Eurographics Workshop on 3D Object Retrieval (2015)
I. Pratikakis, M. Spagnuolo, T. Theoharis, L. Van Gool, and R. Veltkamp (Editors)

SHREC'15 Track: Non-rigid 3D Shape Retrieval†

Z. Lian[1‡], J. Zhang[1‡], S. Choi[2], H. ElNaghy[3], J. El-Sana[4], T. Furuya[5], A. Giachetti[6], R.A. Guler[7]
L. Lai[8], C. Li[9], H. Li[8], F.A. Limberger[10], R. Martin[11], R.U. Nakanishi[12], A.P. Neto[12]
L.G. Nonato[3], R. Ohbuchi[5], K. Pevzner[4], D. Pickup[11], P. Rosin[11], A. Sharf[4]
L. Sun[8], X. Sun[11], S. Tari[13], G. Unal[7], R.C. Wilson[10]

[1]Institute of Computer Science and Technology, Peking University, Beijing, PR China
[2]Seoul National University, Korea, [3]Ain Shams University, Egypt, [4]Ben Gurion University, Israel
[5]University of Yamanashi, Japan, [6]University of Verona, Italy, [7]Sabanci University, Turkey
[8]Beijing Technology and Business University, Beijing, PR China, [9]Duke University, Durham, USA, [10]University of York, UK
[11]Cardiff University, UK, [12]Universidade de São Paulo, Brazil, [13]Middle Eastern Technical University, Turkey

Abstract

Non-rigid 3D shape retrieval has become a research hotpot in communities of computer graphics, computer vision, pattern recognition, etc. In this paper, we present the results of the SHREC'15 Track: Non-rigid 3D Shape Retrieval. The aim of this track is to provide a fair and effective platform to evaluate and compare the performance of current non-rigid 3D shape retrieval methods developed by different research groups around the world. The database utilized in this track consists of 1200 3D watertight triangle meshes which are equally classified into 50 categories. All models in the same category are generated from an original 3D mesh by implementing various pose transformations. The retrieval performance of a method is evaluated using 6 commonly-used measures (i.e., PR-plot, NN, FT, ST, E-measure and DCG.). Totally, there are 37 submissions and 11 groups taking part in this track. Evaluation results and comparison analyses described in this paper not only show the bright future in researches of non-rigid 3D shape retrieval but also point out several promising research directions in this topic.

Categories and Subject Descriptors (according to ACM CCS): H.3.3 [Computer Graphics]: Information Systems—Information Search and Retrieval

1. Introduction

With the rapid development of computer hardware and software, 3D models have become widely-used in our daily lives. Since the number of 3D models is increasing rapidly, the focus of researchers' interests has been shifted from "how to design and create 3D models" to "how to quickly and accurately find 3D models we want". Until now, large numbers of papers on content-based 3D object retrieval have been published. In the last few years, the topic of non-rigid 3D shape retrieval has attracted more and more researchers around the world. Possible reasons are twofold. First, non-rigid 3D objects are commonly-seen in our surroundings. For example, the same type of models shown in Figure 1

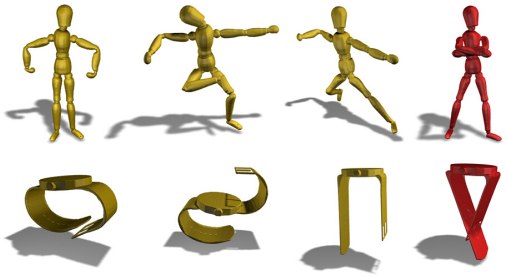

Figure 1: *Examples of non-rigid 3D models. Note that the rightmost two models have different topological structures against other objects in the same class.*

† http://www.icst.pku.edu.cn/zlian/shrec15-non-rigid/
‡ Track organizers. E-mail: lianzhouhui@pku.edu.cn

DOI: 10.2312/3dor.20151064

appear in quite different poses and thus it is hard to classify them into the same categories using many traditional approaches. Second, generating isometry-invariant and discriminative shape descriptors usually requires the exploration of many mathematical theories and the utilization of new techniques in relevant research areas.

Up to now, a large number of algorithms for non-rigid 3D shape retrieval have been proposed. Since each method has its own advantages and disadvantages, it is therefore important to fairly compare the performance of those different methods in a convincing manner. However, most retrieval approaches specifically-designed for non-rigid 3D objects can only process watertight single-component meshes. Mainly due to the fact that designers typically only care about the visual appearances of 3D objects they design, large percentages of existing 3D models are not watertight and often contain inner structures and/or geometric noises. Thereby, building a large-scale database containing such kinds of high-quality 3D watertight meshes is not a trivial task. The McGill Articulation 3D Shape Benchmark (McGill) [SZM*08] is the first publicly available benchmark which is now widely-used for the evaluation of non-rigid 3D shape retrieval methods. However, only 10 categories and totally 255 models are contained in the McGill database, and furthermore the numbers of objects in each pair of categories are unequal. That might cause evaluation bias and make the performance comparison based on the McGill database not convincing enough.

In order to solve this problem and promote the investigations of non-rigid 3D shape retrieval, we organized two SHREC tracks on non-rigid 3D shape retrieval in 2010 and 2011, respectively. The SHREC'10 non-rigid track [LGF*10] is the first attempt in the history of SHREC to specifically focus on the comparison of non-rigid 3D shape retrieval methods. The track only utilized 200 models for evaluation and there were merely 3 participation groups. In the next year, both the size of database (600 models) and the number of participants (9 groups) tripled [LGB*11], indicating that more and more researchers have become interested in this research topic. From then on, more efforts have been devoted to the researches of non-rigid 3D shape retrieval, making this topic a hotpot in content-based 3D object retrieval. This year, we built a new non-rigid 3D shape database consisting of 1200 watertight 3D triangle meshes that could be equally divided into 50 classes. Each group was asked to submit up to five distance matrices obtained by implementing their methods, and finally 11 groups around the world took part in the *SHREC'15 track: Non-rigid 3D Shape Retrieval*. The rest of this paper is organized as follows: Section 2 presents an overview of our database. Section 3 describes the evaluation rules of the track. Section 4 briefly introduces the 11 participation groups and their methods, whose more details are discussed in Section 5. Experimental results and corresponding analyzes are provided in Section 6. Finally, Section 7 concludes the paper and points out some future research directions.

2. Database

The new database built for this track consists of 1200 watertight triangle meshes (see Figure 2) derived from 50 original models that are selected from the following publicly available repositories (e.g., PSB database [SMKF04], McGill database [SZM*08], TOSCA shapes [BBK08], SHREC'11 non-rigid database [LGB*11], Google 3D Warehouse [Goo15], etc.). Given a 3D mesh, we use *Autodesk 3d Max* to build its skeleton and then generate 23 deformed versions of the mesh by articulating around its joints in different ways. To remove the inner structures of those articulated models (see Figure 3), we implement our own codes to first capture 18 depth-buffer views for the normalized object on the vertices of a unit geodesic sphere, and then convert those images into a point cloud. Finally, we wrap the point cloud into a polygon surface and fix it to form a watertight 3D manifold by using *Geomagic*, which can be automatically implemented with recorded macros. Figure 2 shows some examples of those 1200 non-rigid models which have been equally classified into 50 categories.

Note that: 1) The 600 models used in the SHREC'11 non-rigid track [LGB*11] are directly included in the new database; 2) In each category, 20 models have the same topological structures as the original source model while there exist 4 objects whose topological structures are modified by letting some parts be connected. The purpose is to construct a more challenging database so that the robustness of a method against topological errors can also be evaluated.

3. Evaluation

The evaluation rules are exactly the same as previous two SHREC non-rigid tracks. Specifically, participants are asked to implement their algorithms on the database to calculate the dissimilarity between each pair of models, and then generate a distance matrix for each method. The matrix consists of 1200×1200 floating point numbers, where the number at position (i, j) represents the dissimilarity between models i and j.

With the matrices submitted by participants, we evaluate their retrieval performance based on Precision-recall curves (PR-plot) and the following five quantitative measures: Nearest Neighbor (NN), First Tier (FT), Second Tier (ST), E-measure (E), and Discounted Cumulative Gain (DCG). Detailed definitions of these six evaluation measures can be found in [SMKF04].

