Partial Similarity of 3D Shapes Using Cross Recurrence Plot

Rafael Umino Nakanishi*, Jorge Piazentin Ono†, Paulo Pagliosa‡, Luis Gustavo Nonato* and Afonso Paiva*

*ICMC, USP, São Carlos
†School of Engineering, NYU, New York
‡FACOM, UFMS, Campo Grande

Fig. 1. By querying a user-selected segment of flamingo's head (green), our method is capable of retrieving regions (blue) which are most similar to the query region in a large dataset of different 3D models.

***Abstract*—This paper presents a novel 3D partial shape retrieval algorithm based on time-series analysis. Given a piece of a 3D shape, the proposed method encodes the shape descriptor given by the Heat Kernel Signature (HKS) as a time-series, where the time is considered an ordered sequence of vertices provided by the Fiedler vector. Finally, a similarity metric is created using a well-known tool in time-series analysis called Cross Recurrence Plot (CRP). The good performance of our method is also attested in a large collection of shape models.**

***Keywords*-Partial Shape Retrieval; Cross Recurrence Plot; Heat Kernel Signature; Fiedler Vector; Time-Series Analysis; Geometry Processing.**

I. Introduction

Partial similarity measures among different documents are ubiquitous tools in computer science. The main idea here is to find parts in documents — say text, images — that looks similar. It is easy to identify similarities when comparing words inside a text, or images inside a video, for example. However, this problem becomes more complicated when dealing 3D objects. In this scenario, it is necessary to clearly define what is a partial similarity and what is a 3D model part. For our context, we are interested in finding sub-regions of the shapes that are similar under some structure extracted from their surfaces. Assuming that an object can be decomposed as a collection of semantic regions or *partial shapes*, the partial shape similarity compares how similar two different regions are.

This paper introduces a novel partial similarity measure for 3D objects represented by triangular meshes. Firstly, our method computes a local shape descriptor based on the *Heat Kernel Signature* (HKS) for each vertex of the mesh. Then, a spectral segmentation is applied on the mesh and the HKS in each partial shape is compactly encoded into a time-series. The similarity distance between two partial shapes is obtained by comparing their correspondent time-series using *Cross Recurrence Plot* (CRP). This strategy allows to detect and quantify all relevant partial shapes of the mesh through a single matrix. Fig. 1 shows our method in action, given a region on a flamingo model (green) as query shape, the most similar regions are retrieved (blue) from a 3D model collection.

Contributions. In summary, the main contributions of the paper are:

- This paper introduces the concept of representing a local shape descriptor of a partial shape as a time-series;
- Our method enables the comparison of multiple partial shapes using a single CRP matrix. While these matrices are widely explored in time-series analysis, to the best of the author's knowledge it is the first time that CRP is applied in 3D shape retrieval;
- The proposed method does not require semantic mesh segmentation nor pointwise correspondence between the partial shapes.

Paper outline. The paper is organized as follows: Sec. II presents a brief review of techniques for partial shape retrieval of 3D shapes existing in the literature. The main concepts used in the proposed method, namely the Fiedler vector, HKS, and CRP, are introduced in Sec. III, Sec. IV, and Sec. V, respectively. The method is described in Sec. VI. Results are shown and discussed in Sec. VII. We conclude in Sec. VIII.

II. Related work

There are many papers about shape similarity and retrieval of 3D shapes [1], [2] in computer vision and computer graphics literature. In order to better contextualize our method and highlight its properties, we focus on the existing methods for a partial similarity of non-rigid 3D shapes.

Bronstein et al. [3] formalized the perceptual intuition of partial shape similarity using similarity metric based on Gromov-Hausdorff distance. This approach was first introduced by Mémoli et al. [4], based on distances between two metric spaces theory applied to point clouds. Although this method can give good results, the high computational cost [2] makes this approach impracticable.

Ovsjanikov et al. [5] proposed heat kernel maps approach to finding an isometric matching in one single shape and among a pair of shapes. This method works by fixing a point and iteratively searching over all other points looking for the best match. It differs from the method we present in the sense of quantities of matches they can find since our method can show all possible candidate regions. Sun et al. [6] use a similar paradigm, by introducing HKS as isometry invariant features. The HKS descriptor has the advantage of simplicity over heat kernel while being informative and keeping isometric invariance.

Bag-of-words approach is borrowed from text and image analysis for non-rigid shape retrieval [7], which uses geometric words based on multiscale HKS. The loss of spatial information is compensated by using a pair of words to build the shape dictionary. Lavoué [8] has shown some results on partial retrieval by using this approach. All of the works above use the features themselves to compare two different shapes and take into account the spatial relationship between them.

