Recognizing 3D Objects from a Limited Number of Views using Temporal Ensembles of Shape Functions

Karla Brkić*, Siniša Šegvić*, Zoran Kalafatić*, Aitor Aldomà,‡, Markus Vincze‡
*University of Zagreb, Faculty of Electrical Engineering and Computing, Croatia
‡Automation and Control Institute, Vienna University of Technology, Austria
Email: karla.brkic@fer.hr

***Abstract*—We consider the problem of 3D object recognition, assuming an application scenario involving a mobile robot equipped with an RGB-D camera. In order to simulate this scenario, we use a database of 3D objects and render partial point clouds representing depth views of an object. Using the rendered point clouds, we represent each object with an object descriptor called temporal ensemble of shape functions (TESF). We investigate leave-one-out 1-NN classification performance on the considered dataset depending on the number of views used to build TESF descriptors, as well as the possibility of matching the descriptors built using varying numbers of views. We establish the baseline by classifying individual view ESF descriptors. Our experiments suggest that classifying TESF descriptors outperforms individual ESF classification, and that TESF descriptors offer reasonable descriptivity even when very few views are used. The performance remains very good even if the query TESF and the nearest TESF are built using a differing number of views.**

I. Introduction

3D object recognition is one of the essential tasks in practical robotics. In order to interact with the world, robots must be capable of understanding which objects they encounter in it. Although the objects in reality are three-dimensional, the dimensionality of the data that a robot perceives depends on the sensors used to acquire it. In this paper, we assume a scenario in which a robot is equipped with an RGB-D camera (e.g. Kinect or an equivalent sensor). The robot moves around the object of interest, acquiring a number of depth views of the object, as illustrated in Figure 1. We are interested exclusively in the depth channel, so the RGB image is discarded. We have previously proposed [1] a method for integrating multiple depth views of a 3D object into a single descriptor, called *temporal ensemble of shape functions descriptor* (TESF). In order to build a TESF descriptor, depth information for each view is represented as a point cloud. These point clouds are then represented using individual *ensemble of shape functions descriptors* (ESF) [2], and individual ESFs are combined to form TESF descriptors. A TESF descriptor can be built using an arbitrary number of object views. The resulting descriptor is always of the same length. Therefore, TESF descriptors are particularly suitable for object classification in practical robotics, as a robot can acquire varying numbers of views for different objects.

In this paper, we build on earlier work with temporal ensembles of shape functions. Previous experiments, detailed in [1], focused on building TESF descriptors using a *fixed number* of object views. In addition, these views were designed to capture the object from all possible sides. TESF descriptors proved to be very discriminative, performing equally well or slightly better than state of the art solutions [3]. Still, some open questions important in the context of practical robotics remained:

1) how descriptive are TESF descriptors when built on very few object views,
2) can a TESF descriptor classifier built using t_1 views be applied on TESF descriptors built using t_2 views, $t_1 \neq t_2$?

The goal of this work is to provide detailed experiments to answer these open questions. We do so by adhering to experimental setup from [1], where actual depth data is simulated using rendered 3D models.

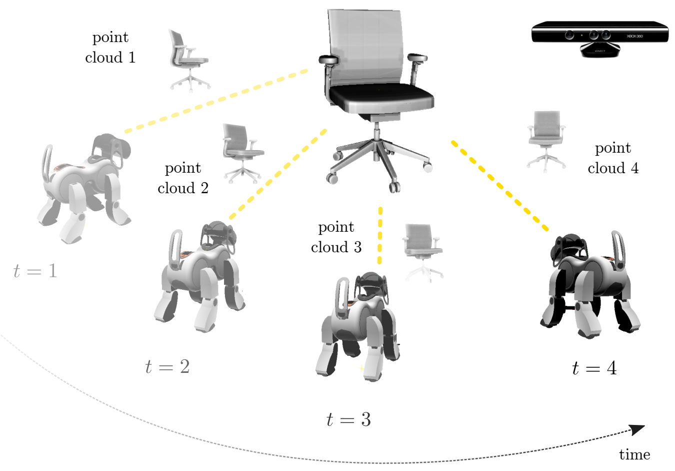

Fig. 1: Our application scenario. The robot moves around the object of interest (in this case, chair) and acquires several views (in this case, four) using a depth camera. These views are represented as point clouds.

