Eurographics Workshop on 3D Object Retrieval (2016)
A. Ferreira, A. Giachetti, and D. Giorgi (Editors)

SHREC'16 Track: Shape Retrieval of Low-Cost RGB-D Captures

Pedro B. Pascoal[†1,2,3], Pedro Proença[†,1,4], Filipe Gaspar[†,1,2], Miguel Sales Dias[†,1,2], Alfredo Ferreira[†,3],

Atsushi Tatsuma[5], Masaki Aono[5], K. Berker Logoglu[6], Sinan Kalkan[7], Alptekin Temizel[6],
Bo Li[8], Henry Johan[9], Yijuan Lu[a], Viktor Seib[b], Norman Link[b] and Dietrich Paulus[b]

[1]ISCTE - Instituto Universitário de Lisboa/ISTAR-IUL, Lisbon, Portugal
[2]Microsoft Language and Development Center, Lisbon Portugal
[3]INESC-ID/ Técnico Lisboa /Universidade de Lisboa, Portugal
[4]Surrey Space Centre, University of Surrey, UK
[5]Department of Computer Science and Engineering, Toyohashi University of Technology, Japan
[6]Informatics Institute, Middle East Technical University, Ankara, Turkey
[7]Department of Computer Engineering, Middle East Technical University, Ankara, Turkey
[8]Department of Mathematics and Computer Science, University of Central Missouri, Warrensburg, USA
[9]Fraunhofer IDM@NTU, Singapore
[a]Department of Computer Science, Texas State University, San Marcos, USA
[b]Active Vision Group (AGAS), University of Koblenz-Landau, Universitätsstr. 1, 56070 Koblenz, Germany

Abstract

RGB-D cameras allow to capture digital representations of objects in an easy and inexpensive way. Such technology enables ordinary users to capture everyday object into digital 3D representations. In this context, we present a track for the Shape Retrieval Contest, which focus on objects digitized using the latest version of Microsoft Kinect, namely, Kinect One. The proposed, track encompasses a dataset of two hundred objects and respective classification.

Categories and Subject Descriptors (according to ACM CCS): H.3.3 [Information Storage and Retrieval]: Information Search and Retrieval—Relevance feedback I.3.5 [Computer Graphics]: Computational Geometry and Object Modeling—Geometric algorithms, languages, and systems

1. Introduction

Due to the growing popularity of low-cost scanners, several RGB-D object datasets have been emerging in the research community [MFP*13, PPG*15]. So far, objects captured by these datasets have proven that such low quality 3D representations, are of little use. Nevertheless, such captures are much faster than other technologies, which enable the usage in scenarios, such as, real-time recognition. For this, it is essential to first identify which 3D shape descriptors provide better performance, when used to retrieve such digitalized objects.

In this context, we present a dataset that provide the research community with a benchmark for the training and evaluation of techniques for digitalized objects. This work is an extension of a previous track done by Pascoal et al. [PPG*15]. In the scope of this track we will use the same automated process for point-cloud capture and registration.

2. Pipeline overview

Our approach presents an easy to build solution, which can be spread not only to the scientific community, but also, to the common users. The whole Capture pipeline can broadly be divided into Capture and Toolkit (Figure 1).

The Capture (*Online process*), encloses the saving of color and depth frames. We capture partial point-clouds from multiple viewpoints and repeat this process for three sessions for each object, in order to cover different elevation angles (30°, 45°and 60°) as depicted in Figure. In each session we capture 90 pairs of RGB and Depth, which in total make 270 RGB-D pairs.

The Toolkit (*Offline process*), uses the captured raw data, and provides post-processing actions. Using the segmented images, we then perform two independent processes: image segmentation for

† Organizer of the SHREC track.

DOI: 10.2312/3dor.20161090

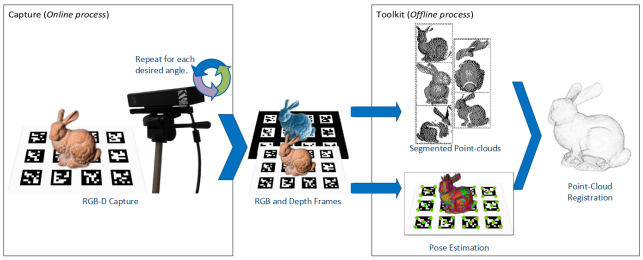


Figure 1: Capture pipeline.

object extraction, and pose calculation using the turntable markers. Finally, using all segmented local point clouds from the object and pose information, we generate the registered point-cloud. Additionally, we apply a filter to smooth the surface of the global point cloud, in order to remove untreated noise.

3. Database

This work is an extension of the work presented by Pascoal et al. [PPG*15]. For this track, we've selected a subset of two hundred objects, from the **RGB-D ISCTE-IUL Dataset** [PPGD15]. Each capture was selected manually in order to offer better captures than those of previous tracks [MFP*13, PPG*15]. Furthermore, instead of the generic office objects, we provide a wider range of classes, in an attempt to provide digitalized matches for models presented in other datasets, such as the Princeton Shape Benchmark [SMKF04] and the Sketch-Based 3D presented by Li et al. [LLL*14].

The dataset is organized according to the type of object. Each object belongs to a specific class, whereas the "class" annotation is a very low-level description, such as the "name of the object". For example, a toy car, belongs to the class "Car". Furthermore, each object of the class, must have but a very small variation from the others. This variation cannot be too great, for instance, a formula-one car needs to have its own unique class, since, although it can be considered a car, its shape is very different from the standard car used by consumers. The complete list of classes and their number of objects is presented in Table 1. The dataset provides for each object, 90 frame pairs of RGB and Depth images, the segmented and registered point clouds and the polygon mesh. All data, from raw data to triangular meshes, was made available to all participants, so that each could use the most appropriate for his algorithm.

