Eurographics Workshop on 3D Object Retrieval (2010)
I. Pratikakis, M. Spagnuolo, T. Theoharis, and R. Veltkamp (Editors)

SHREC 2010: robust large-scale shape retrieval benchmark

A. M. Bronstein[†1], M. M. Bronstein[†,1], U. Castellani[†,3] B. Falcidieno[4], A. Fusiello[3], A. Godil[6],

L. J. Guibas[†,2], I. Kokkinos[5], Z. Lian[6,7], M. Ovsjanikov[†,8], G. Patané[4], M. Spagnuolo[4], R. Toldo[3]

[1]Department of Computer Science, Technion – Israel Institute of Technology
[2]Department of Computer Science, Stanford University
[3]Department of Computer Science, University of Verona
[4]CNR-IMATI Genova
[5]Deparment of Applied Mathematics, École Centrale de Paris
[6]National Institute of Standards and Technology
[7]Beihang University
[8]Institute for Computational and Mathematical Engineering, Stanford University

Abstract

SHREC'10 robust large-scale shape retrieval benchmark simulates a retrieval scenario, in which the queries include multiple modifications and transformations of the same shape. The benchmark allows evaluating how algorithms cope with certain classes of transformations and what is the strength of the transformations that can be dealt with. The present paper is a report of the SHREC'10 robust large-scale shape retrieval benchmark results.

Categories and Subject Descriptors (according to ACM CCS): H.3.2 [Information storage and retrieval]: Information Search and Retrieval—Retrieval models I.2.10 [Artificial intelligence]: Vision and Scene Understanding—Shape

1. Introduction

Today, only a small fraction of Internet repositories of visual and geometric data is tagged and accessible through simple text search. Fast growth of these repositories makes content-based retrieval one of the next grand challenges in search and organization of such information. Particularly difficult is the problem of *invariant shape retrieval*, in which one tries to find a shape that has undergone some transformation, such as changes in scale, orientation, non-rigid deformations, missing parts, etc.

SHREC'10 robust large-scale shape retrieval benchmark simulates a retrieval scenario, in which the queries include multiple modifications and transformations of the same shape. The benchmark allows evaluating how algorithms cope with certain classes of transformations and what is the strength of the transformations that can be dealt with.

† Organizer of the SHREC track. All organizers and participants are listed in alphabetical order. For any information about the benchmark, contact mbron@cs.technion.ac.il

2. Data

The dataset used in this benchmark was aggregated from three public domain collections: TOSCA shapes [BBK08], Robert Sumner's collection of shapes [SP04], and Princeton shape repository [SMKF04]. Each of the datasets is available in the public domain. The shapes were represented as triangular meshes with the number of vertices ranging approximately between 300 and 30,000.

The dataset consisted of two parts: 456 shapes used as negatives, and 13 shapes used as positives, as shown in Figure 1. For each positive shape, 55 simulated transformations of different types and strengths were applied (total 715). The transformed positives were used as queries. The total dataset size (negatives+positives+transformed positives) was 1184.

A separate set with a total of 624 shapes was optionally provided for training. The training set included representative transformations of different classes and strengths. Besides null shapes, the training set contained no shape instances from the test set. The test and training sets are available at http://tosca.cs.technion.ac.il/book/shrec_robustness.html.

Figure 1: *Null shapes used in the benchmark. The first 13 shapes (top left) were used as positives.*

2.1. Queries

The query set consisted of 13 shapes taken from the dataset (null shapes), with simulated transformations applied to them. For each null shape, transformations were split into 11 classes shown in Figure 2: *isometry* (non-rigid almost inelastic deformations), *topology* (welding of shape vertices resulting in different triangulation), *micro holes* and big *holes*, *global* and *local scaling*, additive Gaussian *noise*, *shot noise*, *partial* occlusion (resulting in multiple disconnected components), down *sampling* (less than 20% of original points), and *mixed* transformations.

In each class, the transformation appeared in five different versions numbered 1–5. In all shape categories except scale and isometry, the version number corresponded to the transformation strength levels: the higher the number, the stronger the transformation (e.g., in noise transformation, the noise variance was proportional to the strength number). For scale transformations, the levels 1–5 corresponded to scaling by the factor of 0.5, 0.875, 1.25, 1.625, and 2. For the isometry class, the numbers did not reflect transformation strength.

The total number of transformations per shape was 55, and the total query set size was 715 shapes. Each query had one correct corresponding null shape in the dataset.

