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

SHREC'14 Track: Retrieval and Classification on Textured 3D Models

S. Biasotti[†1], A. Cerri[†1], M. Abdelrahman[2], M. Aono[3], A. Ben Hamza[4], M. El-Melegy[2], A. Farag[2], V. Garro[5], A. Giachetti[5], D. Giorgi[6], A. Godil[7], C. Li[7], Y.-J. Liu[8], H. Y. Martono[3], C. Sanada[3], A. Tatsuma[3], S. Velasco-Forero[9], C.-X. Xu[8]

[1]Istituto di Matematica Applicata e Tecnologie Informatiche "E. Magenes", CNR, Italy
[2]Computer Vision and Image Processing Laboratory (CVIP Lab), University of Louisville, KY, USA
[3]Department of Computer Science and Engineering, Toyohashi University of Technology, Japan
[4]Concordia University, Canada
[5]Dipartimento di Informatica, Università di Verona, Italy
[6]Istituto di Scienza e Tecnologie dell'Informazione "A. Faedo", CNR, Italy
[7]National Institute of Standards and Technology, USA
[8]Department of Computer Science and Technology, Tsinghua University, P. R. China
[9]Department of Mathematics, National University of Singapore, Singapore

Abstract

This paper reports the results of the SHREC'14 track: Retrieval and classification on textured 3D models*, whose goal is to evaluate the performance of retrieval algorithms when models vary either by geometric shape or texture, or both. The collection to search in is made of 572 textured mesh models, having a two-level classification based on geometry and texture. Together with the dataset, a training set of 96 models was provided. The track saw eight participants and the submission of 22 runs, to either the retrieval or the classification contest, or both. The evaluation results show a promising scenario about textured 3D retrieval methods, and reveal interesting insights in dealing with texture information in the CIELab rather than in the RGB colour space.*

Categories and Subject Descriptors (according to ACM CCS): H.3.1 [Information Storage and Retrieval]: Content Analysis and Indexing—Abstracting methods;

Introduction

The aim of SHREC is to evaluate the performance of existing 3D shape similarity methods, by highlighting their strengths and weaknesses, using a common test collection allowing for a direct comparison of methods. In this report the results of the SHREC'14 track: *Retrieval and Classification on Textured 3D Models* are presented. The aim of the track is to assess the performance of retrieval algorithms when models vary either by geometric shape or texture, or both. The track extends the analogous SHREC'13 track [CBA*13] in terms of the number of models and the type of geometric and texture deformations considered, and also for the existence of a training set for learning-based methods. The dataset is built from a set of null textured models, by randomly selecting and coupling geometric and texture perturbations. Geometric perturbations include noise and model re-sampling, non-rigid and non-isometric deformations; texture perturbations include non-uniform transformation of the RGB channels.

1. Data Collection and Queries

The dataset is made of 572 watertight mesh models, grouped in 16 geometric classes. Each class but one is obtained from 12 null models, corresponding to 4 base meshes endowed with 3 different textures. Then, 2 transformations are applied to each null shape, each one randomly combining a geometric deformation (a re-sampling, the addition of Gaussian noise, an affine deformation, a non-rigid and a non-isometric deformations) with a texture one (lightening/darkening, topological deformations in the texture patterns, affine transformations in the RGB channels), for a to-

† Organizer of the track. Dataset and evaluation measures are available at http://wwww.ge.imati.cnr.it/shrec14.

DOI: 10.2312/3dor.20141057

tal of 36 models. The last class contains 32 shapes, built on top of 8 null models by applying 3 deformations each.

An alternative dataset classification my be provided by grouping models according to texture, resulting in 13 classes. Some samples from the dataset are given in Figure 1.

The two contests. We distinguished between two contests, retrieval and classification; participants could opt to take part in both retrieval and classification, or only one of the two challenges. Prior to the publication of the dataset, a training set made of 96 models classified according to both geometry (16 classes) and texture (13 classes) was published. Each participant was asked to submit up to 3 runs for each contest (s)he intended to take part in, each run being either the result of a different algorithm, or of a different parameter setting. Also, the executable/source code used to produce the submitted runs was required.

Retrieval contest: Each model is used as a query against the rest of the dataset, with the goal of retrieving the most similar objects. In particular, two models are considered *highly similar* if they share both geometry and texture; they are *marginally similar* is they share only geometry; otherwise, they are dissimilar. For this track, a dissimilarity matrix (572x572) was requested, each element (i, j) being the dissimilarity value between models i and j in the whole dataset;

Classification contest: The goal is to assign the query to its class. In this case, two different classification matrices, one for the geometric and one for the texture classification were required, being each class one of those specified in the training set release. In each matrix, the element (i, j) is the probability that model i belongs to class j. Participants to the retrieval task could participate also to the classification one with the nearest neighbor (1-NN) classifier derived from their dissimilarities matrices. In this case the element $(i, j) = 1$ if i is classified in class j and 0 otherwise.

2. Participants

Eight groups[†] took part in this Shrec track:

1. M. Abdelrahman, M. El-Melegy and A. Farag (University of Louisville, USA) participated with two runs (**AEF1**, **AEF2**) in both the retrieval and the classification task. Their method is detailed in Section 3.1;
2. V. Garro and A. Giachetti (University of Verona, Italy), participated with three runs (**GG1**, **GG2**, **GG3**) in both the retrieval and the classification task. Their method is detailed in Section 3.2;
3. H. Y. Martono and M. Aono (Toyohashi University of Technology, Japan) participated with three runs (**HA1**, **HA2**, **HA3**) in the retrieval task, and with a further run (**HAc**) in the classification task. Their method is detailed in Section 3.3;
4. C. Li, A. Godil (NIST, USA) and A. Ben Hamza (Concordia University, Canada) participated with three runs (**LBG2**, **LBG3**, **LBG4**) in the retrieval task, and with two further runs (**Lc1**, **Lc2**) in the classification task. Their method is detailed in Section 3.4;
5. A. Tatsuma, M. Aono, C. Sanada (Toyohashi University of Technology, Japan) participated with run **TA** in the retrieval task. Their method is detailed in Section 3.5;
6. S. Velasco-Forero (National University of Singapore, Singapore) participated with three runs (**Ve1**, **Ve2**, **Ve3**) in the retrieval task. His method is detailed in Section 3.6;
7. C.-X. Xu, and Y.-J. Liu (Tsinghua University, the People's Republic of China) participated with run **XL** in the retrieval task. Their method is detailed in Section 3.7;
8. D. Giorgi (CNR, Italy) participated with three runs (**Gi1**, **Gi2** and **Gi3**) in both the retrieval and the classification task. Her method is detailed in Section 3.8;

3. Description of the methods

In this section we describe all the proposed methods.

