Eurographics Workshop on 3D Object Retrieval (2014)
B. Bustos, H. Tabia, J.P. Vandeborre, and R. Veltkamp (Editors)

Temporal Ensemble of Shape Functions

Karla Brkić[1], Aitor Aldomà[2], Markus Vincze[2], Siniša Šegvić[1], Zoran Kalafatić[1†]

[1]University of Zagreb, Faculty of Electrical Engineering and Computing, Croatia
[2]Automation and Control Institute, Vienna University of Technology, Austria

Abstract

This paper proposes novel descriptors that integrate information from multiple views of a 3D object, called Temporal Ensemble of Shape Functions (TESF) descriptors. The TESF descriptors are built by combining per-view Ensemble of Shape Functions (ESF) descriptors with Spatio-Temporal Appearance (STA) descriptors. ESF descriptors provide a compact representation of ten different shape functions per object view (obtained by virtually rendering the object from different viewpoints), and STA descriptors efficiently combine ESF descriptors of multiple object views. The proposed descriptors are evaluated on two publicly available datasets, the 3D-Net database and the Princeton Shape Benchmark. They provide a good performance on both datasets, similar to that of the Spherical Harmonic Descriptor (SHD), with the advantage that because of their view-based nature the TESF descriptors might prove useful for the problem of object classification from limited viewpoints. Such property is of special interest in robotics where the agent is able to move around the object to improve single-view results.

Categories and Subject Descriptors (according to ACM CCS): I.3.3 [Computer Graphics]: Picture/Image Generation—Line and curve generation, I.4.7 [Image Processing and Computer Vision]: Feature Measurement—Feature representation, I.5.4 [Pattern Recognition]: Applications—Computer vision

1. Introduction and related work

3D model retrieval has been extensively pursued in recent decades. These efforts have focused on designing efficient systems that return a list of *similar* models to those provided by the system's user. The problem usually boils down to finding a compact representation of the models' geometrical traits (or functional properties) in order to efficiently compute a similarity coefficient between the *query* model and the model database previously known by the system. The output is simply a sorted list of models based on such coefficients. Retrieval systems have raised much interest within heterogeneous communities thanks to development of several technologies enabling rapid creation and productive sharing of 3D models.

Examples of recent applications have been envisioned in robotics: (i) [WARV12] builds an object recognition system for depth sensor devices entirely trained on CAD models downloaded from the Internet while (ii) [TPBBB13] uses 3D models representation to build a knowledge base on how to manipulate daily objects to fulfill certain tasks. 3D model retrieval systems are a key component to render such systems scalable.

Tangelder et al. provide in [TV08] a survey on 3D object retrieval, including an extensive review of different shape matching paradigms. In [Liu12], a survey of recent view-based methods for object retrieval is presented. In particular, [CTSO03] extracts silhouettes from rendered images of the model to establish a similarity measure between query and target models. [OF09] presents a view-based approach based on image and depth features for the retrieval of models provided with a single-view representation of a query object. [DA10] proposes a unified framework for model retrieval accepting multimodal queries (sketch, 2D images or 3D models).

Similar to the aforementioned methods, our method is also based on single-views obtained by rendering the object from uniformly distributed viewpoints. Instead of 2D images of the object like in [CTSO03], the different views of the object are represented as 3D point clouds and efficiently

† This research has been supported by the Research Centre for Advanced Cooperative Systems ACROSS (EU FP7 #285939) and the University of Zagreb grant VIF2013-26.

encoded by means of a rotationally invariant 3D shape descriptor. The descriptors extracted from each view are combined into a single descriptor before being matched to the database. In contrast to other view-based approaches, our method does not require a pose normalization stage (contrary to [DA10]) and the matching stage is very efficient since the query model and, the models in the database, are represented by a single descriptor and thus, does not require matching of individual images.

In more detail, we propose a shape matching scheme based on the combination of a 3D shape global descriptor (for different views of an object) and a temporal descriptor. The 3D global descriptor compactly encodes geometric statistics of the different views surfaces while the temporal descriptor merges and summarizes that information into a single descriptor that efficiently represents the whole 3D model. The system requires that the user query is provided in the form of a full 3D model. We evaluate the performance of the method on two large datasets of 3D models organized into 55 and 161 categories. The goal of the overall system is to provide a class for a query model based on its similarity to the models within the available classes. As outlined before, such system might be used to minimize human interaction when adding new 3D models to a knowledge-base used for object classification.