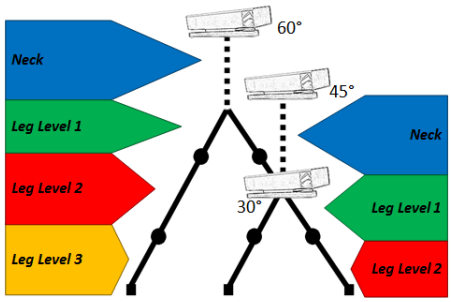


Figure 2: Capture pipeline.

Airplane	3
Animal	10
Bird	1
Book	4
Bowl	10
Car	4
Car Convertible	4
Car Formula1	3
Car Sport	8
Castle	2
Cell phone	5
Coffee cup	10
Dinosaur	6
Game Controller	8
Game Handheld	6
Glasses VR	3
Guitar	10
Headphones	4
Headset	5
Keyboard	4
Motorbike	7
Mug	10
Puncher	5
Remote	5
Shoes lady	10
Soda can	10
Sofa	4
Toy Human	11
Wooden mannequin	10
Wooden puzzle	10
Wooden spoon	8

Table 1: Dataset classes and their number of objects.

Additionally, for each capture we collected a matching "high-quality" 3D model, acquired from the 3D dataset, SketchUp 3D Warehouse [cTNL16]. This was used in the evaluation process, using the captured model as query, to retrieve "high-quality" models of the same class.

4. Evaluation

In the proposed track we adopted the most commonly used methods, precision and recall, to measure and evaluate the submitted algorithms. The relevance assessments where done using only the categorization.

Using each captured object as the query, participants should return a ranked list of the remaining test data according to the similarity score.

For the query there where two distinct retrieval ranked lists requested. One using a capture for the retrieval of captured objects, and another using the same captured object as query but to retrieve "high-quality" similar models from the internet. Each rank list had the length of the whole dataset.

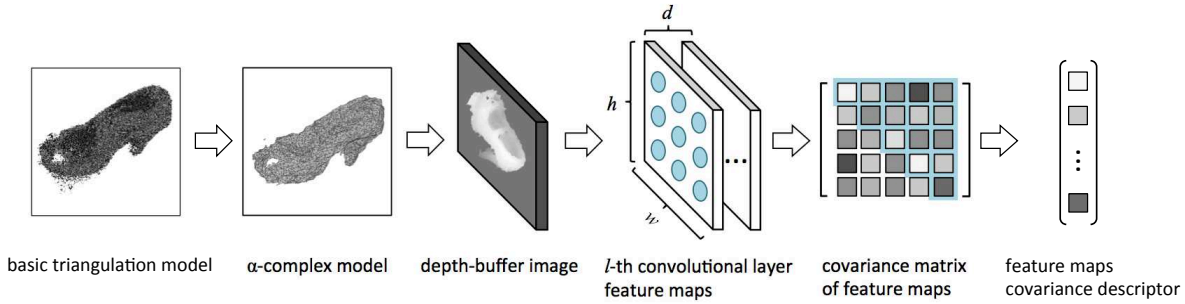


Figure 3: The overview of A. Tatsuma and M. Aono procedures.

5. Submissions

For this contest, four different groups participated with their respective methods.

- Atsushi Tatsuma and Masaki Aono, participated with a method for 3D shape retrieval using pre-trained Convolutional Neural Networks (CNN) [LBD*89];
- K. Berker Logoglu, Sinan Kalkan, Alptekin Temizel, used a local 3D descriptor that leverages the synchronized RGB and depth data provided by RGB-D sensors;
- Bo Li, Henry Johan, Yijuan Lu, used a hybrid shape descriptor **ZFDR** proposed in [LJ13], which is composed of visual and geometrical features of a 3D model;
- Finally, Viktor Seib, Norman Link and Dietrich Paulus present a discrete Hough-space for continuous voting space in order not to lose the feature's descriptiveness.

5.1. 3D Shape Retrieval using Feature Maps Covariance Descriptor, by Atsushi Tatsuma and Masaki Aono

A. Tatsuma and M. Aono proposed a method for 3D shape retrieval using pre-trained Convolutional Neural Networks (CNN) [LBD*89]. The overview of their approach is illustrated in Figure 3. Their method extract the Feature Maps Covariance Descriptor (FMCD) [TA16] from each depth-buffer image of a 3D model.

For this track, they selected the basic triangulation dataset. As a preprocessing of 3D model, by using MeshLab [CCR08], they reduced the number of vertices to about 10,000 points, and reconstructed the 3D model with α-complex algorithm. In addition, they normalized the scale, position and rotation of the 3D model with Point SVD [TA09].

After the preprocessing, they rendered depth-buffer images with 224×224 resolution from each vertex of the unit geodesic sphere. As a results, 38 depth-buffer images were obtained.

To obtain the feature vector of the 3D model, they extracted the FMCD from each depth-buffer image. FMCD comprises covariances of convolutional layer feature maps on the CNN.

Let $F = [\mathbf{f}_1, \ldots, \mathbf{f}_n] \in \mathbb{R}^{d \times n}$ denote the d feature maps of size $n = w \times h$ outputted from the l-th convolutional layer. To obtain a representation of a depth-buffer image, they calculated the covariance matrix of the feature maps

$$C = \frac{1}{n-1} \sum_{i=1}^{n} (\mathbf{f}_i - \mathbf{m})(\mathbf{f}_i - \mathbf{m})^\top,$$

where $\mathbf{m}$ is the mean of the feature maps. The covariance matrix C is a symmetric matrix.