4. Participants

This year, we had 11 groups taking part in the *SHREC'15 Track: Non-rigid 3D Shape Retrieval* and received 37 dissimilarity matrices as follows:

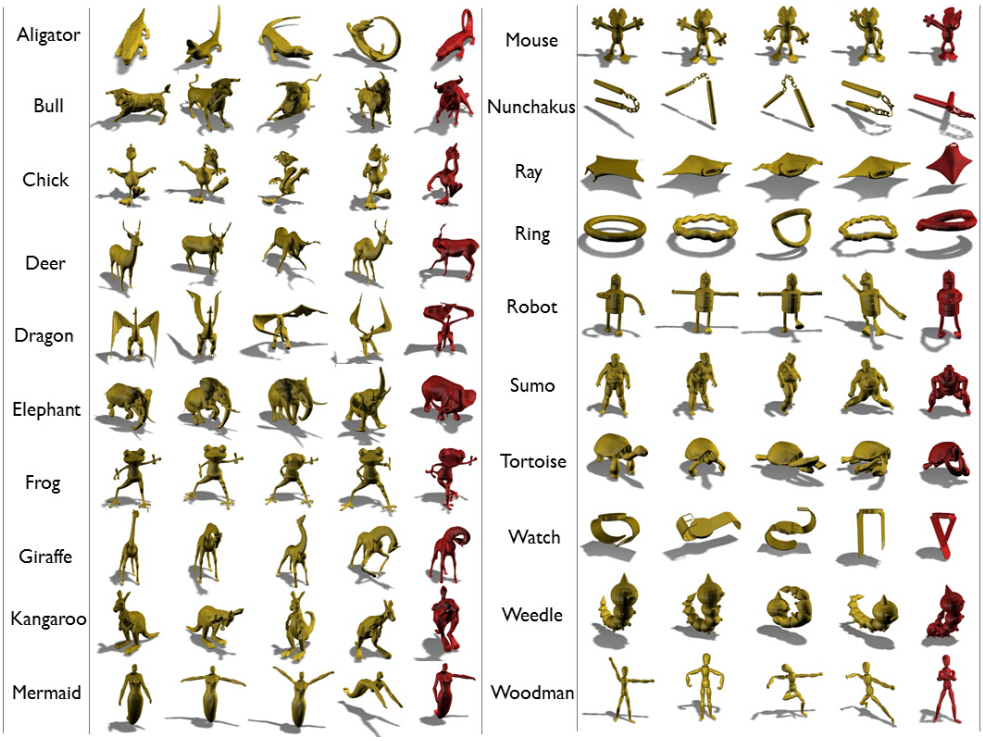

Figure 2: *Examples of the new types of models that are not included in the SHREC'11 non-rigid database.*

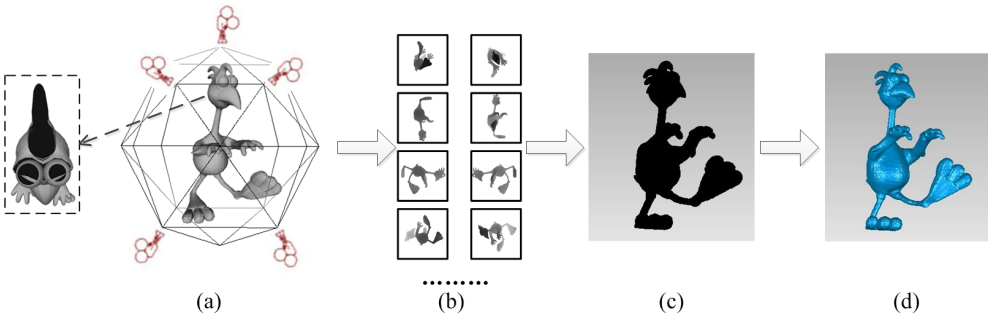

Figure 3: *Creating the watertight manifold for a 3D mesh that may contain inner structures and other noises/errors. We first normalize the original model (a) so that it is located inside a unit geodesic sphere, and then a set of depth-buffer views (b) of the 3D mesh are captured on the 18 vertices of the bounding sphere. Afterwards, we convert those depth-buffer images into a point cloud (c). Finally, by wrapping the point cloud and fixing holes and other errors on the surface, we obtain a watertight manifold (d) without inner structures.*

1. *SNU-1* and *SNU-2* submitted by Sungbin Choi from Seoul National University, Korea.
2. *CompactBoFHKS-4D*, *CompactBoFHKS-5D*, *CompactBoFHKS-10D* and *CompactBoFHKS-19D* submitted by Hanan ElNaghy from Ain Shams University, Egypt.
3. *SV-LSF* and *SV-LSF-kpca50* submitted by Takahiko Furuya and Ryutarou Ohbuchi from University of Yamanashi, Japan.
4. *HAPT-run1*, *HAPT-run2* and *HAPT-run3* submitted by Andrea Giachetti from University of Verona, Italy.
5. *SPH-SparseCoding-256*, *SPH-SparseCoding-1024*, *SPH-SPEM-VLAD-32* and *SPH-SPEM-VLAD-64* submitted by Riza Alp Guler, Gozde Unal from Sabanci

University, Turkey and Sibel Tari from Middle Eastern Technical University, Turkey.
6. *HKS*, *SA*, *WKS* and *Multi-Feature* submitted by Long Lai, Li Sun and Haisheng Li from Beijing Technology and Business University, China.
7. *SG-L1*, *SG-L2*, *SG-L3*, *SG-L4* and *SG-L5* submitted by Chunyuan Li from Duke University, USA.
8. *FVF-HKS*, *FVF-SIHKS* and *FVF-WKS* submitted by Frederico A. Limberger and Richard C. Wilson from University of York, UK.
9. *TSASR256*, *TSASR512* and *TSASR1024* submitted by Rafael Umino Nakanishi, Afonso Paiva Neto and Luis Gustavo Nonato from Universidade de São Paulo, Brazil.
10. *SID-1*, *SID-2*, *SID-3*, *SID-4* and *SID-5* submitted by Kirill Pevzner, Andrei Sharf and Jihad El-Sana from Ben Gurion University, Israel.
11. *EDBCF-AV* and *EDBCF-NW* submitted by David Pickup, Xianfang Sun, Paul Rosin and Ralph Martin from Cardiff University, UK.

5. Methods

5.1. SNU: Geodesic Distance Distribution, by S. Choi

In this method, only geodesic distance distributions of mesh models are utilized. For each mesh model in the data set, 1,600 points are sampled on the surface. Then, all geodesic distances between sampled points are calculated, generating a 1600×1600 geodesic distance matrix (GDM). Afterwards, GDM is converted to a histogram having 200 bins. Dissimilarity between different mesh models is calculated by applying different similarity measure functions (RunID SNU_1: Euclidean distance; RunID SNU_2: Correlation measure) [SFH*09].

5.2. CompactBoFHKS: Bag of Compact HKS-based Feature Descriptors, by H. ElNaghy

The key idea of this method is the synergy between Heat Kernel Signatures (HKS) and Bag of Features (BoF) paradigm [BBGO09]. The method proceeds through five main phases: HKS Computation, Feature Point Detection, Feature Point Description, Bag of Features and finally the Matching phase.

Initially, the HKS computation phase encodes each point in a given 3D model by an m-dimensional HKS feature vector $K_t(x,x)$. It describes the model's local and global geometric properties at m different time values over the time interval $[t_{min},t_{max}]$, using equation (1):

$$K_t(x,x) = \sum_{i=0}^{k} e^{-\lambda_i t} \phi_i(x)^2 \qquad (1)$$

where λ_i and ϕ_i are the i^{th} eigenvalue and eigenfunction of the Laplace-Beltrami operator respectively [SOL09].

The second phase of feature point detection constitutes an initial set of feature points by capturing the HKS critical points for all models over m times scales. Then, an innovative filtering technique is applied on this initial set to carefully pick the most stable and significant feature points. It inspects all the initially detected critical points and discards repeated or insignificant ones. The proposed filtering technique dramatically reduces both time and space required later to construct the visual dictionary during the Bag of Features phase.

After that, the feature description phase normalizes the HKS feature vector $K_t(x,x)$ associated with each feature point in the feature space. This normalization process produces an m-dimensional feature vector $h(x)$, such that:

$$h(x) = (h_1(x),...,h_m(x)); h_i(x) = c(x)K_{t_i}(x,x) \qquad (2)$$

where the constant $c(x)$ is selected in such a way that $||h(x)||_2 = 1$ and $t_i \in [t_{min},t_{max}]$ [BBGO09].Then, a d-dimensional HKS-based feature vector $p(x)$ is selected carefully as a subset of $h(x)$. The feature descriptor $p(x)$ should encode each feature point belonging to the final feature space in a compact, robust and informative way.

Then, the Bag of Features phase is subdivided into two main sub phases: vocabulary construction and object representation. Throughout the vocabulary construction sub phase, a visual dictionary $\mathtt{P} = \{p_1,...,p_v\}$ of size v is built by clustering the feature space. Then, 3D object representation sub phase performs vector quantization in the feature space in order to represent each 3D model by its bag of features vector $F(x)$.

Given a visual dictionary $\mathtt{P} = \{p_1,...,p_v\}$ and a 3D model X with n vertices, each vertex x is represented with a v-dimensional feature distribution $\theta(x)$ such that its associated d-dimensional feature descriptor $p(x)$ is replaced by the closest geometric word p_i from the visual dictionary $\mathtt{P}$. After vector quantization, the final v-dimensional BoF vector $F(x)$ [BBLO11] is computed such that:

$$F(x) = \sum_{i=1}^{n} \theta(x_i) \qquad (3)$$

Finally, the matching phase determines the similarity or dissimilarity of two given 3D models by computing the distance d_{BoF} between their corresponding bags of features. Therefore, 3D object similarity problem is reduced to the problem of comparing vectors of feature frequency.

Since this method is mainly based on the Bag of Features paradigm using a concise feature space of compact HKS-based feature descriptors, it is denoted as "CompactBoFHKS".

Parameter settings are as follows:

- For HKS computation, the number of computed eigenvalues (λ) and eigenvectors (ϕ) was set to 300 ($k = 300$) and HKS ($K_t(x,x)$) is computed for each vertex by uniformly

sampling 100 points ($m = 100$) in the logarithmically scale over time interval $[t_{min}, t_{max}]$, with $t_{min} = 4ln10/\lambda_{300}$ and $t_{max} = 4ln10/\lambda_2$.

