Project Report: Intrinsic Shape Context Descriptors for Deformable Shapes

Juan M. Pérez Rúa and Jilliam M. Díaz Barros, *Member, IEEE*

***Abstract*— In this document an approach to describe and classify non-rigid 3D shapes, represented as triangular meshes, is presented. This approach is based on intrinsic shape context descriptor (ISC). The heat kernel signature is used as field function, and for the shape context descriptor, a histogram with polar system of coordinates is implemented. The inward ray shooting surface charting function is employed to build the polar grids, by using geodesic distances. It is demonstrated that this descriptor is a useful parameter to classify different shapes. The SHREC database was used in order to determine the ISC performance.**

***Index Terms*—3D Processing, heat kernel signature, non-rigid matching, shape context descriptor.**

I. Introduction

THIS document summarizes the results of the implementation of the Shape Context descriptor in 3D shapes, using the Heat Kernel Signature, commonly abbreviated as HKS. The HKS is a largely known intrinsic descriptor for 3D manifolds based on the physics of heat transfer between points on a surface. This is why using HKS along with the shape context (SC) is likely to be an intrinsic, informative, and simplified descriptor for shapes. In order to show this implementation, the present work is organized as follows. In section II, related work with characterization models of 2-D and 3-D shapes are presented. Next, for a better understanding of the design and implementation of the project, a summary of technical terminology is introduced. Then, discussion on implementation is presented in section IV. Different experiments and their results are shown in section V, while at the end, some conclusions are reported.

II. Related Work

Below are presented different works related with the intrinsic shape context descriptors for non-rigid shapes.

Belongie *et al.* presented in [1] 2D and 3D shape descriptor called *shape context*, which enables to obtain information about the distribution of shape contour points in relation to a point. This information can be used as a descriptor to measure similarities among shapes and find correspondences, by using the chi-squared statistic in a cost function (for 2D shapes) or K-medoid algorithm (for 3D objects in 2D images), and the standard least squares methods, respectively. The distribution of points is computed with a histogram, using a log-polar coordinate system. The approach presented high tolerance to deformations, such as translation, scale and rotation.

In [2], Sun *et al.* proposed a point signature named *Heat Kernel Signature* or *HKS*, which considers the heat propagation model in the temporal domain to characterize shapes. The HKS is based in the heat kernel, related with Laplace-Beltrami operator, preserving some of its properties such as being stable against perturbations on the shape. It is used in this case to identify and classify salient features on different shapes. Three of the main advantages of this signature are that it gives a complete representation of intrinsic geometry of the shape; also the information about neighborhoods of a given point, besides it can be implemented in non-rigid shapes with high-accuracy results.

Considering the previous works, Kokkinos *et al.* presented in [3] a new approach to characterize 3D shapes by using intrinsic shape context (ISC) descriptors, based on HKS and an enhanced version of it, scale-invariant HKS (SIHKS). On each shape, they used the cotangent weight scheme to compute the first 300 eigenvalues and eigenvectors of the Laplace-Beltrami operator. A modified log (log-polar) system of coordinates for 3D shapes was implemented to construct the ISC descriptors, with a radius equal to 20, 5 linearly spaced radial bins and 16 angular bins. The surface charting function used for this system was the outward ray shooting, considering the geodesic distance from the analyzed point. In order to eliminate variations due to rotation, they implemented Fourier transform modulus. The system demonstrated to be invariant to surface resolution, as well as for other parameters such as scaling, microholes, noise, modifications in topology, and isometry, with better performance than their counterparts HKS and SIHKS.

III. Theoretical Framework

A. Shape context descriptor

It was introduced in 2000 by Belongie and Malik in [4], as a feature descriptor for planar shapes matching and retrieval.

The main idea of the shape context descriptor is to obtain a representation of the relative distributions of points on a shape with regard to each point, by considering a sampled version of its contours (internal and external). A log-polar histogram is computed in all the points of the edge map, to obtain the descriptors for each one.

A basic example can be observed in Fig. 1, where the log-polar chart is applied in the sampled contour of letter *A*. This chart shifts from point to point, and in each one a histogram is computed, by counting the number of points within each bin.