3.1. Textured 3D models Classification using Scale Invariant Heat Kernels (M. Abdelrahman, M. El-Melegy, A. Farag)

Runs **AEF1** and **AEF2** rely on a combined shape description made of a geometric and a photometric contribution.

Geometry descriptor. For shape geometry description we present an approach based on scale invariant heat kernel signature (SI-HKS). Sun et al. [SOG09] proposed to use the heat kernel signature as a local shape descriptor

$$h(x,t) = H_t(x) = \sum_{i=1}^{\infty} e^{-\lambda_i t} \varphi_i^2(x),$$

where λ_i and φ_i are the eigenvalues and eigenfunctions of the Laplace-Beltrami operator, respectively. The HKS has several desired properties [SOG09]: it is intrinsic and thus isometry-invariant (two isometric shapes have equal HKS), multi-scale and thus captures both local features and global shape structure, and also informative: under mild conditions, if two shapes have equal heat kernel signatures, they are isometric. The proposed descriptor is based on a Bag-of-Feature representation of the HKS in frequency domain combined with the first 15 normalized eigenvalues of the Laplace-Beltrami operator. These eigenvalues have been proposed by Reuter et al. [RWSN09] as intrinsic shape descriptors (shape-DNA).

Scale invariance is a desirable property of the shape descriptor, which can be achieved by many ways. We propose a local scale normalization method based on simple operations. It was shown [BBGO11] that scaling a shape by a factor β results in changing $h(x,t)$ to $\beta^2 h(x,\beta^2 t)$. To deal with

[†] Research groups 3 and 4 could not provide any part of the code producing the submitted results.

Figure 1: *Some models in the dataset. Null models have been processed using MeshLab [Vis] and Remesh [AF06].*

this problem, we propose here to apply the Fourier transform (FT) as follows: taking the amplitude of the FT, the effect of the multiplicative constant β^2 is eliminated by normalizing $|H_t'(w)|$ by the sum of the amplitudes of the FT components. The amplitudes of the first significant FT components (we normally use 20) are employed to construct the scale-invariant shape descriptor. This proposed method eliminates the scale effect without having to use the noise-sensitive derivative operation or the logarithmic transformation that both were used in [BBGO11]. Thus our method is simpler, more computational-efficient and more robust to noise. Eventually we use the L_1-norm for classification.

Photometric Descriptor. The photometric descriptor is based on colour histograms, one for colour band (RGB). We use the L_1-norm as a distance measure between colour histograms. The query colour histogram is rotated to obtain the minimum distance. Then we get the mean distance measure for the three bands. The photometric distance measure is used to re-arrange the retrieval results from the geometric descriptor to retrieve similar texture objects first.

3.2. Colour Data and (colour-weighted) Histograms of Area Projection Transform (V. Garro, A. Giachetti)

To compute textured mesh differences we tested two approaches based on the Histograms of the Multiscale Area Projection Transform (MAPT) [GL12]. The method is based on a spatial map that encodes the likelihood of the points inside the shape of being centers of spherical symmetry. For each radius of interest σ, this map is computed as follows:

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

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):

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

with $\alpha(R) = 1/4\pi R^2$ and $\sigma(R) = c \cdot R$ $(0 < c < 1)$. 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 $R_1,...,R_n$. In the paper it is also shown that histograms of MAPT computed inside the objects are very good global shape descriptors, performing very good results 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) were concatenated creating an unique descriptor. Voxel side and sampled radii were chosen differently for each model, proportionally 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. To compare textured meshes we modified the method in two ways as follows.

Histograms of Area Projection Transform and colour Data. We computed the MAPT histograms with the same radii and sampling grids values as in [GL12]: the isotropic sampling grid is taken as being proportional to the cubic root of the volume of each model ($s = cbrt(V)/30$), and the sampled radii are integer multiples of s (10 values from $2s$ to $11s$). The radius σ is taken, as in the original paper, equal to $R/2$ for all the sampled R. Furthermore, we computed for each mesh the histograms of colour components in CIELab space. The colour quantization is set to 4 bins for the lightness component L^* and 8 bins for both the cromaticity values a^* and b^*, obtaining for each shape a histogram of size $4 \times 8 \times 8$. With this procedure each mesh is described by two histograms, the first one representing the geometric information and the second one representing the texture information. The total dissimilarity between two meshes is computed using a convex combination of the two histogram distances:

$$D(m_i,m_j) = \gamma\, d_j(m_i,m_j) + (1-\gamma) d_c(m_i,m_j)$$

where $0 \le \gamma \le 1$, $d_j(m_i,m_j)$ is the normalized Jeffrey divergence between the two MAPT histograms of meshes m_i and m_j and $d_c(m_i,m_j)$ corresponds to the normalized χ^2-distance of the two CIELab colour histograms. The runs **GG1** and **GG2** are obtained with γ set to 0.4 and 0.6, respectively.

Colour-weighted Histograms of Area Projection Transform. To get run **GG3**, a new descriptor is constructed by concatenating to the standard MAPT histograms, other three similar histograms obtained from colour-weighted MAPT maps, simply computed multiplying the area contribution of the surface elements by the red, green and blue components respectively (scaled in the range $[0,1]$). In this way

also the texture information should be encoded in the descriptor. We set the same radii and sampling parameters as the method described above, for each radius we now have 4 histograms (1 standard MAPT histogram and 3 colour-weighted MAPT histograms) with these choices the final descriptors of shapes are vectors composed by 480 elements. The dissimilarity between two meshes is obtained with the normalized Jeffrey divergence between the two corresponding descriptors.