3. Evaluation methodology

Evaluation simulated matching of transformed shapes to a database containing untransformed (null) shapes. As the database, all 469 shapes with null transformations were used. Multiple query sets according to transformation class and strength were used. For transformation x and strength n, the query set contained all the shapes with transformation x and strength $\leq n$. In each transformation class, the query set size for strengths 1,2,..., 5 was 13, 26, 39, 52, and 65. In addition, query sets with all transformations broken down according to strength were used, containing 143, 286, 429, 572, and 715 shapes. Retrieval results on these query sets are referred to as *average* in the following.

Participants were asked to submit a distance matrix of size

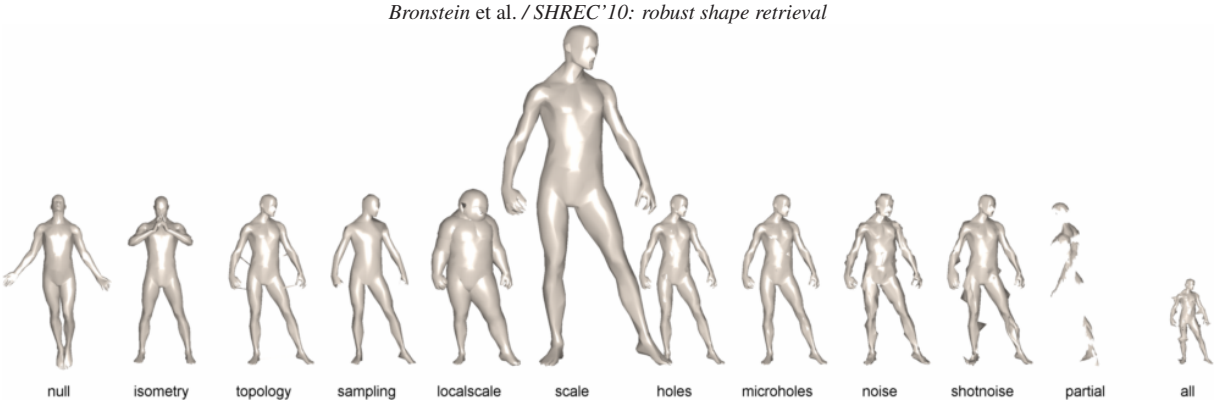

Figure 2: *Transformations of the human shape used as queries (shown in strength 5, left to right): null, isometry, topology, sampling, local scale, scale, holes, micro holes, noise, shot noise, partial, mixed.*

1184×1184 for the whole dataset, representing the dissimilarity between each pair of shapes. Performance was evaluated on subsets of the distance matrix using precision/recall characteristic. *Precision* $P(r)$ is defined as the percentage of relevant shapes in the first r top-ranked retrieved shapes. In the present benchmark, a single relevant shape existed in the database for each query. *Mean average precision* (mAP), defined as

$$mAP = \sum_r P(r) \cdot rel(r),$$

(where $rel(r)$ is the relevance of a given rank), was used as a single measure of performance. Ideal performance retrieval performance results in first relevant match with mAP=100%. Retrieval performance results between negative shapes and between same-class positive shapes (males and females, centaur, horse, and human shapes) were ignored.

4. Methods

Four families of methods were evaluated in this benchmark: *visual similarity* methods [LRS, LGS] (abbreviated as VS), *part-based* (PB) bags of words [TCF09], *ShapeGoogle* (SG) using bags of features constructed from intrinsic local descriptors [OBG09], and ShapeGoogle with *similarity sensitive hashing* (SS) [BBOG10]. The methods are briefly outlined in the following; the reader is referred to the respective papers for additional details.

VS1–3: Visual similarity

The visual similarity based method has been widely considered as the most discriminative approach in the field of content-based 3D object retrieval. Lian *et al.* developed two such kind of methods, referred to as *clock matching bag of features* (CM-BOF) and *geodesic sphere based multi-view descriptor* (GSMD), respectively. These two algorithms utilize a particular visual similarity based framework, and the only difference between them is how to describe the depth-buffer views captured around a 3D object. More specifically, CM-BOF uses a local feature based shape descriptor to represent a view as a histogram, and GSMD describes the view by a global feature vector. Finally, a *modified manifold ranking* (MMR) method is applied to try to further improve the retrieval performance of CM-BOF. Broadly, the visual similarity-based 3D shape retrieval framework is implemented subsequently in four steps:

Pose normalization: Normalize 3D objects with respect to the canonical coordinate frame to ensure that their mass centers coincide with the origin, they are bounded by the unit sphere, and they are well aligned to three coordinate axes. Rotation invariance is achieved by applying the PCA technique to find the principal axes and align them to the canonical coordinate frame.

View rendering: After pose normalization, 66 depth-buffer views with size 256×256 are captured on the vertices of a given unit geodesic sphere whose mass center is also located in the origin, such that a 3D model can be represented by a set of images. The views are rendered using OpenGL.