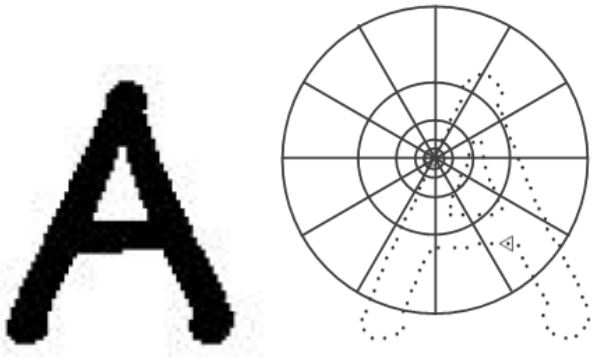

Fig. 1. Left: Original shape. Right: Sampled contours, with log-polar chart. [4]

The resulting descriptors of some points of two slightly-modified versions of letter *A* are displayed in Fig. 2. It can be noted that the descriptors for two points with the same relative position on both shapes are very similar, and they differ from the descriptor of the other point indicated in the figure.

To find the corresponding points between two shapes, a cost function $C_{i,j}$ is employed, by comparing the histograms of each pair of points in both shapes. For a point *i* of the first shape, and a point *j* of the second one, the histograms are normalized and stored in variables *g(k)* and *h(k)* respectively. The cost function, based on the chi-squared statistic, is computed with (1).

$$C_{i,j} = \frac{1}{2}\sum_{k-1}^{K}\frac{[g(k)-h(k)]^2}{g(k)+h(k)} \tag{1}$$

Since the objective is to obtain one correspondence point in shape 2, per each point in shape 1, a function to minimize total cost is applied. This function is not implemented in the present project, but for further explanation document [1] can be consulted. One more step is performed to estimate the transformation between the points. In [1] they only consider affine transformations, which are estimated with standard least squares methods.

B. Heat kernel signature

The heat kernel signature is a descriptor to represent non-rigid shapes based on heat kernel, which gives the fundamental mathematical approach to solve heat equation in a defined domain, considering the boundary conditions in the region of interest. HKS was proposed by Sun *et al.* in [2] to represent 3D shapes, and its invariant to isometric transformations.

HKS is calculated for all or a set of key points in a mesh, by applying in each case the heat kernel $h_t(x,y)$ on the surface, which describes the heat propagation from one point *x* to another one *y* in the shape, in a time *t*. In this case, the heat kernel signature is limited to temporal domain, considering only the heat transfer from one point to itself.

To understand the heat kernel signature, the heat diffusion equation over *M*, a compact Riemannian manifold possibly with boundary, is considered in (2), where Δ_M is the Laplace-Beltrami operator of *M*, and *u(x,t)* is the heat distribution at a point *x* at time *t*.

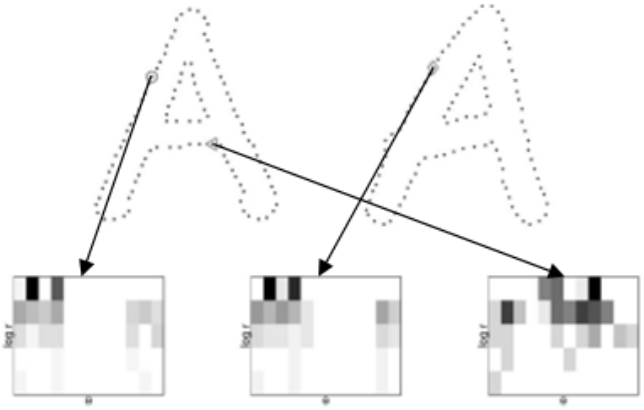

Fig. 2. Shape context descriptors for different points in letters *A*. [1]

$$\Delta_M u(x,t) = -\frac{\partial u(x,t)}{\partial t} \tag{2}$$

The solution of previous equation is presented in (3), where *dy* is the volume form at $y \in M$, and $u_0(y)$ is the initial heat distribution over the surface.

$$u(x,t) = \int h_t(x,y)u_0(y)dy \tag{3}$$

The eigen-decomposition of the heat kernel, for compact *M*, can be denoted by (4). λ_i is the i^{th} eigenvalue and ϕ_i the i^{th} eigenfunction of the Laplace-Beltrami operator.

