computational Geometry. pp. 177–186 (2001)
11. Hoppe, H., DeRose, T., Duchamp, T., McDonald, J., Stuetzle, W.: Surface reconstruction from unorganized points. In: Proceedings of the 19th Annual Conference on Computer Graphics and Interactive Techniques. pp. 71–78 (1992)
12. Knopp, J., Prasad, M., Van Gool, L.: Orientation invariant 3d object classification using hough transform based methods. In: Proceedings of the ACM Workshop on 3D Object Retrieval. pp. 15–20. 3DOR '10 (2010)
13. Knopp, J., Prasad, M., Willems, G., Timofte, R., Van Gool, L.: Hough transform and 3d surf for robust three dimensional classification. In: ECCV (6). pp. 589–602 (2010)
14. Leibe, B., Leonardis, A., Schiele, B.: Combined object categorization and segmentation with an implicit shape model. In: ECCV' 04 Workshop on Statistical Learning in Computer Vision. pp. 17–32 (2004)
15. Leibe, B., Leonardis, A., Schiele, B.: Robust object detection with interleaved categorization and segmentation. International journal of computer vision 77(1-3), 259–289 (2008)
16. Liu, Y., Zha, H., Qin, H.: Shape topics: A compact representation and new algorithms for 3d partial shape retrieval. In: Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on. vol. 2, pp. 2025–2032. IEEE (2006)

17. Markley, F.L., Cheng, Y., Crassidis, J.L., Oshman, Y.: Quaternion averaging. Journal of Guidance Control and Dynamics 30(4), 1193–1197 (2007)
18. Ohbuchi, R., Osada, K., Furuya, T., Banno, T.: Salient local visual features for shape-based 3d model retrieval. In: Shape Modeling and Applications, 2008. SMI 2008. IEEE International Conference on. pp. 93–102. IEEE (2008)
19. Salti, S., Tombari, F., Di Stefano, L.: On the use of implicit shape models for recognition of object categories in 3d data. In: ACCV (3). pp. 653–666. Lecture Notes in Computer Science (2010)
20. Seib, V., Link, N., Paulus, D.: Implicit shape models for 3d shape classification with a continuous voting space (2014), international Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications
21. Shilane, P., Min, P., Kazhdan, M., Funkhouser, T.: The princeton shape benchmark. In: Shape modeling applications, 2004. Proceedings. pp. 167–178. IEEE (2004)
22. Sivic, J., Zisserman, A.: Video google: A text retrieval approach to object matching in videos. In: Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on. pp. 1470–1477. IEEE (2003)
23. Stanford University Computer Graphics Laboratory: Stanford 3d scanning repository (Nov 2014), http://graphics.stanford.edu/data/3Dscanrep/, http://graphics.stanford.edu/data/3Dscanrep/
24. Toldo, R., Castellani, U., Fusiello, A.: A bag of words approach for 3d object categorization. In: Computer Vision/Computer Graphics CollaborationTechniques, pp. 116–127. Springer (2009)
25. Tombari, F., Di Stefano, L.: Object recognition in 3d scenes with occlusions and clutter by hough voting. In: Image and Video Technology (PSIVT), 2010 Fourth Pacific-Rim Symposium on. pp. 349–355. IEEE (2010)
26. Tombari, F., Salti, S., Di Stefano, L.: Unique signatures of histograms for local surface description. In: Proc. of the European conference on computer vision (ECCV). pp. 356–369. ECCV'10, Springer-Verlag, Berlin, Heidelberg (2010)
27. Wittrowski, J., Ziegler, L., Swadzba, A.: 3d implicit shape models using ray based hough voting for furniture recognition. In: 3DTV-Conference, 2013 International Conference on. pp. 366–373. IEEE (2013)