Eurographics Workshop on 3D Object Retrieval (2014), pp. 1–10
B. Bustos, H. Tabia, J.P. Vandeborre, and R. Veltkamp (Editors)

SHREC'14 Track: Shape Retrieval of Non-Rigid 3D Human Models

D. Pickup[1*], X. Sun[1*], P. L. Rosin[1*], R. R. Martin[1*], Z. Cheng[2*], Z. Lian[3*], M. Aono[4], A. Ben Hamza[9], A. Bronstein[5], M. Bronstein[14], S. Bu[6], U. Castellani[7] S. Cheng[6], V. Garro[7], A. Giachetti[7], A. Godil[8], J. Han[6], H. Johan[10], L. Lai,[13] B. Li[11], C. Li[8], H. Li,[13], R. Litman[5], X. Liu,[13] Z. Liu[6], Y. Lu[11], A. Tatsuma[4], J. Ye[12]

[1]Cardiff University, UK
[2]National University of Defense Technology, China
[3]Peking University, China
[4]Toyohashi University of Technology, Japan
[5]Technion - Israel Institute of Technology, Israel
[6]Northwestern Polytechnical University, China
[7]University of Verona, Italy
[8]National Institute of Standards and Technology, USA
[9]Concordia University, Canada
[10]Fraunhofer IDM@NTU, Singapore
[11]Texas State University, San Marcos, USA
[12]Penn State University, USA
[13]Beijing Technology and Business University, China
[14]University of Lugano, Switzerland
[*]Track organisers

Abstract

We have created a new benchmarking dataset for testing non-rigid 3D shape retrieval algorithms, one that is much more challenging than existing datasets. Our dataset features exclusively human models, in a variety of body shapes and poses. 3D models of humans are commonly used within computer graphics and vision, and so the ability to distinguish between body shapes is an important shape retrieval problem. In this track nine groups have submitted the results of a total of 22 different methods which have been tested on our new dataset.

1. Introduction

The ability to recognise a deformable object's shape, regardless of the pose of the object, is an important requirement for modern shape retrieval methods. Many state-of-the-art methods achieve extremely high accuracy when evaluated on the most recent benchmark [LGB*11]. It is therefore hard to distinguish between good methods, and there is little room to demonstrate improvement in approaches. There is thus a need for a more challenging benchmark for non-rigid 3D shape retrieval. Many novel apporaches have been published since the previous benchmark, and therefore a new comparison of state-of-the-art methods is also beneficial.

We have created a new, more challenging, benchmarking dataset for testing non-rigid 3D shape retrieval algorithms. Our dataset features exclusively human models, in a variety of body shapes and poses. 3D models of humans are commonly used within computer graphics and vision, therefore the ability to distinguish between body shapes is an important shape retrieval problem. The shape differences between humans are much more subtle than the differences between the shape classes used in current benchmarks (e.g. ants and birds), yet humans are able to visually recognise specific individuals. Successfully performing shape retrieval on a dataset of human models is therefore a far more challenging, but relevant task. We use our dataset to evaluate the retrieval performance of 22 different methods, submitted by nine different research groups. The track's website is available at [Tra].

Figure 1: *A selection of models included in our datasets. Above: Real dataset, below: Synthetic dataset.*

2. Datasets

Our track uses two datasets, a *Real* dataset, obtained by scanning real human participants, and a *Synthetic* dataset, created using 3D modelling software. The latter may be useful for testing algorithms intended to retrieve synthetic data, with well sculpted local details, while the former may be more useful to test algorithms that are designed to work even in the presence of noisy coarsely captured data lacking in local detail.

2.1. Real Dataset

The *Real* dataset was built from point-clouds contained within the Civilian American and European Surface Anthropometry Resource (CAESAR) [cae]. This dataset comprises 400 meshes, representing 40 human subjects (20 male, 20 female) in 10 different poses. The point-cloud models were manually selected from CAESAR to be models with significant visual differences. We employed SCAPE (shape completion and animation of people) [ASK*05] to build articulated 3D meshes, by fitting a template mesh to each subject. Realistic deformed poses of each subject were built using a data-driven deformation technique [CLC*13]. We remeshed the models using freely available software [VC04, VCP08]. The resulting models have approximately 15,000 vertices.

2.2. Synthetic Dataset

We also used the DAZ Studio [DAZ13] 3D modelling/animation software to create a dataset of synthetic human models. The software includes a parametrized human model, where parameters control body shape. We used this to produce a dataset consisting of 15 different human models (5 male, 5 female, 5 child), each with its own unique body shape. We generated 20 different poses for each mode, resulting in a dataset of 300 models. The same poses were used for each body shape, and models are considered to belong to the same class if they share the same body shape. All models were remeshed using the same method as for the *Real* dataset. The resulting models have approximately 60,000 vertices. A selection of both real and synthetic models is shown in Figure 1.

3. Evaluation

We assessed two different retrieval tasks:

1. Returning a list of all models, ranked by shape similarity to a query model.
2. Returning a list of models that all share the same shape as the query model.

For both tasks, every model in the database was used as a separate query model. In the first task, for each query we asked the participants to order all other models in the dataset in terms of similarity to the query model. In the second task, for each query the participants were asked to submit a list of arbitrary length of all models which they classify as 'the same shape' as the query model. Both tasks were evaluated separately.

The evaluation procedure for Task 1 is identical to that used in several previous SHREC tracks [LGB*11]. We evaluated the results using various statistical measures: precision and recall, nearest neighbour (NN), first tier (1-T), second tier (2-T), e-measure (E-M), discounted cumulative gain (DCG), and precision and recall curves. Definitions of these measures are given in [SMKF04]. The results for Task 2 were evaluated using the F-Measure [BYRN11].

4. Methods

4.1. Simple shape measures, and Euclidean distance based canonical forms *D. Pickup, X. Sun, P. L. Rosin and R. R. Martin*

This section presents two techniques, simple shape measures based on surface area, and skeleton driven canonical forms.

4.1.1. Simple shape measures

Two simple shape measures were tested separately on the datasets. The first is the total surface area of the mesh. This measure is not scale independent, and all human models were assumed to be properly scaled. In order to present a scale independent result, the second measure used is compactness. This is calculated as $\text{Volume}^2/\text{SurfaceArea}^3$. Both methods are trivial to implement, and are very efficient to compute.