On a general level, this work explores strategies to accumulate information for the description of object's shapes over time. In this paper, we limit ourselves to testing the performance of the method using a fixed number of views providing a uniform coverage of the viewpoint space of objects. This serves as an initial estimate of its applicability for the more challenging case where the availability of multiple viewpoints is limited by physical constraints of the robot embodiment as well as constraints posed by different environment configurations.

2. Ensemble of Shape Functions

The Ensemble of Shape Functions (ESF) descriptor introduced in [WV11] is an ensemble of ten 64-bin sized histograms (resulting in a total descriptor size of 640 bins) of shape functions describing characteristic properties of the point cloud. The shape functions consist of angle (point triplets), point distance (point pairs) and area shape (point triplets) distributions. A voxel grid ($64 \times 64 \times 64$) serves as an approximation of the real surface and is used to efficiently trace the line joining a point-pair sample. By tracing a line within the voxel grid, the statistics related to the different shape functions can be classified to be either "on the surface", "off the surface" or a combination of both (see Figure 1). By design, ESF is invariant to translation and rotation. Since CAD models are not always represented in the real scale of the object, scale invariance is obtained by scaling the point cloud under consideration to the unit sphere before the voxel grid construction. The public implementation of ESF available in the Point Cloud Library (PCL) is used in this work. The number of point-pair and point-triplet samples is set to 40000, resulting in a fixed computational cost, regardless of the number of points in the input point cloud.

ESF was designed for the problem of object classification [WARV12] on range data. For this purpose, during an offline training stage, CAD models of objects belonging to different classes were virtually rendered from different viewpoints and the resulting point clouds encoded using the ESF descriptor. During recognition, objects within the scene under consideration are segmented out and represented by means of an ESF histogram which is used to efficiently retrieve the k closest matches from the offline generated database, effectively providing a classification result for each object in the scene [CH67].

Even though originally designed to describe partial views of an object, the ESF descriptor can be computed without any modification on the full 3D point cloud of a model (i.e, obtained by densely sampling the surface mesh of a CAD model). Within this paper, the ESF descriptor computed on full object point clouds is dubbed *ESF-Full*.

3. Spatio-temporal appearance descriptors

Spatio-temporal appearance (STA) descriptors [BPSK11] are local, histogram-based descriptors that compactly represent appearance of an object of interest through time. They are used in computer vision to build a representation of an object of interest in a video sequence. This representation is available in every frame of the sequence, and it encodes information about the object available up to and including the considered frame.

In order to build an STA descriptor of an object of interest, it is assumed that a bounding box of the object is known in every point in time. In each considered frame of the video sequence, the bounding box around the object is divided into a regular grid of a predefined size $n \times m$, where n is the number of rows of the grid and m is the number of columns. Each patch of the grid is represented by a k_1-bin histogram of an arbitrary image function (e.g. hue, saturation, gradient etc.). This intermediate representation, called the grid-of-histograms (GoH) representation, consists of $n \times m$ k_1-bin histograms. By discarding bin boundary information and concatenating the GoH histograms into a single feature vector, we obtain a vector of $n \times m \times k_1$ elements in each frame θ, which we call *grid vector* and denote as $\mathbf{g}^{(\theta)}$.

Two types of STA descriptors exist: STA descriptors of the first order (STA1 descriptors) and STA descriptors of the second order (STA2). The two types of descriptors differ in the ability to model complexity of the underlying spatio-temporal phenomena. STA1 descriptors are based on simple averaging, and therefore more suitable for representing simpler spatio-temporal structure, such as objects with little

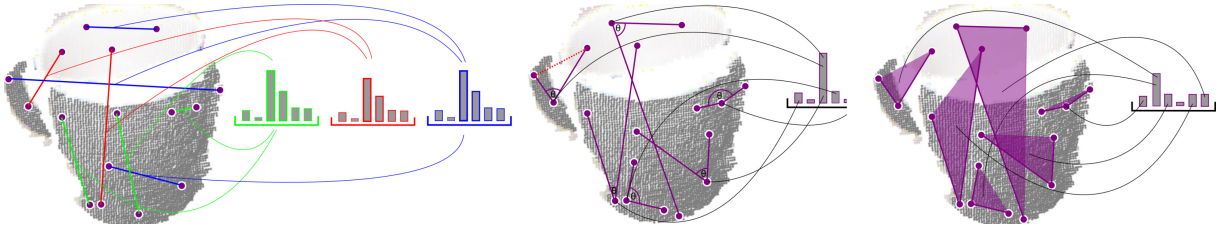

Figure 1: Calculation of the shape functions on an example point cloud of a mug. Left: point distance distributions; Middle: angle distributions; Right: area spanned by triplets of sampled points. Measurement are classified into "on the surface", "off surface" as well as a combination of "both", respectively depicted as green, red and blue lines in the left part of the figure. Pictures borrowed from [WV11].

variance. On the other hand, STA2 descriptors model *distributions* of spatio-temporal appearance, and are better to use on more complex problems (e.g. objects with a lot of appearance change, deformable objects and similar).