II. Related work

The problem of 3D object recognition appears in computer (i.e. robot) vision and in computer graphics. From the computer graphics perspective, the focus is on 3D object retrieval,

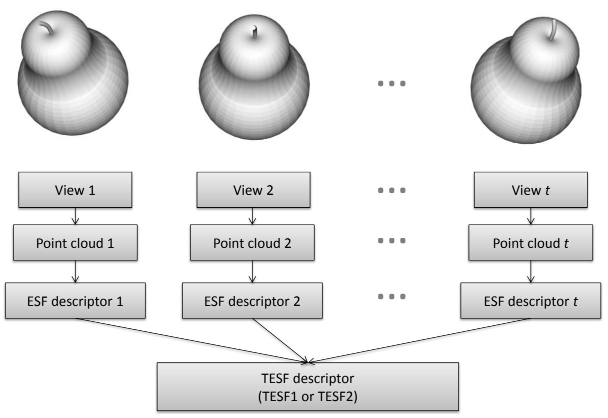

Fig. 2: From object views to TESF descriptors.

i.e. finding 3D objects similar to a query object. From the computer vision perspective, the focus is on individual real world 3D object classification.

Combining these two perspectives is becoming increasingly common, and it is difficult to draw a hard line between them. For example, Wohlkinger et al. [4] propose a 3D object recognition framework for depth sensors that is trained exclusively on CAD models downloaded from the Internet. Tenorth et al. [5] propose decomposing CAD models of daily used objects to learn about their functional parts, which could be useful in robotic applications. Ohbuchi and Furuya [6] propose a method for retrieving 3D models based on a single-view query using bags of visual features, which is an idea from computer vision. Chen et al. propose measuring similarity of contours extracted from rendered images of models, a purely computer graphics approach that could easily be generalized to computer vision. Daras and Axenopoulos [7] introduce a 3D object retrieval framework that supports multimodal queries (sketches, images or 3D models). Further details on 3D object retrieval can be found in a number of surveys, e.g. [8] or [9].

Temporal ensembles of shape functions (TESF descriptors) [1], used in this paper, are based on ensembles of shape functions [2] and spatio-temporal appearance descriptors [10]. They combine a single-view object recognition method with an idea of temporal integration from the domain of video analysis. They can be readily applied in computer graphics. When working with real images, however, RGB data itself is not sufficient, as depth channel is necessary to generate ensembles of shape functions. Different views of an object are represented as 3D point clouds, encoded as ensembles of shape functions and efficiently combined into a single descriptor. This procedure is illustrated in Figure 2.

III. Temporal Ensembles of Shape Functions

To build a TESF descriptor of an object, we assume that a number of partial point clouds representing the object are available. In our envisioned application, these partial point clouds could be obtained e.g. by a Kinect-like sensor, so that each partial point cloud represents one depth view of the object as seen by the sensor. Each of these partial views is first represented using ensembles of shape functions (ESF), introduced by Wohlkinger and Vincze [2].

A. The ESF descriptor

The basic idea of the ensemble of shape functions descriptor is to represent a partial point cloud by distributions of values of characteristic shape functions. Each point cloud represents a (partial) surface of the object. The descriptor is an extension of the idea of shape functions, as introduced by Osada et al. [11].

In order to represent the partial point cloud by a characteristic shape function, we randomly sample pairs of points on the partial point cloud surface and measure their distance. We then categorize the connecting line segment as lying mostly on the surface, lying mostly off the surface, or being a combination of both cases. For each of the cases, we maintain a 64-bin histogram of recorded distances. For instance, if we measure point distance of 12.27, and the line segment connecting the two selected points is lying mostly off the partial surface, then the value 12.27 will be entered into the histogram for lines lying off surface.

The described procedure is repeated for randomly sampled point triplets. However, instead of measuring point distances, we now measure the area of the spanned triangle, as well as a predefined angle in the triangle. The triangle is again characterized as lying mostly on the surface, lying mostly off the surface, or a combination of both, and the measured area and angle are entered into the appropriate histogram.

By randomly sampling point pairs and triplets and measuring the obtained distances, areas and angles, we obtain a total of 9 histograms. Additionally, we measure the ratios of point triplet line distances, and enter them into another histogram. This results in a total of 10 64-bin histograms that are representative of individual partial point clouds. In order to obtain the ESF descriptor of a partial point cloud, we concatenate the described ten histograms into a single 640-dimensional descriptor which is then normalized. Note that the ESF computation algorithm ensures invariance to translation and rotation.