Feature extraction: For each view, a specific image processing technique is applied to represent the view as a compact feature vector. Based on the different 2D shape descriptors used, the algorithms are classified as the following two categories: local feature based and global feature based methods. In the CM-BOF algorithm, each view is described as a word histogram obtained by the vector quantization of the view's salient local features. In the GSMD method, each view is represented as a global feature vector with 47 elements including 35 Zernike moments, 10 Fourier coefficients, eccentricity and compactness.

Dissimilarity calculation: The last step of this framework is the dissimilarity calculation for two shape descriptors. The basic idea is that, after obtaining the principal axes of an object, instead of completely solving the problem of fixing the exact positions and directions of these three axes to the

canonical coordinate frame, all possible poses are taken into account during the shape matching stage.

The dissimilarity between the query model Y and the source model X is defined as,

$$D(Y,X) = \min_{0\le i\le 23} \sum_{k=0}^{65} d\left(\mathbf{f}_Y(\pi_0(k)), \mathbf{f}_X(\pi_i(k))\right),$$

where $\mathbf{f}_m = \{\mathbf{f}_m(k)\}_{k=0}^{65}$ denotes the shape descriptor of 3D object m, $\mathbf{f}_m(k)$ stands for the feature vector of view k, the permutations $\pi_i = \{\pi_i(k)\}_{k=0}^{65}$, $0 \le i \le 23$ indicate the arrangements of views for all 24 possible poses of a normalized model, and $d(\cdot,\cdot)$ measures the dissimilarity between two views. For more details about the multi-view shape matching scheme, the reader is referred to [LRS, LGS].

In the presented experiments, average descriptor computation time was about 4 sec for CM-BOF and CM-BOF-MRR and 0.5 for GSMD.

Given the distance matrix calculated using CM-BOF, MMR is applied to exploit the intrinsic global geometric structure of the target feature space. That results in the third method named as CM-BOF+MMR. In the following, for notation brevity, CM-BOF+MMR is denoted as VS1; CM-BOF as VS2, and GSMD as VS3.

PB1–3: Part-based bags of words

Toldo *et al.* [TCF09] presented a method is inspired by the *bag of words* (or features) framework for textual document classification and retrieval. The "words" are from a 3D visual vocabulary, defined by extracting and grouping the geometric features of the object sub-parts from the dataset, after 3D object segmentation. In the following, this class of algorithms is referred to as *part based* (PB). The main steps of the algorithm are as follows:

Object sub-parts extraction: Spectral clustering is used for the selection of seed-regions. Being inspired by the *minima-rule*, the adjacency matrix is defined purposely in order to allow convex regions to belong to the same segment. Furthermore, a multiple-region growing approach is introduced to expand the selected seed-regions. In particular, a weighted fast marching is proposed by guiding the front propagation according to local geometry properties. In practice, the main idea is to reduce the speed of the front for concave areas which are more likely to belong to the region boundaries.

Object sub-parts description: Local region signature is computed as a compact representation of each sub-part. Working at the part level, as opposed to the whole object level, enables a more flexible class representation and allows scenarios in which the query model is significantly transformed (e.g., deformed) to be classified.

3D visual vocabulary construction: The set of region descriptors are properly clustered in order to obtain a fixed number of 3D visual *words* (i.e., the set of clusters centroids). The clustering defines a vector quantization of the whole region descriptor space. Note that the vocabulary should be large enough to distinguish relevant changes in image parts, but not so large as to distinguish irrelevant variations such as noise.

Object representation and matching: Each 3D object is encoded by assigning to each object sub-part the corresponding visual word. A bag of words representation is defined by counting the number of object sub-parts assigned to each word. A histogram of visual words occurrences is build for each 3D object which represent its *global* signature. The objects matching is obtained by comparing their respective signature.

Three settings were used in this benchmark. In PB2, the visual vocabulary was computed from the training set. In PB3, the visual vocabulary was computed from the test set. In PB1, a different parameter settings was defined, increasing the number of clusters in the bag of words descriptors computation (i.e., from 8 to 14 centroids per descriptor).