To calculate the STA1 descriptor at the point in time t, one does a weighted averaging of the grid vectors from all frames θ, $1 \leq \theta \leq t$,

$$\mathrm{STA}_1(t) = \sum_{\theta=1}^{t} \alpha_\theta \mathbf{g}^{(\theta)}. \tag{1}$$

In the STA1 representation, the information about the distribution of individual grid vector components through time is lost. For example, assume that three grid vectors are available. Let us consider the first component of the grid vector, and assume that (i) it is constantly 0.4 throughout the three observations, and (ii) it takes on the values of 0.1, 0.9 and 0.2. In both cases, assuming equal weighting, the computed first component of the STA1 descriptor will be 0.4, although a look at the underlying grid vectors indicates two different spatio-temporal behaviors (constancy vs. sharp change).

The STA2 descriptor is designed to solve the problem of losing underlying component distribution by explicitly modeling the distribution of grid vector components over time. Let us define a component vector, $\mathbf{c}_i^{(t)}$, as a vector of values of the i-th component $\mathbf{g}^{(\theta)}(i)$ of the grid vector $\mathbf{g}^{(\theta)}$ up to and including time t, $1 \leq \theta \leq t$:

$$\mathbf{c}_i^{(t)} = \left[\mathbf{g}^{(1)}(i), \mathbf{g}^{(2)}(i), \mathbf{g}^{(3)}(i), \ldots, \mathbf{g}^{(t)}(i)\right]^T. \tag{2}$$

To obtain the STA2 descriptor in time t, one builds a k_2-bin histogram, called the STA2 histogram, out of each of the $m \times n \times k_1$ component vectors. The STA2 descriptor is a concatenation of the bin frequencies of all $m \times n \times k_1$ STA2 histograms, which can be written as

$$\mathrm{STA}_2(t) = \left[\mathcal{H}_{k_2}(\mathbf{c}_1^{(t)}), \mathcal{H}_{k_2}(\mathbf{c}_2^{(t)}), \ldots, \mathcal{H}_{k_2}(\mathbf{c}_{mnk_1}^{(t)})\right]^T. \tag{3}$$

The function $\mathcal{H}_{k_2}(\mathbf{c})$ builds a k_2-bin histogram of values contained in the vector $\mathbf{c}$ and returns a vector of histogram bin frequencies. Given that the individual components of the grid vector correspond to bins of individual grid histograms, STA2 descriptors can be thought of as building histograms of the second order, i.e. histograms of histograms.

As the grid vectors are normalized, the maximum possible value that a grid vector component can take is 1, in case when all other components are 0. Therefore, in the original work on STA descriptors it is proposed to obtain the bin boundaries of STA2 histograms by uniformly dividing the interval $[0,1]$ into k_2 bins [BPSK11].

4. Temporal Ensembles of Shape Functions

It is our intention to generalize STA descriptors to the problem of 3D object retrieval by replacing the notion of frames θ, $1 \leq \theta \leq t$ (as defined in the original framework) with individual views of a 3D object. Instead of building a GoH representation of an object in each video frame, a 3D object is represented by an ESF descriptor in each of its views. Essentially, we combine STA and ESF descriptors by having the ESF descriptor take the role of a grid vector in the original STA framework:

$$\mathbf{g}^{(\theta)} = \mathrm{ESF}(\theta),\ \theta = 1, \ldots, N, \tag{4}$$

where $\mathrm{ESF}(\theta)$ denotes the ESF descriptor calculated for view θ of the object, and N is the total number of available views. Following this assumption, STA1 and STA2 descriptors can be calculated from Equations 1 and 3. Different views of the object can be viewed as different points in time. The calculated STA1 and STA2 descriptors built on top of ESF descriptors are therefore called Temporal Ensembles of Shape Functions of the First Order (TESF1) and Temporal Ensembles of Shape Functions of the Second Order (TESF2).