- For vocabulary construction, K-means algorithm is chosen for clustering the feature space and the size of the visual dictionary was set to 64 (v=64).
- For 3D object representation, hard vector quantization was adopted for computing the feature distribution $\theta(x)$.
- For final matching, Manhattan distance (L1-norm) is used for computing the distance d_{BoF} between two given bags of features.
- Four runs have been proposed by changing the criteria of choosing the d-dimensional feature descriptor as $p(x)$ a subset of the m-dimensional HKS-based feature vector h(x).

5.3. SV-LSF: Super Vector of Localized Statistical Features, by T. Furuya and R. Ohbuchi

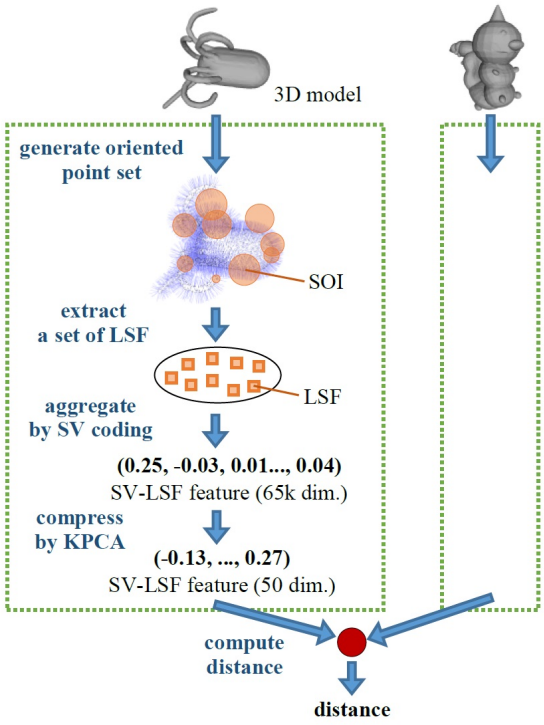

Figure 4: *Pipeline of SV-LSF algorithm.*

The algorithm aims at extracting 3D shape descriptors that are robust against articulation of 3D models, are applicable to diverse shape representations, and are compact for efficient comparison among 3D models. The SV-LSF is an improved version of the LSF algorithm by Ohkita et al. [OOFO12] [LGB*11]. Figure 4 describes a pipeline of our algorithm. The algorithm consists of the following steps.

(1) Generating oriented point set: Given a surface-based 3D model, it is first converted into a 3D oriented point set by sampling the surfaces of the 3D model. This sampling process gains invariances to shape representations of 3D model. We use the algorithm by Osada et al. [OFCD02] for converting a polygonal model into an oriented point set. The algorithm randomly and uniformly samples points on the surfaces of the 3D polygonal model. Each point is associated with the normal vector of the triangle on which the point is sampled. We sample $N_p = 5000$ oriented points per 3D model. The oriented point set of the 3D model is scaled to fit a sphere having diameter 1.

(2) Extracting local features: We densely extract a set of local 3D geometric features from the oriented point set. For each oriented point, a Sphere-Of-Interest (SOI), in which a local feature is computed, is defined. For robustness against scale change of local 3D shapes, we use multi-scale SOIs. Radius of each SOI is randomly selected from a range [0.01, 0.15]. The set of oriented points enclosed in the SOI is described by LSF [OOFO12], which is a 625-dimensional feature vector having invariance against 3D rotation. The dimensionality of LSF is then reduced from 625 down to 64 by using PCA to make LSF more discriminative and to accelerate subsequent process (i.e., feature aggregation). A 3D model is represented by a set of $N_p = 5000$ LSF features.

(3) Aggregating local features: The set of LSF features is aggregated into a feature vector per 3D model for efficient comparison among 3D models. We use Super Vector (SV) coding [ZYZH10] for more accurate aggregation than commonly-used Bag-of-Features [CDF*04]. In this SHREC track, we use soft-assignment variant to SV coding [FO14]. A codebook for SV is learned by using Gaussian Mixture Model (GMM) clustering algorithm using a set of 250,000 LSF features randomly selected from all the LSF features extracted of 3D models in the test database. We learn 1,000 clusters, or codewords, for the codebook. The SV-LSF feature is power-normalized by applying the following equation to each element z of the SV-LSF feature vector.

$$f(z) = sign(z)|z|^p \tag{4}$$

we use $p = 0.3$ for the track. The power-normalized feature is then normalized by its L2 norm.

(4) Reducing dimensionality of aggregated features: The SV-LSF is a high-dimensional feature vector. As the dimensionality of LSF is 64 and the number of clusters is 1000, SV-LSF is $(64+1)1,000 = 65,000$-dimensional vector. To reduce the cost for feature comparison among 3D models, dimensionality of the SV-LSF is reduced down to 50 by using Kernel PCA (KPCA) with dot kernel as with [FO14]. To learn a projection of KPCA, we use 1200 SV-LSF features extracted from 1200 3D models in the test dataset. A distance between a pair of 3D models is computed by using Cosine distance between a pair of SV-LSF features of the 3D models. Cosine distance is computed as $-(c+1)/2$, where c is Cosine similarity between two SV-LSF features.

All the parameters for the SV-LSF were tuned through preliminary experiments using the SHREC 2011 Non-rigid 3D Watertight Meshes dataset [LGB*11]. We determined the combination of parameters so that retrieval accuracy became the highest among the combinations of parameters we tried. We submit the following two runs for this track;

- **SV-LSF:** 65,000-dimensional SV-LSF features are compared to generate ranking results.
- **SV-LSF-kpca50:** 50-dimensional SV-LSF features produced by KPCA are compared.

5.4. HAPT: Histograms of Area Projection Transform, by A. Giachetti

The method proposed is based on Histograms of Area Projection Transform (HAPT) [GL12]. It is based on the estimation of a discretized spatial map (Multiscale Area Projection Transform) that encodes the likelihood of the points inside the shape of being centers of spherical symmetry. This map is obtained by computing for each radius of interest the value:

$$APT(\vec{x},S,R,\sigma) = Area(T_R^{-1}(k_\sigma(\vec{x}) \subset T_R(S,\vec{n}))) \quad (5)$$

where S is the surface of interest, $T_R(S,\vec{n})$ is the parallel surface of S shifted along the normal vector (only in the inner direction) and $k_\sigma(\vec{x}$ is a sphere of radius σ centered in the generic point $\vec{x}$ where the map is computed. Values at different radii are normalized in order to have a scale-invariant behavior, creating the Multiscale APT (MAPT):

$$MAPT(x,y,z,R,S) = \alpha(R)\, APT(x,y,z,S,R,\sigma(R)) \quad (6)$$

where $\alpha(R) = 1/4\pi R^2$ and $\sigma(R) = c \cdot R \quad (0 < c < 1)$.

A discretized MAPT is easily computed, for selected values of R, on a voxelized grid including the surface mesh, with the procedure described in [GL12]. The map is computed in a grid of voxels with side s on a set of corresponding sampled radius value $R_1,\ldots,R_n$.

In this track, discrete MAPT maps were quantized in 12 bins and histograms computed at different scales (radii) considered were concatenated creating an unique descriptor. Voxel side and sampled radii were chosen differently for each model and proportional to the cubic root of the object volume, in order to have the same descriptor for scaled versions of the same geometry. In our test we computed the map at 12 different radii, using the same parameters for radius step discretization and voxelization applied in [GL12], e.g. voxel sixe $s = cbrt(volume)/25$, first radius equal to $2s$, step between radii equal to s, $c = 0.5$.

We however submitted 3 different distance matrices obtained from the same MAPT computation, obtained as follows: the first one by concatenating all the histograms, then measuring the resulting histogram distances with the Jeffreys' divergence [Jef46]. The second one by concatenating all the histograms but the first (computed at the finer scale, assumed to be possibly noisy), then measuring histogram distances with the Jeffreys divergence. The third one was obtained by concatenating the histograms computed at the first eight scales, in order to have exactly the same parameters used in the original paper [GL12] in the experiments done on SHREC'11 Nonrigid dataset [LGB*11].

The MAPT/histograms extraction was performed using the a C++ implementation of the method available at http://www.andreagiachetti.it. The computation time of a map depends on the model features, on the discretization and on the radius range, and was 3 min on average for the tested dataset on a Dell XPS L-701 X laptop (Intel Corei7 CPU Q740) running Ubuntu Linux. A single query takes around 1.2 msec. using a Matlab implementation of the Jeffrey divergence distance.

5.5. SPH-SPEM-VLAD: Screened Poisson Hyperfield features with Sparse Coding and VLAD, by R.A. Guler, S. Tari and G. Unal

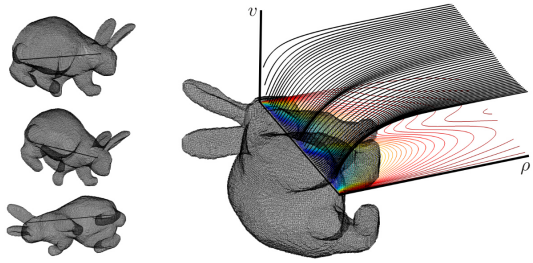

Figure 5: ***Left:*** *Various poses of the bunny and a line inside it.* ***Right:*** *The rows of Y for the nodes that are on the line visualized as isocontours and line plots.*

The screened Poisson hyperfield [GTU14] provides a volumetric representation technique where the shape information is encoded inside the shape by exploiting internal distance relationships via the screened Poisson PDE:

$$\begin{aligned} \Delta v^{\rho_i}(\mathbf{x}) - \frac{v^{\rho_i}(\mathbf{x})}{{\rho_i}^2} &= 0 \\ v^{\rho_i}(\mathbf{x})|_{\mathbf{x}\in\partial\Omega} &= 1. \end{aligned} \quad (7)$$

The screened Poisson hyperfield(SPH), solution to (7), can be considered as continuum of shape fields that sweeps the ρ^2 dimension for each node $\mathbf{x} \in \Omega$ on the lattice on which the shape is described. The hyperfield $v^{\rho_i}_{i=1\ldots N}$ is obtained by varying the screening parameter ρ^2 .