SG1–3: ShapeGoogle

ShapeGoogle is a framework presented in [OBG09] for shape representation and retrieval using bags of geometric words computed from dense intrinsic descriptors. The framework consists of the following main stages:

Feature extraction: In ShapeGoogle, this stage consists of the computation of a dense feature descriptor $\mathbf{h}(x)$ at each point of the shape X. Different instances SG1–3 of the ShapeGoogle algorithm differ in the definition of the dense descriptor. In SG1, the heat kernel signature (HKS) [SOG09] $\mathbf{h}(x) = (h_{t_1}(x,x), \ldots, h_{t_n}(x,x))$ was used as a local descriptor, where $h_t(x,y) \approx \sum_{i=0}^{K} e^{-\lambda_i t}\phi_i(x)\phi_i(y)$ is the heat kernel associated with the positive-semidefinite Laplace-Beltrami operator Δ_X, and λ and ϕ are the eigenvalue and eigenfunctions of Δ_X. The cotangent weight scheme was used to discretize Δ_X. Values of $K = 100$, $n = 6$ were used, and $t_1, \ldots, t_6$ were chosen as $1024, 1351, 1783, 2353, 3104$ and 4096 (these are setting identical to [OBG09]).

In SG2, eigenpairs and the mass matrix obtained by the linear finite elements method (FEM) described in [PSF10] were used to compute the heat kernel signatures. Same settings for K, n and $t_1, \ldots, t_n$ as in SG1 were used. Such a discretization is known to be less sensitive to geometric and topological noise, irregular sampling, and local shape deformations.

In SG3, the scale-invariant heat kernel signature (SI-HKS) [BK10] was used. SI-HKS is

$$\hat{\mathbf{h}}(x) = |\mathcal{F}_\tau \mathrm{diff}_\tau \log(h_{\alpha^{\tau_1}}(x,x), \ldots, h_{\alpha^{\tau_n}}(x,x))|,$$

where diff denotes the finite difference operator and $\mathcal{F}$ is the Fourier transform. Cotangent weights and $K = 100$ first

eigenpairs were used to obtain $\mathbf{h}$. Value of $\alpha = 2$ and τ ranging from 1 to 25 with increments of $1/16$ were used. The first six discrete frequencies of the Fourier transform were taken (these are settings identical to [BK10]).

Vector quantization: Given a vocabulary of representative local descriptors $\mathcal{V} = \{\mathbf{h}_1, \ldots, \mathbf{h}_V\}$, the dense descriptor at each point of the shape is replaced by the V-dimensional distribution $\theta(x)$, with $\theta_k(x) = e^{-\|\mathbf{h}_k - \mathbf{h}(x)\|^2/2\sigma^2}$.

Bag of features computation: Finally, a bag of feature vector is computed by integrating the distributions over the entire shape, $\mathbf{f}_X = \sum_x \theta(x)$. Bag of feature vectors are normalized by the L_1 norm. The distance between two shapes X and Y is computed as

$$D(X,Y) = \|\mathbf{f}_X - \mathbf{f}_Y\|_1.$$

In the presented experiments, average computation time for a bag of feature was about 2 sec. For additional details about the ShapeGoogle framework and heat kernel descriptors, the reader is referred to [OBG09, BK10, SOG09]

SS1: Similarity sensitive hashing

An extension to the ShapeGoogle framework based on similarity sensitive hashing approach was presented in [BBOG10], by means of which bags of features are embedded into a Hamming space, which allows to represent shapes as binary codes. The approach includes training performed offline once and projection computation step performed for each shape.

Training (pre-computation): Given a representative set comprising pairs of V-dimensional ShapeGoogle bags of features of knowingly similar shapes differing only by a transformation drawn from the class of transformations invariance to which is desired (referred to as *positives* and denoted by P), and given a set of bags of features of knowingly dissimilar pairs of shapes (*negatives* N), a representation of the bags of features as n-dimensional binary codes is computed by means of a projection of the form

$$\mathbf{y}(\mathbf{f}) \;=\; \mathrm{sign}(\mathbf{A}\mathbf{f} + \mathbf{b}),$$

where $\mathbf{A}$ and $\mathbf{b}$ are an $m \times V$ matrix and an $m \times 1$ vector, respectively. The projection parameters $\mathbf{A}$ and $\mathbf{b}$ are selected in such a way that the Hamming metric between two projections,

$$d_{\mathbb{H}}(\mathbf{y},\mathbf{y}') \;=\; \frac{m}{2} - \frac{1}{2}\sum_{i=1}^{m} \mathrm{sign}(\mathbf{y}_i\mathbf{y}'_i),$$

reflects the desired similarity of the underlying interval descriptors. This is done by minimizing the loss function

$$\min_{\mathbf{A},\mathbf{b}} \quad \mathrm{E}_P\left\{e^{\mathrm{sign}(d_{\mathbb{H}}\circ(\mathbf{y}\times\mathbf{y}) - d_0)}\right\} + \mathrm{E}_N\left\{e^{\mathrm{sign}(d_0 - d_{\mathbb{H}}\circ(\mathbf{y}\times\mathbf{y}))}\right\}.$$

where empirical expectations on P and N are used. A modification of the approach introduced in [SVD03] based on Adaboost iterations are used to perform the training. At k-th iteration, the k-th row of the matrix a and the k-th element of the vector b are found by an LDA-type procedure minimizing a weighted version of the loss function. Weights of false positive and false negative pairs are increased, and weights of true positive and true negative pairs are decreased, using the standard Adaboost reweighting scheme.