TESF descriptors have two important properties. First, the ordering of the used views does not influence the built TESF descriptor. Permuting views in any way always results in the same TESF2 descriptor, and results in the same TESF1 descriptor if we assume equal weighting. Second, the TESF descriptor can be built using any number of views, and the resulting representation will always have the same length.

5. Solving the STA2 binning problem in TESF

As mentioned previously, ESF descriptors consist of ten concatenated histograms of 64 bins, resulting in a total vector length of 640. There is one important issue to consider when building TESF2 descriptors, concerning bin boundaries of the STA2 histograms. Let us consider the original formulation of STA2 descriptor calculation, where STA2 descriptors are calculated using grid vectors generated by the grid-of-histograms representation. In this formulation, STA2 histogram bin boundaries are obtained by equally dividing the interval $[0,1]$ into k_2 bins. Component vectors used to build STA2 histograms represent bin values of normalized grid histograms that have k_1 bins. As we have no prior knowledge on the grid histograms, let us assume that they are uniform, i.e. that the frequency of each bin is equal, $1/k_1$. Therefore, the values in component vectors will also group around $1/k_1$. If one builds a histogram of such vectors, one can expect the most filled bin to be the one that contains the value $1/k_1$, with the bin frequency decreasing the further we get from this value towards 1.

Assuming a large value of k_1, it makes sense to reconsider the originally proposed uniform division of the interval $[0,1]$ into k_2 bins, as it can be expected bins further from $1/k_1$ will be sparsely populated. When the number of considered grid histogram bins k_1 is low (up to 10), the sparsity of the histogram is not very pronounced. Additionally, a grid vector is normalized part-wise, so that each concatenated histogram is normalized. Its $n \times m \times k_1$ components sum to the total number of histograms in a grid, $n \times m$. On the other hand, an ESF descriptor consists of 10 64-bin histograms, and it is normalized so that its 640 components sum to 1, so it can effectively be viewed as a single histogram of 640 bins.

The expected value in the component vectors built using ESF descriptors as a basis is 1/640, assuming a uniform distribution of the ESF descriptor components. Earlier research on STA2 descriptors suggests to use up to 10 bins for STA2 histograms. However, it is clear that using only 10 bins for the STA2 histograms in this case would yield STA2 histograms with the value of 1 in the first bin, and zeros in all remaining bins, as the values in component vectors are typically around 1/640. Therefore, an adaptation of the original STA2 binning scheme is needed. One could consider drastically increasing the number of bins in STA2 histograms (up to 1000), but this approach would significantly increase the length of the resulting STA2 descriptor and introduce a lot of unnecessary sparsity. We use a different, data-driven approach, in which we estimate a prior of bin values and adjust the bin boundaries accordingly. Our approach is as follows:

1. For each object view in the training set, we build an ESF descriptor.
2. We generate a set of values by adding all 640 components of all calculated ESF descriptors into the set.
3. For the generated set, we find the mean μ, the standard deviation σ and the maximum value M.
4. We divide the interval $[0, \mu + 3\sigma]$ into k_2 - 2 bins. The remaining two bins are $(\mu + 3\sigma, M]$ and $(M, 1]$.

This procedure is illustrated in Fig. 2

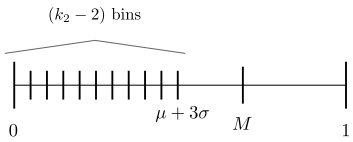

Figure 2: Calculating bin boundaries for the STA2 descriptor built on top of the ESF descriptor.

6. Experiments

In the experiments, we evaluate the performance of the TESF1 and TESF2 descriptors in the task of 3D object retrieval, and compare it to the performance of ESF-Full and Spherical Harmonic Descriptor (SHD). Our goal is to obtain an estimate of the performance of TESF for global 3D object retrieval, serving as a proof-of-concept to motivate investigating it for classification based on a limited amount of viewpoints. The SHD signatures are computed directly on the 3D models using the code available at the author's website [Kaz13].

6.1. Datasets

Two datasets were used in the experiments: the 3D-Net database [WV11] and the Princeton Shape Benchmark (PCB) [SMKF04].

The 3D-Net database consists of more than 200 classes of 3D models stored in a hierarchy according to WordNet [Fel98]. In our experiments, we used a subset of 55 classes containing a total of 1267 objects. The object classes vary from common household objects (e.g. mug, chair, shoe), to vehicles (e.g. car, convertible, airplane) and animals (e.g. elephant, shark, horse). A list of all the used object categories is shown in the first column of Table 2.