A weakness of the proposed method, that is common to the original method, is that histograms do not encode information on the relative position of the symmetric parts and on their number.

3.3. Local texture descriptors (H. Y. Martono, M. Aono)

In our aproach, we have investigated a good combination of Histogram of Oriented Gradients (HOG) [DT05], Local Binary Pattern (LBP) [OPM02], Local Ternary Pattern (LTP) [ZJHM13], and Weber Local Descriptor (WLD) [CSH*10].

As pre-processing, we apply pose normalization [TA09] and generate multiple colour shaded images for texture features by using jrman [jrm], where we take the rotation angles of -90, -45, +45, +90 on each axis x, y, z plus original pose to obtain shade images from 13 different views in total. On the other hand, for shape features we generate Fourier spectra from MFSD [TA09] for each 3D object. After preprocessing, we apply Gabor filters to the images, attempting to sharpen the texture difference. RGB channels are separated after filtering. For each colour channel, we extract features based on LPB, LTP, WLD, and HOG respectively.

In this event 3 run methods are proposed: HOG is combined in turn with LPB (run **HA1**), LTP (run **HA2**) and WLD (run **HA3**). All methods are sketched in Figure 2.

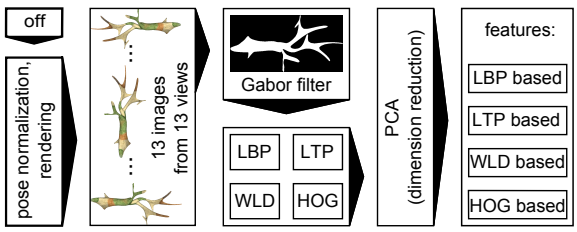

Figure 2: *Flow feature extraction process.*

LBP is a powerful feature for texture classification. The basic idea for developing the LBP operator was that 2-dimensional surface textures can be described by two complementary measures: local spatial patterns and colour contrast. The original LBP operator form labels for the image pixels by thresholding the 3×3 neighborhood of each pixel with the center value and considering the result as a binary number. At this time, we use another extension of the original operator called LBP uniform pattern. In this method, there are a total of 256 patterns, 58 of which are uniform, which yield in 59 different labels.

LTP is an extension of LBP. Unlike LBP, it does not threshold the pixels into 0 and 1, rather it uses a threshold constant to threshold pixels into three values.

WLD is also a very powerful and robust local descriptor, consisting of two components: differential excitation and orientation. It is computed based on the ratio between the two terms: one is the relative intensity differences of a current pixel against its neighbor (3×3); the other is the intensity of the current pixel.

HOG is a feature descriptor counting the occurrences of gradient orientation in localized portions of an image. By using this descriptor, local object appearance and shape within an image can be described by the distribution of intensity gradients or edge direction. The implementation of these descriptors can be achieved by dividing the image into small connected regions called cells, and for each cell compiling a histogram of gradient directions or edge orientation for the pixels within the cells.

3.4. Spectral geometry based methods for textured 3D shape retrieval (C. Li, A. Godil, A. Ben Hamza)

We use the spectral geometry based framework in [Li13] for textured 3D 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) feature extraction. We use spectral graph wavelet signature [LH13b] to capture geometry information, and colour histogram for texture information; (2) spatial sensitive shape comparison via intrinsic spatial pyramid matching [LH13a]. The cotangent weight scheme was used to discretize LBO. The eigenvalues λ_i and associated eigenfunctions φ_i can be computed by solving the generalized problem $C\varphi_i = \lambda_i A\varphi_i, \quad i = 1,2,\ldots,m$, where A is a positive-definite diagonal area matrix and C is a sparse symmetric weight matrix. We set $m = 200$ in our experiments.

Feature extraction. The first stage consists of the computation of an informative descriptor $h(x)$ at each vertex of the triangle meshed shape X. We use spectral graph wavelet signature [LH13b] to capture geomtery information, and colour histogram for texture information.

Geometry information. In general, any one of spectral descriptors with the eigenfunction-squared form reviewed in [LH13c] can be used in our spatial partition context 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 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 (cf. [LH13b, Eq. (20)] for a precise formulation of g). The resolution level is set as 2. With a dictionary learned by k-means, the descriptor $S = \{s_t,\ t = 1,2,\ldots,T\}$ at each point of the shape is replaced by the Gaussian kernel based soft assignment $Q = \{q_k,\ k = 1,2,\ldots,K\}$. $k = 100$ in our experiment.

Texture information. We simply choose colour histogram (CH) to characterize texture information on the surface. Each channel is discretized into 5 bins.

Intrinsic spatial pyramid matching. 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

$$\lambda_2 = \inf_{f \perp \varphi_1} \frac{f'Cf}{f'Af}.$$

We use the isocontours of the second eigenfunction (Figure 3) to cut the shape into R patches, thus the shape description is the concatenation of R sub-histograms of SGWS and CH 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 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].

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

The partition resolution level is set as 5 (R = 16 patches) in this contest. Therefore, the description power of SGWS and CH is enhanced by incorporating this spatial information. Finally, we get ISPM induced histograms for shape representation. The dissimilarity between two shapes is computed as the L_2 distance. We manipulate geometry and texture information seperately to obtain one dissimilarity matrix for each. Run **LBG3** represents spatial partition level 1 (1 patch), runs **LBG2** and **LBG4** represents spatial partition level 5 (the latter is a weighted sum of geometric and texture contribution, with coefficients 0.8 and 0.2, respectively). In classification, the distance between query and class is represented as the minimum distance between the query and instances in the class. The distance is then transformed to similarity via an exponential kernel. The label probability is obtained by normalizing the similarity with L_1 norm (Run **Lc1** and **Lc2** represents spatial partition levels 1 and 5, respectively).

Running time. The method is implemented in MATLAB. The time consuming steps are the computation of LBO and k-means dictionary learning. For a mesh with 14,000 vertices, it takes 7.7 seconds to compute the LBO. To learn a dictionary with 100 geometric words, it takes 14 minutes.

3.5. Multiresolution Representation Local Binary Pattern Histograms (A. Tatsuma, M. Aono, C. Sanada)

Run **TA** is based on the Multiresolution Representation Local Binary Pattern Histograms (MRLBPH), a new 3D model feature that captures textured features of rendered images from 3D model by analyzing multiresolution representations using Local Binary Pattern (LBP) [OPH96].