$$h_t(x,y) = \sum_{i=0}^{\infty} e^{-\lambda_i t}\phi_i(x)\phi_i(y) \tag{4}$$

As stated previously, the heat kernel signature is restricted to temporal domain. The resulting heat kernel equation is presented in (5).

$$h_t(x,x) = \sum_{i=0}^{\infty} e^{-\lambda_i t}{\phi_i}^2(x) \tag{5}$$

The heat kernel is invariant under isometric deformations. It is also informative, with multi-scale property and stable against noise. The isometric invariance of the heat kernel is related to invariance in Laplace-Beltrami operator.

The discrete version of Laplace-Beltrami operator for a mesh can be represented as in (6), where *A* is a positive diagonal matrix and all the values in the diagonal *A(i,i)* depict the area of the triangles which are connected to vertex *i*. *W* is a symmetric semi-definite matrix.

$$L = A^{-1}W \tag{6}$$

The equation (6) can be also represented as (7), where Λ is a diagonal matrix of eigenvalues and Φ is a matrix which columns are the orthonormal eigenvectors of *L*.

Fig. 3. Surface charting. Left: Local multidimensional scaling. Center: Inward ray shooting. Right: Outward ray shooting. [3]

$$L = \Phi\Lambda\Phi^{\mathrm{T}}A \tag{7}$$

The discrete heat kernel then can be written as (8). $k_t(i,j)$ corresponds to the discrete version of $h_t(x,y)$, and its diagonal entries gives the HKS, sampled at discrete time intervals.

$$K_t = \Phi \mathrm{e}^{-\mathrm{t}\Lambda}\Phi^{\mathrm{T}} \tag{8}$$

The mathematical proofs of previous equations are fully detailed in [2].

C. *Laplace-Beltrami Operator*

The Laplace-Beltrami operator is a generalized concept to apply the Laplace operator on Riemannian and pseudo-Riemannian manifolds. This operator corresponds to the divergence of the gradient

Further explanation can be found in [5] and [6].

D. *Shape context descriptor in surfaces*

The approach presented in subsection *A.* can only be implemented in planar shapes. For this reason, Kokkinos *et al.* introduced in [3] a generalization of this descriptor to surfaces, called *Intrinsic shape context descriptor* (ISC), combined with HKS or Scale-invariant HKS, SIHKS. In order to adapt the shape context to 3D shapes, it was necessary to implement some modifications in the descriptor, such as defining a surface charting around a point and eliminating orientation ambiguity.

Regarding to surface charting, they proposed three methods to apply intrinsic polar grid on a surface, illustrated in Fig. 3: Local multidimensional scaling (MDS), shooting geodesic directions from vertex and shooting directions from a geodesic circle. They implemented the second method in their work. The paths created with the geodesic distance function in the mesh, were not necessarily along the edges of the triangles.

In the local MDS a planar grid is created by projecting the paths in the surface to the plane. In order to do so, "a planar configuration of points which pairwise Euclidean distance are as close as possible to the pairwise geodesic distances between points" is computed. On the other hand, in the outward ray shooting, the grid is built from the vertex, by shooting geodesic outwards from it. These paths are used as reference to define the polar system of coordinates in the vertex with equal angular segments. The last method consists in creating a geodesic circle around a vertex, divide it into segments with the same length, and propagate rays from them to the vertex.

Due to polar nature of the coordinate system, rotation ambiguity is presented in the descriptor. To eliminate this orientation ambiguity, the authors proposed to implement the *Fourier transform modulus* (FTM) in the descriptor, taking only the absolute value of it.

During the implementation, the cotangent weight scheme was used to compute the first 300 eigenvalues and eigenvectors of the Laplace-Beltrami operator on each shape, with Neumann boundary conditions. For HKS, six time scales t were used: t = 512, 675, 891, 1176, 1152 and 2048, whereas for the computation of SIHKS, 400 logarithmically sampled scales from 10^{-5} to $5\text{x}10^{5}$ were implemented. In the ISC descriptor, the radius R of polar grid was 20, with 16 angular bins, and 5 linearly spaced radial bins.