In practice, in order to perform the described calculations we approximate the partial point surface with a voxel grid. Decisions whether a line or a triangle is on the surface, off the surface or both are made on the basis of that grid approximation. Further implementation details can be found in [2].

B. Integrating individual ESF descriptors

ESF descriptors are built on a per-view basis, meaning that for t partial views of an object we will obtain t ESF descriptors. In a real robotics scenario, we would like to classify the object that the robot is seeing using all available descriptors. However, most standard classifiers expect a single fixed-length feature vector as an input, so we need a way to integrate t ESF descriptors into a single descriptor.

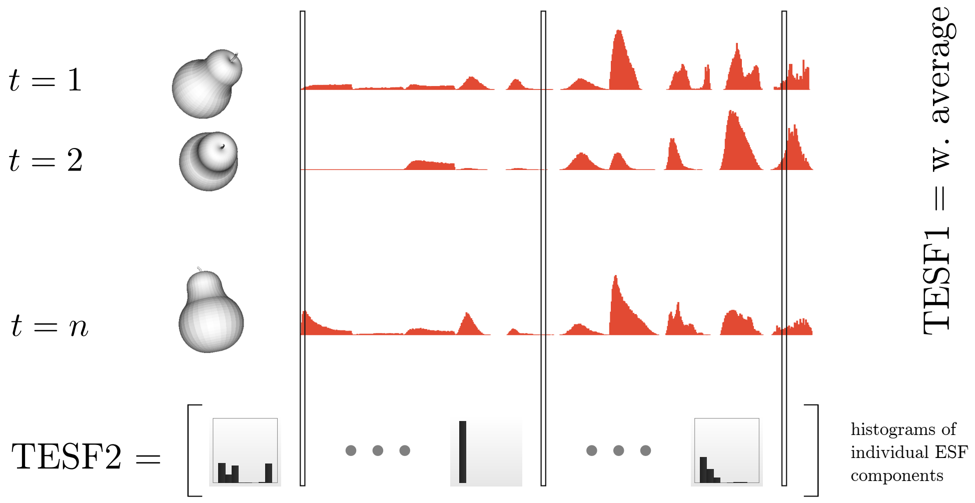

Fig. 3: Calculating TESF1 and TESF2 descriptors. Given a number of views and their corresponding ESF descriptors (drawn in orange), TESF1 is obtained by weighted averaging of all the available ESFs. TESF2 is obtained by histogramming each ESF component (represented here by framing three example components with a rectangle and building a histogram out of the collected data).

In [1] we proposed two variants of integrating ESF descriptors into a single descriptor, namely

1) Temporal Ensembles of Shape Functions of the First Order (TESF1), and
2) Temporal Ensembles of Shape Functions of the Second Order (TESF2).

Let us assume that the robot is moving around the object and looking at it, so that in each point in time θ we obtain a view v_θ and its corresponding ESF descriptor $\mathrm{ESF}(v_\theta)$. We now proceed to describe the algorithms for calculating TESF1 and TESF2 descriptors.

C. The TESF1 descriptor

Assume that we wish to build the TESF1 descriptor at the point in time t. We have already obtained t views of the object and their corresponding ESF descriptors $\mathrm{ESF}(v_\theta)$, $1 \leq \theta \leq t$. In order to obtain the TESF1 descriptor from these views, we simply do a weighted averaging of the t ESF descriptors:

$$\mathrm{TESF1}(t) = \sum_{\theta=1}^{t} \alpha_\theta \mathrm{ESF}(v_\theta). \tag{1}$$

Factors α_θ are arbitrary, and should be set depending on a particular application. For instance, it might be the case that all the views are equally relevant, so it makes sense to set $\alpha_\theta = 1 \; \forall \theta$. On the other hand, it might be the case that we know something important about the way the views were acquired, for instance that the robot is moving away from the object. In that particular case it makes sense to decrease the importance of latter views, e.g. by setting $\alpha_\theta = \frac{c}{\theta}$, where c is an arbitrary constant. Different strategies for choosing α_θ can be derived by analogous reasoning.