Recently, Tabia et al. [9] introduced a different approach by computing covariance matrices of features and comparing those matrices to find similarities between a pair of shapes. Our method uses a similar approach by not comparing the raw features but using an auxiliary matrix structure to find similarities between query and target shapes.

III. Fiedler vector

Fiedler vector provides an approximate solution for the *Minimum Linear Arrangement* (MLA) problem proposed by Harper [10]. MLA is a NP-hard problem and it tries to find permutations $\pi : \mathcal{V} \rightarrow \{1, 2, ..., n\}$, where $\mathcal{V}$ is the set of vertices of a triangle mesh $\mathcal{M}$ and $n = |\mathcal{V}|$, which minimizes the sum of all neighboring pairs of vertices. In other words, the MLA problem sorts the vertices of $\mathcal{V}$ on a positive integer line minimizing the cost function:

$$\sum_{\forall (i,j) \in \mathcal{E}} w_{i,j} \left| \pi(i) - \pi(j) \right| ,$$

where $\mathcal{E}$ denotes the set of edges of $\mathcal{M}$ and $w_{i,j}$ is the weight associated to each edge $(i, j) \in \mathcal{E}$.

In order to obtain the Fiedler vector, we compute the spectrum of the discrete Laplace-Beltrami operator (LBO) [11] for a mesh surface. The LBO matrix is defined as

$$\mathbf{L} = \mathbf{D}^{-1}\mathbf{A} ,$$

where the diagonal matrix $\mathbf{D} = \mathrm{diag}(d_1, d_2, \ldots, d_n)$ is known as *lumped mass matrix* and the *stiffness matrix* $\mathbf{A} = \mathbf{U} - \mathbf{W}$, where the *weighted adjacency matrix* $\mathbf{W}$ is given by:

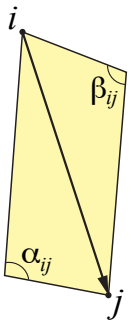
$$\mathbf{W} = \begin{cases} w_{i,j} & (i,j) \in \mathcal{E} \\ 0 & \text{otherwise} \end{cases}$$

and the diagonal matrix $\mathbf{U} = \mathrm{diag}\left(u_1, u_2, \ldots, u_n\right)$ with

$$u_i = \sum_{j \in \mathcal{N}_i} w_{i,j} ,$$

where $\mathcal{N}_i$ is the set of 1-ring neighbor vertices of the vertex i.

The trivial choice for values for the weights and mass is the unitary constant, i.e., $w_{i,j} = 1$ and $d_i = 1$. Setting this value, the discrete LBO relies only on the connectivity information of the graph. Although this property is well studied in graph theory [12], it does not give any geometric information of the discrete surface.



One way to introduce the geometric information was proposed by Pinkall and Polthier [13], using cotangent weights:

$$w_{i,j} = \frac{\cot(\alpha_{ij}) + \cot(\beta_{ij})}{2},$$

where α_{ij} and β_{ij} are the opposite angles of edge (i, j). In addition to the weight values, the mass values are computed using an approximation proposed by Meyer et al. [14]:

$$d_i = \frac{a_i}{3} ,$$

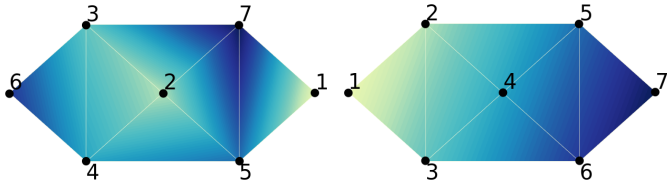
where a_i represents the area of all triangles incident at vertex i.

Computing the spectrum of the LBO matrix $\mathbf{L}$, it is possible to find mesh connectivity information. Since $\mathbf{L}$ is symmetric semi-positive definite, its eigenvalues are:

$$0 = \lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_n .$$

The Fiedler vector is given by the eigenvector associated with the first nonzero eigenvalue.

The Fiedler vector gives the minimum variation possible in a mesh. Fig. 2 shows a toy example of vertices sorted using the Fiedler vector, the vertices are sorted according to the entries of the Fiedler vector. This property is isometry invariant, then similar regions have the same arrangement. In order to simulate a time-series, we consider as "time" the consistent sorting produced by the Fiedler vector.