The covariance matrix lies on the Riemannian manifold of symmetric positive semi-define matrices. To project the covariance matrix onto a point in the Euclidean space, they used the mapping method proposed by Pennec et al. [PFA06]. The mapping method first projects the covariance matrix onto the Euclidean space that is tangent to the Riemannian manifold at the tangency point P. The projected vector $\mathbf{y}$ of the covariance matrix C is given by

$$\mathbf{y} = \log_P(C) = P^{\frac{1}{2}} \log(P^{-\frac{1}{2}} C P^{-\frac{1}{2}}) P^{\frac{1}{2}},$$

where $\log(\cdot)$ is the matrix logarithm operators. The mapping method extracts the orthonormal coordinates of the projected vector that are given by the following vector operator

$$\mathrm{vec}_P(\mathbf{y}) = \mathrm{vec}_I(P^{-\frac{1}{2}} \mathbf{y} P^{-\frac{1}{2}}),$$

where I is the identity matrix, and the vector operator at identity is defined as

$$\mathrm{vec}_I(\mathbf{y}) = \left[y_{1,1} \sqrt{2} y_{1,2} \sqrt{2} y_{1,3} \ldots y_{2,2} \sqrt{2} y_{2,3} \ldots y_{d,d} \right]^\top.$$

From the observation in some studies [TSCM13, SGMC14], they choose the identity matrix for P. Consequently, the vectorized covariance matrix is given by

$$\mathbf{c} = \mathrm{vec}_I(\log(C)).$$

Finally, they obtained the depth-buffer image representation to normalize the vector $\mathbf{c}$ with the signed square rooting normalization [JC12] and ℓ_2 normalization.

For the pre-trained CNN, they used the VGG-M networks [CSVZ14]. The final feature vector is obtain by concatenating the fully connected layer activations and FMCD extracted from the first convolutional layer. The Euclidean distance is used for the dissimilarity between two feature vectors. To compare two 3D models, they apply the Hungarian method [Kuh55] to all pair dissimilarities between their feature vectors.

Run Number	Support Radius (cm)	# of CoSPAIR Levels	Keypoint Extraction Method
1	18	10	Sub-sampling at 1cm
2	18	7	Sub-sampling at 1cm
3	15	7	Sub-sampling at 1cm
4	20	10	ISS
5	18	7	ISS

Table 2: Parameters for different runs.

For the high-quality model dataset, they extract 3D shape feature vector with the same procedures excluding the mesh simplification and reconstruction processing.

5.2. Colored Histograms of Spatial Concentric Surflet-Pairs, by K. Berker Logoglu, Sinan Kalkan, Alptekin Temizel [LKT16]

Logoglu et al. [LKT16] recently introduced a local 3D descriptor, Colored Histograms of Spatial Concentric Surflet-Pairs (CoSPAIR) descriptor, that leverages the synchronized RGB and depth data provided by RGB-D sensors. They showed that the CoSPAIR is among the best performing methods for RGB-D object recognition. Thus, chosen CoSPAIR as the basis of their method. The extraction of the CoSPAIR descriptor is shown in Figure 4.

For the tests, all the provided (segmented) scans were used. For each scan of a query object, the descriptors were extracted from the detected keypoints using either sub-sampling or Intrinsic Shape Signatures (ISS) and matched to the test object's scans one by one. The Euclidean distances between the best matching descriptors are averaged. Thus, eventually, for each test object, 200 distances are obtained. The distances are then converted to similarity scores. The test procedure is depicted in Figure 5. Finally, the same algorithm was run 5 times with different support radii, number of levels and keypoint extraction methods (Table 2) to produce 5 different ranked lists.

5.3. Hybrid Shape Descriptor ZFDR, by Bo Li, Henry Johan, Yijuan Lu [LJ13]

Considering the fact that there are many inaccuracies in the low-cost captures, such as normals, curvatures, connectivity, and topology, B. Li et al. employed a more robust hybrid-based approach rather than a purely geometry-based algorithm, whose performance is more likely to be affected by the inaccuracies. Their hybrid approach extracts both visual features (Zernike moments and Fourier descriptors) and geometrical features (Depth and Radius length) to characterize a 3D object.

Their algorithms and the corresponding five runs for each task are mainly based on the hybrid shape descriptor **ZFDR** proposed in [LJ13], which is composed of the following four visual or geometrical features of a 3D model. (1) Thirteen sample silhouette views' **Z**ernike moments and **F**ourier descriptor features; (2) Six depth buffer views' **D**epth information; and (3) A model's **R**ay-based features which are generated by measuring the lengths of a set of rays shot from the center of the model to the utmost intersections on the surface of the model. Based on the four component features in the **ZFDR** shape descriptor, they also test **ZFDR**'s three variations: **ZF**, **ZFD** and **ZFR** to observe the impacts when they completely or partially drop the geometrical component features. **DESIRE** [Vra04] (also mentioned as **DSR**, that is **D+S+R**) is a well-known hybrid shape descriptor, where **S** denotes the one-dimensional Fourier transform features of three canonical **S**ilhouette views of a 3D model. Their two component features **D** and **R** are based on **DESIRE**. To find out whether the performance will be improved further, we combine our hybrid shape descriptor **ZFDR** and **DESIRE** together to form a new hybrid shape descriptor, that is **ZFDSR**. The pipeline to generate the above five shape descriptors is shown in Figure 6. For more details about the feature extraction and retrieval process, please refer to [LJ13].