4.1.2. Skeleton driven canonical forms

A variant on the canonical forms presented by Elad and Kimmel [EK03] is used to normalise the pose of all the models in the dataset, and then the rigid view-based method by Lian et al. [LGSX13] is used for retrieval. A canonical form is produced by extracting a curve skeleton from a mesh, using the method by Au at al. [ATC*08]. The SMACOF Multidimensional Scaling method used by [EK03] is then applied to the skeleton, to put the skeleton into a canonical pose. The skeleton driven shape deformation method by Yan et al. [YHMY08] is then used to deform the mesh to the new pose defined by the canonical skeleton. This produces a similar canonical form to [EK03], but with the local features better preserved. The models in the *Synthetic* dataset are simplified to approximately 15000 vertices, and any holes are filled, before computing the canonical form.

4.2. Hybrid shape descriptor and meta similarity generation for non-rigid 3D model retrieval, *B. Li, Y. Lu, A. Godil and H. Johan*

A hybrid shape descriptor [LGJ13] has been proposed to integrate both geodesic distance-based global features and curvature-based local features. An adaptive algorithm based on Particle Swarm Optimization (PSO) is developed to adaptively fuse different features to generate a meta similarity between any two models. The approach can be generalized to similar approaches which integrate more or other features. It first extracts three component features of the hybrid shape descriptor: curvature-based local feature, geodesic distance-based global feature, and Multidimensional scaling (MDS) based ZFDR [LJon] global feature. Based on the extracted features, corresponding distance matrices are computed and they are fused into a meta distance matrix based on PSO. Finally, the distances are sorted to generate the retrieval lists.

Curvature-based local feature vector: V_C**.** First, the Curvature Index feature of a vertex p is computed, which characterizes local geometry: $CI = \frac{2}{\pi}\log(\sqrt{\frac{K_1^2+K_2^2}{2}})$, where K_1 and K_2 are the two principal curvatures in the x and y directions respectively at p. Then the Curvature Index deviation feature of the adjacent vertices of p is computed: $\delta CI = \sqrt{\frac{\sum_{i=1}^{n}(CI_i-\widetilde{CI})}{n}}$, where $CI_1, CI_2, \ldots, CI_n$ are the Curvature Index values of the adjacent vertices of p and $\widetilde{CI}$ is the mean Curvature Index of all the adjacent vertices. Next, to describe the local topological property, the Shape Index feature of p is computed: $SI = \frac{2}{\pi}\arctan(\frac{K_1+K_2}{|K_1-K_2|})$. After that, a combined local shape descriptor is formed by concatenating the above three local features: $F = (CI, \delta CI, SI)$. Finally, based on the Bag-of-Words framework, the local feature vector $V_C = (h_1, h_2, \cdots, h_{N_C})$ is generated, where the number of cluster centres N_C is set to 50.

Geodesic distance-based global feature vector: V_G**.** First, to avoid the high computational cost involved in the geodesic distance computation among many vertices, the models are simplified to 1000 vertices. Next, the geodesic distances among all the vertices of a simplified model are generated to form a geodesic distance matrix GDM. Finally, the GDM is decomposed based on Singular Value Decomposition and the first largest k eigenvalues are used as the global feature vector. In experiments, k is set to 50.

MDS-based ZFDR global feature vector: V_Z**.** To leverage pose and deformation variations of non-rigid models, Multidimensional scaling (MDS) techniques are utilized to map the non-rigid models into a 3D canonical form. The previously computed geodesic distances among the 1000 vertices of each simplified 3D model are used as the input of MDS for the feature space transformation. Finally, the hybrid global shape descriptor ZFDR [LJon] is used to characterize the features of the transformed 3D model in the new feature space. There are four feature components in ZFDR: **Z**ernike moments feature, **F**ourier descriptor feature, **D**epth information feature and **R**ay-based feature. This approach is named as MDS-ZFDR and Stress MDS is adopted in the experiments. It was also found that for 3D human retrieval using **R** feature only (that is MDS-R) can always achieve better results than other combinations such as **ZF**, **DR** or **ZFDR**. The reason should be related to the more salient feature of the geometry-related 'thickness' variations in the human models, such as fat versus slim bodies which are better characterized by the **R** feature, compared to other visual-related features like **ZF** and **D**.

Retrieval algorithm: (1) Computation of Curvature-based local feature vector V_C based on the original models and local feature distance matrix M_C generation; (2) Computation of Geodesic distance-based global feature vector V_G and global feature distance matrix M_G. (3) MDS-based ZFDR global feature vector V_Z and MDS-ZFDR global feature distance matrix M_Z computation; (4) PSO-based meta distance matrix generation and ranking. A meta distance matrix $M = w_C M_C + w_G M_G + w_Z M_Z$ is generated, where w_C, w_G and w_Z fall in [0,1]. As a swarm intelligence optimization technique, PSO-based approach is robust and fast in solving problems that are non-linear and non-differentiable. It includes four steps: initialization, particles' velocity and positions update, search evaluation and result verification. The number of particles N_P=10; the maximum number of search iterations N_t=10; and First Tier is selected as the fitness value for search evaluation. Please note that the PSO-based weight assignment preprocessing step is only performed once for each of the test sets.

The 'Hybrid_R' runs only use 'MDS-R' features, compared to the original 'Hybrid' approach presented in [LGJ13] which uses 'MDS-ZFDR'. Besides comparing the component features including 'Curvature', 'Geodesic' distance and 'MDS-ZFDR' based features, the performance of 'MDS-R' is compared with 'MDS-ZFDR'.

4.3. Histograms of Area Projection Transform, *A. Giachetti and V. Garro*

Human characters are recognised with the Histograms of Area Projection Transform (HAPT), general purpose shape descriptors proposed in [GL12]. The method is based on a spatial map (Multiscale Area Projection Transform) that encodes the likelihood of the points inside the shape of being centres of spherical symmetry. This map is obtained by computing for each radius of interest the value:

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

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 centred in the generic point $\vec{x}$ where the map is computed. Values at different radii are normalized in order to have a scale-invariant behaviour, creating the Multiscale APT (MAPT):

$$\mathrm{MAPT}(x,y,z,R,S) = \alpha(R)\ \mathrm{APT}(x,y,z,S,R,\sigma(R)) \quad (2)$$

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 values $R_1,\dots,R_n$. In the paper it is also shown that histograms of MAPT computed inside the objects are very good global shape descriptors, showing very good performances on the SHREC 2011 Non-Rigid Watertight contest data [LGB*11]. For that recognition task discrete MAPT maps were quantized in 12 bins and histograms computed at the different scales (radii) considered were concatenated creating a 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. c was always taken equal to 0.5.