(a) Input mesh. (b) Sorted vertices.

Fig. 2. Vertex sorting using Fiedler vector.

IV. Heat Kernel Signature

In this section, we briefly introduce the concept of Heat Kernel Signature (HKS) [6], which is the basic mechanism used by our method to characterize three-dimensional shapes.

Let $\mathcal{S}$ be a compact manifold without boundary. The heat diffusion over $\mathcal{S}$ at time t for some initial heat distribution $f : \mathcal{S} \to \mathbb{R}^+$ is given by

$$\Delta_{\mathcal{S}} u(\mathbf{x}, t) = -\frac{\partial u}{\partial t}(\mathbf{x}, t) \quad \text{and} \quad \lim_{t \to 0} u(\mathbf{x}, t) = f \,,$$

where $\Delta_{\mathcal{S}}$ is the LBO of $\mathcal{S}$. The heat operator $H_t : L_2 \to L_2$, where L_2 is the space of all squared integrable functions on $\mathcal{S}$, is given by the solution of the heat equation at time t, that is,

$$(H_t f)(\mathbf{x}) = u(\mathbf{x}, t) = (e^{-t\Delta_{\mathcal{S}}} f)(\mathbf{x}) \,.$$

For any $\mathcal{S}$ one can show [15] that the heat kernel is a function $k_t(\mathbf{x}, \mathbf{y}) : \mathbb{R}^+ \times \mathcal{S} \times \mathcal{S} \to \mathbb{R}$ such that, for all $f \in L_2$, $t > 0$, and $\mathbf{x}, \mathbf{y} \in \mathcal{S}$,

$$(H_t f)(\mathbf{x}) = \int_{\mathcal{S}} k_t(\mathbf{x}, \mathbf{y}) f(\mathbf{y}) d\mathbf{y} \,.$$

The heat kernel can be understood as the amount of heat dissipated from a heat source point $\mathbf{x}$ to another point $\mathbf{y}$ at a given time t. The heat kernel has the following eigendecomposition for compact $\mathcal{S}$:

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

where λ_i is the i-th eigenvalue and ϕ_i the corresponding eigenfunction of the LBO.

Heat kernel has interesting properties that make it a suitable geometry tool for many applications [5], [6], [16], e.g., it is isometry-invariant and insensitive to noise, and provides multi-scale information.

The HKS, proposed by Sun et al. [6], is a shape descriptor based on heat kernel defined as

$$\text{HKS}_t(\mathbf{x}) = h_t(\mathbf{x}, \mathbf{x}) = \sum_{i \geq 0} e^{-\lambda_i t} \phi_i(\mathbf{x})^2 \,.$$

In the discrete scenario, i.e., when a manifold $\mathcal{S}$ is represented by a mesh $\mathcal{M}$, the HKS for a mesh vertex $\mathbf{v}$ is written in terms of the eigenvalues and eigenvectors of the LBO matrix $\mathbf{L}$:

$$\text{HKS}_t(\mathbf{v}) = \sum_{i=1}^{n} e^{-\lambda_i t} \phi_i(\mathbf{v})^2 \,. \tag{1}$$

In particular, we compute the discrete LBO by using cotangent weights as discussed in Sec. III. Note that the HKS is a function of time t. Thus, for each vertex $\mathbf{v}$, we consider time instants $(t_1, t_2, \ldots, t_m)$ and evaluate Eq. (1) for each t_i. The resulting m-dimensional vector is taken as the HKS of $\mathbf{v}$.

In our tests, we used a logarithmic scale with 50 time instants in the range $[t_{\min}, t_{\max}]$. We adopted $t_{\min} = 0.3 \times 10^{-2}$ and $t_{\max} = 1.5$ for all meshes (corresponding to the means of the minimum and maximum eigenvalues of LBO, respectively, for the entire test database). Longer time intervals have not changed the results significantly in our experiments.

V. Cross Recurrence Plot

Recurrence Plot (RP) is a technique which enables the analysis of nonlinear and nonstationary data in a dynamical system. In order to display and compare characteristic structures (e.g., similarities and alignments) between two dynamical systems, RP was extended into Cross Recurrence Plot (CRP) [17]. In other words, CRP provides a matrix that shows all equivalent states which occur simultaneously in two time-series at different times.