Considering ρ as an additional dimension gives rise to a two-dimensional scale-space where smoothing increases with a decrease in v and an increase in ρ^2. By concatenating the spatial dimensions, the hyperfield can be represented as a 2D matrix $Y_{|\Omega|\times N}$, where each row is a 1D curve(see

Figure 5) describing the behavior of $v(\mathbf{x})$ in ρ-dimension:

$$Y_{|\Omega|\times N} = \begin{bmatrix} v^{\rho_1}(x_1) & v^{\rho_2}(x_1) & \dots & v^{\rho_N}(x_1) \\ v^{\rho_1}(x_2) & v^{\rho_2}(x_2) & \dots & v^{\rho_N}(x_2) \\ \vdots & \vdots & & \vdots \\ v^{\rho_1}(x_{|\Omega|}) & v^{\rho_2}(x_{|\Omega|}) & \dots & v^{\rho_N}(x_{|\Omega|}) \end{bmatrix}. \quad (8)$$

5.5.1. Screened Poisson Encoding Maps + VLAD

The covariance matrix of Y is decomposed to yield an orthogonal set of bases: the eigen maps $\Phi_j, j = 1,...6$ of the field: $Y^T Y = \Phi\Lambda\Phi$. After the new bases are calculated, the field can be projected to form 6 mappings, where each mapping $\mathcal{P}_j$, is related to a measure of the variance explained by the j_{th} basis: $\mathcal{P}_j = Y\Phi_j$. Two significant properties of SPEM-Screened Poisson Encoding Maps, $\mathcal{P}$, are robustness to deformation and adaptation to scale. The eigenvectors Φ for each shape adapts in ρ domain such that the representation is unaltered in case of global scale change. For details regarding SPEM, the reader is referred to [GTU14].

Vector of Locally Aggregated Descriptors (VLAD) [JDSP10] characterizes the distribution of vectors with respect to the pre-computed centers. The difference vectors from each feature to assigned center are aggregated.

First, a codebook, $\mathcal{C} = \{\mathbf{c}_1,,\dots,\mathbf{c}_k\}$, of k representative points $\mathbf{c}_i \in \mathbb{R}^6$ is calculated using k-means clustering algorithm. Each node in the shape domain is assigned to the nearest cluster center, as a hard vector quantization to obtain $\mathcal{NN} \in \mathbb{R}^6$.

The differences $\mathcal{P}(\mathbf{x}) - \mathbf{c}_i$ of the vectors assigned to each center, $\mathbf{c}_i$, are accumulated to obtain *residual* sum vectors $\mathbf{R}_i \in \mathbb{R}^6$:

$$\mathbf{R}_i = \sum_{\mathbf{x}\in\Omega:\, \mathcal{NN}_{(\mathbf{x})}=\mathbf{c}_i} \mathcal{P}(\mathbf{x}) - \mathbf{c}_i. \quad (9)$$

To form the final descriptor, the aggregated residual vectors $\mathbf{R}_i$ for all centers are concatenated and normalized so that they have unit L_2 norm.

5.5.2. Sparse Coding

Unlike SPEM+VLAD, the sparse coding approach utilizes the values of the hyperfield, v^ρ, directly. The rich shape information characterized by measurements for each voxel, rows of Y, are analyzed in their original dimensionality N, albeit the loss of global scale adaptation. First, a dictionary $D \in \mathbb{R}^{N\times d}$ is trained using measurements of uniformly sampled shape voxels in an unsupervised manner, minimizing elastic-net [ZH05] cost function (10). The hyperfield for each shape is represented using S, again solved using (10), whose columns are the contribution of each atom in D to reconstruct the measurements. Columns of S are sum pooled to obtain a histogram which is used as the descriptor.

$$\min_{D_{N\times d}, S_{d\times|\Omega|}} \frac{1}{2}\|Y^T - DS\|_2^2 + \lambda\|S\|_1 + \lambda_2\|S\|_F^2 \quad (10)$$

5.5.3. Implementation

For the retrieval problem each shape is represented on a lattice with a fixed volume of 250.000 voxels. The screened Poisson operator is represented as a sparse matrix: $P_{|\Omega|\times|\Omega|} = (\Delta - \frac{1}{\rho^2}I)\chi_{(B.C.)}$, where Δ is the seven-point discretization of the Laplace operator and $\chi_{(B.C.)_{|\Omega|\times|\Omega|}} \rightarrow \{0,1\}$ is an indicator function for the edges that are allowed by the desired boundary condition. The gradient descent solution to the screened Poisson field is then obtained by iterating the following multiplication: $v^{n+1} = v^n(I + \tau P)$, where τ is the artificial time step. The sparse matrix vector multiplication is done using CUSP library [BG12]. The unsupervised dictionary learning and pursuit problem is solved using SPAMS toolbox [MBPS10].

Both VLAD and sum pooled sparse codes are power normalized (see [GTU14].) by taking their square-root. The L_1 distance between the descriptors of two shapes is used as the dissimilarity measure. The results for SPEM+VLAD are presented for k=32,64 resulting in descriptors of length 192 and 384. The results for the sparse coding approach are obtained using dictionaries that contain 256 and 1024 atoms.

5.6. Multi-Feature: Multi-feature Descriptor, by L. Lai and L. Sun

Large numbers of feature descriptors of non-rigid 3D models have been proposed, such as Heat Kernel Signature (HKS) [SOL09], Wave Kernel Signature (WKS) [ASC11a] and Surface Area (SA) [PSea14] of the Model. However, the single feature descriptor has limitations when corresponding between non-rigid shapes. The entropy is introduced to calculate the weight according to the characteristics of HKS, WKS and SA for constructing multi-feature. Multi-feature will be used to improve the accuracy when correspondence.

We have submitted 4 runs. The HKS, WKS and SA are traditional methods for non-rigid 3D model. Here, we fuse f_1, f_2 and f_3 (representing the HKS, WKS and SA respectively) to construct the multi-feature descriptor.

(1) Select m models from SHREC'11 non-rigid dataset, marked as dataset A. The models which are not included in A are marked as dataset T. Pick a model q from T. Calculate the similarity between q and the models in A based on different features and sort the results. We just take the first k similar models. We call it $R^i{}_{qk}$ (i=1,2,3).

(2) Classify the models in $R^i{}_{qk}$ and calculate the probability distribution of different categories in $R^i{}_{qk}$. $\left\{p_1{}^i, p_2{}^i, ..., p_N{}^i\right\}$ is the probability distribution of f_i with N categories.

(3) According to the definition of information entropy, the formula (11) can be used to get the entropy of each feature,

$$E(f_i) = -\sum_{j=1}^{n} p_j{}^i log_2 p_j{}^i \quad (11)$$

Then, we get the weight of each feature by,

$$W_i = \frac{1-E(f_i)}{3-\sum E(f_i)}, i = 1,2,3 \quad (12)$$

where W_1, W_2 and W_3 are the weights of f_1 , f_2 and f_3 respectively.

(4) Extract the feature vectors of each model in SHREC'15 dataset. Calculate similarity distance matrix ${W^1}_D$, ${W^2}_D$ and ${W^3}_D$ for f_1 , f_2 and f_3. The weights ${W^1}_D$, ${W^2}_D$ and ${W^3}_D$ are also normalized.

(5) Calculate the final similarity distance matrix of multi-feature ${M_D}^{multi}$ according to formula (13).

$${M_D}^{multi} = \sum_{i=1}^{3} {M^i}_D W_i \quad (13)$$

5.7. SG: Spectral Geometry, by C. Li

We use the framework in [Li13] for nonrigid shape representation and retrieval. It consists of two main stages: (1) spectral graph wavelet signature [LH13b] for descriptor extraction, and (2) intrinsic spatial pyramid matching [LH13a] for shape comparison. The cotangent weight scheme was used to discretize Laplace Beltrami Operator (LBO), with the eigenvalues λ_i and associated eigenfunctions φ_i, and $m = 200$.

Spectral graph wavelet signature: The first stage is the computation of spectral descriptor $h(x)$ at each vertex of the triangle meshed shape X. In general, any one of spectral descriptors with the eigenfunction-squared form reviewed in [LH13c] can be used for isometric invariant representation. We used the recently proposed spectral graph wavelet signature (SGWS) as the local descriptor; it provided a general and flexible interpretation for the analysis and design of spectral descriptors $S_x(t,x) = \sum_{i=1}^{m} g(t,\lambda_i)\varphi_i^2(x)$. In particular, a multi-resolution shape descriptor was obtained by setting $g(t,\lambda_i)$ as a cubic spline wavelet generating kernel and considering the scaling function. The resolution level is set as 2.

Intrinsic spatial pyramid matching: Given a vocabulary $P = \{p_k\}_{k=1}^{K}, K = 100$, learned by k-means, the dense descriptor $S = \{s_t\}_{t=1}^{T}$ at each vertex of the shape is replaced by the hard assignment $Q = \{q_k\}_{k=1}^{K}$.