For the recognition of different human subjects, however, scale invariance is not wanted. For this reason a fixed voxel size and a fixed set of radii are used.

The values for these parameters have been chosen differently for the *Real* and the *Synthetic* datasets, using simple heuristics. The algorithm was tested using three different parameter configurations for each dataset (*Real* and *Synthetic*). The results were then compared, and the best configurations for each dataset were submitted to the track. The voxel size was taken similar to the size of the smaller details well defined in the meshes. For the *Synthetic* dataset, where fingers are visible and models are smaller, $s = 4\,mm$ is used and 11 increasing radii have been computed starting from $R_1 = 8\,mm$ and iteratively adding a fixed step of $4\,mm$ for the remaining values $\{R_2,\dots,R_{11}\}$. For the *Real* dataset, where models are bigger and details are smoothed, $s = 12\,mm$ is used applying 7 different radii starting from $R_1 = 24\,mm$ with a constant radius increasing of $12\,mm$.

The procedure for model comparison then simply consists in concatenating the histograms computed at the different scales and measuring distances between shapes by evaluating the Jeffrey divergence of the corresponding concatenated vectors.

In the tests this 'general purpose' shape comparison procedure is applied without specific adaptations to the task. A possible way to specialize it for human body recognition may consist in learning discriminative sets of radii with a feature selection procedure or in recognizing and comparing specific body regions.

The MAPT/histograms extraction (using the c++ implementation available at http://www.andreagiachetti.it) for the *Real* dataset takes around 46 min, with a mean of 7 sec. for each model; the computation for the *Synthetic* dataset is much longer dealing with more detailed meshes: 2 hours for the entire dataset, 25 sec. for each shape. A single query takes around 1.2 msec. using a Matlab implementation of the Jeffrey divergence distance.

4.4. R-BiHDM, *J. Ye*

The R-BiHDM [YYY13] method is a spectral method used for general non-rigid shape retrival. Using modal analysis, the method projects Biharmonic distance [LRF10] map into a low-frequency representation which operates on the modal space spanned by the lowest eigenfunctions of shape Laplacian [RWP06, OBCS*12], and then computes its spectrum as an isometric shape descriptor.

Let $\psi_0,\psi_1,\dots,\psi_m$ be the eigenfunctions of Laplacian Δ corresponding to its smallest eigenvalues $0 = \lambda_0 < \lambda_1 \leq \dots \leq \lambda_m$. Let $d(x,y)$ be the Biharmonic distance between two points on mesh, which is defined as

$$d(x,y)^2 = \sum_{i=1}^{m} \frac{1}{\lambda_i^2} \left(\psi_i(x) - \psi_i(y)\right)^2. \quad (3)$$

The squared Biharmonic distance map $\mathcal{D}^2$ is a functional map defined by

$$\mathcal{D}^2[f](x) = \int_{x\in S} d^2(x,y) f(y) \mathrm{d}y, \quad (4)$$

where S is the differential manifold of shape. The reduced matrix version of $\mathcal{D}^2$ is denoted by $A = \{a_{i,j}\}$, where $a_{i,j} = \left\langle \psi_i, \mathcal{D}^2\psi_j \right\rangle_S = \int_S \psi_i(x)\mathcal{D}^2[\psi_j](x)\mathrm{d}x$ for $0 \leq i,j \leq m$. Note that $\mathrm{tr}(A) = 0$ and all eigenvalues of A, denoted by $\mu_0,\dots,\mu_m$ are in a magnitude descending order, where $\mu_0 > 0$ and $\mu_i < 0$ for $i > 0$. The shape descriptor is defined as a vector $[\mu_1,\dots,\mu_m]^T$ (scale dependent) or $[\frac{\mu_1}{\mu_0},\dots,\frac{\mu_L}{\mu_0}]^T$ (scale independent). For this shape contest, we choose $L = 30$ and $m = 100$. Finally, a normalized Euclidean distance is used for nearest neighbour queries. The descriptor is insensitive to a number of perturbations, such as isometry, noise, and remeshing. It has superior discrimination capability regarding globally change of shape and is very efficient to compute. It has been shown that scale independent descriptor (R-BiHDM) is more reliable for generic nonrigid shape tasks, while scale dependent descriptor (R-BiHDM-s) is more suitable for this human shape task.

4.5. HKS-TS and SIHKS-H, *L. Lai, X. Lui and H. Li*

The HKS-TS (heat kernel signature based on time serial) is an application of HKS [SOG09], which adds the statistics of dynamic HKS on a shape according to an appropriate time serial chosen using a subset of the *Real* data. The SIHKS-H (scale invariant heat kernel signature based on statistics histogram) is an application of SI-HKS [BK10]. The SI-HKS is calculated on the shape to form a histogram. Then the similarity between different shapes can be calculated according to the SIHKS-H. Different similarity will be found by using different methods. Finally, the ranking list can be produced according to the similarity. For Task 2 two methods are used, HKS-TS-HC and SIHKS-H-HC. They add a further processing step to the methods used for Task 1. HC means the hierarchical clustering algorithm. The hierarchical clustering algorithm is added for classification according to the similarity calculating in Task 1. Intuitively, the algorithm is improved to be fit for this task.

4.6. High-level Feature Learning for 3D Shapes, *S. Bu, S. Chen, Z. Lui and J. Han*

The proposed high-level feature learning method for 3D shapes is carried out in the following three stages.

1. Low-level feature extraction: three representative intrinsic features, scale-invariant heat kernel signature (SI-HKS) [BK10], shape diameter function (SDF) [GSCO07], and averaged geodesic distance (AGD) [HSKK01], are adopted as low-level descriptors.
2. Middle-level feature extraction: To tackle the spatial information missing in the low-level features, a middle-level position-independent Bag-of-Features (BoF) is first extracted from the above low-level 3D descriptors. In order to compensate the lack of structural relationship, the original BoF is further extended into a geodesic-aware BoF (GA-BoF), which considers the geodesic distance between each pair of BoF on the 3D surface.
3. High-level feature learning: Finally, a deep learning based approach is introduced to further learn high-level features from the GA-BoF, which is able to discover the intrinsic relationship among GA-BoF and provide high discriminative features for 3D shape retrieval.