Consider a dynamical system modeled by d variables at instant of time t, if the set of variables are sufficient to describe future configurations of the system, they are called *state variables*. The vector formed by coupling these variables belongs to a d-dimensional space called *phase space*, which contains all the possible states of the system. The evolution of the state variables in time forms an *orbit* in phase space that characterizes the system and can be displayed for a visual analysis [17], [18]. In particular, we represent the matrix provided by HKS as multidimensional time-series $h_{ij} = \text{HKS}_{t_j}(\mathbf{x}_i)$ for $i = 1, \ldots, n_v$ and $j = 1, \ldots, n_t$, where n_v is the number of sampling points (vertices) and n_t is the number of time steps.

Unfortunately, in real applications, neither all state variables are known nor can be measured. Thus, it is not possible to completely describe the phase space only from measured observations. Therefore, according to Takens' embedding theorem [19], the phase space from a multidimensional time-series can be reconstructed using an embedding dimension m and a time delay τ:

$$\mathbf{x} = \{\mathbf{x}_i\} \quad \text{with} \quad i = 1, \ldots, n_w - (m-1)\tau \,,$$

where n_w is the number of windows (samples) and

$$\begin{aligned} \mathbf{x}_i \quad = \quad & (x_{1,i}, x_{1,i+\tau}, \ldots, x_{1,i+(m-1)\tau}, \ldots, \\ & x_{2,i}, x_{2,i+\tau}, \ldots, x_{2,i+(m-1)\tau}, \ldots, \\ & x_{d,i}, x_{d,i+\tau}, \ldots, x_{d,i+(m-1)\tau}) \,. \end{aligned}$$

Moreover, theorem states that the reconstructed phase space $\mathbf{x}$ from an unknown phase space embedded in $\mathbb{R}^d$ preserves the topological structures of the original phase space, if $m \geq 2d+1$. The choice of the parameters m and τ are usually estimated by the methods *False Nearest Neighbours* and *Mutual Information*, respectively [20]. Using these methods, the suited parameters for our application are $m = 5$ and $\tau = 2$.

A reconstructed phase space will commonly have an embedding dimension greater than three, then it will not be possible to directly visualize when a state in the phase space is almost recurrent. CRP solves these problems encoding the m-dimensional phase space into a binarized distance matrix $\mathbf{R}$, whose each entry $r_{i,j}$ is 1 if the orbit of state i is close to state j and 0 otherwise. Mathematically, comparing the similarity between two orbits (time-series) $\mathbf{x}$ and $\mathbf{y}$ is given by

$$r_{i,j} = \Theta\left(\epsilon_i^x - \text{sim}(\mathbf{x}_i, \mathbf{y}_j)\right) \; \Theta\left(\epsilon_j^y - \text{sim}(\mathbf{x}_i, \mathbf{y}_j)\right) \,,$$

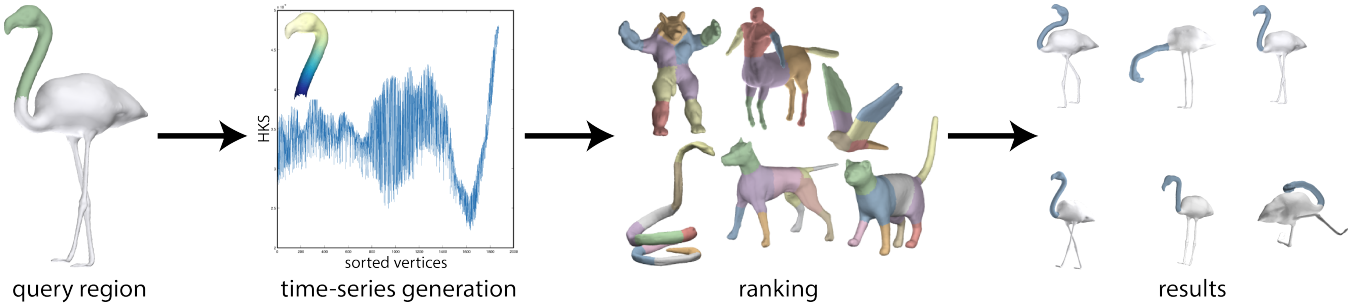


Fig. 3. Overview of our pipeline.

where Θ is the Heaviside function, sim is the cosine similarity measure

$$\mathrm{sim}(\mathbf{x},\mathbf{y}) = \frac{\mathbf{x}\cdot\mathbf{y}}{\|\mathbf{x}\|_2\|\mathbf{y}\|_2}$$

and ϵ_i^x and ϵ_i^y are the binarization thresholds. The values of ϵ_i^x and ϵ_i^y can be either fixed or adaptive, so that a percentage of nearest neighbors are considered recurrences [17], [21]. In our experiments, we used a fixed value $\epsilon_i^x = \epsilon_i^y = 0.45 \times 10^{-2}$, this value is an average of the maximum values reached in each CRP matrix and presented good results in most of the cases tested.