IV. Proposed Approach

The first step in order to implement the ISC based on the heat kernel signature is to extract the HKS for every vertex on the mesh. As the mesh is represented as triangular faces, (8) must be implemented.

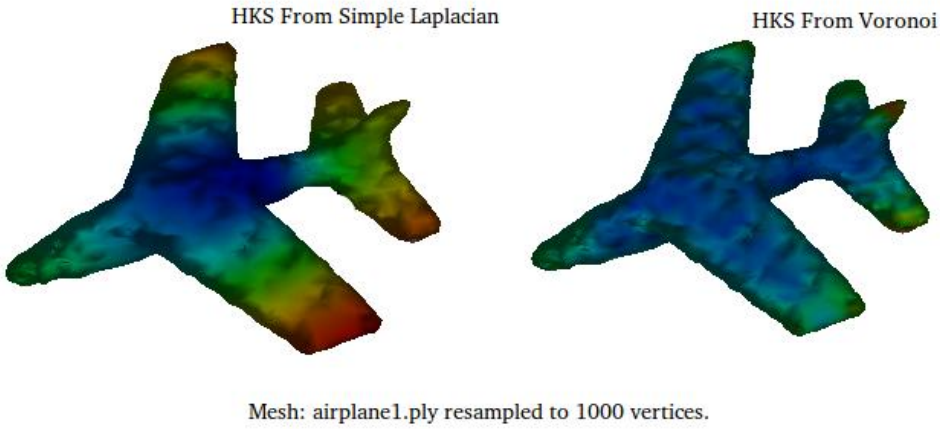

Fig. 4. Visual comparison between the HKS computed from the Laplacian matrix (*Left*), and the HKS computed from the Voronoi (cotangent) approach (*Right*). The left image seems to be more informative, since the HKS shows a more clear propagation, and a more clear differentiation between points with different characteristics.

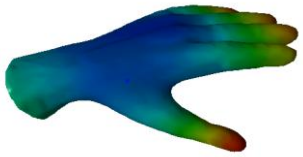

Fig. 5. Detail of the HKS computed for a hand. The image represents the difference from every point to a randomly selected point in the surface. As the curvatures start to increase, the values start increasing as well (finger tips are red, while hand body is blue).

Thus, the first problem in order to solve the HKS is to extract the Laplace-Beltrami operator from the mesh. Meyer *et al.* in [5] and Bobenko *et al.* in [6], show 3 different ways to compute and approximate the Laplace-Beltrami operator for manifolds represented as triangular meshes. The first one, and simpler approach, is to compute a matrix L as the difference between the adjacency matrix and the degree diagonal matrix ($L=D-A$). This approach is known as Laplacian matrix, or Kirchhoff matrix. Thus, the result is a symmetric and sparse matrix that approximately shows the operator applied on the mesh.

Some authors consider a problem that the Laplacian matrix is not normalized, defining the second approach to compute the Laplace-Beltrami operator [5]. This is the inverse of the square root of the diagonal degree matrix, multiplied by the Laplacian matrix and itself. The result of this approach is a symmetric and normalized matrix.

Finally, the last approach is proposed in Meyer *et al.* in [5]. in a more complex way, by using the Voronoi area computation when the face is not obtuse. An approximation to the intrinsic area of the face is implemented when this is obtuse. The result is definitive positive normalized matrix that approximates better the Laplace Beltrami operator.

The three approaches are implemented in order to compare performance of the HKS. Fig. 4 shows a comparison of the HKS computed with the simple Laplacian Matrix, and the Voronoi approach. As seen, as the Voronoi (and also the normalized Laplacian matrix, which is not shown) ends up with a normalized matrix, the results are not as informative, as the one computed with the Laplacian.