TESF1 is a simple and effective descriptor [1]. However, when averaging individual ESF descriptors, a lot of information is lost. Two very different sets of ESF descriptors can produce the same TESF1 descriptor. To address this problem, we have proposed the TESF2 descriptor.

D. The TESF2 descriptor

The TESF2 descriptor is designed to be more expressive than TESF1, by explicitly modeling the distribution of ESF descriptor components over time.

Let us define a component vector, $\mathbf{c}_i(t)$, as a vector of values of the i-th component $\mathrm{ESF}(v_\theta)[i]$ of the ESF descriptor $\mathrm{ESF}(v_\theta)$ up to and including time t, $1 \leq \theta \leq t$:

$$\mathbf{c}_i(t) = \left(\mathrm{ESF}(v_1)[i],\ \mathrm{ESF}(v_2)[i],\ \ldots,\ \mathrm{ESF}(v_t)[i]\right)^T. \tag{2}$$

For a given point in time t, we will have a total of 640 component vectors $\mathbf{c}_i(t)$, where $1 \leq i \leq 640$. There will be one component vector for each of the 640 components of the modeled ESF descriptors. In time t, the length of each of these component vectors will be equal to t.

To obtain the TESF2 descriptor in time t, each of the 640 component vectors is treated as a set of measurements and modeled with a k-bin histogram. The TESF2 descriptor is a concatenation of the bin frequencies of all 640 histograms, which can be written as

$$\mathrm{TESF2}(t) = \left[\mathcal{H}_k(\mathbf{c}_1(t)), \mathcal{H}_k(\mathbf{c}_2(t)), \ldots, \mathcal{H}_k(\mathbf{c}_{640}(t))\right]^T. \tag{3}$$

The function $\mathcal{H}_k(\mathbf{c})$ builds a k-bin histogram of values contained in the vector $\mathbf{c}_i$ and returns a vector of histogram bin frequencies. Given that the individual components of the grid vector correspond to bins of individual ESF histograms, TESF2 descriptors can be thought of as building histograms of the second order, i.e. histograms of histograms.

Figure 3 illustrates the differences in calculating TESF1 and TESF2 descriptors.

E. Solving the TESF2 binning problem

In practical implementations of the histogramming function $\mathcal{H}_k(\mathbf{c})$, there is a problem in defining bin bounds of the built histograms. The maximum theoretical value that can be encountered in the components of an ESF descriptor is 1, as ESF descriptors are normalized. In practice, ESF descriptor components are typically of the order of magnitude of 10^{-3}, given that the normalization is performed over 640 vector components. Therefore, it makes sense to employ non-linear binning, in order to increase the binning resolution around typically encountered values. In our experiments, we employ the non-linear binning scheme from [1]. Further details on the binning are omitted from this work, and the interested reader is referred to [1]

IV. Experiments with a limited number of views

In order to test the performance of TESF descriptors built using varying numbers of views, we adhere to the experimental setup from [1]. To simulate the performance of a Kinect-like sensor, we use a database of 3D object models and partially render each model from 20 different angles. The 20 views are evenly distributed on a sphere around the object, and each rendered view is assigned an index from 1 to 20. We employ a 1-nearest neighbor classifier [12] using Euclidean distance and leave-one-out cross-validation to obtain performance estimates. We use the publicly available implementation of ESF available in the Point Cloud Library (PCL) [13]. The number of point-pair and point-triplet samples is set to 40000, and a $64 \times 64 \times 64$ voxel grid is used.

A. The 3D-Net database

In our experiments, we use a subset of the 3D-Net database [2]. The considered subset consists of 1267 objects grouped into 55 classes organized in a hierarchy according to the WordNet database [14]. The object classes range from common household objects to vehicles and animals. A word cloud visualization of all the used object categories is shown in Figure 4, and a few example objects in Figure 5.

B. The baseline: individual ESF descriptor matching

The simplest way to classify an object seen from multiple views is to ignore the fact that the views are related and to treat each view as an individual observation. In the case of 1-NN classification, object class is then determined by comparing a view with a database of all stored views of all the training objects. Assume that we are using the described subset of 1267 objects from 3D-Net as our training set and 20 views per object. This effectively means comparing the ESF descriptor of a view of a new object with $1267 \times 20 = 25430$ stored ESF descriptors. This procedure will be 20 times slower than analogous matching of TESF1 descriptors and $\frac{20}{k}$ times slower than analogous matching of TESF2 descriptors, where k denotes the number of bins used for building TESF2 descriptors.