The Princeton Shape Benchmark dataset consists of 3D models of various diverse categories, such as buildings, household objects, vehicles, animals, body parts, furniture, plants etc. The models are also organized in a hierarchy, with the intention of simplifying the use of finer and coarser classifications. The original dataset is divided into a train and a test database, which we merged for our experiments. We use in total 1807 models, divided into 161 object classes (the finest possible classification granularity).

6.2. Evaluating retrieval performance

We evaluated four descriptors that represent 3D objects: ESF-Full, Spherical Harmonic Descriptor (SHD) [KFR03], and TESF1 and TESF2 descriptors. To build the TESF descriptors, we rendered 20 views of the objects, represented

Table 1: Retrieval performance on the 3D-Net database.

	1-NN	10-NN
TESF1	77.03%	92.89%
TESF2	82.64%	94.71%
ESF-Full	68.40%	88.10%
SHD [KFR03]	84.71%	94.56%

each view with an ESF descriptor, and used the obtained 20 ESF descriptors for TESF calculation. The 20 viewpoints locations are at the center of the triangular faces of a regular icosahedron which encloses the model under consideration. When building TESF1, we used equal weights for all the views. For TESF2, 12 bins were used, and bin boundaries were computed according to the procedure proposed in Section 5. The computed mean value used in the binning strategy was exactly $0.0015625 = 1/640$, justifying our assumption of a uniform histogram prior.

In order to evaluate the performance of the considered descriptors, we measured the 1-NN and the 10-NN retrieval performance for each sample in the dataset, using Euclidean distance as a distance function. In the 1-NN retrieval, for each sample we find its nearest neighbor, and count the retrieval as succesful if the neighbor is of the same class as the sample. In the 10-NN retrieval, we retrieve the 10 nearest neighbors, and count the retrieval as successful if at least one object of the same class is retrieved among the neighbors. The experiment was repeated for both 3D-Net and PSB datasets.

Table 1 summarizes the retrieval performance on the 3D-Net database. In 1-NN retrieval, the best performing descriptor is SHD, obtaining a correct retrieval rate of 84.71%. However, TESF2 achieves a comparable retrieval rate of 82.64%. In 10-NN retrieval, TESF2 is better than SHD, with a retrieval rate of 94.71%. A few examples of objects from 3D-Net that are misclassified with TESF2 using 10-NN retrieval are shown in Figure 3. As there is no correct object class in the 10 nearest neighbors, the closest of the neighbors is shown. It can be seen that the objects are visually quite similar, although for some objects the similarity it is not immediately apparent when analyzing just the labels (e.g. an elephant misclassified as a chair). A detailed per-class analysis of 1-NN and 10-NN retrieval performance of TESF2 on 3D-Net is shown in Table 2.

It is interesting to consider how the retrieval performance changes when different numbers of neighbors are considered. As shown in Figure 4, relative performance of different descriptors remains similar as for the 1-NN and 10-NN case. TESF2 descriptors perform almost equivalent to SHD, while TESF1 and ESF-Full descriptors are slightly worse, with TESF1 outperforming ESF-Full.

The PSB dataset is more demanding than 3D-Net, having more objects (1807 compared to 1267) and significantly

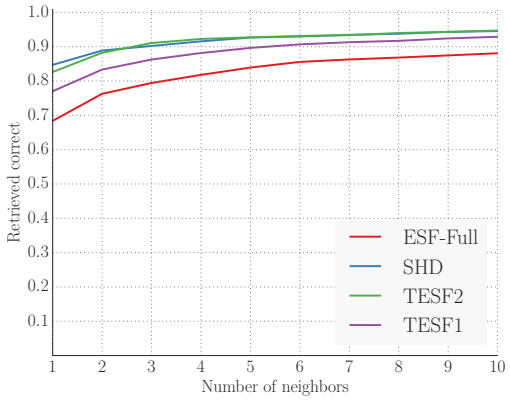

Figure 4: Retrieval performance on the 3D-Net database depending on the number of neighbors.

Table 3: Retrieval performance on the Princeton Shape Benchmark dataset.

	1-NN	10-NN
TESF1	46.20%	72.49%
TESF2	54.73%	77.86%
ESF-Full	39.45%	65.57%
SHD [KFR03]	55.06%	77.86%

more object classes (161 compared to 55). The performance of all the descriptors is worse than on the 3D-Net dataset in both 1-NN and 10-NN retrieval, as shown in Table 3. The best performance of 77.86% is obtained when using 10-NN retrieval, both with TESF2 and SHD descriptors. Curiously, the performance rate is exactly the same (1407 out of 1807 object correctly classified), although per-class performance varies. TESF1 is the third best-performing descriptor, and ESF-Full the fourth.