However, in this point an extra problem is faced. The size of this matrix is V by V, where V is the number of vertices of the mesh. Thus, for a mesh of 50000 vertices, and using a 32 bit variable to represent each digit, the size of the matrix is 9.31 Giga-bytes. This is a large amount of RAM memory, taking into account that the most new computers have between 4 and 6 Giga-bytes installed. Two possible solutions to face this problem were taken into account. First, as usually the square of the number of vertices is much bigger than the number of connections between them; this matrix is likely to be sparse. Matrix libraries for C++ commonly have a special class to store sparse matrices. However, is not usual to find an eigen-solver that allows the user to compute more than 1 eigen-pair for each sparse matrix. At this point, after testing the speed and capacities of computing eigen-decomposition of several libraries, it was decided to look forward another solution. This is, re-sampling the meshes. Quadric Edge Decimation technique appears as a fast and precise method for this decimation, as seen in [7]. Moreover, re-sampling the

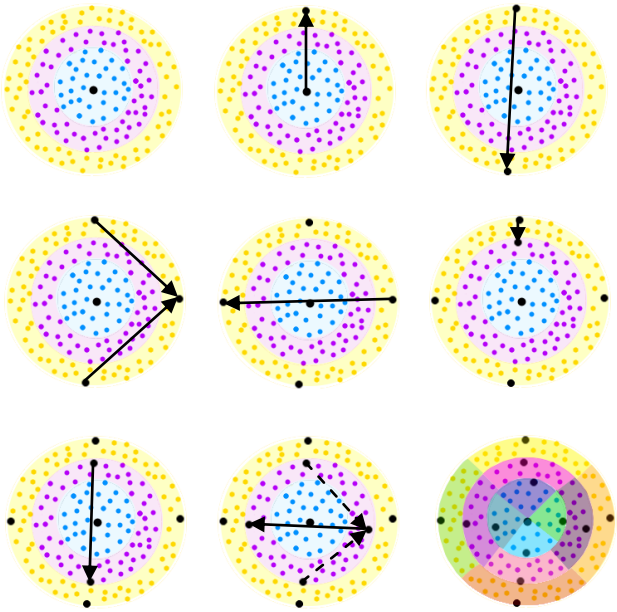

Fig. 6. Surface charting process for a given point. First, radial bins are defined by geodesic closer point computations. Then, a point in the last bin border is randomly selected. The farter point from this last point is obtained. The process continues until a complete binning is performed.

input meshes in a homogeneous form for all the studied meshes brings the advantage of invariance to sampling. Thus, although some details are lost, these are not sufficient to lost capacity to classify the object. All input meshes are resized to 2000 vertices, no matter what is the original size.

After computing the Laplace Beltrami Operator, the next problem to solve is the eigen-decomposition of the matrix. This eigen-decomposition is eased because of the symmetry property of the Laplacian matrix. Eigen library is the tool chosen to solve this problem. Furthermore, as eigen-library does not have a partial eigen-solver, the 2000 eigen-vectors and eigen-values are obtained. This step is recommended in order to avoid more description loss than the one generated by the decimation. Computing 2000 eigen-vectors takes around 20 seconds in a 1.7GHz processor. Recalling equation (8), a HKS matrix is computed for 1 time each time. However, as there is no need to use all the information on the matrix, since the concept of heat transfer from one point to itself is intrinsic enough to describe a shape, the diagonal of this matrix is the only information extracted. This computation is developed a few times for values from 100 to 1000 seconds (100, 200, 300 and so on). After this, to add some scaling invariance, the HKS is normalized, with values between 0 and 1. Also, this approach eases the visualization by color coding the transition (Fig. 5). So, it is established that as values increases, color changes from blue to green, then to yellow, and finally from yellow to red.

Computing the HKS completes the first part of the process of finding a strong intrinsic descriptor for 3D shapes. The next step is now to compute the shape context based on the inward shooting proposed on [3]. The process, is however, modified to improve speed and ease the concept of the surface charting.

The proposed method starts taking 500 equally spaced points from a mesh, in order to normalize the descriptor for all