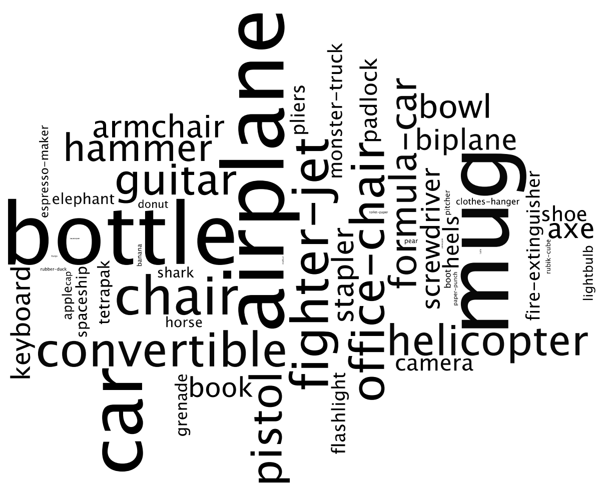

Fig. 4: A word cloud visualization of the considered classes from the 3D-Net database. Word size is proportional to the number of instances of the given class.

Fig. 5: Examples of objects from the 3D-Net database.

In order to speed up processing, one could consider matching only corresponding views. For example, if we assume our setup where 20 views are spaced at predefined positions around the object, view at position 14 of a query object could be compared only to views at position 14 of all other objects. The major problem with this kind of setup is that objects in the 3D-Net database are not necessarily equally aligned in the 3D coordinate system, and their orientation is not known. Taking the same view of two objects of the same class in the 3D-Net database might yield quite different surfaces, as illustrated in Figure 6. Therefore, we instead opt for comparing the ESF

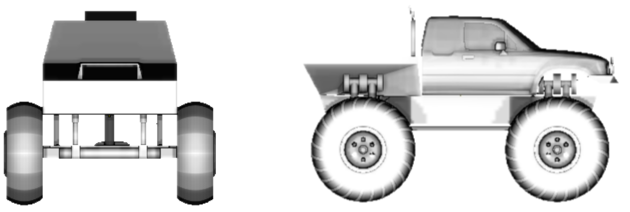
Fig. 6: The same view (view 3) of two different trucks from the 3D-Net database. The trucks are oriented differently in the 3D space, so viewing them from the same position yields two different surfaces.

descriptor of a query view with all other views of all other objects.

To investigate classification performance of individual ESF descriptor matching, we set up the following experiment. For each of the 1267 objects in the training set, we consider all 20 views of the object, and try to match each view's ESF descriptor with the remaining $(1267 - 1) \times 20$ ESF descriptors of all views of all other objects.

This kind of matching yields a 1-NN leave-one-out classification accuracy of 65.99%. In [1], we have shown that by using TESF1 descriptors on the same problem one obtains a classification accuracy of 77.03%, and by using TESF2 descriptors a classification accuracy of 82.64%. Hence, using TESF descriptors we obtain not only faster, but also more accurate classification.

C. Varying the number of views for query TESFs

Although TESF descriptors perform very well when sufficient number of views is sampled around the object of interest, there are two essential questions that need to be addressed when applying TESF descriptors in a practical robotics scenario: (i) what is the influence of the number of used views on classification accuracy, and (ii) is this approach robust to a changing number of views?

As mentioned before, we are assuming that a robot is moving around an object of interest and acquiring a number of views of the object. These views are then represented by their ESF descriptors. The robot need not be able to see the object from all angles, and the number of acquired views can vary depending on the situation. Using these views, the robot builds a TESF descriptor and tries to determine which class the object belongs to by finding the nearest neighbor to the built TESF descriptor in its *object descriptor database*. The object descriptor database contains precomputed TESF descriptors of training objects. The stored TESF descriptors need not be built on the same views as the query descriptor. Both view angles and the total number of views might differ.