Figure 4 illustrates the generation of our proposed MRLBPH feature. We enclose the 3D model within a unit geodesic sphere after normalizing the 3D model via Point SVD [TA09]. From each vertex of the unit geodesic sphere, we render depth and colour buffer images with 256×256 resolution; a total of 38 viewpoints are defined. We process a depth channel and each channel of the RGB colour space as described below.To obtain multiresolution representations, we apply a Gaussian filter with varying scale parameters to an image. In our implementation, we select 3, 7, 11, 19, and 43 for the scale parameters. For each scale image, we cal-

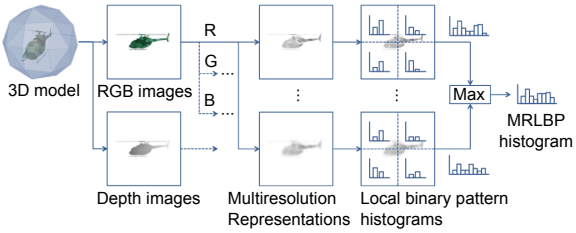

Figure 4: *Overview of our Multiresolution Representation Local Binary Pattern Histograms (MRLBPH)*

culate an LBP histogram. Let g_c denote the image value at arbitrary pixel (x,y), and let $g_1,\ldots,g_8$ be the image values of each of the eight neighborhood pixels. The LBP value is then calculated as

$$LBP(x,y) = \sum_{i=1}^{8} s(t, g_i - g_c) \cdot 2^{i-1},$$

where $s(t,g)$ is a threshold function defined as 0 if $g < t$ and 1 otherwise. In our implementation, we set the threshold value t to 0, and quantize LBP values into 64 bins. To incorporate spatial location information, we partition the image into 2×2 blocks and calculate the LBP histogram at each block. The LBP histogram of each scale image is obtained

by concatenating the histograms of these blocks. We merge the histograms of the scale images by selecting the maximum value of each histogram bin.

For each viewpoint, an MRLBP histogram is obtained by concatenating the histograms of depth and colour buffer images and then normalized by L_1 normalization.

To compare two 3D models, we apply the Hungarian method [Kuh55] to all pair dissimilarities between their MRLBP histograms. To calculate the dissimilarity between two histograms **a** and **b** with n bins, we use Jeffrey's divergence defined as

$$d(\mathbf{a},\mathbf{b}) = \sum_{i=1}^{n} \left(a_i \log \frac{a_i}{m_i} + b_i \log \frac{b_i}{m_i} \right),$$

where $m_i = (a_i + b_i)/2$.

3.6. Colour + Shape descriptors (S. Velasco-Forero)

The proposed method is a modification of the 3D Shape + colour descriptor proposed in [CBA*13]. We make use of two main components of the textured shape.

1. First, we compute the geodesic distance matrix in the mesh information [SSK*05], denoted by **G**. Following the same intuition than in [SFH*09] we use a spectral representation of the geodesic distance as descriptor. However, we have used a centralised geodesic matrix [MSS*99], defined as follows, $\mathbf{D} = \mathbf{G} - \mathbf{1}_N\mathbf{G} - \mathbf{G}\mathbf{1}_N + \mathbf{1}_N\mathbf{G}\mathbf{1}_N$ where $\mathbf{1}_N$ denotes a N by N matrix that for each component takes value $1/N$ and N is the number of vertices in the mesh. As in [CBA*13], the first 40 eigenvalue are used as shape descriptor. This vector of eigenvalues is denoted by $\lambda(\mathbf{D})$. Many distances can be computed from two shapes from $\lambda(\mathbf{D_i})$ and $\lambda(\mathbf{D_j})$ [SFH*09]. However, we have followed the recommendation on [SFH*09] by using the *mean normalized Manhattan distance*, i.e.,
$$\mathtt{D}_{eig}(S_i,S_j)) = \sum_{k=1}^{40} \frac{2|\lambda_k(\mathbf{D_i}) - \lambda_k(\mathbf{D_j})|}{\lambda_k(\mathbf{D_i}) + \lambda_k(\mathbf{D_j})}.$$
2. Second, we incorporate texture information in the shape descriptor. From [CBA*13], histograms of colour information in RGB space have shown good performance as texture descriptor. Accordingly, we define the distance from the colour representation by
$$\mathtt{D}_{RGB}(S_i,S_j)) = \sum_{k=R,G,B} \frac{EMD(\mathtt{hist}(S_i^k), \mathtt{hist}(S_j^k))}{3},$$
where *EMD* denotes the Earth mover's distance between the histogram in the colour space. In our experiments, we have used the fast implementation provided by [PW09].
3. Finally, for a given couple of textured shapes, the proposed distance is computed as follows:
$$\mathtt{D}(S_i,S_j) = (\mathtt{D}_{RGB}(S_i,S_j))^{1-p} + (\mathtt{D}_{eig}(S_i,S_j))^{p},$$
where p is a parameter to control the trade-off between colour and shape information. In the experiment, $p = 0.7$ (run **Ve1**), 0.8 (run **Ve2**), 0.95 (run **Ve3**), in accordance with the intuition that geometry is more important than colour in how humans interpret shape similarity. A scheme of the proposed descriptors is shown in Fig. 5.

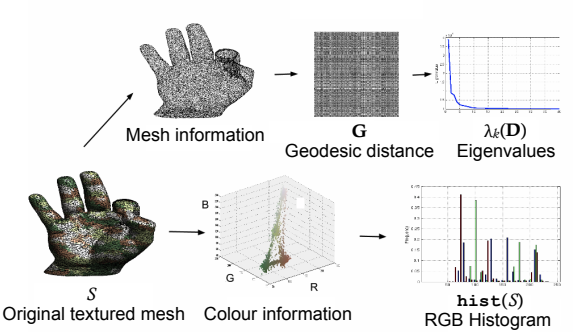

Figure 5: *Proposed method includes a shape descriptor from the geodesic distance matrix and a colour descriptor from the histogram representation of RGB colour information. Details are included in section 3.6.*

3.7. Measuring Distance between 3D Models Based on Geometry and colour Features (C.-X. Xu, Y.-J. Liu)

There are serveral categories of retrieval methods based on the input, including keywords, 2D images and 3D models [LZL*12]. [LLJ*13] and [LFLF12] propose a sketch-based method, which belongs to the 2D image manners but applies to some specific cases such as 3D CAD design process.