5.4. Shape Retrieval with Hough-Voting in a Continuous Voting Space, by Viktor Seib, Norman Link and Dietrich Paulus [LJ13]

V. Seib et al. method is related to the Implicit Shape Model formulation by Leibe et al. [LLS04]. Adaptations of this method to 3D data were proposed [KPW*10, STDS10, WZS13]. In contrast to the original formulation, the adaptations to 3D data all use a discrete Hough-space for voting. V. Seib et al. use a continuous voting space and omit the vector quantization of features in order not to lose the feature's descriptiveness. To be able to generalize from learned shapes, they match each extracted feature with the k best matches in the learned dictionary. Their algorithm works on point cloud data. Thus, when using the mesh model, it is required to first convert it back to point clouds by densely sampling the surface.

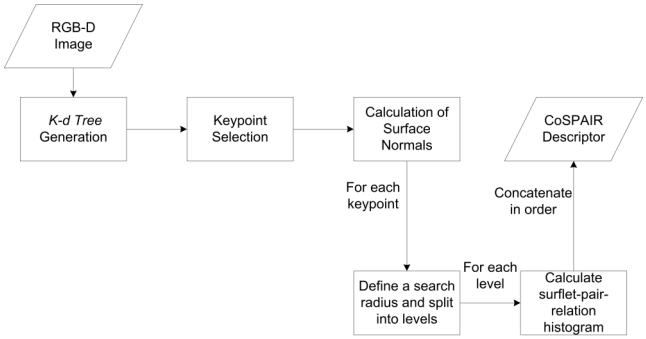


Figure 4: CoSPAIR extraction flow.

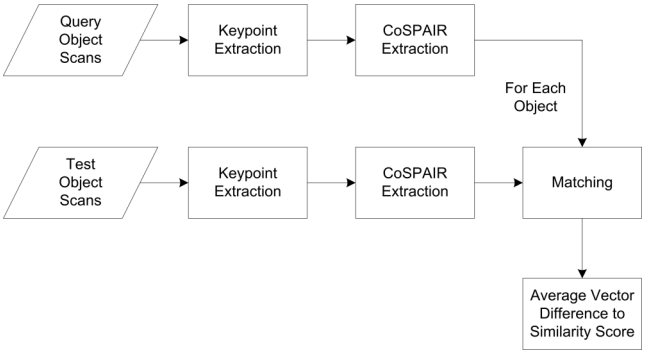


Figure 5: Shape retrieval flow.

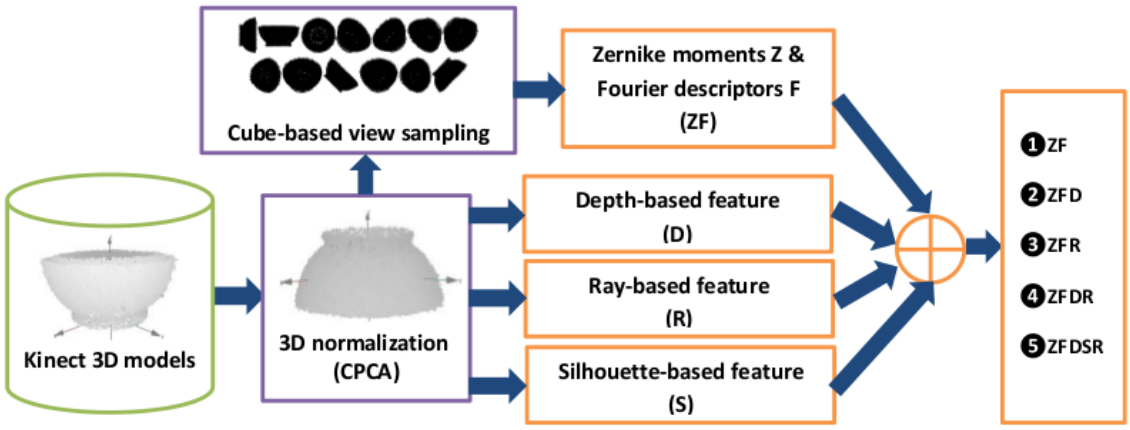


Figure 6: Flowchart of computing five hybrid shape descriptors: **ZFDR**, **ZF**, **ZFD**, **ZFR** and **ZFDSR**.

In training, key points are extracted from full 3D models using a uniform voxel grid and a SHOT descriptor [TSDS10] is computed for each key point. In the next step, spatial relations between detected features on the training model are computed. For each feature, a vector pointing from the feature to the object's centroid is obtained, in the following referred to as *center vector*. The final data pool after training contains all features that were computed on all training models. Along with each feature, a center vector and the class of the corresponding object is stored.

To classify objects, features are detected on the input data 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 $d(f_l, f_d) = \|f_l - f_d\|_2$. Since it cannot be expected to encounter the same objects during classification as were used in training, each detected feature is associated with the k best matching features from the learned data pool.

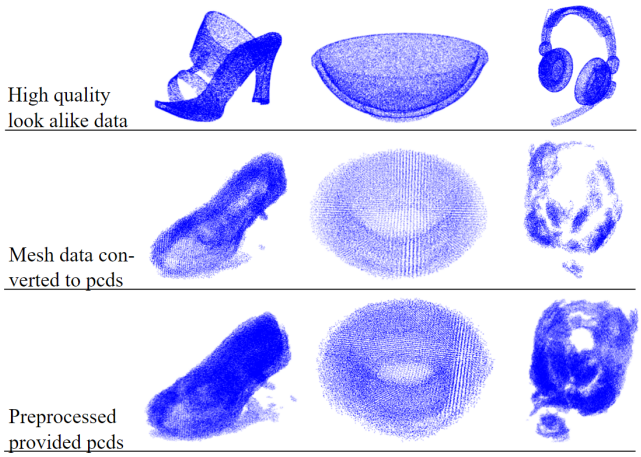


Figure 7: Data that was used for each run/evaluation.