In general, we see that TESF2 and SHD perform similarly, as illustrated on both datasets. The differences in performance between TESF1 and TESF2 validate the proposed scheme to accumulate *temporal* information, indicating that the added complexity of TESF2 provides improved performance. The lower performance of ESF-Full motivates the use of single-views (the kind of representation for which ESF was originally designed) and temporal accumulation.

6.3. Ranking ESF sub-histogram importance

The ESF descriptors are formed by concatenating ten 64-bin histograms, which we refer to as ESF sub-histograms. When building the TESF representation, all ten 64-bin sub-histograms of each underlying ESF descriptor are used. It is interesting to consider how the overall retrieval performance changes if we instead select only *some* of the ESF sub-histograms.

Table 2: 1-NN and 10-NN retrieval performance of TESF2 on the 3D-Net database.

class	instances	correct (10-NN)	top confusions (10-NN)	correct (1-NN)	top confusions (1-NN)
total	1267	94.71%		82.64%	
airplane	78	98.72%	fighter jet	89.74%	fighter jet
apple	12	100.00%	-	91.67%	bottle
armchair	29	89.66%	mug	41.38%	chair
axe	26	84.62%	guitar	42.31%	hammer
banana	6	100.00%	-	100.00%	-
banjo	2	100.00%	-	50.00%	guitar
biplane	27	92.59%	stapler, airplane	70.37%	stapler
book	25	100.00%	-	100.00%	-
boot	8	100.00%	-	37.50%	shoe
bottle	79	92.41%	mug, car	72.15%	fire extinguisher, car
bowl	30	93.33%	mug, pistol	86.67%	cap
camera	21	90.48%	chair, book	71.43%	chair, book
can	25	100.00%	-	100.00%	-
cap	8	75.00%	mug	50.00%	mug
car	69	100.00%	-	95.65%	convertible
chair	49	85.71%	office chair	55.10%	armchair
clothes hanger	8	87.50%	airplane	62.50%	airplane
convertible	44	100.00%	-	88.64%	car
donut	10	100.00%	-	100.00%	-
elephant	14	92.86%	office chair	78.57%	office chair
espresso maker	11	90.91%	camera	54.55%	armchair
fighter jet	47	93.62%	airplane	74.47%	airplane
fire extinguisher	17	94.12%	heels	70.59%	bottle
flashlight	17	94.12%	espresso maker	88.24%	espresso maker, bottle
formula car	34	100.00%	-	97.06%	spaceship nx class
grenade	15	93.33%	mug	73.33%	apple, mug, camera, pitcher
guitar	40	100.00%	-	100.00%	-
hammer	35	100.00%	-	97.14%	axe
heels	20	100.00%	-	90.00%	boot, padlock
helicopter	39	97.44%	office chair	94.87%	fire extinguisher, office chair
horse	13	100.00%	-	92.31%	fighter jet
keyboard	24	100.00%	-	100.00%	-
ladle	2	0.00%	saucepan, cap	0.00%	saucepan, cap
light bulb	12	91.67%	office chair	75.00%	office chair, pear, bottle
monster truck	18	94.44%	airplane	88.89%	airplane, office chair
mug	82	100.00%	-	100.00%	-
office chair	43	100.00%	-	93.02%	armchair, chair, elephant
padlock	21	90.48%	heels, fire extinguisher	85.71%	heels
paper punch	5	40.00%	stapler, saucepan, car	20.00%	stapler, armchair, saucepan, car
pear	6	83.33%	grenade	66.67%	grenade
pineapple	2	50.00%	office chair	0.00%	office chair, convertible
pistol	40	97.50%	guitar	87.50%	guitar
pitcher	7	57.14%	rubber duck, camera, can	42.86%	rubber duck
pliers	18	77.78%	stapler	66.67%	stapler
rubber duck	4	100.00%	-	50.00%	pitcher, grenade
rubik cube	6	100.00%	-	100.00%	-
saucepan	2	100.00%	-	50.00%	paper punch
screwdriver	24	100.00%	-	100.00%	-
shark	13	100.00%	-	84.62%	fire extinguisher, pistol
shoe	19	100.00%	-	89.47%	boot, car
spaceship nx class	14	78.57%	fighter jet, elephant, office chair	71.43%	fighter jet
stapler	27	88.89%	pistol, banana, car	77.78%	shoe, pistol, banana, convertible, car, biplane
starfruit	2	0.00%	camera, grenade	0.00%	camera, grenade
tetra pak	15	100.00%	-	93.33%	shoe
toilet paper	3	66.67%	mug	33.33%	mug