Since the input here is 3D models, we use a method similar with the one proposed in [LZL*12], using geodesic distance to replace simple Euclidean distance. We sample the model on its surface in N-dimensional space $(N > 3)$, which includes both geometric and textural information, then these sampling points are optimally clustered. A generally defined geodesic distance is computed among the points, and we can get the shape distribution of the model. By comparing the shape distributions we get the dissimilarity between two models, resulting in run **XL**.

Extend Geodesic to N-dimension. We extend the concept of geodesic to N-dimension $(N > 3)$ by constructing the model as follows: if a model is constructed in a N-dimensional space $(N \geq 1)$ where the metric between any two points are defined (such as L_2 norm), then the edge lengths follows the metric definition while other topological relationships (such as connections between vertices) are kept the same.

In this case, a tuple (x,y,z,r,g,b,a) is used to represent a vertex where x, y z are its 3D coordinates and r, g, b, a are its colour components. For each edge, the L_2 norm distance between the two endpoints (vertices) are calculated as

its length. Thus we get a model which has the same connections between vertices in a higher dimension ($N = 7$).

We use a window-propagation based method to calculate the geodesic distance between any two points on the surface of a model, see [Liu13] for details. Since this algorithm is based on the unfolding of faces, the calculation of geodesic remains mostly the same with colour information included in a high dimension space.

Construct and Compare Shape Distribution. The main procedure of our method is as follows: We sample points on the surface of the model in the constructed 7-dimensional space. These sampling points are clustered using a modified ISODATA algorithm. Lastly, we calculate the feature histogram of each model using these clustered sampling points, and we can get a shape distribution for each model. See more details in [LZL*12].

Suppose that for a model there are c clusters and F_i is the set of feature points in the ith cluster with n_i points. For each point$f_{ip} \in F_i$ and $f_{jq} \in F_j$, we compute the geodesic distance $d_{i_p j_q} = \|f_{ip} - f_{iq}\|$, where $i \neq j$, $i, j = 1, 2, ..., c$, $p = 1, 2, ..., n_i$, $q = 1, 2, ..., n_j$, and store all the distances in an array D. Then we convert the normalized array D into a histogram and a shape distribution for each model is constructed.

There are some ways mentioned in [OFCD01] to measure the difference between shape distributions. In our experiment, we choose the Minkowski L_2 norm of the probability density functions to measure the shape distributions, i.e. $D(f,g) = \sqrt{\int |f-g|^2}$, which is obviously a metric.

Since the probability density function is represented by the shape distribution generated above (combined with another parameter to create different levels of approaching, see details in [OFCD01]), the function is piecewise linear, which leads to a direct calculation of a second (or first) degree polynomial and is easy to implement.

3.8. Textured shape distribution, joint histograms and persistence (D. Giorgi)

Run Gi1. The Textured Shape Distribution (TSD) descriptor is a variant on classical Shape Distributions (SD) [OFCD01]. TSD consists of the distribution of colour-aware geodesic distances computed between points sampled over the surface mesh representing the 3D model.

First, the surface mesh is embedded in the three-dimensional CIELab colour space, so that the coordinates of the vertices are (L,a,b), where L specifies luminosity and a, b specify colour. The length of an edge is the distance between its endpoints, namely, the CIE94 distance defined for CIELab coordinates [Fai05]. Then, the colour-aware geodesic distances are computed in the embedding space with the metric induced by the CIE94 distance. The use of the CIELab space rather than the RGB space for representing colour is suggested by the fact that CIELab is a perceptually uniform space, that is, uniform changes of coordinates in the CIELab space correspond to uniform changes in the colour perceived by the human eye. The use of the CIE94 distance instead of a classical Euclidean distance is also aimed at respecting perceptual uniformity.

TSD encodes the distribution of colour distances, yet it also takes into account the connectivity of the underlying model, as distances are computed by walking on the surface model. In this sense, TSD can be considered as an hybrid descriptor, taking into account both colorimetric and geometric information. In the current implementation, a set of 1024 points was sampled over each surface model following a farthest-point criterion. The colorimetric geodesic distances between pairs of samples were computed following the Dijkstra algorithm . The distribution of these distances was discretized using a histogram of 64 bins. Histograms were compared using the L_2 norm.

Run Gi2. Since TSD does not take into account the length of edges in the Euclidean space, it loses part of the geometric information. This information can be added through a joint distribution of colorimetric geodesic distances and classical geodesic distances computed on the surface embedded in the Euclidean space. In this run, a 16×16 bi-dimensional, joint histogram (JH) was computed for each 3D model. The L_2 norm was used for comparison. The distance matrix is the sum of the distance matrix obtained using the TSD descriptor and the distance matrix obtained using the JH descriptor.

Run Gi3. AS TSD can be seen as an hybrid descriptor, it makes sense to combine it with a purely geometric descriptor and a purely colorimetric descriptor, in line with what proposed in [BCGS13]. Hence Run Gi3 combines TSD with the popular Spherical Harmonic (SH) descriptor [KFR03], and the persistence-based purely colorimetric descriptor of the PHOG signature in [BCGS13], which computes persistence spaces based on the coordinates in the CIELab colour space. The distance matrix corresponding to this run is the sum of the three distance matrices obtained using the TSD descriptor, the SH descriptor, and the persistence-based descriptor of PHOG, respectively.

4. Evaluation measures and results

Retrieval performance measures. The retrieval performance of each submitted run has been evaluated according to the following relevance scale: If a retrieved object shares both shape and texture with the query, then it is highly relevant; if it shares only shape, it is considered marginally relevant; otherwise, it is not relevant. The evaluation process has been based on the following evaluation measures: Average precision-recall curves, Nearest Neighbor (NN), First Tier (FT), Second Tier (ST), Normalized Discounted Cumulated Gain (NDCG) and Average Dynamic Recall (ADR).