To investigate the robustness of TESF classification to these changes in views, we vary the number of views used in building TESF descriptors of query objects and measure classification performance. At the start of each experiment, we randomly select n view indices from the rendered 20 views of objects in the 3D-Net (e.g. when selecting three views, views 3, 6 and 17 might be chosen). When evaluating leave-one-out classification accuracy, we select individual objects from the training set and build their TESF descriptors using only the chosen n view indices. 1-NN classification of these descriptors is then performed by searching for the nearest TESF descriptor in the rest of the set. However, for the remainder of the set (which simulates the object descriptor database of the robot) we use the full 20-view range TESF descriptors. In other words, when finding the nearest neighbor the left out object is represented by a TESF descriptor using n views, and this descriptor is then compared to TESF descriptors of other objects built using 20 views to find the closest match. We test both TESF1 and TESF2 descriptors.

Randomly selecting view numbers can be biased. It is possible to select very close views, leading to small information gain over a single view scenario. On the other hand, it is also possible to select views that are very far apart, which might not be the case in real robotic scenarios, leading to overly optimistic classification rates. To alleviate this problem, we repeat the random selection of views 10 times for each considered number of views n, and we report the average obtained classification accuracy and the standard deviation. Results are summarized in Table I.

Our first observation is that TESF1 descriptors consistently perform worse than TESF2 descriptors, as is to be expected given that TESF2 descriptors are more expressive. However, TESF1 descriptors built using 8 views and above offer an improvement over individual ESF descriptor matching. For TESF2, we see that even with using very few views (3), we are still likely to see an increase in performance over individual ESF 1-NN classification (although the standard deviation of performance is quite large). The classification performance seems to steadily increase as we add more views, and the standard deviation of the performance drops. As our views are equally positioned around the object, selecting more and more views means obtaining more information about the object. At 8 views the object is already reasonably well represented, and at 15 views the standard deviation is very small, which means that regardless of which exact 15 views we choose, the majority of the surface of the object will be seen and captured in the resulting TESF descriptor.

D. Building the object descriptor database from fewer views

In previous experiments, we considered matching TESF descriptors of query objects built using randomly selected 3, 5, 8 and 15 views with an object descriptor database built using 20 object views. Now, we investigate how classification performance would change if our object descriptor database instead contained descriptors built using the same number of views as the query object. Our goal is to see whether good classification accuracy can be obtained even if the objects stored in the database were seen from a limited number of views, which is s realistic expectation in robotic applications. To that end, we repeat the experiment described in the previous

Number of views n	TESF1 accuracy [%]	TESF2 accuracy [%]
1 (ESF)	65.99	
3	57.64 ($\sigma = 4.27$)	70.19 ($\sigma = 4.10$)
5	62.00 ($\sigma = 5.75$)	75.94 ($\sigma = 3.09$)
8	71.87 ($\sigma = 1.93$)	80.64 ($\sigma = 1.14$)
15	76.55 ($\sigma = 0.81$)	82.09 ($\sigma = 0.46$)
20 [1]	77.03	82.64

TABLE I: Influence of the number of views used to build TESF descriptors on classification accuracy. For each number of views (3, 5, 8, 15), we randomly select view indices 10 times and run leave-one-out 1-NN cross-validation. Nearest neighbors are found from an object database of 20-view TESF2 descriptors. We report mean classification accuracy and standard deviation over the 10 iterations.

Number of views n	Accuracy over 10 runs [%]
3	54.97 ($\sigma = 4.01$)
5	66.85 ($\sigma = 7.34$)
8	81.31 ($\sigma = 0.61$)
15	82.09 ($\sigma = 0.46$)

TABLE II: Using an object descriptor database containing descriptors built from fewer than 20 views. For each number of views (3, 5, 8, 15), we randomly select view indices 10 times and run leave-one-out 1-NN cross-validation. Nearest neighbors are found in a database built using the same number of views as the query objects, but differing in view indices. We report mean classification accuracy and standard deviation over the 10 iterations.

section, but this time with an object descriptor database built using the same number of views as the query: 3, 5, 8 and 15. Given that TESF2 consistently performed better than TESF1 in previous experiments, this experiment is done with TESF2 descriptors only.

In this experiment, we again measure the leave-one-out classification accuracy over 10 iterations, randomly selecting the views used for building the query objects in each iteration. The object descriptor database is built only once for each considered number of views, and view indices for building the object descriptor database are selected randomly (e.g. for 5 views they are 17, 12, 14, 19, and 2).