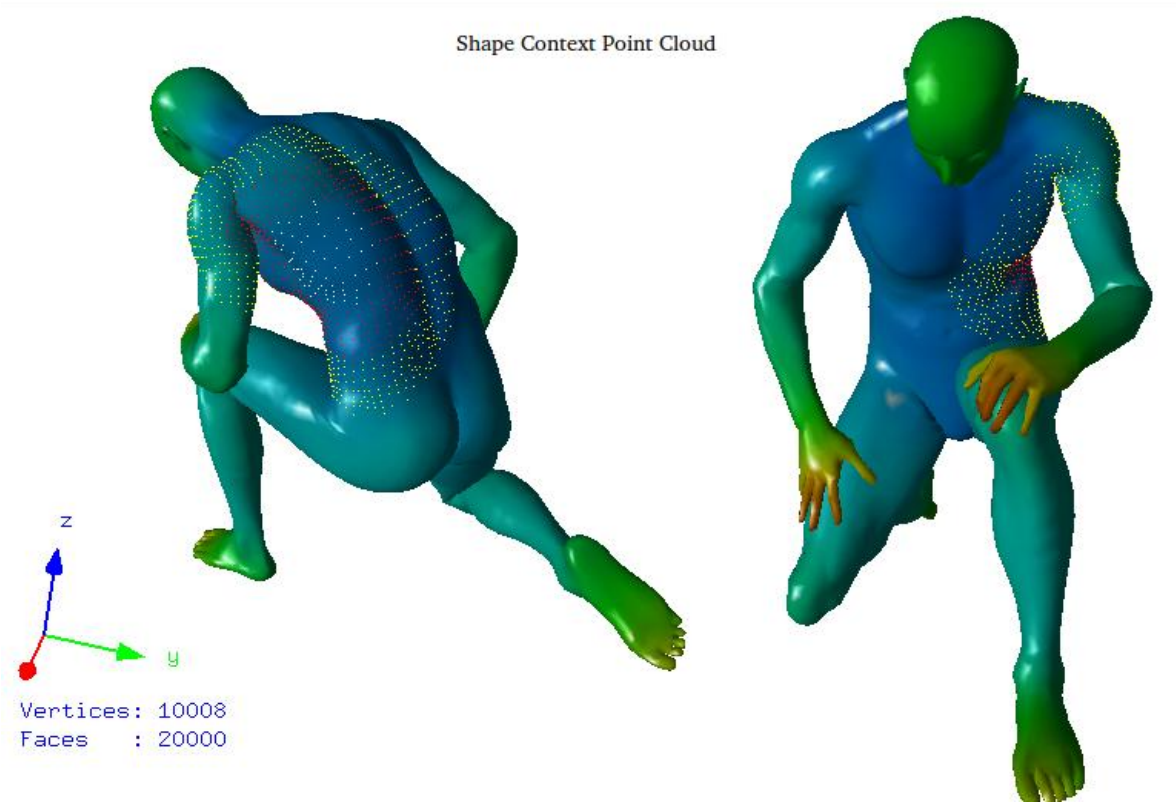

Fig. 7. Shape context formed by a cloud of points closer to a given point by distance of 10 (blue), 20 (red) and 30 (yellow) observed from two different points of view.

meshes. Each one of these points is representing the center of a shape context chart, as seen in Fig. 6. Now, to show the concept, assume that the mesh is formed by an infinitely large amount of vertices. Given a single point, from the 500 points previously selected, a set of other points closer to it by a radius *L1* can be selected. Also, a set of points with geodesic distance between *L2* and *L3* is defined. Each set of points defines a radial bin from the reference point. For this work, 3 bins are established, with geodesic radius of 10, 20 and 30. For each set of points representing the radial bins, 4 points (in this case) from the border with the next bin are selected, in order to represent landmarks for the angular binning. These 4 points are selected by choosing the 4 points that are further from each other (approximately). So, to each one of this 4 landmarks point for a radial bin, can be associated a Cartesian direction (N, S, W, E). The N point of the last bin is chosen first. Then, the S, followed by the E, and finally the W. The N point in the next radial bin is the closer point in the border of that bin to the N point of the current bin. At the end, the chart should look like the one in the Fig. 6.

Now, the angular binning is formed, by grouping the closer of the points in that radial bin to each one of its 4 landmarks. For a continuous mesh, this surface is the perfect shape context chart, but for the discrete case is just a very good approximation. The procedure from this point, is as expected and shown in the works [3] and [4]. The HKS of all points closer to one landmark are averaged. As 12 bins were defined, a matrix of size 500x12 is obtained, evaluating the SC for all the previously selected points. The magnitude of the *Fast Fourier Transform* (*FFT*) of the vectors of this matrix is extracted, in order to search invariance to rotation of the mesh, and the fact that it was not selected a standard axis system. Fig. 7 shows the radial binning for a real mesh for a given point. This approach to compute the SC was preferred among the local multidimensional scaling because of its lack of precision. And a similar concept could be applied to develop a SC shooting outwards.