Any function f on X can be written as the linear combination of the eigenfunctions. Using the variational characterizations of the eigenvalues in terms of the Rayleigh-Ritz quotient, the second eigenvalue is given by

$$\lambda_2 = \inf_{f \perp \varphi_1} \frac{f'Cf}{f'Af} \quad (14)$$

We use the isocontours of the second eigenfunctionto cut the shape into $\mathtt{R} = 2^{l-1}$ patches, thus the shape description is the concatenation of R sub-histograms of Q along eigenfunction value in the real line. To consider the two-sign possibilities in the concatenation, we invert the histogram order, and consider the scheme with the minimum cost as a better matching. The second eigenfunction is the smoothest mapping from the manifold to th real line, resulting in this intrinsic partition quite stable. It provably extends the property of popular SPM in image domain to capture spatial information for meshed surfaces, so is referred as intrinsic spatial pyramid matching (ISPM) in [LH13a]. The partition number is set as $l = 2,\cdots,6$ in this track. These ISPM induced histograms are used for shape comparison.

5.8. FVF: Fisher Vector Encoding Framework, by F. Limberger and R. Wilson

The method uses a statistical and discriminative framework (Fisher Vector) which combines local spectral signatures that describe local characteristics of the shape into a global descriptor for shape retrieval. Summarizing, this method computes local descriptors for each model in the database, and then it computes a dictionary by selecting a subset of models and estimating a Gaussian Mixture Model (GMM) for each layer of the concatenated descriptors from the subset. After that, each shape signature is encoded to a global representation of the shape using Fisher Vector (FV) [JH98]. This step results in a high dimensional vector that describes each shape according to its deviation from the dictionary. This vector is used as a global signature of the shape being effective for retrieval and classification tasks.

5.8.1. Local Descriptor

Our approach was tested with three different local descriptors: the Heat Kernel Signature (HKS), the scale-invariant Heat Kernel Signature (SI-HKS) and the Wave Kernel Signature (WKS). The HKS [SOL09] estimates the behaviour of the heat diffusion over the surface of a model while the SI-HKS [BK10] is its scale-invariant version. The WKS [ASC11b] represents the average probability of measuring a quantum mechanical particle at shape vertices and it is scale invariant in the same way as the SI-HKS. In our framework we interpret each frequency of the local descriptors as a layer that describes the entire shape.

For the computation of the local signatures we use the cotangent scheme to compute the first 300 eigenvalues of the Laplace-Beltrami operator (LBO). We evaluate the HKS and SI-HKS at the time interval $[4\ln(10)/\lambda_{300}, 4\ln(10)/\lambda_2]$ logarithmic scaled and the WKS at the energy interval $[\log(\lambda_2), \log(\lambda_{300})]$, where λ_i corresponds to the ith eigenvalue of the LBO. The increments were chosen accordingly to sample 100 points in the interval. In the SI-HKS, the first 6 discrete lowest frequencies were used to represent the shape after scale normalization.

5.8.2. Fisher Vector

Fisher Vector encodes a set of local descriptors into a high dimensional array which corresponds to the direction that the local descriptors should be modified to fit the parameters of a model. To compute the FV we estimate the parameters of a GMM by using the Expectation Maximization algorithm to optimize a Maximum Likelihood criterion. Then, we write the local descriptors $X = \{x_t, x_t \in \mathbb{R}^D, t = 1...T\}$ wrt. the GMM u_λ, which means expressing X by its gradient in respect to u_λ with parameters $\lambda = \{w_k, \mu_k, \Sigma_k, k = 1...K\}$.

This is done by computing the association strength (soft assignment) of each vector x_t to a mode k in the GMM, which is given by the posterior probability [PSM10]

$$q_{tk} = \frac{exp[-\frac{1}{2}(x_t-\mu_k)^T\Sigma_k^{-1}(x_t-\mu_k)]}{\Sigma_{i=1}^K exp[-\frac{1}{2}(x_t-\mu_i)^T\Sigma_k^{-1}(x_t-\mu_i)]}. \quad (15)$$

For each mode k and each descriptor dimension $j = 1..D$, we compute the deviation vectors (gradient) with respect to the mean and covariance, respectively

$$u_{jk} = \frac{1}{T\sqrt{w_k}} \sum_{i=1}^{T} q_{ik} \frac{x_{ji}-\mu_{jk}}{\sigma_{jk}}, \quad (16)$$

$$v_{jk} = \frac{1}{T\sqrt{2w_k}} \sum_{i=1}^{T} q_{ik} \left[\left(\frac{x_{ji}-\mu_{jk}}{\sigma_{jk}} \right)^2 - 1 \right]. \quad (17)$$

Finally, the FV representation of a shape S is the concatenation of the vectorization of the matrices u_{jk} and v_{jk}

$$\Gamma_\lambda^X(S) = \left[\dots u_k \dots v_k \dots \right]^T. \quad (18)$$

We define the distance between two shapes R and S as the ℓ_1 norm between Fisher Vectors:

$$d_{FV} = ||\Gamma_\lambda^{X_1}(R) - \Gamma_\lambda^{X_2}(S)||_1. \quad (19)$$

All experiments were performed in Matlab in a Windows PC (Intel Core i7 3.4GHz, 8GB RAM), taking around 6 hours to compute the three dissimilarity matrices. For the local descriptors we have used the original codes from the authors. We compute a dictionary using the first 29 models of the database and estimate a GMM with 38 Gaussians for each layer. For models with 10K vertices, the three local descriptors are computed in around 18 seconds. The creation of the dictionary using 29 models takes around 50 seconds. Writing each local descriptor using the FV framework takes on average 0.3 seconds per model.

5.9. TSASR: Time series analysis for shape retrieval, by R.U. Nakanishi, A. Paiva and L.G. Nonato

The key idea of the proposed method is to represent the feature vector of a 3D model as a time series for shape retrieval. As illustrated in Figure 6, the proposed method comprises three main steps: vertex sorting, time series generation and scattering coefficients of the time series. In the first step, given an input triangle mesh, we need to sample the vertices as a sorted vector that inflicts a spatial dependency to their features. In general, the feature is invariant to isometry, such as Gaussian and mean curvatures. The mean curvature is usually approximated in point-set surfaces using the *surface variation* [PKG03], which performs PCA and KNN algorithm in each point. In triangle meshes, the vertex neighbors is easily computed using vertex-rings. After computing the feature for each vertex, the vertex sorting stage is obtained using the *Fiedler vector* [LZ10], i.e., the second smallest eigenvector of discrete Laplace-Beltrami matrix. The Fiedler vector provides the algebraic connectivity of the surface mesh which can be used to guide the vertex sorting. Figure 6 shows the vertex sorting of the hand model from low (blue) to high (red) indexes. At second step, the time series of the feature vector is simply created replacing the time intervals by the sorted indexes (sorted index $\times$ feature). Finally, for each model is extracted the *coefficients* [BM11] of its corresponding time series. The coefficients are acquired using the *scattering wavelets technique* [AM12], this technique relies on multiscale co-occurrence of coefficients with wavelets filter banks on windowed intervals of the time series. Moreover, the scattering wavelets is invariant under translations and stable to small signal deformations, which are well suited properties to find similar structures in digital signals (e.g., audio and music data [CR13]).

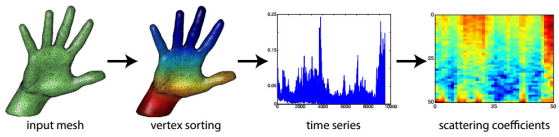

Figure 6: *Pipeline of the TSASR based shape retrieval framework.*

In order to compute the similarity between the 3D models, we apply the *cross correlation method* [Cha03] in the time series that represent the models, in other words, this method measures the similarity between coefficients sets. Given two models (coefficients) S and T, we defined a distance metric by:

$$dist(S,T) = 1 - cc(S,T),$$

where $cc(S,T)$ is the normalized cross correlation between S and T.

In our experiments, we use the symmetric cotangent weights to approximate the Laplace-Beltrami matrix [LZ10] and a vertex smoothing [BK04] to improve the results. We adopt 3-ring of each vertex as vertex neighborhood for the surface variation estimation. The scattering wavelets window size are set to 256, 512 and 1024.

5.10. SID: Sphere Intersection Descriptor, by K. Pevzner, A. Sharf and J. El-Sana

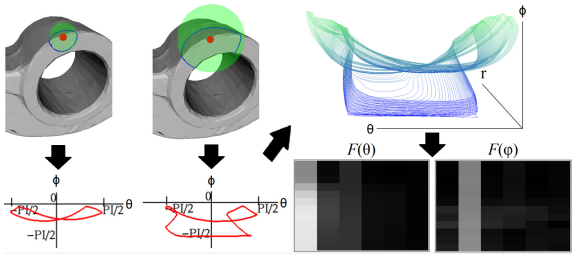

Figure 7: *Overview of Sphere-Intersection Descriptor (SID) for a single feature. Top-left: two sphere intersections with different radii. Bottom-left: corresponding parameterized intersection curves. Top-right: multiple intersections aggregated. Bottom-right: frequency domain transformation yielding a compact 2D image representation.*

In essence, this method tracks spheres' evolutions and aggregate intersection curves together into one descriptor. By transforming to frequency domain, we obtain a 2D image representation of the local surface geometry sampling.