4.6.1. Low-level 3D Shape Descriptors

In this research, the scale-invariant heat kernel signature, shape diameter function, and average geodesic distance are adopted as the low-level 3D shape descriptors which are used for generating middle-level features, since these three local descriptors are robust against non-rigid and complex shape deformations. The first six frequency components of the SI-HKS, SDF and AGD descriptors are concatenated to form a low-level shape descriptor as

$$F(x_i) = (\mathrm{SIHKS}(x_i)[\omega_1, \ldots, \omega_6], \mathrm{SDF}(x_i), \mathrm{AGD}(x_i)), \quad (5)$$

where the dimension of the feature is $M = 8$.

4.6.2. Middle-level Features

In this step, Bag-of-Features (BoFs) are computed to represent the occurrence probability of geometric words, and Minkowski metric is adopted as feature weighting [CdAM12] for k-means to generate geometric words more precisely.

After the geometric words $\mathcal{C} = \{c_1, c_2, .., c_K\}$ of size K is obtained, the next step is to quantize the low-level descriptor space in order to obtain a compact representation. For each point $x \in X$ with the descriptor $F(x)$, feature distribution $\phi(x)$ is defined as

$$\phi_i(x) = c(x) \exp\left(-\frac{||F(x) - c_i||_2^2}{k_{\mathrm{BoF}}\ \sigma_{\min}^2}\right), \quad (6)$$

where the constant $c(x)$ is selected to satisfy $||\phi(x)||_1 = 1$.

The geodesics on the mesh are used to measure the spatial relationship between each pair of BoFs on vertices, and introduce the geodesics-aware Bag-of-Features (GA-BoF):

$$\mathbf{v}(X) = N(X) \sum_{x_i \in X} \sum_{x_j \in X} \phi(x_i)\phi(x_j)^T \exp\left(-k_{\mathrm{gd}} \frac{g_d(x_i, x_j)}{\sigma_{\mathrm{gd}}}\right), \quad (7)$$

where $N(X)$ is a normalization factor which makes features have a fixed maximum value of 1, σ_{gd} is the maximal geodesic distance of any pair of vertices on the mesh, and k_{gd} denotes the decay rate of distances, which is selected empirically. The resulting $\mathbf{v}$ is a $K \times K$ matrix, which represents the frequency of geometric words i and j appearing within a specified geodesic distance. This expression provides occurrence probability of geometric words and relationship between them.

4.6.3. Feature Learning via Deep Learning

In order to further deeply mine the relationship of features from intra-class shapes and inter-class shapes in a large dataset, deep learning is introduced into our framework, which will result in high-level features with strong generalization. Due to the fact that deep belief networks (DBN) [HOT06] has shown good performance and is a probabilistic approach, DBN is adopted as the feature learning method to extract high-level features for the 3D shapes.

Stacking a number of the restricted Boltzmann machines (RBMs) and learning layer by layer from bottom to top gives rise to a DBN. It has been shown that the layer-by-layer greedy learning strategy [HOT06] is effective, and the greedy procedure achieves approximate maximum likelihood learning. In this method, the bottom layer RBM is trained with the input data of GA-BoF, and the activation probabilities of hidden units are treated as the input data for training the upper-layer RBM, and so on.

In the shape retrieval task, unlabelled 3D shape data are used to train the DBN layer-by-layer. After obtaining the optimal parameters, the input GA-BoFs are processed layer-by-layer till the final layer which are used as the high-level features. In the retrieval, L_2 distance of the features is used to measure the similarity of two shapes $\mathbf{X}$ and $\mathbf{Y}$ as

$$d_s(\mathbf{X},\mathbf{Y}) = ||o(\mathbf{X}) - o(\mathbf{Y})||_2. \tag{8}$$

4.7. Bag-of-Features approach with Augmented Point Feature Histograms, *A. Tatsuma and M. Aono*

The developed Augmented Point Feature Histograms (APFH) expands Point Feature Histograms (PFH) [RMBB08] by adding the statistics of their geometric features. PFH is known as a local feature vector for 3D point clouds. PFH constructs a histogram of geometric features extracted from neighbouring oriented points. Improving the discriminant power of PFH by adding the mean and covariance of its geometric features is investigated. Because APFH is a local feature vector as well as PFH, it is invariant to the global deformation and articulation of the 3D model.

The overview of how the method defines the proposed APFH is illustrated in Figure 2. With APFH, the first step is to randomly generate oriented points on the triangle surface of a 3D model using Osada's method [OFCD02]. To generate a random point $\mathbf{p}$ on an arbitrary triangle surface composed of vertices $\mathbf{v}_a$, $\mathbf{v}_b$, and $\mathbf{v}_c$, the following formula is employed:

$$\mathbf{p} = (1-\sqrt{r_1})\mathbf{v}_a + \sqrt{r_1}(1-r_2)\mathbf{v}_b + \sqrt{r_1}r_2\mathbf{v}_c. \tag{9}$$

In the implementation, two random variables, r_1 and r_2 in the above equation, are computed using the Niederreiter pseudo-random number generator [BFN94]. The oriented point is generated by inheriting the normal vector of the surface as an orientation of the point.