The center vectors of the created correspondences are used to create hypotheses on object center locations in a continuous voting space. A separate voting space for each class is used.

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. In a final step the found maxima positions from all voting spaces of individual classes are merged. In case multiple maxima are found at the same position, i.e. if they are closer than half of the kernel bandwidth, only the maximum with the highest probability is retained.

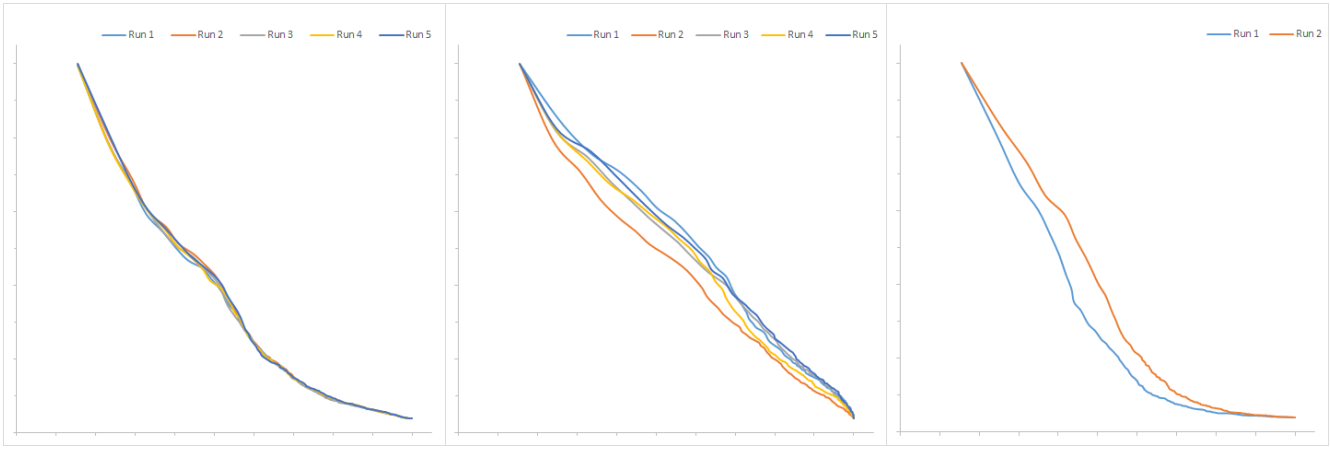
The presented algorithm returns a list of results ranked by the common weight of the contributing votes. For the shape similarities, they apply a simple transformation from weights to similarities for each object i: $s = \frac{\omega_i}{\omega_{max}}$ (where ω_{max} is the weight of the most likely object hypothesis).

Finally, to evaluate their approach, they performed two individual runs.

For the first run (run ID 1), the provided mesh data was converted into point clouds (pcds). The point clouds were converted to meters (the provided data was in millimeters) and downsampled with a uniform grid so that the resulting files contained 4 points per centimeter. Further, a statistical outlier removal was applied to each object. For the first run they used these objects from converted mesh data as query and for retrieval. For the second run (run ID 2) they used the provided point cloud data, which was also converted to meters and downsampled. Again, outliers were removed before using the data. For the second run they used these objects from preprocessed point clouds as query and for retrieval.

Finally, for the look alike data provided, these meshes were converted to point clouds and were scaled to the same size as the corresponding object from the first run. Further, these data was downsampled so that the resulting files contained 10 points per centimeter. For this evaluation, the objects from the converted meshes were used as query where the converted look alike objects were retrieved. A comparison of some of the objects used in these runs is given in Figure 7.

All runs were performed with SHOT features using the radii



(a) B. Logoglu et al. (b) B. Li et al. (c) V. Seib et al.

Figure 8: Precision-Recall curves of different runs of each participant.

0.1 m and a k of 5 for nearest neighbor matching. The bandwidth was set to 0.1 m for all runs.

6. Results

All SHREC participants submitted, at least one rank list for the evaluations, with the exception of Logoglu et al., since their algorithm uses synchronized RGB and depth data provided by RGB-D sensors to compare objects. As such, the scenario of retrieving "high-quality" similar models from the internet was impractical for technique. Each rank list has the whole collection of 200 objects ordered by their dissimilarities. In Figure 8, we present the Precision-Recall curves, of each run, for the participants that provided more than one. As we can clearly perceive, the curves of the runs of each method, all follow a similar path, but **B. Li et al. run 1** and **V. Seib et al. run 2** are clearly superior than their other runs.

Using the best run of each method, we compiled the Precision-Recall curves presented in Figure 9. Based on these results, **A. Tatsuma et al.** method provided the best precision of all. Similar to previous tracks [MFP*13, PPG*15], we can conclude that view-based methods generally work better for such objects, since such methods are proven to be more robust to topology errors, surface deformations and noise, which are frequent in such models.

However, by scoping Precision-Recall curves for each class (Figure 10), we're able to better extract each method's strengths. For instance, **B. Logoglu et al.** outperforms all for **Game Controller**, **Wooden puzzle**, and the non-rigid **Wooden mannequin**. All these captures share holes, and empty spaces. Although, some other captures also share such feature, they're too small. For bigger objects in general the results are clearly better, but not as good as for the previously named classes.