Finally, now that a complete intrinsic shape context descriptor has been defined for a single mesh, an approach to classify it is provided. A much more complete and exact way

Fig. 8. Patterns created by showing the image formed by the SCD matrix of 2 meshes of the same class. Colored pairs show 10 best matches for feature points.

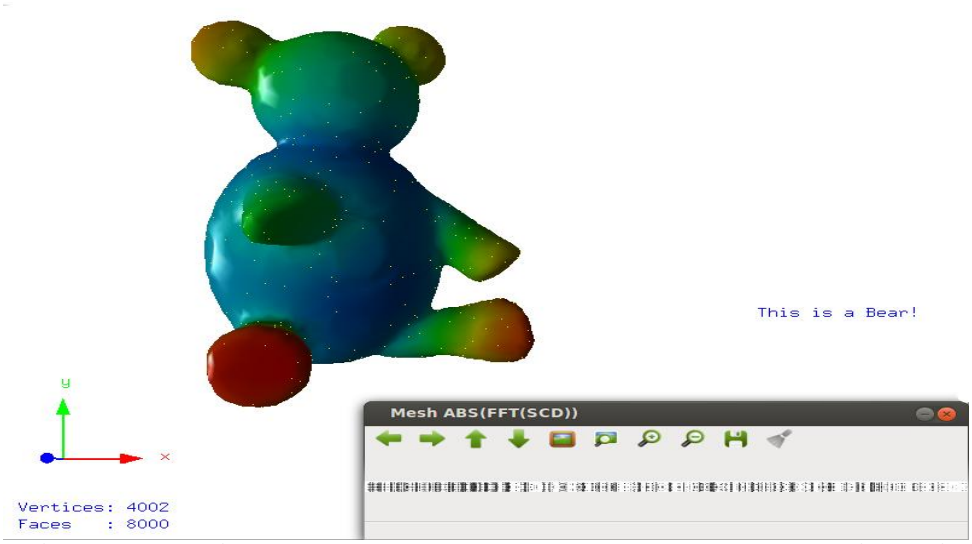

Fig. 9. HKS of the mesh of a bear (UP), image formed with the ISCDHKS matrix, and letters in blue show a correct classification of the mesh.

of implementing this computation would be the extraction of features points along the mesh, and compare them point to point with given mesh models (Refer to Fig. 8 to see the result of a prototype of an algorithm that makes this comparison). This problem would be a complex cost minimization that could be solvable with the Hungarian approach [4] [1]. However, for this project a much simpler (sacrificing accuracy) approach is developed by comparing templates (Human, Bear and Ant) to a given mesh intrinsic shape context descriptor based on HKS (ISC-HKS). The next section shows the results of this comparison.

V. Results

To analyze performance of the ISC-HKS, five metrics are implemented, comparing a template T of size wxh (representing an ISC-HKS of one of 3 models: Human, Bear and Ant), with a matrix I of size WxH (representing the ISC-HKS for a given mesh):

a. Squared difference

$$R(x,y) = \sum_{x'y'} \left(T(x',y') - I(x+x',y+y')\right)^2 \tag{9}$$

b. Normalized squared difference

$$R(x,y) = \frac{\sum_{x'y'} \left(T(x',y') - I(x+x',y+y')\right)^2}{\sqrt{\sum_{x'y'} T(x',y')^2 \cdot \sum_{x'y'} I(x+x',y+y')^2}} \tag{10}$$

c. Cross-correlation

$$R(x,y) = \sum_{x'y'} \left(T(x',y') \cdot I(x+x',y+y')\right) \tag{11}$$

d. Normalized correlation coefficient

$$R(x,y) = \frac{\sum_{x'y'} \left(T'(x',y') \cdot I'(x+x',y+y')\right)}{\sqrt{\sum_{x'y'} \left(T'(x',y')\right)^2 \cdot \sum_{x'y'} \left(I'(x+x',y+y')\right)^2}} \tag{12}$$

e. Normalized Cross-correlation coefficient

$$R(x,y) = \frac{\sum_{x'y'} \left(T(x',y') \cdot I'(x+x',y+y')\right)}{\sqrt{\sum_{x'y'} T(x',y')^2 \cdot \sum_{x'y'} I(x+x',y+y')^2}} \tag{13}$$