Next a PFH for each oriented point is constructed. The PFH finds the k-neighbourhood for each oriented point, and calculates a four-dimensional geometric feature $\mathbf{f} = [f_1, f_2, f_3, f_4]^{\mathrm{T}}$ as proposed in [WHH03]. The four-dimensional geometric feature is defined as follows for every pair of points $\mathbf{p}_a$ and $\mathbf{p}_b$ in the k-neighbourhood, and for their normal vectors $\mathbf{n}_a$ and $\mathbf{n}_b$:

$$\begin{aligned} f_1 &= \arctan(\mathbf{w}\cdot\mathbf{n}_b, \mathbf{u}\cdot\mathbf{n}_a), \\ f_2 &= \mathbf{v}\cdot\mathbf{n}_b, \\ f_3 &= \mathbf{u}\cdot\frac{\mathbf{p}_b-\mathbf{p}_a}{d}, \\ f_4 &= d, \end{aligned}$$

where $\mathbf{u} = \mathbf{n}_a$, $\mathbf{v} = (\mathbf{p}_b - \mathbf{p}_a)\times\mathbf{u}/||(\mathbf{p}_b-\mathbf{p}_a)\times\mathbf{u}||$, $\mathbf{w} = \mathbf{u}\times\mathbf{v}$, and $d = ||\mathbf{p}_b - \mathbf{p}_a||$. The PFH collects the four-dimensional geometric features in a 16-bin histogram $\mathbf{f}_h$. The index of the histogram bin h is defined by the following formula:

$$h = \sum_{i=1}^{4} s(t, f_i)\cdot 2^{i-1},$$

where $s(t, f)$ is a threshold function defined as 0 if $f < t$ and 1 otherwise. The threshold value of f_1, f_2, and f_3 are set to 0, and set the threshold value of f_4 to the average value of f_4 in the k-neighbourhood.

Furthermore, the mean and covariance of the four-dimensional geometric features is calculated. Let $\mathbf{f}_i$ be the four-dimensional geometric feature of an oriented point in the k-neighbourhood. The mean feature $\mathbf{f}_m$ and covariance feature $\mathbf{f}_c$ in the k-neighbourhood are defined as follows:

$$\mathbf{f}_m = \frac{1}{k}\sum_{i=1}^{k}\mathbf{f}_i,$$

$$\mathbf{f}_c = \mathrm{Upper}\left(\frac{1}{k-1}\sum_{i=1}^{k}(\mathbf{f}_i-\mathbf{f}_m)(\mathbf{f}_i-\mathbf{f}_m)^{\mathrm{T}}\right),$$

where $\mathrm{Upper}(\cdot)$ concatenates the upper triangular part of the matrix. Our APFH $\mathbf{f}_{APFH}$ is composed $\mathbf{f}_h$, $\mathbf{f}_m$, and $\mathbf{f}_c$.

Finally, APFH $\mathbf{f}_{APFH}$ is normalized with the power and the L2 normalization [PSM10].

To compare 3D models, the set of APFH features of a 3D model is integrated into a feature vector of the 3D model using the Bag-of-Features (BoF) approach [BBGO11, SZ03]. Moreover, the BoF is projected onto Jensen-Shannon kernel space using the homogeneous kernel map method [VZ12]. This approach is called BoF-APFH.

In addition, similarity between features is calculated using the manifold ranking method with the unnormalized graph Laplacian [ZBS11]. This approach is called MR-BoF-APFH.

The parameters of each algorithm are fixed empirically. For the APFH, the number of points is set to 20000, and the number of the neighbourhood to 55. For the BoF-APFH approach, a codebook of 1200 centroids is generated using K-means clustering, and used the SHREC'11 Non-rigid 3D Watertight dataset for training of the codebook.

4.8. BoF and SI-HKS, *R. Litman, A. Bronstein, M. Bronstein and U. Castellani*

All shapes were down-sampled to have $4,500$ triangles. For each shape $\mathcal{S}$ in the data-set, an SI-HKS [BK10] descriptor $\mathbf{x}_i$ was calculated in every point $i\in\mathcal{S}$. Unsupervised dictionary learning was done over randomly selected descriptors from all of the shapes using the SPAMS toolbox [MBPS09], with dictionary size of 32. The resulting 32 atom dictionary $\mathbf{D}$ was, in essence, the *bag-of-features* of this method. Next, in every point descriptor $\mathbf{x}_i$ was 'replaced' with a sparse code $\mathbf{z}_i$ by solving pursuit problem

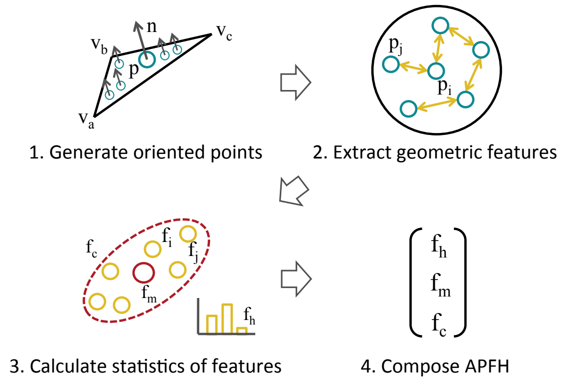

Figure 2: *Overview of the Augmented Point Feature Histograms (APFH).*

$$\min_{\mathbf{z}_i} \frac{1}{2}\|\mathbf{x}_i - \mathbf{D}\mathbf{z}_i\|_2^2 + \lambda\|\mathbf{z}_i\|_1. \quad (10)$$

The resulting codes $\mathbf{z}_i$ were subsequently pooled into a single histogram using mean pooling $\mathbf{h} = \sum_i \mathbf{z}_i w_i$, with w_i being element area of point i.

Lastly, the main contribution of this method: the initial $\mathbf{D}$ has undergone supervised training using about 30% of the shape-classes. Training was done using stochastic gradient descent of the loss-function defined in [WS09].

Present are the performance of the supervised training (supDLtrain), and for reference, the performance of initial unsupervised $\mathbf{D}$ (UnSup32). Additionally, a similar unsupervised method used in [BBGO11] is also presented (softVQ48). This method uses k-means clustering (with $k = 48$) and soft vector-quantization, instead of dictionary learning and pursuit, respectively.

4.9. Spectral Geometry, *C. Li, A. Godil and A. Ben Hamza*

The spectral geometry based framework in [Li13] is used for human body shape representation and retrieval. This framework is based on the eigendecomposition of the Laplace-Beltrami operator (LBO), which provides a rich set of eigenbases that are invariant to isometric transformations. It consists of two main stages: (1) spectral graph wavelet signature [LH13b] for descriptors extraction, and (2) intrinsic spatial pyramid matching [LH13a] for shape comparison. The cotangent weight scheme was used to discretize LBO. The eigenvalues λ_i and associated eigenfunctions $\boldsymbol{\varphi}_i$ can be computed by solving the generalized problem $C\boldsymbol{\varphi}_i = \lambda_i A\boldsymbol{\varphi}_i, \quad i = 1,2,\ldots,m$, where A is a positive-definite diagonal area matrix and C is a sparse symmetric weight matrix. In the experiments m is set to 200.