Projection: Given the optimal projection parameters $\mathbf{A}^*, \mathbf{b}^*$ found at the training stage and a bag of features $\mathbf{f}$, the bitcode representation is computed as $\mathbf{y}(\mathbf{f}) = \mathrm{sign}(\mathbf{A}^*\mathbf{f} + \mathbf{b}^*)$. The comparison of two shapes with bitcodes $\mathbf{y}(\mathbf{f}_X)$ and $\mathbf{y}(\mathbf{f}_Y)$ is performed by computing the Hamming distance $D(X,Y) = d_{\mathbb{H}}(\mathbf{y}(\mathbf{f}_X), \mathbf{y}(\mathbf{f}_Y))$.

In the experiments in this report, $m = 96$, $|P| = 10^4$, and $|N| = 10^5$ were used. The training set was constructed from the separate training set optionally provided as part of the benchmark. Training time was about 20 minutes. The projection computation time was negligible.

5. Results

Tables 1–10 present the performance of each of the methods compared in the benchmark. Table 11 summarizes all the best performing algorithms.

In average performance on all classes of transformations, the best performance is achieved by ShapeGoogle using SI-HKS local descriptor and similarity sensitive hashing (SS1). Second best in all strengths is CM-BOF (VS2). In the isometric deformations class, all ShapeGoogle methods (SG1–3 and SS1) produce 100% mAP. Next best results are significantly inferior. Topological noise and holes are dealt with efficiently by VS2 and SS1. VS2, PB1 and all ShapeGoogle methods are insensitive to micro holes. VS2, PB1 and SS1 are insensitive to scale. SS1 shows the best performance with local scale. VS2 and SS2 achieve the best performance in the sampling class. In the noise class, CM-BOF and all ShapeGoogle methods (SG1–3 and SS1) produce 100% mAP at maximum strength. In the shot noise class, all ShapeGoogle methods (SG1–3 and SS1) produce 100% mAP at maximum strength. On mixed transformations, VS2 achieves the best performance.

The sensitivity of different methods tested in this benchmark to different classes of transformations is summarized in Table 12.

6. Conclusions

There is no absolute winner in SHREC'10 robust large-scale shape retrieval benchmark, as different method showed different performance across transformation classes. On the average, ShapeGoogle using SI-HKS local descriptor and similarity sensitive hashing (SS1) showed the best performance (98.27% mAP on the full query set, second place CM-BOF with 94.33% mAP, third place SG3 with 90.79% mAP). SS1

	Strength		
Transform.	**$\leq$ 1**	**$\leq$ 3**	**$\leq$ 5**
Isometry	VS2,PB1,SG1–3,SS1	SG1–3,SS1	SG1–3,SS1
Topology	VS2,PB1–3,SG1–2,SS1	VS2,SS1	VS2,SS1
Holes	VS2,PB1,SG1–3,SS1	VS2,SG2–3,SS1	VS2,SS1
Micro holes	VS2,PB1–2,SG1–3,SS1	VS2,PB1,SG1–3,SS1	VS2,PB1,SG1–3,SS1
Scale	VS2,PB1,SS1	VS2,PB1,SS1	VS2,PB1,SS1
Local scale	VS1–2,PB1–2,SG1–3,SS1	SS1	SS1
Sampling	VS2,PB2,SG1–3,SS1	VS2,SG1–3,SS1	VS2,SG2
Noise	VS2,PB1–2,SG1–3,SS1	VS2,SG1–3,SS1	VS2,SG1–3,SS1
Shot noise	VS2,SG1–3,SS1	VS2,SG1–3,SS1	SG1–3,SS1
Partial	SS1	SS1	SS1
Mixed	VS2	VS2	VS2
Average	SS1	SS1	SS1

Table 11: *Winning algorithms across transformation classes and strengths. VS1=CM-BOF+MMR, VS2=CM-BOF, VS3=GSMD, PB1=part-based bag of words with large number of visual words, PB2=part-based bag of words with visual vocabulary computed from the training set, PB3=part-based bag of words with visual vocabulary computed from the test set, SG1=ShapeGoogle with HKS descriptor using cotangent weights, SG2=ShapeGoogle with HKS descriptor using FEM, SG3=ShapeGoogle with SI-HKS descriptor using cotangent weights, SS1=ShapeGoogle with SI-HKS descriptor and similarity-sensitive hashing.*