5.10.1. Feature Detection

Feature point detection is done in an iterative manner. We start with all mesh vertices as candidates for being considered feature points and reduce the amount of candidates in each iteration. At each step of the feature detection loop we perform the following operations for each candidate:

1. Increase the spheres' radius.
2. Sphere-with-surface intersection curve extraction.
3. Transformation of the intersection curve to spherical coordinations.
4. Parametrization of the intersection curve as $(\phi(t),\theta(t))$.
5. Computation of saliency metric.

At the end of each iteration we consider the intersection of neighboring spheres and upon sphere-to-sphere intersection we only keep the one with the ones with the stronger saliency metric. Thus, the algorithm converges from all mesh vertices to keypoints.

5.10.2. Saliency Metric

Given a vertex v let $\theta_v(t)$, $\phi_v(t)$ be defined for $t \in 1,\ldots,n$ then the *integral over the intersection curve* of v is defined by:

$$i_v = \sum_{t=1}^{n-1} \frac{\|\theta_v(t+1)-\theta_v(t)\| \cdot (\phi_v(t+1)+\phi_v(t))}{2} \qquad (20)$$

This formulation gives high saliency values to features at local extrema.

5.10.3. Feature Description

Given a keypoint and a support sphere radius from the previous step, the sphere intersection curve with the surface is projected to spherical coordinates and parameterized to $\phi(t),\theta(t)$ components. Lastly, $\phi(t),\theta(t)$ are transformed to frequency domain thus producing rotational invariant local feature descriptor. Figure 7 demonstrates the sphere intersection with the mesh, the resulting curve on the (ϕ,θ) plane and the transformation to frequency domain.

5.10.4. Shape Similarity

For similarity computation between two meshes, we perform a min cost maximal matching on a bipartite graph generated from the set of descriptors of each mesh. Similarity between two given feature descriptors is given by L_1 distance of their magnitudes. Thus, model correspondence is performed by minimizing the cost of the matching on the feature descriptors sets.

5.11. EDBCF: Euclidean Distance Based Canonical Forms, by D. Pickup, X. Sun, P.L. Rosin and R.R. Martin

We compute a canonical form of a mesh by stretching out its limbs so that its extremities are distant from one another. We achieve this effect more efficiently than the common method of computing multidimensional scaling (MDS) on the geodesic distances [EK03]. We instead maximise the Euclidean distances between feature points on the extremities of the mesh, whilst preserving the original edge lengths to ensure isometric deformation.

We first scale the mesh so that the maximum distance of any point on its surface to the centroid of all vertices is one. We then use the method of Ben-Chen and Gotsman [BCG08] to calculate the conformal factor of the mesh. The conformal factor increases along the length of mesh protrusions, which results in high values at the extremities of the mesh. To obtain a set of feature points for a mesh with N vertices, we sample the $\sqrt{N}$ vertices which have the largest conformal factors and also satisfy the requirement that they are local maxima. A vertex is defined to be a local maximum if its conformal factor is greater than that of all its neighbours in a 2-ring neighbourhood. We select $\sqrt{N}$ feature points, as this is the largest number of features we can have whilst being able to compute the Euclidean distances between all pairs of feature points in linear time (with respect to the number of mesh vertices).

We compute the canonical form of the mesh by adapting the least-squares MDS formulation used by [EK03]

$$S(X) = \sum_{i=1}^{N} \sum_{j=i+1}^{N} w_{i,j}(\delta_{i,j} - d_{i,j}(X))^2, \qquad (21)$$

where N is the number of vertices, $w_{i,j}$ are weighting coefficients, and $d_{i,j}$ is the Euclidean distance between vertices

i and j of the resulting canonical mesh X. We set the value of $\delta_{i,j}$ for all connected vertices i and j equal to the length of the edge connecting them. This aims to preserve the edge lengths of the mesh, to ensure isometric deformation. In order to maximise the distance between feature points, the value of $\delta_{i,j}$ for each pair of the $\sqrt{N}$ sampled vertices is set to a high value α. We want this value to be large enough to straighten all the limbs of the model, and our experiments show 10 is large enough. As long as α is large enough and the parameter β discussed below is optimised accordingly, any value of α can be chosen.

If the two vertices i and j are neither a pair of feature points nor connected by an edge, we do not enforce a target distance between them, so $\delta_{i,j}$ and $w_{i,j}$ are both set to zero for such cases. Not having to compute and optimise the distances between these points is crucial in keeping the linear time complexity of our distance calculations. The weights $w_{i,j}$ in Equation (21) for all i and j that are connected by an edge are set to $\beta/\delta_{i,j}^2$, where β is a user defined parameter for preserving edge lengths. We divide by the square of the edge length $\delta_{i,j}^2$ so that the distance in Equation (21) becomes a relative, rather than absolute, difference, making the weighting independent of the length of the edge. The conformal factor is normalised to lie in the interval $[0,1]$, and the entries in the weighting matrix $w_{i,j}$ for each pair of feature points are set to the mean of their conformal factors. This results in vertices which are nearer the ends of long 'limbs' of the object having a higher impact on the resulting canonical form, and avoids stretching out inappropriate parts of the mesh. The SMACOF algorithm can then be used to minimise Equation (21) as described in [EK03].

In many cases the number of local maxima of conformal factor is less than $\sqrt{N}$. We want the number of feature points to be exactly $\sqrt{N}$ so that the number of edges connecting pairs of feature points grows at the same rate as the number of mesh vertices. This in turn ensures that we can use the same value for the parameter β regardless of mesh resolution. We offer two different solutions to handling this issue. The first is to increase the number of feature points to $\sqrt{N}$ by adding extra randomly selected vertices as feature points.

The second is to separately normalise the weightings $w_{i,j}$ used for pairs of feature points, and for adjacent vertices. We normalise the weights for adjacent vertices by dividing by the total number of edges, and we normalise the feature point pair weights by dividing by the sum of all feature point pair weights. Thus, we may rewrite the final functional to be minimised as

$$S(X) = \sum_{i\in F}\sum_{j\in F, j\neq i}\frac{0.5(\Phi_i+\Phi_j)}{\sum_{i\in F}\sum_{j\in F,j\neq i}0.5(\Phi_i+\Phi_j)}(\delta_{i,j}-d_{i,j}(X))^2 + \sum_{(i,j)\in E}\frac{\beta}{|E|\delta_{i,j}^2}(\delta_{i,j}-d_{i,j}(X))^2, \quad (22)$$

Table 1: *Retrieval performance of all runs evaluated using five quantitative measures on the whole database.*

PARTICIPANT	METHOD	NN	FT	ST	E	DCG
Choi	SNU_1	0.8983	0.5638	0.6690	0.5167	0.8326
	SNU_2	0.8992	0.5657	0.6706	0.5181	0.8335
ElNaghy	CompactBoFHKS-4D	0.9817	0.8722	0.9080	0.7476	0.9581
	CompactBoFHKS-5D	0.9775	0.8582	0.8988	0.7356	0.9533
	CompactBoFHKS-10D	0.9842	0.8714	0.9082	0.7465	0.9582
	CompactBoFHKS-19D	0.9825	0.8672	0.9059	0.7440	0.9575
Furuya	SV-LSF	1.0000	0.9797	0.9974	0.8292	0.9977
	SV-LSF_kpca50	1.0000	0.9972	0.9997	0.8357	0.9997
Giachetti	HAPT_run1	0.9983	0.9657	0.9821	0.8150	0.9919
	HAPT_run2	0.9975	0.9658	0.9818	0.8150	0.9918
	HAPT_run3	1.0000	0.9613	0.9795	0.8117	0.9915
Guler	SPH_SparseCoding_256	0.9967	0.9510	0.9662	0.8008	0.9868
	SPH_SparseCoding_1024	0.9975	0.9568	0.9696	0.8047	0.9885
	SPH_SPEM_VLAD_32	0.9942	0.9106	0.9460	0.7778	0.9757
	SPH_SPEM_VLAD_64	0.9958	0.9343	0.9570	0.7911	0.9811
Lai	HKS	0.0650	0.0637	0.1245	0.0745	0.3911
	SA	0.0650	0.0677	0.1282	0.0788	0.3936
	WKS	0.1342	0.0744	0.1374	0.0833	0.4089
	Multi-Feature	0.4508	0.1864	0.2625	0.1846	0.5259
Li	SG_L1	0.9725	0.7596	0.8143	0.6597	0.9192
	SG_L2	0.9683	0.7400	0.7945	0.6428	0.9104
	SG_L3	0.9550	0.7067	0.7661	0.6167	0.8949
	SG_L4	0.9475	0.6910	0.7561	0.6065	0.8859
	SG_L5	0.9367	0.6681	0.7364	0.5879	0.8750
Limberger	FVF-HKS	0.9608	0.7254	0.8093	0.6446	0.9136
	FVF-SIHKS	0.9800	0.8249	0.8826	0.7178	0.9503
	FVF-WKS	0.9767	0.8225	0.8945	0.7242	0.9518
Nakanishi	TSASR256	0.8133	0.4638	0.5445	0.4203	0.7498
	TSASR512	0.8608	0.4758	0.5498	0.4292	0.7673
	TSASR1024	0.8958	0.5316	0.5960	0.4722	0.7975
Pevzner	SID-1	0.7958	0.4841	0.6141	0.4596	0.7781
	SID-2	0.9725	0.6811	0.7853	0.6157	0.9031
	SID-3	0.9683	0.6796	0.7838	0.6142	0.9039
	SID-4	0.9767	0.7188	0.8213	0.6482	0.9200
	SID-5	0.9767	0.7109	0.8164	0.6422	0.9171
Pickup	EDBCF_AV	0.9750	0.7699	0.8680	0.6899	0.9358
	EDBCF_NW	0.9775	0.7931	0.8839	0.7076	0.9431

where F is the set of all feature points, E is the set of all edges, and Φ_i is the conformal factor of vertex i.