Spectral graph wavelet signature: The first stage consists of the computation of a dense 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 in the human body retrieval contests for isometric invariant representation. In this work the recently proposed spectral graph wavelet signature (SGWS) is used as the local descriptor; it provides 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 a bid to capture the global and local geometry, 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 of representative local descriptors $P = \{p_k,\ k = 1,2,\ldots,K\}$ learned by k-means, the dense descriptor $S = \{s_t,\ t = 1,2,\ldots,T\}$ at each point of the shape is replaced by the Gaussian kernel based soft assignment $Q = \{q_k,\ k = 1,2,\ldots,K\}$.

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

$$\lambda_2 = \inf_{f \perp \boldsymbol{\varphi}_1} \frac{f'Cf}{f'Af} \quad (11)$$

The isocontours of the second eigenfunction (Figure 3) are used to cut the shape into R 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, the histogram order is inverted, and the scheme with the minimum cost is considered as a better matching. The second eigenfunction is the smoothest mapping from the manifold to the 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 2 in this contest.

Finally, the result is ISPM induced histograms for shape representation. The dissimilarity between two shapes is computed as the L_1 distance.

Running time The method is implemented in MATLAB. The time consuming steps of the method are the computation of LBO and k-means dictionary learning. For a mesh with 15,000 vertices, it takes 8 seconds to compute the LBO. For a mesh with 60,000 vertices, it takes 37 seconds to compute the LBO. To learn a dictionary with 100 geometric words, it takes 45 minutes. Therefore, it averagely takes at most 24 hours (less than one day) to run the program for each dataset.

Figure 3: *The isocontours of the second eigenfunction.*

5. Results

Here we evaluate the retrieval results of the methods described in Section 4, applied to the datasets described in Section 2. Table 1 shows the results of Task 1 evaluated using the NN, 1-T, 2-T, E-M and DCG measures discussed in Section 3. All methods performed better on the *Synthetic* dataset, with most methods performing poorly on the *Real* data. This shows that it is potentially easier to distinguish between synthetically generated objects, rather than objects captured from the 'real world', and that testing on synthetic data is not a reliable way to predict performance on real data. The different classes in the *Synthetic* data may also be more easily distinguished because they have been manually designed to be different for this competition, whereas the models in the *Real* dataset were generated from body scans of human participants taken from an existing dataset, who may or may not have had very different body shapes. There is in fact a much higher similarity between the classes in the *Real* dataset. Figure 4 shows the precision-recall curve for the best performing methods submitted by each participant.

On the more challenging *Real* dataset, three methods, due to Litman et al., Ye, and Giachetti and Garro, performed significantly better than the others. The best performing method by Litman et al. was trained on a subset of the test set, and therefore has an advantage over their other submissions, but the unsupervised variants of their method still perform well.

The performance of different methods is far closer on the *Synthetic* dataset. The organisers (Pickup et al.) submitted two very simple methods, Surface Area and Compactness. It is interesting to note that they perform better than many of the more sophisticated methods submitted, including their own, and Surface Area is one of the top performing methods on the *Synthetic* dataset. These measures are obviously not novel, but they highlight that sophistication does not always lead to better performance, and a simpler and computationally very efficient algorithm may suffice. Algorithms should concentrate on what is truly invariant for each class.

Table 2 shows the results of Task 2 evaluated using the F-Measure. As for Task 1, the performance of all methods is much higher for the *Synthetic* dataset. All but one of the methods used pre-existing knowledge of the size of each class.

Author	Method	NN	1-T	2-T	E-M	DCG
Giachetti	APT†	0.845	0.534	0.681	0.355	0.795
Lai	HKS-TS†	0.245	0.259	0.461	0.314	0.548
	SIHKS-H†	0.125	0.090	0.186	0.145	0.388
B. Li	Curvature	0.083	0.076	0.138	0.099	0.347
	Geodesic	0.070	0.078	0.158	0.113	0.355
	Hybrid†	0.045	0.080	0.164	0.117	0.354
	Hybrid-R†	0.043	0.092	0.173	0.123	0.363
	MDS-R	0.035	0.066	0.129	0.090	0.330
	MDS-ZFDR	0.030	0.040	0.091	0.075	0.310
C. Li	Spectral Geom.	0.313	0.206	0.323	0.192	0.488
Litman	supDLtrainR†	0.793	0.727	0.914	0.432	0.891
	UnSup32	0.583	0.451	0.659	0.354	0.712
	softVQ48	0.598	0.472	0.657	0.356	0.717
Pickup	Surface Area	0.263	0.289	0.509	0.326	0.571
	Compactness	0.275	0.221	0.384	0.255	0.519
	Canonical	0.010	0.012	0.040	0.043	0.279
Bu	3DDL	0.225	0.193	0.374	0.262	0.504
Tatsuma	BoF-APFH	0.053	0.100	0.226	0.162	0.383
	MR-BoF-APFH	0.048	0.071	0.131	0.084	0.327
Ye	R-BiHDM	0.275	0.201	0.334	0.217	0.492
	R-BiHDM-s	0.685	0.541	0.742	0.387	0.781

Real Dataset

Author	Method	NN	1-T	2-T	E-M	DCG
Giachetti	APT†	0.970	0.733	0.927	0.655	0.936
Lai	HKS-TS	0.467	0.476	0.743	0.504	0.729
	SIHKS-H	0.427	0.206	0.332	0.219	0.562
B. Li	Curvature	0.620	0.485	0.710	0.488	0.774
	Geodesic	0.540	0.362	0.529	0.363	0.674
	Hybrid†	0.460	0.503	0.743	0.512	0.773
	Hybrid-R†	0.413	0.518	0.767	0.532	0.774
	MDS-R	0.267	0.284	0.470	0.314	0.594
	MDS-ZFDR	0.207	0.228	0.407	0.265	0.559
C. Li	Spectral Geom.	0.993	0.832	0.971	0.706	0.971
Litman	supDLtrainS†	0.960	0.887	0.991	0.721	0.975
	UnSup32	0.893	0.754	0.918	0.657	0.938
	softVQ48	0.910	0.729	0.949	0.659	0.927
Pickup	Surface Area	0.807	0.764	0.987	0.691	0.901
	Compactness	0.603	0.544	0.769	0.527	0.773
	Canonical	0.113	0.182	0.333	0.217	0.507
Bu	3DDL	0.923	0.760	0.911	0.641	0.921
Tatsuma	BoF-APFH	0.650	0.592	0.740	0.528	0.824
	MR-BoF-APFH	0.880	0.672	0.871	0.601	0.887
Ye	R-BiHDM	0.737	0.496	0.673	0.467	0.778
	R-BiHDM-s	0.793	0.572	0.760	0.533	0.836