Transform.	**VS1**	**VS2**	**VS3**	**PB1**	**PB2**	**PB3**	**SG1**	**SG2**	**SG3**	**SS1**
Isometry	H	M	M	M	M	M	**L**	**L**	**L**	**L**
Topology	M	**L**	M	**L**	**L**	**L**	**L**	**L**	**L**	**L**
Holes	M	**L**	**L**	M	H	M	**L**	**L**	M	**L**
Micro holes	**L**	**L**	M	**L**	**L**	**L**	**L**	**L**	**L**	**L**
Scale	**L**	**L**	**L**	**L**	**L**	**L**	H	H	**L**	**L**
Local scale	H	**L**	M	M	M	M	M	M	M	**L**
Sampling	M	**L**	M	M	M	M	**L**	**L**	**L**	**L**
Noise	M	**L**	**L**	M	H	H	**L**	**L**	**L**	**L**
Shot noise	M	**L**	**L**	**L**	M	M	**L**	**L**	**L**	**L**
Partial	H	H	H	H	H	H	H	H	H	M
Mixed	M	**L**	**L**	H	H	H	H	H	M	**L**

Table 12: *Sensitivity of different methods to classes of transformations tested in this benchmark (L: mAP>90% in strength 5; M: 50%$\leq$mAP$\leq$90% in strength 5; H: mAP<50% in strength 5).*

	Strength				
Transform.	**1**	**$\leq$2**	**$\leq$3**	**$\leq$4**	**$\leq$5**
Isometry	94.87	65.52	54.13	48.42	45.72
Topology	94.87	94.87	94.87	90.75	88.27
Holes	89.10	89.10	88.78	84.20	79.06
Micro holes	94.87	92.95	91.82	91.25	91.00
Scale	94.87	94.87	94.87	94.87	94.87
Local scale	100.00	69.06	53.59	40.56	32.68
Sampling	88.46	80.46	78.20	76.32	69.94
Noise	94.87	86.55	72.17	63.20	56.44
Shot noise	94.87	93.60	85.50	72.49	62.39
Partial	23.24	16.16	14.69	11.71	9.42
Mixed	37.92	50.01	59.95	64.60	65.39
Average	95.92	90.81	86.48	81.94	77.78

Table 1: *Performance of VS1: clock matching bag of features with modified manifold ranking (mAP in %).*

	Strength				
Transform.	**1**	**$\leq$2**	**$\leq$3**	**$\leq$4**	**$\leq$5**
Isometry	100.00	86.67	79.24	77.46	72.58
Topology	100.00	100.00	100.00	100.00	100.00
Holes	100.00	100.00	100.00	100.00	100.00
Micro holes	100.00	100.00	100.00	100.00	100.00
Scale	100.00	100.00	100.00	100.00	100.00
Local scale	100.00	100.00	98.72	96.47	92.95
Sampling	100.00	100.00	100.00	100.00	100.00
Noise	100.00	100.00	100.00	100.00	100.00
Shot noise	100.00	100.00	100.00	100.00	98.46
Partial	54.22	47.45	46.28	40.57	35.49
Mixed	100.00	100.00	100.00	98.56	97.31
Average	99.03	97.73	96.71	95.66	94.33

Table 2: *Performance of VS2: clock matching bag of features (mAP in %).*

	Strength				
Transform.	**1**	**≤2**	**≤3**	**≤4**	**≤5**
Isometry	90.38	62.51	65.23	64.31	63.06
Topology	89.10	88.57	88.62	88.53	88.52
Holes	94.23	94.23	94.23	94.23	94.15
Micro holes	91.03	90.38	89.96	89.74	89.62
Scale	90.38	90.38	90.38	90.38	90.38
Local scale	89.10	89.10	88.72	86.50	80.24
Sampling	90.38	89.74	89.32	89.10	89.23
Noise	90.38	90.38	91.67	91.35	91.15
Shot noise	90.38	90.71	90.04	90.99	91.56
Partial	50.95	46.73	43.61	36.26	35.32
Mixed	89.10	89.74	89.53	89.65	91.72
Average	96.93	93.89	92.24	90.56	89.29

Table 3: *Performance of VS3: geodesic sphere based multi-view descriptor (mAP in %).*