For equations (12) and (13), the parameters I' and T' are defined in (14) and (15), respectively.

$$I' = I(x+x',y+y') - \frac{1}{w \cdot h} \sum_{x''y''} I(x+x'',y+y'') \tag{14}$$

$$T' = T(x',y') - \frac{1}{w \cdot h} \sum_{x''y''} T(x'',y'') \tag{15}$$

A comparative scheme with the metrics error rates are presented in Table I. As seen, from the evaluated set of metrics, the best one to classify meshes based on proximity to a set of ISC-HKS models is the Normalized Cross-correlation coefficient. The error rate on the Table 1 is computed by counting the number of fails of classification for 15 different meshes. Fig. 9 and Fig. 10 show some results of the developed algorithm.

TABLE I
Metric for Classification

Metric	Error Rate
Squared difference	37%
Normalized squared difference	29%
Cross-correlation	22%
Normalized correlation coefficient	23%
Normalized Cross-correlation coefficient	15%

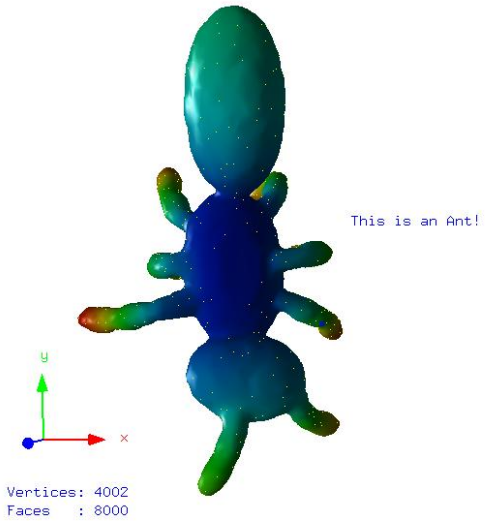

Fig. 10. HKS of the mesh of an ant, and letters in blue show a correct classification of the mesh.

VI. Conclusion

The present work shows the powerful tool that the intrinsic shape context descriptor based on HKS represents for meshes analysis. Even when smart metrics were not used to classify meshes, a relatively low error rate is reached. Also, the implemented shape context only have 12 bins (4 angular and 3 radial bins). So, the performance of this descriptor would increase even without changing the used metrics. However, there are still some disadvantages on the implementation of the modified inward shooting proposed. One of its disadvantages would be that it would not be a good descriptor for simple meshes, due to the low vertices amount the surface charting would be corrupted.

References

[1] S. Belongie, J. Malik, and J. Puzicha, "Shape context: A new descriptor for shape matching and object recognition," in *Advances in neural information processing systems*, 2001, pp. 831-837.

[2] J. Sun, M. Ovsjanikov, and L. Guibas, "A Concise and Provably Informative Multi-Scale Signature Based on Heat Diffusion," in *Eurographics Symposium on Geometry Processing*, vol. 28, 2009.

[3] I. Kokkinos, M. M. Bronstein, R. Litman, and A. M. Bronstein, "Intrinsic shape context descriptors for deformable shapes," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012, pp. 159-166.

[4] S. Belongie and J. Malik, "Matching with shape contexts," in *IEEE Workshop on Content-based Access of Image and Video Libraries*, 2000, pp. 20-26.

[5] M. Meyer, M. Desbrun, P. Schröder, and A. H. Barr, "Discrete Differential-Geometry Operators for Triangulated 2-Manifolds," *Visualization and mathematics*, vol. 3, no. 7, pp. 34-57, 2002.

[6] A. I. Bobenko and B. A. Springborn, "A discrete Laplace-Beltrami operator for simplicial surfaces," *Discrete & Computational Geometry*, vol. 38, no. 4, pp. 740-756, 2007.

[7] G. Gahm, "Mesh decimation," Linkoping University Modeling and Animation, 2010.