Note that, because of the multi-level relevance assessment of each query, most of the evaluation measures have been split up as well. "Highly relevant" evaluation measures relate to the highly relevant items only, while "relevant" evaluation measures are based on all the relevant items (highly relevant items + marginally relevant items). We provide a brief explanation of each evaluation measure, together with the associated evaluation results. The runs of all the track's participants are labeled as specified in Section 2.

Average precision-recall curves. Precision is the fraction of retrieved items that are relevant to the query. Recall is the fraction of the items relevant to the query that are successfully retrieved. Being A the set of all the relevant objects and B the set of all the retrieved object,

$$\text{Precision} = \frac{|A \cap B|}{|B|}, \quad \text{Recall} = \frac{|A \cap B|}{|A|}.$$

Plotting the two quantities in the reference frame recall vs. precision, we get a curve: The larger the area below such a curve, the better the performance under examination. In particular, the precision-recall curve of an ideal retrieval system would result in a constant curve equal to 1. For each query, we thus have a precision-recall curve. By taking the average on all the queries, we get the average precision-recall curve. Figure 6 shows the performances of the best run for each participant with respect to the average precision-recall curve, both "relevant" and "highly relevant".

Nearest Neighbor, First tier and Second tier. These evaluation measures aim at checking the fraction of models in the query's class also appearing within the top k retrievals. Specifically, for a class with $|C|$ members, $k = 1$ for the nearest neighbor (NN), $k = |C| - 1$ for the first tier (FT), and $k = 2(|C| - 1)$ for the second tier (ST). The final score, always ranging in $[0, 1]$, is an average over all the models in the database. Table 1 reports the performances for all the runs according to these measures, with respect to the "relevant" and "highly relevant" classifications.

Average dynamic recall. The idea is to measure how many of the items that should have appeared before or at a given position in the result list actually have appeared. The average dynamic recall (ADR) at a given position averages this measure up to that position. Precisely, for a given query let A be the set of highly relevant classified items, and let B be the set of relevant items. Obviously $A \subseteq B$. The ADR is computed as:

$$\text{ADR} = \frac{1}{|B|} \sum_{i=1}^{|B|} \frac{r_i}{i},$$

where r_i is defined as

$$r_i = \begin{cases} \dfrac{|\{\text{highly relevant items in the first } i \text{ retrieved items}\}|}{i}, & \text{if } i \leq |A|; \\ \dfrac{|\{\text{relevant items in the first } i \text{ retrieved items}\}|}{i}, & \text{if } i > |A|. \end{cases}$$

For all participants, the last column of Table 1 reports the ADR measure averaged on all queries.

	Relevant			Highly Relevant			
Run	**NN**	**FT**	**ST**	**NN**	**FT**	**ST**	**ADR**
AEF1	0.732	0.408	0.521	0.098	0.226	0.350	0.205
AEF2	0.735	0.408	0.521	0.123	0.228	0.351	0.207
GG1	0.711	0.240	0.324	0.696	0.404	0.530	0.349
GG2	0.740	0.265	0.365	0.722	0.432	0.557	0.368
GG3	0.822	0.344	0.469	0.665	0.384	0.504	0.336
HA1	0.721	0.271	0.382	0.424	0.265	0.350	0.249
HA2	0.736	0.282	0.387	0.468	0.278	0.362	0.260
HA3	0.737	0.277	0.386	0.462	0.272	0.358	0.257
LBG2	0.841	0.402	0.553	0.676	0.412	0.565	0.353
LNG3	0.804	0.374	0.512	0.512	0.306	0.406	0.280
LBG4	0.909	0.430	0.559	0.394	0.325	0.437	0.284
TA	**0.963**	**0.436**	**0.562**	0.563	0.336	0.456	0.294
Ve1	0.902	0.404	0.512	0.735	0.396	0.540	0.342
Ve2	0.918	0.398	0.499	0.593	0.338	0.469	0.300
Ve3	0.909	0.372	0.469	0.336	0.275	0.369	0.248
XL	0.348	0.175	0.272	0.108	0.149	0.192	0.159
Gi1	0.895	0.259	0.374	0.890	0.324	0.401	0.313
Gi2	0.906	0.280	0.388	**0.894**	0.366	0.448	0.340
Gi3	0.958	0.383	0.504	0.814	**0.455**	**0.590**	**0.383**

Table 1: *Retrieval performances on the whole dataset. For each evaluation measure, best results are in bold text.*

Normalized discounted cumulated gain. It is first convenient to introduce the *discounted cumulated gain (DCG)*. Its definition is based on two assumptions. First, highly relevant items are more useful if appearing earlier in a search engine result list (have higher ranks); Second, highly relevant items are more useful than marginally relevant items, which are in turn more useful than irrelevant items. Precisely, the DCG at a position p is defined as:

$$\text{DCG}_p = \text{rel}_1 + \sum_{i=2}^{p} \frac{\text{rel}_i}{\log_2(i)},$$

with rel_i the graded relevance of the result at position i. Obviously, the DCG is query-dependent. Therefore, we normalize the DCG to get the *normalized discounted cumulated gain (NDCG)*. This is done by sorting elements of a retrieval list by relevance, producing the maximum possible DCG till position p, also called *ideal DCG (IDCG)* till that position. For a query, the NDCG is computed as

$$\text{NDCG}_p = \frac{\text{DCG}_p}{\text{IDCG}_p}.$$

In the present evaluation, the NDCG values for all queries are then averaged to obtain a measure of the average performance for each submitted run. Note that for an ideal run, we would have $\text{DCG}_p = \text{IDCG}_p$ and hence $\text{NDCG} = 1$.

Figure 7 shows the performance evaluation for all runs according to the NDCG measure as a function of the rank p.