	Strength				
Transform.	**1**	**≤2**	**≤3**	**≤4**	**≤5**
Isometry	100.00	88.57	88.54	88.04	88.04
Topology	100.00	98.08	97.44	97.12	97.69
Holes	100.00	91.94	76.24	67.65	61.58
Micro holes	100.00	100.00	100.00	100.00	100.00
Scale	100.00	100.00	100.00	100.00	100.00
Local scale	100.00	97.12	94.40	85.97	77.61
Sampling	94.87	95.51	90.68	85.06	79.30
Noise	100.00	88.78	69.27	58.23	52.67
Shot noise	92.31	94.23	93.59	93.27	93.85
Partial	1.35	1.49	1.46	1.48	1.43
Mixed	39.13	40.30	38.20	36.59	33.55
Average	95.28	92.11	88.41	85.06	82.20

Table 4: *Performance of PB1: part-based bag of words with large number of visual words (mAP in %).*

	Strength				
Transform.	**1**	**≤2**	**≤3**	**≤4**	**≤5**
Isometry	96.15	86.07	85.37	84.56	84.40
Topology	100.00	97.44	98.29	98.72	98.21
Holes	86.15	80.45	63.86	53.80	47.16
Micro holes	100.00	98.08	98.72	98.08	98.46
Scale	96.15	96.15	97.44	98.08	97.69
Local scale	100.00	92.63	87.46	75.38	67.74
Sampling	100.00	100.00	92.46	85.96	74.01
Noise	100.00	80.93	62.04	52.36	46.17
Shot noise	92.31	93.59	89.74	90.38	89.62
Partial	0.76	0.75	0.79	0.75	0.76
Mixed	25.94	23.49	22.44	21.17	18.22
Average	94.64	90.52	86.19	82.35	78.72

Table 5: *Performance of PB2: part-based bag of words with visual vocabulary computed from the training set (mAP in %).*

	Strength				
Transform.	**1**	**≤2**	**≤3**	**≤4**	**≤5**
Isometry	94.87	84.83	85.73	85.70	83.53
Topology	100.00	95.51	95.30	94.83	92.84
Holes	83.33	75.72	64.73	57.47	54.82
Micro holes	96.15	94.23	92.95	94.71	95.00
Scale	94.87	94.87	94.87	94.87	94.87
Local scale	94.87	94.02	91.05	84.35	76.11
Sampling	89.42	88.94	85.90	80.36	72.25
Noise	83.59	77.41	60.01	50.90	45.06
Shot noise	90.38	90.71	87.48	87.04	87.48
Partial	1.64	1.77	1.62	1.49	1.60
Mixed	36.05	33.98	31.98	32.48	31.49
Average	94.25	90.16	86.09	82.78	79.57

Table 6: *Performance of PB3: part-based bag of words with visual vocabulary computed from the test set (mAP in %).*

	Strength				
Transform.	**1**	**≤2**	**≤3**	**≤4**	**≤5**
Isometry	100.00	100.00	100.00	100.00	100.00
Topology	100.00	98.08	97.44	96.79	96.41
Holes	100.00	100.00	97.44	95.19	90.13
Micro holes	100.00	100.00	100.00	100.00	100.00
Scale	0.98	40.68	43.31	33.72	27.42
Local scale	100.00	100.00	98.72	89.38	80.22
Sampling	100.00	100.00	100.00	100.00	99.23
Noise	100.00	100.00	100.00	100.00	100.00
Shot noise	100.00	100.00	100.00	100.00	100.00
Partial	7.54	5.70	4.51	3.58	2.95
Mixed	53.13	55.86	47.77	37.54	30.34
Average	94.94	93.12	90.84	87.82	85.00

Table 7: *Performance of SG1: ShapeGoogle using HKS local descriptor computed with cotangent weight discretization (mAP in %).*

	Strength				
Transform.	**1**	**≤2**	**≤3**	**≤4**	**≤5**
Isometry	100.00	100.00	100.00	100.00	100.00
Topology	100.00	100.00	98.72	98.08	97.69
Holes	100.00	100.00	100.00	99.04	95.23
Micro holes	100.00	100.00	100.00	100.00	100.00
Scale	0.34	28.05	21.67	16.60	13.52
Local scale	100.00	100.00	95.51	85.19	75.63
Sampling	100.00	100.00	100.00	100.00	100.00
Noise	100.00	100.00	100.00	100.00	100.00
Shot noise	100.00	100.00	100.00	100.00	100.00
Partial	20.90	15.75	11.23	8.71	7.07
Mixed	77.51	76.39	53.25	40.25	32.28
Average	95.73	93.81	90.46	87.40	84.71

Table 8: *Performance of SG2: ShapeGoogle using HKS local descriptor computed with FEM discretization (mAP in %).*