To produce the retrieval results we use the view based method described in [LGSX13]. We have submitted results produced using both variants of our canonical forms. More details about this method can be found in [PSRM15].

6. Results

In this section, we present and compare the results of 37 runs submitted by 11 different groups. Given the 37 dissimilarity matrices, evaluations are carried out for these methods on the average performance of the whole database. This track employs six commonly-used evaluation measures including the five quantitative statistics *(i.e. NN, FT, ST, E, and DCG)* and the *Precision-recall curves* mentioned in Section 3.

Table 1 shows the retrieval accuracies of all 37 runs evaluated on the whole database. We can see that most of these methods perform quite well in this track. For instance, NN values of 25 runs are greater than 0.950 and 3 runs obtain NN values that are equal to 100%, which means that some of these methods are able to perfectly retrieve correct

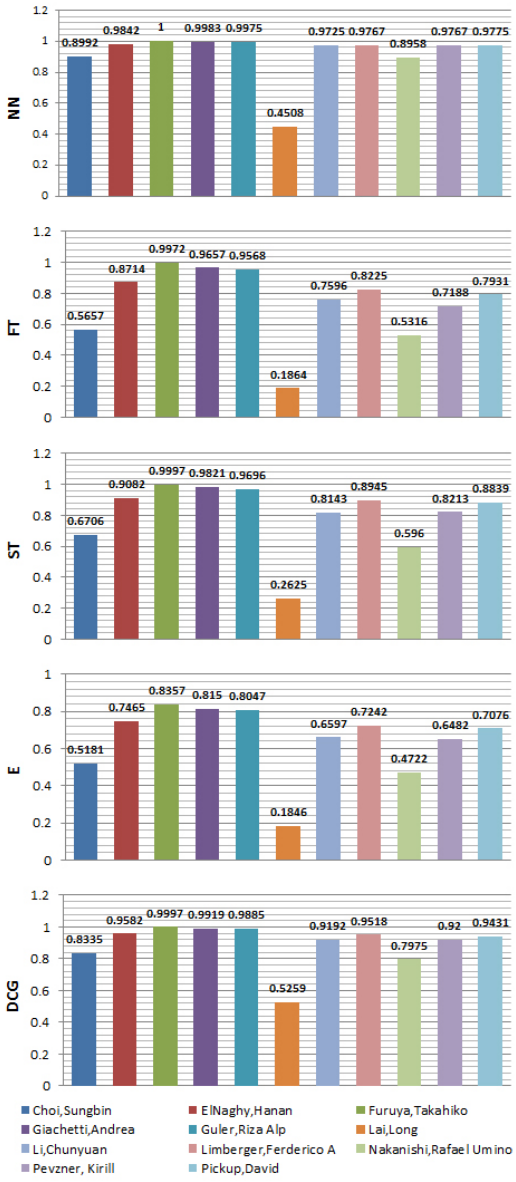

Figure 8: *Column charts of the best retrieval accuracies of each group evaluated on the whole database using five standard measures, respectively.*

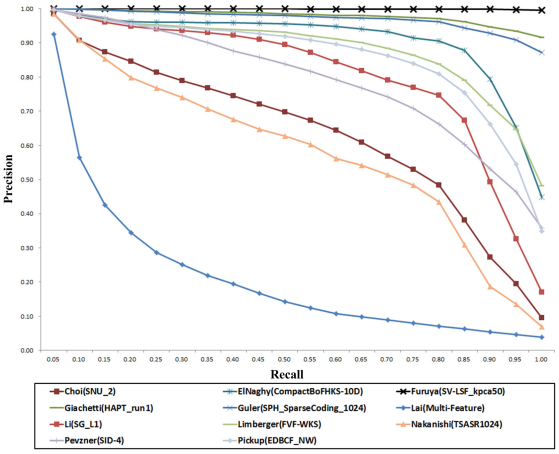

Figure 9: *Precision-recall curves of the best runs of each group evaluated for the whole database.*

nearest neighbors for any queries. Furthermore, 24 runs' DCG values (well known as the most stable retrieval measure for performance evaluation) are larger than 0.900 and 5 runs have DCG values that are above 0.990. Somehow surprisingly, Furuya's SV-LSF_kpca50 obtains almost perfect retrieval accuracies (e.g., $NN = 1.0000$, $FT = 0.9972$, $ST = 0.9997$ and $DCG = 0.9997$) on the database utilized in this track. The method not only clearly outperforms all other methods but also obtains much better retrieval accuracies compared to the best approach (i.e., Smeets's SD-GDM-meshSIFT) in SHREC'11 non-rigid track [LGB*11]. What is more, considering that 16.7% models in each category have different topological structures against other objects in this year's database while models in the same category of the non-rigid database used in SHREC'11 do not have topological differences, Furuya's SV-LSF_kpca50 method indeed possesses excellent capabilities in representing pose-invariant features for non-rigid 3D models.

In Figure 8, column charts of the five quantitative measures are also provided to intuitively compare the best results of each group. Clearly, as mentioned before, Furuya's method performs best in this track. Also obviously, Giachetti's HAPT method takes the second place considering all evaluation measures and Guler's SPH_SparseCoding approach performs slightly worse than Giachetti's method. It is worthy of mentioning that, despite of its simplicity and compactness, ElNaghy's CompactBoFHKS also performs quite well and undoubtedly takes the fourth place among all 11 groups. The same observations can be made from Figure 9 that depicts Precision-recall curves of the best runs submitted by each group on the whole database.

Based on the analyses in [LGX13], existing non-rigid 3D shape retrieval methods can be roughly classified into algorithms employing local features, topological structures, isometry-invariant global geometric properties, direct shape matching, or canonical forms. Mainly due to the computing complexity, no participant utilizes the method involving topological information. We find that using local features to generate isometry-invariant shape descriptors is still the most popular way (18 runs) to achieve effective non-rigid 3D shape retrieval. Carefully designing of local features and coding strategies (e.g., Furuya's SV-LSF) can lead to surprisingly good retrieval accuracies on this database. The second

most popular kind of method (9 runs including Nakanishi's TSASR) is to exploit the isometry-invariant global properties of 3D meshes. There are also methods that use direct shape matching on the set of descriptors (i.e., Pevzner's SID) or utilize 3D canonical forms (i.e., Pickup's EDBCF). In this track, we witness a number of new ideas in non-rigid 3D shape retrieval. For example, Nakanishi's TSASR represents the feature vector of a 3D model as a time series and Guler's SPH-SPEM-VLAD adopts the sparse coding technique which has been widely used in 2D domain. We also observe that traditional approaches including geodesic distance distribution (i.e., Choi's SNU), Lai's HKS, SA and WKS are not competitive against other currently-proposed methods. However, performance of those classic approaches can be significantly improved by integrating with other techniques. For example, Limberger's methods (i.e., FVF-HKS, FV-SIHKS and FVF-WKS) result in much better performance than the original HKS, SI and WKS algorithms via the utilization of the Fisher vector encoding framework.

For more information about this track, please refer to the track's official website [PKU15] in which the new database and the corresponding evaluation code are also freely-available for academic use.

7. Conclusion

In this paper, we first analyzed and discussed the background of non-rigid 3D shape retrieval. Then, we described how to build the new database and how to carry out performance evaluations. Afterwards, we briefly introduced all methods (37 runs) employed by 11 groups who successfully participated in this track. Finally, we presented experimental results to compare the retrieval performance of different methods.

With growing interests in non-rigid 3D shape retrieval, the demands of developing more challenging non-rigid 3D shape benchmarks have also markedly increased. Compared to previous two SHREC non-rigid tracks and other non-rigid 3D shape benchmarks, the *SHREC'15 Track: Non-rigid 3D Shape Retrieval* provides a large-scale database consisting of more 3D watertight meshes with more complicated shape variations. Furthermore, the track has attracted a large number of participants (11 groups and 37 runs) who are currently active in the community and thus methods described in this paper most likely represent the state of the art in non-rigid 3D shape retrieval. We believe that the organization of this track will further promote the investigation of this important research topic.

Several promising research directions in non-rigid 3D shape retrieval are listed as follows: 1) Build a new benchmark database containing an enormous amount of non-rigid 3D models that have different and complex topological structures. 2) Develop preprocessing algorithms to automatically generate watertight manifolds from ordinary 3D models that may possess geometric noises or/and inner structures. 3) Employ state-of-the-art techniques (e.g., Deep Learning, Sparse Coding, etc.) emerged in computer vision, artificial intelligent and image processing to enhance the performance of non-rigid 3D shape retrieval methods. 4) Design more discriminative local/global 3D shape descriptors and create 3D canonical forms with more details preserved.

ACKNOWLEDGMENTS

This work was supported by National Natural Science Foundation of China (Grant No.: 61202230 and 61472015), National Hi-Tech Research and Development Program (863 Program) of China (Grant No.: 2014AA015102) and Beijing Natural Science Foundation (Grant No.: 4152022).