Classification performance measures. In the classification task, the participants were asked to submit a classification matrix C, both for geometry and texture: the element $C(i, j)$ is the probability that model i belongs to the class j. As a performance measure, we consider the *classification rate* (CR), i.e. the percentage of models correctly

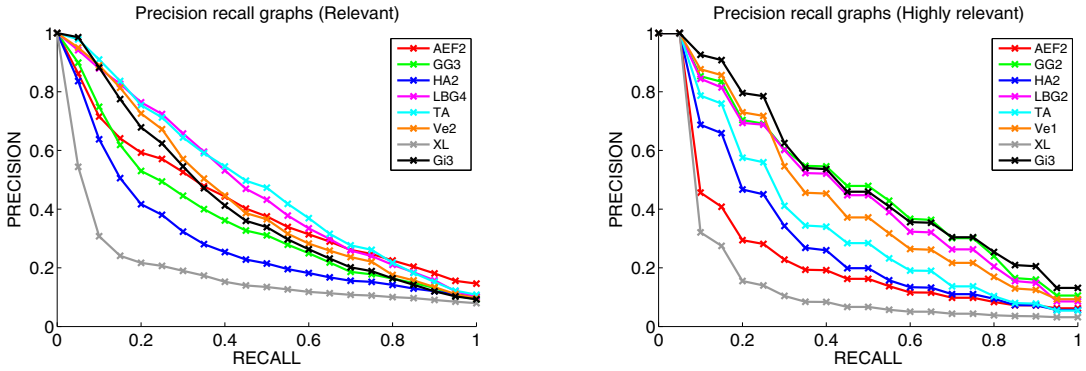

Figure 6: *Performances of the best run for each participant w.r.t to the average precision-recall curve, both relevant and highly relevant.*

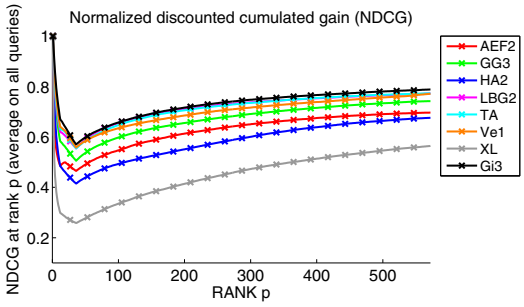

Figure 7: *Performances of all the runs w.r.t. the NDCG measure (run **LBG2** is almost totally covered by run **Gi3**).*

classified, either for geometry or texture. A score 1.0 means that every item is correctly classified. For a percentage matrix C, each row is normalized so that $\max_j C(i,j) = 1$. In case C is obtained with the 1-NN classifier, $C(i,j)$ is 1 if model i is classified as belonging to the class j, and 0 otherwise. Then, we consider as classification rate the number $R = \frac{1}{N}\sum_{i=1}^{N} C(i, ground(i))$, with N the number of elements in the dataset, and $C(i, ground(i))$ the "probability" that model i is recognized as belonging to its correct class. Table 2 reports the performances of those methods involved in the classification task.

Run	***GCR*** / ***TCR***	**GG1**	0.710 / **0.937**	**GG2**	0.740 / 0.934	**GG3**	0.822 / 0.734
HAc	0.524 / 0.661	**Gi1**	0.895 / 0.906	**Gi2**	0.906 / 0.916	**Gi3**	**0.968** / 0.843
AEF1	0.733 / 0.128	**AEF2**	0.736 / 0.149	**Lc1**	0.792 / 0.854	**Lc2**	0.915 / 0.911

Table 2: *For each involved method, the geometric classification rate (GCR) and the texture classification rate (TCR) are displayed. Best results are in bold text.*

5. Discussion and conclusions

In this paper, the SHREC'14 track on *Retrieval and Classification on Textured 3D Models* is introduced, whose aim was to evaluate retrieval algorithms dealing with an emerging type of content, namely textured 3D objects, which we believe deserve attention from the research community. Indeed, the abundance of textured models in Computer Graphics, the advance in 3D shape acquisition technology which makes it possible to obtain textured 3D shapes of even moving objects, the importance of colour features in 3D Shape Analysis applications, together call for shape descriptors which take into consideration colorimetric information. This track extends the SHREC'13 track [CBA*13] in terms of number of models considered, of type of deformations (both in geometry and texture) and performance evaluation (retrieval and classification).

We believe that the success of this track on textured 3D model retrieval, which saw the participation of eight research groups, demonstrated that this field of research is drawing increasing attention. Also, the experimental results offer several hints for discussion.

NDCG and ADR provide an *overall* evaluation of the proposed methods in interpreting the 2-level classification of the dataset. On the one hand, the NDCG results in Figure 7 show encouraging results from almost all the runs submitted to the track. On the other hand, the ADR results in Table 1 emphasize that the dataset was challenging: even if we argue that this is in part due to the intrinsic dataset structure, characterized by highly populated classes, such result also reveals that there is still a long road ahead in the challenging field of textured shape analysis. An interesting insight is offered by the "highly relevant" analysis shown in Tables 1 and 2, and Figure 6: indeed, very good results were achieved by those methods dealing with texture information in the CIELab rather than in the RGB colour space, allowing for a representation of colour that is more robust to the texture deformations proposed in this track.

Finally, we hope this new benchmark may help other researchers improve their techniques, as well as promote further investigation on the benchmarking of algorithms dealing with textured shapes.

Acknowledgments. Work developed in the CNR research activity ICT.P10.009, and partially supported by VISIONAIR, European project "FP7 INFRASTRUCTURES" (2011-2015), and the Italian CNR Flagship project INTEROMICS: InterOmics PB05, research unit WP 15.