	Strength				
Transform.	**1**	**≤2**	**≤3**	**≤4**	**≤5**
Isometry	100.00	100.00	100.00	100.00	100.00
Topology	96.15	96.15	94.87	93.27	92.69
Holes	100.00	100.00	100.00	94.71	89.97
Micro holes	100.00	100.00	100.00	100.00	100.00
Scale	91.03	95.51	97.01	97.76	98.21
Local scale	100.00	100.00	97.44	89.38	82.08
Sampling	100.00	100.00	100.00	100.00	97.69
Noise	100.00	100.00	100.00	100.00	100.00
Shot noise	100.00	100.00	100.00	100.00	100.00
Partial	17.43	10.31	9.57	8.06	6.61
Mixed	56.47	57.44	63.59	67.47	65.07
Average	97.05	95.16	94.03	92.54	90.79

Table 9: *Performance of SG3: ShapeGoogle using SI-HKS local descriptor computed with cotangent weight discretization (mAP in %).*

	Strength				
Transform.	**1**	**≤2**	**≤3**	**≤4**	**≤5**
Isometry	100.00	100.00	100.00	100.00	100.00
Topology	100.00	100.00	100.00	100.00	100.00
Holes	100.00	100.00	100.00	100.00	100.00
Micro holes	100.00	100.00	100.00	100.00	100.00
Scale	100.00	100.00	100.00	100.00	100.00
Local scale	100.00	100.00	100.00	97.18	94.98
Sampling	100.00	100.00	100.00	100.00	99.23
Noise	100.00	100.00	100.00	100.00	100.00
Shot noise	100.00	100.00	100.00	100.00	100.00
Partial	96.15	88.81	86.52	86.80	78.86
Mixed	96.15	96.15	97.44	98.08	95.35
Average	99.84	99.48	99.30	99.10	98.27

Table 10: *Performance of SS1: ShapeGoogle using HKS local descriptor computed with cotangent weight discretization and 96 bit similarity sensitive hash (mAP in %).*

was also among the best in all transformation classes excepting sampling and mixed transformation. CM-BOF and ShapeGoogle using HKS local descriptor computed with FEM discretization (SG2) showed the best robustness to sampling change. CM-BOF showed the best performance in mixed transformations class. The ShapeGoogle framework showed significantly better robustness to non-rigid deformations compared to other methods.

A more detailed version of this report presenting additional details and experiments will be published separately.

References

[BBK08] BRONSTEIN A. M., BRONSTEIN M. M., KIMMEL R.: *Numerical geometry of non-rigid shapes*. Springer, 2008. 1

[BBOG10] BRONSTEIN A. M., BRONSTEIN M. M., OVSJANIKOV M., GUIBAS L. J.: ShapeGoogle: geometric words and expressions for invariant shape retrieval. *TOG (in review)* (2010). 3, 5

[BK10] BRONSTEIN M. M., KOKKINOS I.: Scale-invariant heat kernel signatures for non-rigid shape recognition. In *Proc. CVPR* (2010). 4, 5

[LGS] LIAN Z., GODIL A., SUN X.: Visual similarity based 3D shape retrieval using bag-of-features. *(in review)*. 3, 4

[LRS] LIAN Z., ROSIN P. L., SUN X.: Rectilinearity of 3D meshes. *International Journal of Computer Vision (in press)*. 3, 4

[OBG09] OVSJANIKOV M. BRONSTEIN A. M., BRONSTEIN M. M., GUIBAS L. J.: ShapeGoogle: a computer vision approach to invariant shape retrieval. *Proc. NORDIA* (2009). 3, 4, 5

[PSF10] PATANÉ G., SPAGNUOLO M., FALCIDIENO B.: Multi-scale feature spaces for shape analysis and processing. In *Proc. Shape Modeling International (SMI)* (2010). to appear. 4

[SMKF04] SHILANE P., MIN P., KAZHDAN M., FUNKHOUSER T.: The Princeton shape benchmark. In *Shape Modeling International* (2004), vol. 105, p. 179. 1

[SOG09] SUN J., OVSJANIKOV M., GUIBAS L. J.: A concise and provably informative multi-scale signature based on heat diffusion. In *Computer Graphics Forum* (2009), vol. 28, pp. 1383–1392. 4, 5

[SP04] SUMNER R. W., POPOVIĆ J.: Deformation transfer for triangle meshes. In *ACM SIGGRAPH* (2004), p. 405. 1

[SVD03] SHAKHNAROVICH G., VIOLA P., DARRELL T.: Fast pose estimation with parameter-sensitive hashing. In *Proc. CVPR* (2003), p. 750. 5

[TCF09] TOLDO R., CASTELLANI U., FUSIELLO A.: Visual vocabulary signature for 3D object retrieval and partial matching. In *Proc. Eurographics Workshop on 3D Object Retrieval* (2009). 3, 4