B. Li et al. technique similar to **A. Tatsuma et al.** method, performed better with classes that had very distinct shapes from the others, such as **Bowl**, **Guitar**, and **Soda can**. Both their performances are similar, where each outperforms the other in different classes.

V. Seib et al. method performed best the bigger the objects were, such as **Airplane** and **Keyboard**, and worst against smaller objects with littler details and limbs, such as **Animal**, **Dinosaur** and **Toy-Human**.

Finally, for the look alike evaluation (Figure 9 (b)), the results were considerably low when compared to the typical evaluation results. The major rationale to this fact are that low-cost captures are unable to provide a degree of accuracy of designer made models, which makes them far too different to be considered identical or similar.

7. Conclusions

In this work, we presented a comparison of 3D object retrieval techniques from four research groups. Each participant was presented with a collection of 200 objects, captured using a Microsoft Kinect One.

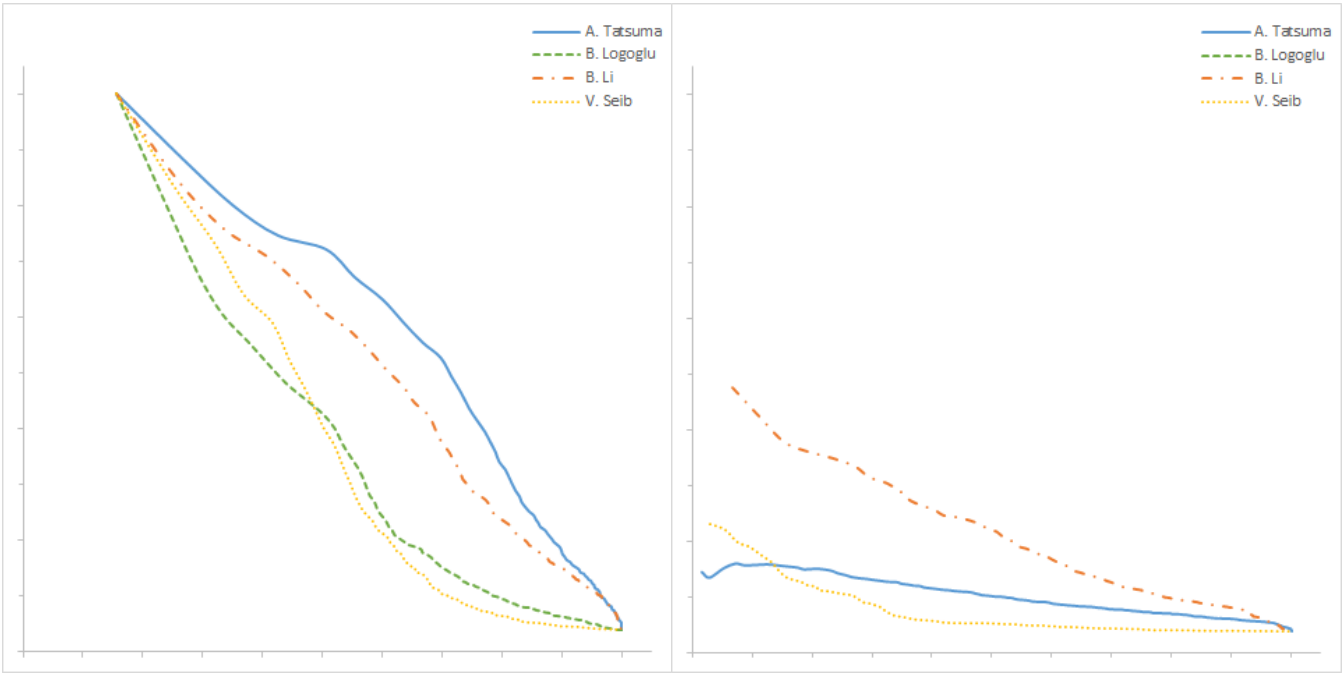
Each participant submitted two different evaluations, with at least one ranked list of results. One using the capture objects as query to retrieve other captured objects, and another using the same captured object as query but to retrieve "high-quality" similar models from the internet.

Analyzing the results we could surmise that some of the techniques used by view based methods are the ones that best performed. However, each algorithm has shown better results for specific classes of objects, and further study on this topic could highlight their specific strengths.

8. Acknowledgements

The work described in this paper was supported by the following projects:

- The RGB-D ISCTE-IUL Dataset has been carried out in the scope of the OLA project (AAL 2014-076), co-financed by the



(a) Low-cost (b) Look alike

Figure 9: Precision-Recall curves of all participants.

AAL Joint Programme (AAL JP) and the following National Authorities in Portugal, Hungary, Poland and Sweden.

- Atsushi Tatsuma and Masaki Aono were supported by Kayamori Foundation of Informational Science Advancement and JSPS KAKENHI Grant Numbers 26280038, 15K12027, and 15K15992.
- The research done by Henry Johan is supported by the National Research Foundation, Prime MinisterâĂŹs Office, Singapore under its International Research Centres in Singapore Funding Initiative.