References

[AF06] ATTENE M., FALCIDIENO B.: Remesh: An interactive environment to edit and repair triangle meshes. In *Proc. SMI'06* (2006), IEEE Computer Society, p. 41. 3

[BBGO11] BRONSTEIN A. M., BRONSTEIN M. M., GUIBAS L. J., OVSJANIKOV M.: Shape google: Geometric words and expressions for invariant shape retrieval. *ACM Trans. Graph. 30*, 1 (2011), 1:1–1:20. 2, 3

[BCGS13] BIASOTTI S., CERRI A., GIORGI D., SPAGNUOLO M.: Phog: Photometric and geometric functions for textured shape retrieval. *Comput. Graph. Forum 32*, 5 (2013), 13–22. 7

[CBA*13] CERRI A., BIASOTTI S., ABDELRAHMAN M., ANGULO J., BERGER K., CHEVALLIER L., EL-MELEGY M., FARAG A., LEFEBVRE F., GIACHETTI A., GUERMOUD H., LIU Y.-J., VELASCO-FORERO S., VIGOUROUX J., XU C.-X., ZHANG J.-B.: SHREC'13 Track: Retrieval on Textured 3D Models. In *Proceedings of the 6th Eurographics Conference on 3D Object Retrieval* (Girona, Spain, 2013), EG 3DOR'13, Eurographics Association, pp. 73–80. 1, 6, 9

[CSH*10] CHEN J., SHAN S., HE C., ZHAO G., PIETIKAINEN M., CHEN X., GAO W.: Wld: A robust local image descriptor. *IEEE T. Pattern Anal. 32*, 9 (2010), 1705–1720. 4

[DT05] DALAL N., TRIGGS B.: Histograms of oriented gradients for human detection. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on* (2005), vol. 1, pp. 886–893 vol. 1. 4

[Fai05] FAIRCHILD M. D.: *Color appearance model*. 2005. 7

[GL12] GIACHETTI A., LOVATO C.: Radial symmetry detection and shape characterization with the multiscale area projection transform. *Comput. Graph. Forum 31*, 5 (2012), 1669–1678. 3

[jrm] URL: http://www.jrman.org/. 4

[KFR03] KAZHDAN M., FUNKHOUSER T., RUSINKIEWICZ S.: Rotation invariant spherical harmonic representation of 3d shape descriptors. In *Proceedings of the Eurographics/ACM SIGGRAPH Symposium on Geometry Processing* (2003), pp. 156–164. 7

[Kuh55] KUHN H. W.: The hungarian method for the assignment problem. *Naval Research Logistics Quarterly 2* (1955), 83–97. 6

[LFLF12] LIU Y.-J., FU Q.-F., LIU Y., FU X.-L.: A distributed computational cognitive model for object recognition. *Science China (Series F): Information Sciences 56*, 9 (2012), 1–13. 6

[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* (Aire-la-Ville, Switzerland, Switzerland, 2011), EG 3DOR'11, Eurographics Association, pp. 79–88. 3

[LH13a] LI C., HAMZA A. B.: Intrinsic spatial pyramid matching for deformable 3d shape retrieval. *Int. J. Multimed. Inf. Retr. 2*, 4 (2013), 261–271. 4, 5

[LH13b] LI C., HAMZA A. B.: A multiresolution descriptor for deformable 3d shape retrieval. *Visual Comput.* (2013), 1–12. 4, 5

[LH13c] LI C., HAMZA A. B.: Spatially aggregating spectral descriptors for nonrigid 3d shape retrieval: a comparative survey. *Multimedia Syst.* (2013), 1–29. 4

[Li13] LI C.: *Spectral Geometric Methods for Deformable 3D Shape Retrieval*. Master's thesis, Concordia University, 2013. 4

[Liu13] LIU Y.-J.: Exact geodesic metric in 2-manifold triangle meshes using edge-based data structures. *Comput. Aided Design 45*, 3 (2013), 695 – 704. 7

[LLJ*13] LIU Y.-J., LUO X., JONEJA A., MA C.-X., FU X.-L., SONG D.: User-adaptive sketch-based 3-d cad model retrieval. *IEEE T. Autom. Sci. Eng. 10*, 3 (2013), 783–795. 6

[LZL*12] LIU Y.-J., ZHENG Y.-F., LV L., XUAN Y.-M., FU X.-L.: 3d model retrieval based on color + geometry signatures. *Visual Comput. 28*, 1 (2012), 75–86. 6, 7

[MSS*99] MIKA S., SCHÖLKOPF B., SMOLA A., MÜLLER K.-R., SCHOLZ M., RÄTSCH G.: Kernel pca and de-noising in feature spaces. In *Advances in Neural Information Processing Systems II* (1999), MIT Press, pp. 536–542. 6

[OFCD01] OSADA. R., FUNKHOUSER. T., CHAZELLE. B., DOBKIN D.: Matching 3d models with shape distributions. In *Proc. SMI'01, IEEE Computer Society* (2001), pp. 154–166. 7

[OPH96] OJALA T., PIETIKÄINEN M., HARWOOD D.: A comparative study of texture measures with classification based on featured distributions. *Pattern Recogn. 29*, 1 (1996), 51–59. 5

[OPM02] OJALA T., PIETIKAINEN M., MAENPAA T.: Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE T. Pattern Anal. 24*, 7 (2002), 971–987. 4

[PW09] PELE O., WERMAN M.: Fast and robust earth mover's distances. In *ICCV* (2009), IEEE, pp. 460–467. 6

[RWSN09] REUTER M., WOLTER F.-E., SHENTON M., NIETHAMMER M.: Laplace-beltrami eigenvalues and topological features of eigenfunctions for statistical shape analysis. *Comput. Aided Design 41*, 10 (2009), 739–755. 2

[SFH*09] SMEETS D., FABRY T., HERMANS J., VANDERMEULEN D., SUETENS P.: Isometric deformation modelling for object recognition. In *Proc. CAIP '09* (Berlin, Heidelberg, 2009), Springer-Verlag, pp. 757–765. 6

[SOG09] SUN J., OVSJANIKOV M., GUIBAS L.: A concise and provably informative multi-scale signature based on heat diffusion. In *Proc. SGP '09* (2009), Eurographics Association, pp. 1383–1392. 2

[SSK*05] SURAZHSKY V., SURAZHSKY T., KIRSANOV D., GORTLER S. J., HOPPE H.: Fast exact and approximate geodesics on meshes. *ACM Trans. Graph. 24*, 3 (July 2005), 553–560. 6

[TA09] TATSUMA A., AONO M.: Multi-fourier spectra descriptor and augmentation with spectral clustering for 3d shape retrieval. *Visual Comput. 25*, 8 (2009), 785–804. 4, 5

[Vis] VISUAL COMPUTING LAB – ISTI – CNR: Meshlab. http://meshlab.sourceforge.net/. 3

[ZJHM13] ZHAO Y., JIA W., HU R.-X., MIN H.: Completed robust local binary pattern for texture classification. *Neurocomputing 106* (2013), 68 – 76. 4