Many techniques were developed for the automatic evaluation of CRPs. These techniques are called Recurrence Quantification Analysis and, in general, they work by measuring the length of the diagonals formed in the CRP matrices. In our method, we will use the metric S_{max}, which measures the maximum length of the diagonals in the CRP, considering possible time shifts in the orbit (which results in curved lines) [21]. S_{max} can be computed as the maximum value of the matrix $\mathbf{S}$, defined by: $s_{1,j} = s_{2,j} = s_{i,1} = s_{i,2} = 1$ and

$$s_{i,j} = \begin{cases} \max\{s_{i-1,j-1}, s_{i-2,j-1}, s_{i-1,j-2}\} + 1 & r_{i,j} = 1 \\ 0 & r_{i,j} = 0. \end{cases}$$

Summarizing, the bigger the S_{max} value, bigger is the matrix diagonal, and by consequence, the recurrence segments are more similar to each other.

VI. Method

In this section, we explain the pipeline of our method of finding similarities between 3D shapes. The method encompasses the steps shown in Fig. 3 and, basically, is divided into three parts: treatment of query object; database model processing; and comparison of the query with all shapes individually. A query mesh is extracted from a model that may or may not be present in the given dataset. For that model, our method performs a computation of Fiedler vector, that is used to sort the vertices like a time-series, and extracts the HKS feature for all mesh vertices. More precisely, we build a time series from a mesh, where each "observation", corresponding to a vertex of the mesh, has a *value* and a *time-stamp*. The value of the observation corresponds to the HKS of that vertex (note that, it is a vector due to each vertex has 50 distinct HKS entries) and the time-stamp is the vertex position in the Fiedler vector.

After rearranging the vertices, the user can select a segment that will be used as the actual query. Only the vertices that belong to the selected region are compared to database models. Selecting only a segment for future comparison prevents over computation of CRP matrix.

A. Dataset segmentation

For all models present in the dataset, the following steps are performed: given a shape $\mathcal{M}$, we divide feature extraction in segmentation step and describing step. For the segmentation step, we apply consensus segmentation [22] to a shape, so it is possible to divide the mesh in n_s segments, where m_i represents the segment i in the shape, $i = 1, 2, ..., n_s$. It is important to observe that consensus segmentation can create more segments than necessary as it resolves the segmentation problem by electing the most stable segmentation to deformations. For instance, Fig. 4 shows the direct result given by Rodola's algorithm [22] in the left. This over-segmentation is not desired by our method, since we want only to separate significant parts, like legs, arms, or antennae. To adjust the result to our approach, we implemented a modification that merges the segments that have only one neighbor segment, ending with a bigger part. This merging process continues until more than one neighbor segment are found, or all segments are merged. In the snake example, our modification starts at the tail end (or head) and merges with the next segment to tail (green and pink segments) and goes recursively until the stop criterion are reached.

Fig. 4. Segmentation of a 3D model: the original result produced by [23] (left) and the modified result (right).

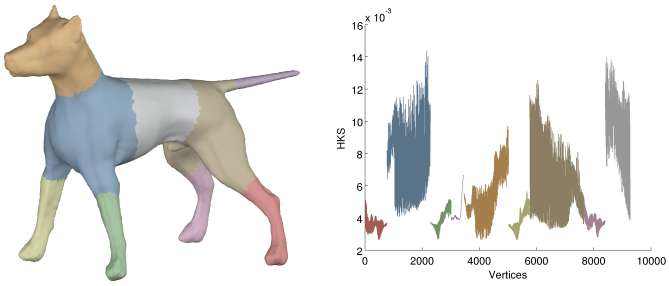


Fig. 5. Example of a segmented mesh and resulting sorted vertices. Each color codes a region of mesh in the sorted feature vector. The y-axis represents the feature value; the x-axis, vertices.

For the describing step, the heat kernel signature, $k_t(x, x)$, is computed for every segment, for all points in the segment over logarithm time interval $t \in [t_{min}, t_{max}]$. Then, the points are sorted by connectivity and then rearranged to become a bigger vector that describes the entire shape, as illustrated in Fig. 5. Note in the figure that each different segment presents different behaviors in the series, as the color scheme in the mesh represent the same regions in the graphic plot. The sorting is necessary to simulate time dependency between every vertex feature.