References

[CCR08] CIGNONI P., CORSINI M., RANZUGLIA G.: MeshLab: an open-source 3D mesh processing system. *ERCIM News*, 73 (April 2008), 45–46. 3

[CSVZ14] CHATFIELD K., SIMONYAN K., VEDALDI A., ZISSERMAN A.: Return of the devil in the details: Delving deep into convolutional nets. In *Proc. of the 25th British Machine Vision Conference* (2014), BMVC'14, pp. 1–12. 3

[cTNL16] ©2016 TRIMBLE NAVIGATION LIMITED: SketchUp 3D Warehouse. http://3dwarehouse.sketchup.com/, 2016. [Online; accessed 31-March-2016]. 2

[JC12] JÉGOU H., CHUM O.: Negative evidences and co-occurences in image retrieval: The benefit of PCA and whitening. In *Proc. of the 12th European Conference on Computer Vision* (2012), vol. 2 of *ECCV'12*, pp. 774–787. 3

[KPW*10] KNOPP J., PRASAD M., WILLEMS G., TIMOFTE R., VAN GOOL L.: Hough transform and 3d surf for robust three dimensional classification. In *ECCV (6)* (2010), pp. 589–602. 4

[Kuh55] KUHN H. W.: The hungarian method for the assignment problem. *Naval Research Logistics Quarterly 2* (1955), 83–97. 3

[LBD*89] LECUN Y., BOSER B., DENKER J. S., HENDERSON D., HOWARD R. E., HUBBARD W., JACKEL L. D.: Backpropagation applied to handwritten zip code recognition. *Neural Computation 1*, 4 (Dec. 1989), 541–551. 3

[LJ13] LI B., JOHAN H.: 3D model retrieval using hybrid features and class information. *Multimedia Tools Appl. 62*, 3 (2013), 821–846. 3, 4

[LKT16] LOGOGLU K. B., KALKAN S., TEMIZEL A.: Cospair: Colored histograms of spatial concentric surflet-pairs for 3d object recognition. *Robotics and Autonomous Systems 75, Part B* (2016), 558 – 570. 4

[LLL*14] LI B., LU Y., LI C., GODIL A., SCHRECK T., AONO M., FU H., FURUYA T., JOHAN H., LIU J., OHBUCHI R., TATSUMA A., ZOU C.: SHRECâĂŹ14 track: Extended large scale sketch-based 3d shape retrieval. In *Proc. EG Workshop on 3D Object Retrieval, 2014 (to appear)* (2014). 2

[LLS04] 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* (2004), pp. 17–32. 4

[MFP*13] MACHADO J., FERREIRA A., PASCOAL P. B., ABDELRAHMAN M., AONO M., EL-MELEGY M., FARAG A., JOHAN H., LI B., LU Y., TATSUMA A.: Shrec'13 track: Retrieval of objects captured with low-cost depth-sensing cameras. In *Proceedings of the Sixth Eurographics Workshop on 3D Object Retrieval* (Aire-la-Ville, Switzerland, Switzerland, 2013), 3DOR '13, Eurographics Association, pp. 65–71. 1, 2, 6

[PFA06] PENNEC X., FILLARD P., AYACHE N.: A riemannian framework for tensor computing. *International Journal of Computer Vision 66*, 1 (2006), 41–66. 3

[PPG*15] PASCOAL P. B., PROENÃĞA P., GASPAR F., DIAS M. S., TEIXEIRA F., FERREIRA A., SEIB V., LINK N., PAULUS D., TATSUMA

A., AONO M.: Retrieval of Objects Captured with Kinect One Camera. In *Eurographics Workshop on 3D Object Retrieval* (2015), Pratikakis I., Spagnuolo M., Theoharis T., Gool L. V., Veltkamp R., (Eds.), The Eurographics Association. 1, 2, 6

[PPGD15] PASCOAL P. B., PROENÇA P., GASPAR F., DIAS M. S.: RGB-D ISCTE-IUL DATASET. http://http://dataset.mldc.pt/, 2015. [Online; accessed 31-March-2016]. 2

[SGMC14] SERRA G., GRANA C., MANFREDI M., CUCCHIARA R.: Covariance of covariance features for image classification. In *Proc. of the International Conference on Multimedia Retrieval* (2014), ICMR '14, pp. 411–414. 3

[SMKF04] SHILANE P., MIN P., KAZHDAN M., FUNKHOUSER T.: The princeton shape benchmark. In *Shape modeling applications, 2004. Proceedings* (2004), IEEE, pp. 167–178. 2

[STDS10] 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)* (2010), Lecture Notes in Computer Science, pp. 653–666. 4

[TA09] TATSUMA A., AONO M.: Multi-fourier spectra descriptor and augmentation with spectral clustering for 3D shape retrieval. *The Visual Computer 25*, 8 (2009), 785–804. 3

[TA16] TATSUMA A., AONO M.: Food image recognition using covariance of convolutional layer feature maps. *IEICE Transactions on Information and Systems E99-D*, 6 (2016). Online First. 3

[TSCM13] TOSATO D., SPERA M., CRISTANI M., MURINO V.: Characterizing humans on riemannian manifolds. *IEEE Transactions on Pattern Analysis and Machine Intelligence 35*, 8 (Aug 2013), 1972–1984. 3

[TSDS10] 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)* (Berlin, Heidelberg, 2010), ECCV'10, Springer-Verlag, pp. 356–369. 5

[Vra04] VRANIC D.: *3D Model Retrieval*. PhD thesis, University of Leipzig, 2004. 4

[WZS13] 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* (2013), IEEE, pp. 366–373. 4

Figure 10: Precision-Recall graph of each category (Low-cost).