Results are summarized in Table II. We see that for 3 and 5 views the results are worse than when 20-view descriptors are used in the object descriptor database, while for 8 and 15 views the results are quite similar. Adequate accuracy is obtained if the object descriptor database is built using a sufficient number of views that need not necessarily include all view angles. This finding means that TESFs could be useful in scenarios where learning, i.e. building the object descriptor database, is conducted *on the robot*. Assume a robot equipped with some kind of grasper, for instance a humanoid robot with arms. If the robot is holding an object in its hand, it cannot see it from all sides without moving it to the other hand and manipulating it. However, by simply rotating its hand to obtain more views of the object, it could learn the object well enough to be able to classify similar objects later.

V. Conclusion and outlook

We have shown that TESF descriptors retain a lot of descriptivity when built using only a few views, and offer performance increases over simple matching of ESF descriptors. Our method can effectively combine any number of views into a single descriptor, and the more surface of the object is covered with the views, the better TESF performs. TESF descriptors can be compared to one another in a 1-NN classification setting even if they were built using varying number of views. This makes them especially interesting in robotic applications, where the robot sees an object of interest from a number of views t_1, but stores representations of similar objects built using t_2 views, where in general $t_1 \neq t_2$. Further analysis should be performed regarding the usability of TESF descriptors in the worst case scenario, where the robot sees an object from a limited viewpoint, i.e. from a series of very close views.

Acknowledgments

This research has been supported by the research project Research Centre for Advanced Cooperative Systems (EU FP7 #285939).

References

[1] K. Brkić, A. Aldomá, M. Vincze, S. Šegvić, and Z. Kalafatić, "Temporal Ensemble of Shape Functions," in *Eurographics Workshop on 3D Object Retrieval*, (Strasbourg, France), pp. 53–60, Eurographics Association, 2014.

[2] W. Wohlkinger and M. Vincze, "Ensemble of Shape Functions for 3D Object Classification," in *IEEE International Conference on Robotics and Biomimetics (IEEE-ROBIO)*, 2011.

[3] M. Kazhdan, T. Funkhouser, and S. Rusinkiewicz, "Rotation invariant spherical harmonic representation of 3D shape descriptors," in *Symposium on Geometry Processing*, 2003.

[4] W. Wohlkinger, A. Aldoma, R. B. Rusu, and M. Vincze, "3DNet: Large-scale object class recognition from CAD models," in *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, pp. 5384–5391, IEEE, 2012.

[5] M. Tenorth, S. Profanter, F. Balint-Benczedi, and M. Beetz, "Decomposing CAD models of objects of daily use and reasoning about their functional parts," in *In IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2013.

[6] R. Ohbuchi and T. Furuya, "Scale-weighted dense bag of visual features for 3D model retrieval from a partial view 3D model," in *Computer Vision Workshops (ICCV Workshops), 2009 IEEE 12th International Conference on*, pp. 63–70, Sept 2009.

[7] P. Daras and A. Axenopoulos, "A 3D shape retrieval framework supporting multimodal queries," *Int. J. Comput. Vision*, vol. 89, pp. 229–247, Sept. 2010.

[8] J. W. Tangelder and R. C. Veltkamp, "A survey of content based 3D shape retrieval methods," *Multimedia Tools Appl.*, vol. 39, pp. 441–471, Sept. 2008.

[9] Q. Liu, "A survey of recent view-based 3D model retrieval methods," *CoRR*, vol. abs/1208.3670, 2012.

[10] K. Brkić, A. Pinz, S. Šegvić, and Z. Kalafatić, "Histogram-based description of local space-time appearance," in *Proceedings of the 17th Scandinavian Conference on Image Analysis*, SCIA'11, pp. 206–217, 2011.

[11] R. Osada, T. Funkhouser, B. Chazelle, and D. Dobkin, "Shape distributions," *ACM Trans. Graph.*, vol. 21, pp. 807–832, Oct. 2002.

[12] T. Cover and P. Hart, "Nearest neighbor pattern classification," *Information Theory, IEEE Transactions on*, vol. 13, pp. 21–27, January 1967.

[13] R. B. Rusu and S. Cousins, "3D is here: Point Cloud Library (PCL)," in *IEEE International Conference on Robotics and Automation (ICRA)*, (Shanghai, China), May 9-13 2011.

[14] C. Fellbaum, ed., *WordNet: an electronic lexical database*. MIT Press, 1998.