Synthetic Dataset

Table 1: *Retrieval results for Task 1. The 1st, 2nd and 3rd highest scores of each column are highlighted. † means the method has used part of the test data for training or parameter optimisation.*

		Real	*Synthetic*
Participant	Method	F-Measure	F-Measure
Giachetti	APT†‡	0.534	0.733
Lai	HKS-TS-HC†‡	0.063	0.244
	SIHKS-H-HC†‡	0.038	0.089
C. Li	Spectral Geometry‡	0.204	0.828
Litman	supDLtrainR†	0.640	0.814
Pickup	Surface Area‡	0.301	0.759
Bu	3DDL‡	0.193	0.760

Table 2: *Retrieval results for Task 2. The 1st, 2nd and 3rd highest scores of each column are highlighted. ‡ signifies the method is aware of the class size, other annotation as for Table 1.*

6. Conclusion

This paper compared non-rigid retrieval results obtained by 22 different methods, submitted by nine research groups, on two new datasets of human body models. These datasets are much more challenging than previous non-rigid datasets [LGB*11], as evidenced by lower success rates. The data obtained by scanning real human participants proved more challenging than the synthetically generated data. This shows that there is a lot of room for future research to improve the analysis of 'real' data. If the performance of methods is to be improved for real data, then more real datasets are needed for testing purposes, as synthetic datasets do not adequately mimic the same challenge.

All methods submitted were designed for generic non-rigid shape retrieval. Our new dataset has created the potential for new research into methods which specialise in shape retrieval of humans.

Acknowledgements

This work was supported by EPSRC Research Grant EP/J02211X/1.

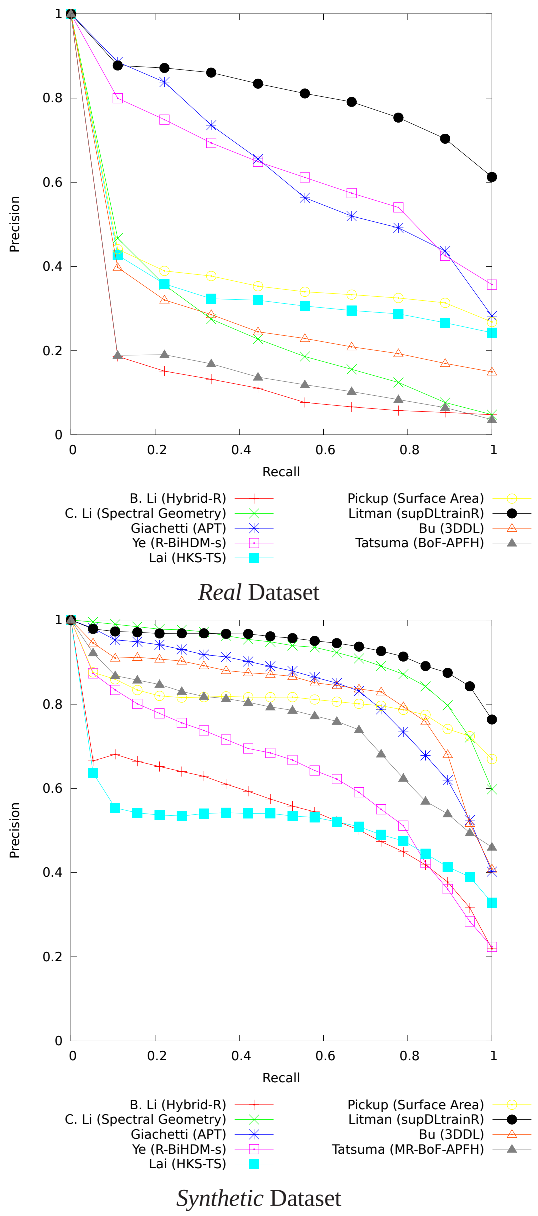

Figure 4: *Precision and Recall curves for the best performing method of each group.*

References

[ASK*05] ANGUELOV D., SRINIVASAN P., KOLLER D., THRUN S., RODGERS J., DAVIS J.: Scape: Shape completion and animation of people. In *ACM SIGGRAPH 2005 Papers* (2005), SIGGRAPH '05, ACM, pp. 408–416. 2

[ATC*08] AU O. K.-C., TAI C.-L., CHU H.-K., COHEN-OR D., LEE T.-Y.: Skeleton extraction by mesh contraction. In *ACM SIGGRAPH 2008 Papers* (New York, NY, USA, 2008), SIGGRAPH '08, ACM, pp. 44:1–44:10. 3

[BBGO11] BRONSTEIN A. M., BRONSTEIN M. M., GUIBAS L. J., OVSJANIKOV M.: Shape google: Geometric words and expressions for invariant shape retrieval. *ACM Transactions on Graphics 30*, 1 (Feb. 2011), 1–20. 6, 7

[BFN94] BRATLEY P., FOX B. L., NIEDERREITER H.: Programs to generate niederreiter's low-discrepancy sequences. *ACM Transactions on Mathematical Software 20*, 4 (Dec. 1994), 494–495. 6

[BK10] BRONSTEIN M., KOKKINOS I.: Scale-invariant heat kernel signatures for non-rigid shape recognition. In *2010 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (June 2010), pp. 1704–1711. 5, 6

[BYRN11] BAEZA-YATES R. A., RIBEIRO-NETO B. A.: *Modern Information Retrieval - the concepts and technology behind search, Second edition*. Pearson Education Ltd., Harlow, England, 2011. 2

[cae] CAESAR. http://store.sae.org/caesar/. 2

[CdAM12] CORDEIRO DE AMORIM R., MIRKIN B.: Minkowski metric, feature weighting and anomalous cluster initializing in k-means clustering. *Pattern Recognition 45*, 3 (2012), 1061–1075. 5