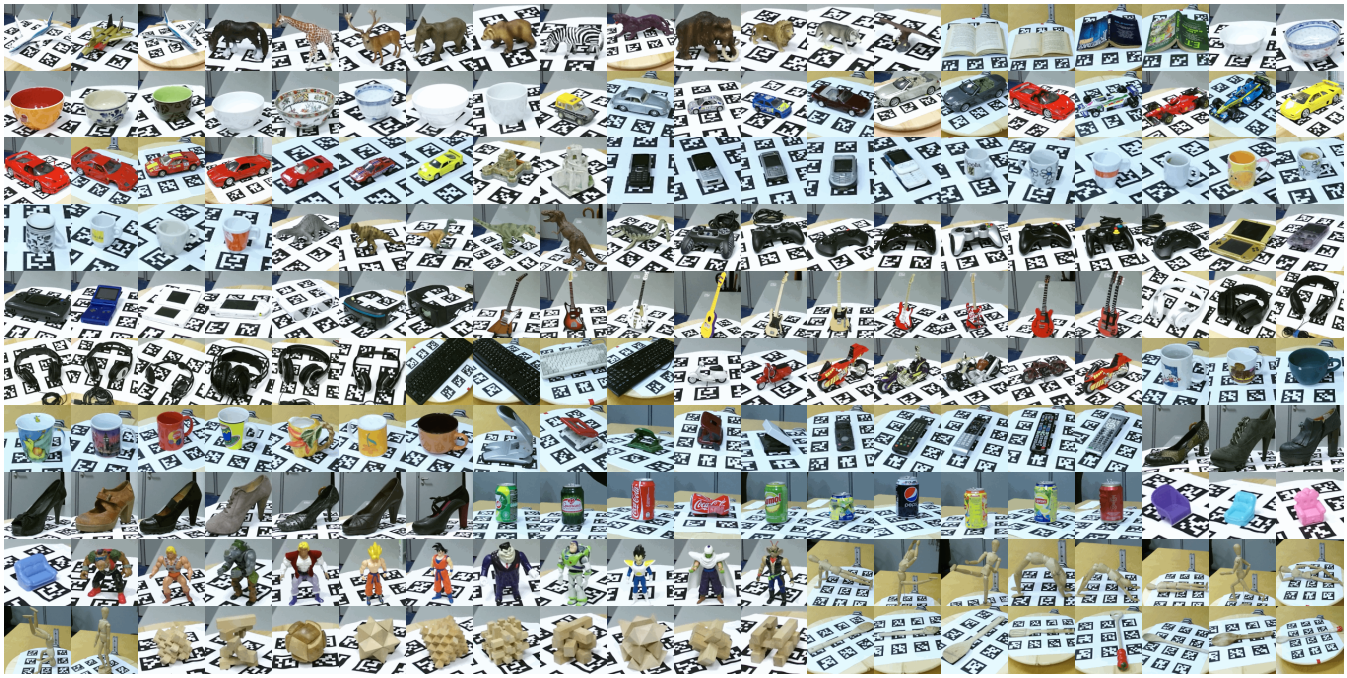


Figure 11: The 200 objects that were captured.

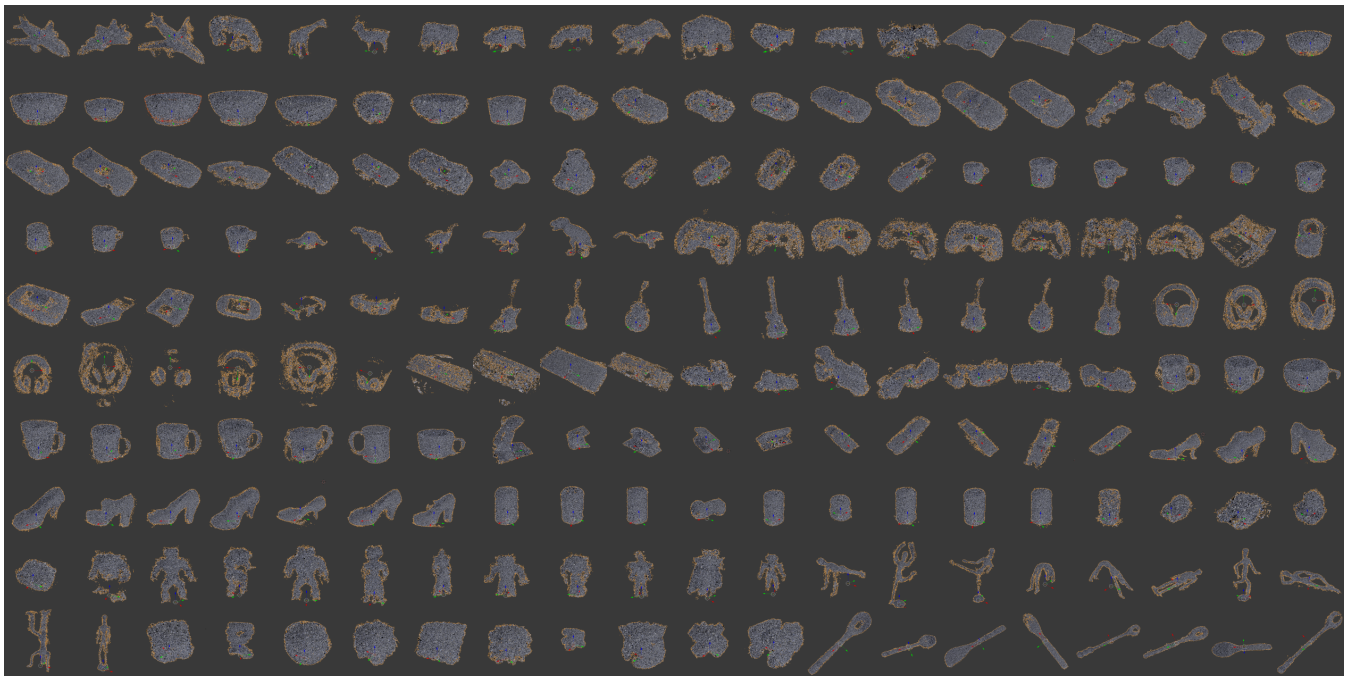


Figure 12: The 200 captures, registered using basic triangulation.