References

[AM12] ANDÉN J., MALLAT S.: Scattering representation of modulated sounds. In *15th DAFx* (2012). 9

[ASC11a] AUBRY M., SCHLICKEWEI U., CREMERS D.: Image classification using super-vector coding of local image descriptors. In *2011 IEEE International Conference on Computer Vision Workshops (ICCV Workshops)* (2011), pp. 1626– 1633. 7

[ASC11b] AUBRY M., SCHLICKEWEI U., CREMERS D.: The wave kernel signature: A quantum mechanical approach to shape analysis. In *IEEE International Conference on Computer Vision Workshops (ICCV Workshops)* (2011), pp. 1626–1633. 8

[BBGO09] BROSTEIN A. M., BRONSTEIN M. M., GUIBAS L. J., OVSJANIKOV M.: Shape google: A computer vision approach to isometry invariant shape retrieval. In *2009 IEEE 12th International Conference on Computer Vision Workshops (ICCV Workshops)* (2009), pp. 320–327. 4

[BBK08] BRONSTEIN A. M., BRONSTEIN M. M., KIMMEL R.: *Numerical geometry of non-rigid shapes*. Springer, 2008. 2

[BBLO11] BROSTEIN A. M., BRONSTEIN M. M., LEONIDAS J. G., OVSJANIKOV M.: Shape google: Geometric words and expressions for invariant shape retrieval. *ACM Transactions on Graphics (TOG) 30* (2011). 4

[BCG08] BEN-CHEN M., GOTSMAN C.: Characterizing shape using conformal factors. In *Proceedings of the 1st Eurographics conference on 3D Object Retrieval* (2008), pp. 1–8. 10

[BG12] BELL N., GARLAND M.: Cusp: Generic parallel algorithms for sparse matrix and graph computations, 2012. Version 0.3.0. 7

[BK04] BOTSCH M., KOBBELT L.: A remeshing approach to multiresolution modeling. In *Symposium on Geometry Processing* (2004). 9

[BK10] BRONSTEIN M. M., KOKKINOS I.: Scale-invariant heat kernel signatures for non-rigid shape recognition. In *2010 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2010), pp. 1704–1711. 8

[BM11] BRUNA J., MALLAT S.: Classification with scattering operators. In *IEEE CVPR* (2011), p. 1561ÍC1566. 9

[CDF*04] CSURKA G., DANCE C. R., FAN L., WILLAMOWSKI J., BRAY C.: Visual categorization with bags of keypoints. In *ECCV 2004 workshop on Statistical Learning in Computer Vision* (2004), pp. 59–74. 5

[Cha03] CHATFIELD C.: *The analysis of time series: an introduction*. CRC, 2003. 9

[CR13] CHEN X., RAMADGE P. J.: Scattering representation of modulated sounds. In *47th CISS* (2013), pp. 1–6. 9

[EK03] ELAD A., KIMMEL R.: On bending invariant signatures for surfaces. *IEEE Transactions on Pattern Analysis and Machine Intelligence 25*, 10 (2003), 1285–1295. 10, 11

[FO14] FURUYA T., OHBUCHI R.: Fusing multiple features for shape-based 3d model retrieval. In *British Machine Vision Conference (BMVC) 2014* (2014). 5

[GL12] GIACHETTI A., LOVATO C.: Radial symmetry detection and shape characterization with the multiscale area projection transform. *Computer Graphics Forum 31*, 5 (2012), 1669–1678. 6

[Goo15] GOOGLE: Google 3d warehouse. *https://3dwarehouse.sketchup.com/* (2015). 2

[GTU14] GULER R. A., TARI S., UNAL G.: Screened poisson hyperfields for shape coding. *SIAM Journal on Imaging Sciences 7*, 4 (2014), 2558–2590. 6, 7

[JDSP10] JÉGOU H., DOUZE M., SCHMID C., PÉREZ P.: Aggregating local descriptors into a compact image representation. In *2010 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2010), pp. 3304–3311. 7

[Jef46] JEFFREYS H.: An invariant form for the prior probability in estimation problems. *Proceedings of the Royal Society of London. Series A. Mathematical and Physical Sciences 186*, 1007 (1946), 453–461. 6

[JH98] JAAKKOLA T. S., HAUSSLER D.: Exploiting generative models in discriminative classifiers. In *Proceedings of the 1998 Conference on Advances in Neural Information Processing Systems II* (1998), pp. 487–493. 8

[LGB*11] LIAN Z., GODIL A., BUSTOS B., DAOUDI M., HERMANS J., KAWAMURA S., KURITA Y., LAVOUÉ G., NGUYEN H. V., OHBUCHI R., OHKITA Y., OHISHI Y., PORIKLI F., REUTER M., SIPIRAN I., SMEETS D., SUETENS P., TABIA H., VANDERMEULEN D.: Shrec'11 track: Shape retrieval on non-rigid 3d watertight meshes. In *Proc. Eurographics Workshop on 3D Object Retrieval 2011 (3DOR 2011)* (2011), pp. 79–88. 2, 5, 6, 12

[LGF*10] LIAN Z., GODIL A., FABRY T., FURUYA T., HERMANS J., OHBUCHI R., SHU C., SMEETS D., SUETENS P., VANDERMEULEN D., WUHRER S.: SHREC'10 Track: Non-rigid 3D Shape Retrieval. In *Proc. 3DOR'10* (2010), pp. 101–108. 2

[LGSX13] LIAN Z., GODIL A., SUN X., XIAO J.: CM-BOF: visual similarity-based 3d shape retrieval using clock matching and bag-of-features. *Machine Vision and Applications 24*, 8 (2013), 1685–1704. 11

[LGX13] LIAN Z., GODIL A., XIAO J.: Feature-preserved 3d canonical form. *International journal of computer vision 102*, 1-3 (2013), 221–238. 12

[LH13a] LI C., HAMZA A. B.: Intrinsic spatial pyramid matching for deformable 3d shape retrieval. *International Journal of Multimedia Information Retrieval 2*, 4 (2013), 261–271. 8

[LH13b] LI C., HAMZA A. B.: A multiresolution descriptor for deformable 3d shape retrieval. *The Visual Computer* (2013), 1–12. 8

[LH13c] LI C., HAMZA A. B.: Spatially aggregating spectral descriptors for nonrigid 3d shape retrieval: a comparative survey. *Multimedia Systems* (2013), 1–29. 8

[Li13] LI C.: *Spectral Geometric Methods for Deformable 3D Shape Retrieval*. Master's thesis, Concordia University, 2013. 8

[LZ10] LÉVY B., ZHANG H. R.: Spectral mesh processing. In *ACM SIGGRAPH 2010 Courses* (2010), pp. 8:1–8:312. 9

[MBPS10] MAIRAL J., BACH F., PONCE J., SAPIRO G.: Online learning for matrix factorization and sparse coding. *The Journal of Machine Learning Research 11* (2010), 19–60. 7

[OFCD02] OSADA R., FUNKHOUSER T., CHAZELLE B., DOBKIN D.: Shape distributions. In *ACM Trans. on Graphics* (2002), pp. 807–832. 5

[OOFO12] OHKITA Y., OHISHI Y., FURUYA T., OHBUCHI R.: Non-rigid 3d model retrieval using set of local statistical features. In *Proc. IEEE ICME 2012 Workshop on Hot Topics in 3D Multimedia (Hot3D)* (2012), pp. 593–598. 5

[PKG03] PAULY M., KEISER R., GROSS M.: Multi-scale feature extraction on point-sampled surfaces. *Computer Graphics Forum 22*, 3 (2003), 281–289. 9

[PKU15] PKU: Shrec 2015 - non-rigid 3d shape retrieval. *http://www.icst.pku.edu.cn/zlian/shrec15-non-rigid/* (2015). 13

[PSea14] PICKUP D., SUN X., ET AL. P. L. R.: Shrec'14 track: Shape retrieval of non-rigid 3d human models. In *Eurographics Workshop on 3D Object Retrieval (2014)* (2014), pp. 1–10. 7

[PSM10] PERRONNIN F., SÁNCHEZ J., MENSINK T.: Improving the fisher kernel for large-scale image classification. In *Proceedings of the 11th European Conference on Computer Vision: Part IV* (2010), pp. 143–156. 9

[PSRM15] PICKUP D., SUN X., ROSIN P. L., MARTIN R. R.: Euclidean-distance-based canonical forms for non-rigid 3d shape retrieval. *Pattern Recognition (to appear)* (2015). 11

[SFH*09] SMEETS D., FABRY T., HERMANS J., VANDERMEULEN D., SUETENS P.: Isometric deformation modeling for object recognition. In *Computer Analysis of Images and Patterns* (2009), pp. 757–765. 4

[SMKF04] SHILANE P., MIN P., KAZHDAN M., FUNKHOUSER T.: The princeton shape benchmark. In *Proc. SMI'04* (2004), pp. 167–178. 2

[SOL09] SUN J., OVSJANIKOV M., LEONIDAS G.: A concise and provably informative multi-scale signature based on heat diffusion. In *Proceedings of the Symposium on Geometry Processing* (2009), pp. 1383–1392. 4, 7, 8

[SZM*08] SIDDIQI K., ZHANG J., MAXRINI D., SHOKOUFANDEH A., BOUIX S., DICKINSON S.: Retrieving articulated 3d models using medial surfaces. *Machine Vision and Applications 19*, 4 (2008), 261–274. 2

[ZH05] ZOU H., HASTIE T.: Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society: Series B (Statistical Methodology) 67*, 2 (2005), 301–320. 7

[ZYZH10] ZHOU X., YU K., ZHANG T., HUANG T. S.: Image classification using super-vector coding of local image descriptors. In *Proc. European conference on Computer vision 2010 (ECCV 2010)* (2010), pp. 141–154. 5