[CLC*13] CHEN Y., LAI Y., CHENG Z., MARTIN R., SHIYAI J.: A data-driven approach to efficient character articulation. In *Proceedings of IEEE CAD/Graphics* (2013). 2

[DAZ13] DAZ Studio. http://www.daz3d.com/, 2013. 2

[EK03] ELAD A., KIMMEL R.: On bending invariant signatures for surfaces. *IEEE Transactions on Pattern Analysis and Machine Intelligence 25*, 10 (2003), 1285–1295. 3

[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. 4

[GSCO07] GAL R., SHAMIR A., COHEN-OR D.: Pose-oblivious shape signature. *IEEE Transactions on Visualization and Computer Graphics 13*, 2 (2007), 261–271. 5

[HOT06] HINTON G. E., OSINDERO S., TEH Y.-W.: A fast learning algorithm for deep belief nets. *Neural computation 18*, 7 (2006), 1527–1554. 5

[HSKK01] HILAGA M., SHINAGAWA Y., KOHMURA T., KUNII T. L.: Topology matching for fully automatic similarity estimation of 3d shapes. In *Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques* (New York, NY, USA, 2001), SIGGRAPH '01, ACM, pp. 203–212. 5

[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 *Proceedings of the 4th Eurographics conference on 3D Object Retrieval* (2011), EG 3DOR'11, Eurographics Association, pp. 79–88. 1, 2, 4, 9

[LGJ13] LI B., GODIL A., JOHAN H.: Hybrid shape descriptor and meta similarity generation for non-rigid and partial 3D model retrieval. *Multimedia Tools and Applications (Online First version))* (2013), 1–30. 3

[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* (2013), 1–20. 3

[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. 7

[LH13b] LI C., HAMZA A. B.: A multiresolution descriptor for deformable 3d shape retrieval. *The Visual Computer* (2013), 1–12. 7

[LH13c] LI C., HAMZA A. B.: Spatially aggregating spectral descriptors for nonrigid 3d shape retrieval: a comparative survey. *Multimedia Systems* (2013), 1–29. 7

[Li13] LI C.: *Spectral Geometric Methods for Deformable 3D Shape Retrieval*. Master's thesis, Concordia University, 2013. 7

[LJon] LI B., JOHAN H.: 3D model retrieval using hybrid features and class information. *Multimedia Tools and Applications* (2011, Online First version), 1–26. 3

[LRF10] LIPMAN Y., RUSTAMOV R. M., FUNKHOUSER T. A.: Biharmonic distance. *ACM Trans. Graph. 29*, 3 (July 2010), 27:1–27:11. 4

[MBPS09] MAIRAL J., BACH F., PONCE J., SAPIRO G.: Online dictionary learning for sparse coding. In *Proceedings of the 26th Annual International Conference on Machine Learning* (New York, NY, USA, 2009), ICML '09, ACM, pp. 689–696. 6

[OBCS*12] OVSJANIKOV M., BEN-CHEN M., SOLOMON J., BUTSCHER A., GUIBAS L.: Functional maps: A flexible representation of maps between shapes. *ACM Trans. Graph. 31*, 4 (July 2012), 30:1–30:11. 4

[OFCD02] OSADA R., FUNKHOUSER T., CHAZELLE B., DOBKIN D.: Shape distributions. *ACM Transactions on Graphics 21* (2002), 807–832. 6

[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* (Berlin, Heidelberg, 2010), ECCV '10, Springer-Verlag, pp. 143–156. 6

[RMBB08] RUSU R. B., MARTON Z. C., BLODOW N., BEETZ M.: Persistent point feature histograms for 3D point clouds. In *Proceedings of the 10th International Conference on Intelligent Autonomous Systems* (2008). 6

[RWP06] REUTER M., WOLTER F.-E., PEINECKE N.: Laplace-Beltrami spectra as âĂŸshape-dnaâĂŹ of surfaces and solids. *Computer-Aided Design 38*, 4 (2006), 342 – 366. 4

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

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

[SZ03] SIVIC J., ZISSERMAN A.: Video google: A text retrieval approach to object matching in videos. In *Proceedings of the Ninth IEEE International Conference on Computer Vision* (Washington, DC, USA, 2003), vol. 2 of *ICCV '03*, IEEE Computer Society, pp. 1470–1477. 6

[Tra] Track website. http://www.cs.cf.ac.uk/shaperetrieval/shrec14/. 1

[VC04] VALETTE S., CHASSERY J.-M.: Approximated centroidal voronoi diagrams for uniform polygonal mesh coarsening. *Computer Graphics Forum 23*, 3 (2004), 381–389. 2

[VCP08] VALETTE S., CHASSERY J.-M., PROST R.: Generic remeshing of 3d triangular meshes with metric-dependent discrete voronoi diagrams. *IEEE Transactions on Visualization and Computer Graphics 14*, 2 (2008), 369–381. 2

[VZ12] VEDALDI A., ZISSERMAN A.: Efficient additive kernels via explicit feature maps. *IEEE Transactions on Pattern Analysis and Machine Intelligence 34*, 3 (March 2012), 480–492. 6

[WHH03] WAHL E., HILLENBRAND U., HIRZINGER G.: Surflet-pair-relation histograms: A statistical 3D-shape representation for rapid classification. In *Proceedings of International Conference on 3D Digital Imaging and Modeling* (2003), pp. 474–482. 6

[WS09] WEINBERGER K. Q., SAUL L. K.: Distance metric learning for large margin nearest neighbor classification. *J. Mach. Learn. Res. 10* (June 2009), 207–244. 7

[YHMY08] YAN H.-B., HU S.-M., MARTIN R., YANG Y.-L.: Shape deformation using a skeleton to drive simplex transformations. *Visualization and Computer Graphics, IEEE Transactions on 14*, 3 (May 2008), 693–706. 3

[YYY13] YE J., YAN Z., YU Y.: Fast nonrigid 3d retrieval using modal space transform. In *Proceedings of the 3rd ACM Conference on International Conference on Multimedia Retrieval* (New York, NY, USA, 2013), ICMR '13, ACM, pp. 121–126. 4

[ZBS11] ZHOU X., BELKIN M., SREBRO N.: An iterated graph laplacian approach for ranking on manifolds. In *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (2011), KDD '11, pp. 877–885. 6