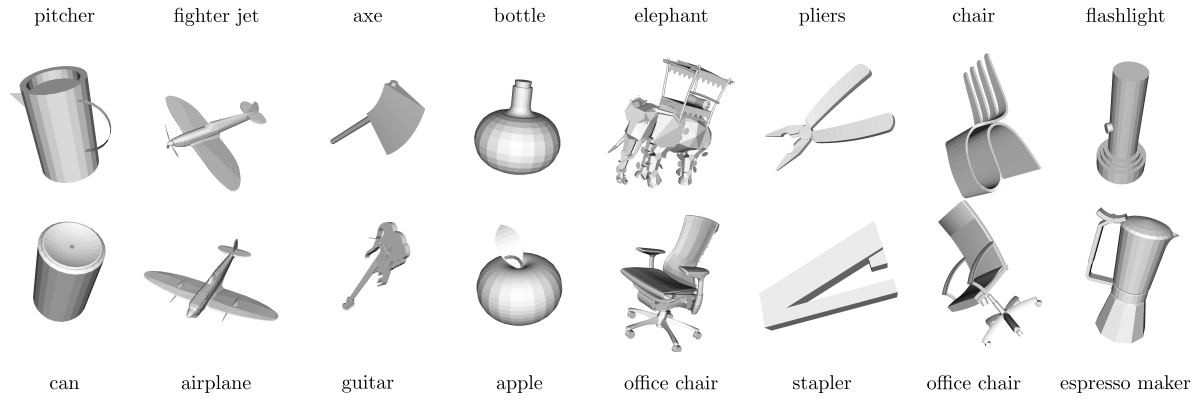

Figure 3: Examples of misclassifications using TESF2 and 10-NN retrieval. The correct class was not found among the 10 nearest neighbors. The top row shows query objects, and the bottom row the closest of the 10 nearest neighbors.

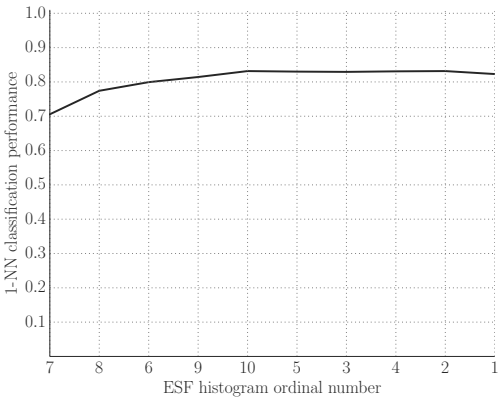

Figure 5: Changes in 1-NN retrieval performance on the 3D-Net dataset when ESF histograms are incrementally added.

In order to investigate the influence of individual ESF sub-histograms on total performance, we perform an experiment in which we incrementally select ESF sub-histograms to be used in building TESF2 descriptors. Initially, we consider each of the 10 ESF sub-histograms as individual feature vectors, and perform 10 experiments. In each experiment, we build TESF2 descriptors using one of the 10 ESF sub-histograms as a basis, and evaluate the 1-NN retrieval performance. We select the ESF sub-histogram with the best performance as a first sub-histogram in our partial ESF descriptor. In the second step, we consider the concatenations of the selected ESF sub-histogram with each of the remaining 9 histograms, measure the retrieval performance, and select the best among the 9 considered histograms. So, in the second step we are considering partial ESF descriptors of length $2 \times 64 = 128$. In the third step, the procedure is repeated for the remaining 8 histograms to find the third histogram to use in the concatenation, and we are considering partial ESF descriptors of length $3 \times 64 = 192$. The procedure is repeated until 10 histograms in total are selected, resulting in a full-size ESF descriptor length of 640.

Figure 5 illustrates the change of performance rate as more ESF sub-histograms are added to the descriptor. It can be seen that the most discriminative ESF sub-histogram is sub-histogram 7, that corresponds to the D2:Distance(in) shape function. Using this sub-histogram alone as a basis for building TESF2 descriptors, we are able to obtain 1-NN performance of 70.57% on the 3D-Net database. Performance steadily increases as sub-histograms 8, 6, 9 and 10 are added. These sub-histograms correspond to the D2:Distance, D3:Area(mixed) and ratio of line distances shape functions. Peak performance of 83.16% is reached for the first five selected sub-histograms. Adding the remaining sub-histograms does not improve the performance further. By adding sub-histograms 5, 3, 4 and 1 the performance slightly decreases, and at the point when sub-histogram 2 is added it is restored to the peak. Sub-histograms 1, 2, 3, 4 and 5 correspond to shape functions A3:Angle(in, out, mixed) and D3:Area(in, out). We conclude that the angle and the area-based shape functions seem to be not as discriminative as distance-based shape functions. Wohlkinger and Vincze [WARV12] report that weighting of the individual sub-histograms could be used to improve performance of ESF descriptors, but specific weights for individual sub-histograms are not given.

7. Conclusion and future work

The Temporal Ensemble of Shape Functions (TESF) descriptor has been presented. It is based on a combination of a single view 3D shape descriptor and a temporal descriptor (initially used to accumulate appearance descriptors of an object over time) that accumulates the different shape properties of the object (in form of single view 3D descriptors)

over time. The proposed method has been experimentally validated on two large datasets with good results.

Because of the view-based nature of TESF as well as other interesting properties such as its invariance to the order in which the individual views are presented, exploring its performance for object classification from a limited number of viewpoints seems a promising future direction of work. To this end, several challenges need to be carefully investigated. In particular, since the specific viewpoints of an object during online recognition are not known at training time, a new training scheme needs to be deployed to consider the availability of limited viewpoints.

References

[BPSK11] BRKIC K., PINZ A., SEGVIC S., KALAFATIC Z.: Histogram-based description of local space-time appearance. In *Proceedings of the 17th Scandinavian Conference on Image Analysis* (2011), SCIA'11, pp. 206–217. 2, 3

[CH67] COVER T., HART P.: Nearest neighbor pattern classification. *Information Theory, IEEE Transactions on 13*, 1 (January 1967), 21–27. doi:10.1109/TIT.1967.1053964. 2

[CTSO03] CHEN D.-Y., TIAN X.-P., SHEN Y.-T., OUHYOUNG M.: On visual similarity based 3d model retrieval. *Computer Graphics Forum 22*, 3 (2003), 223–232. 1

[DA10] DARAS P., AXENOPOULOS A.: A 3d shape retrieval framework supporting multimodal queries. *Int. J. Comput. Vision 89*, 2-3 (Sept. 2010), 229–247. URL: http://dx.doi.org/10.1007/s11263-009-0277-2, doi:10.1007/s11263-009-0277-2. 1, 2

[Fel98] FELLBAUM C. (Ed.): *WordNet: an electronic lexical database*. MIT Press, 1998. 4

[Kaz13] KAZHDAN M. M.: Screened Poisson Surface Reconstruction. http://www.cs.jhu.edu/~misha/Code/Matching/, 2013. [Online; accessed 09-December-2013]. 4

[KFR03] KAZHDAN M., FUNKHOUSER T., RUSINKIEWICZ S.: Rotation invariant spherical harmonic representation of 3D shape descriptors. In *Symposium on Geometry Processing* (2003). 4, 5

[Liu12] LIU Q.: A survey of recent view-based 3d model retrieval methods. *CoRR abs/1208.3670* (2012). 1

[OF09] OHBUCHI R., FURUYA T.: 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* (Sept 2009), pp. 63–70. doi:10.1109/ICCVW.2009.5457716. 1

[SMKF04] SHILANE P., MIN P., KAZHDAN M., FUNKHOUSER T.: The Princeton shape benchmark. In *Shape Modeling International* (June 2004). 4

[TPBBB13] TENORTH M., PROFANTER S., BALINT-BENCZEDI F., BEETZ M.: Decomposing cad models of objects of daily use and reasoning about their functional parts. 1

[TV08] TANGELDER J. W., VELTKAMP R. C.: A survey of content based 3d shape retrieval methods. *Multimedia Tools Appl. 39*, 3 (Sept. 2008), 441–471. 1

[WARV12] WOHLKINGER W., ALDOMA A., RUSU R. B., VINCZE M.: 3DNet: Large-scale object class recognition from CAD models. In *Robotics and Automation (ICRA), 2012 IEEE International Conference on* (2012), IEEE, pp. 5384–5391. 1, 2, 7

[WV11] WOHLKINGER W., VINCZE M.: Ensemble of Shape Functions for 3D Object Classification. In *IEEE International Conference on Robotics and Biomimetics (IEEE-ROBIO)* (2011). 2, 3, 4