B. Comparisons

After extracting query's and target shape's feature, the comparison between them is done using CRP matrix and Recurrence Quantification Analysis metric S_{max} over binarized matrix. It is important to note that the CRP matrix does not consider the context of the parts nor the semantic behind each mesh. To rank the best parts, we use the value given by S_{max}: values increase as the diagonal size increases and we consider the bigger diagonal represents bigger similar regions. This criterion may lead to mistakes, but for the most tests, satisfactory regions were found.

The first thing to note is the usage of descriptor vector as sequence dependent vector. By doing so, we assure that all shapes features are going to be rearranged equally, and the feature vectors are comparable by using a time-series descriptor, as CRP. Moreover, the HKS proposed by [6] is isometry invariant and consequently, our method inherits this property. The second thing is that CRP matrices can find multiple similarities between the segment of interest and the target shape. An example of CRP matrix can be seen in Fig. 6 in the left, where the query is a centaur arm, shown in green, and the most similar regions (two regions were asked, in this case) are the two arms in blue. In the matrix we highlight those two regions that represent the arms, that have greatest S_{max} values. Other candidate regions are shown with lighter blue color in the matrix and no-matching regions are white.

VII. Results and Discussion

Our code was implemented in MATLAB, making the computation of the discrete LBO spectrum easier due to built-in packages. Given the high computational cost of evaluating

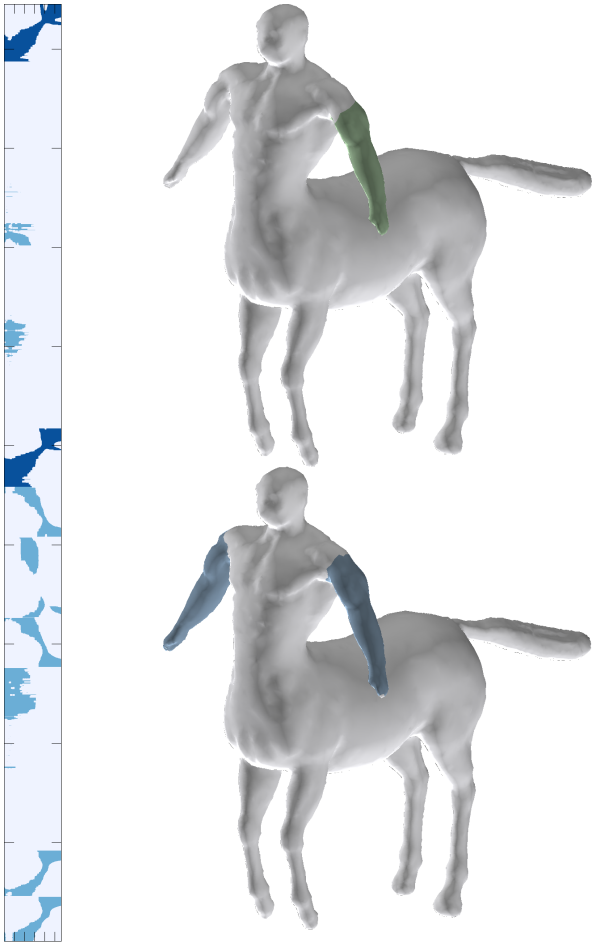


Fig. 6. CRP example of a centaur arm used as query. The resulting top-two regions are darker in the left matrix and correspond to the blue regions in bottom centaur, while the lighter blue ones are possible candidates.

HKS for all test meshes, those descriptors were computed once and stored on a disk for later access.

Our experiments were performed using object models present on SHREC'11 non-rigid database [24]. This base contains 600 models divided into 30 classes with 20 instances each. Individual meshes contain about 9000 vertices randomly sorted. Fig. 7 shows an example of the models with one instance of each class. For each query, the user selected interactively the region of interest, as described in Sec. VI.

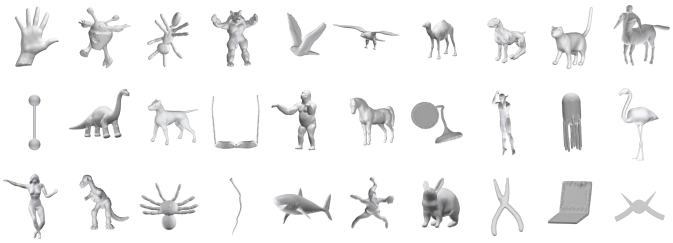
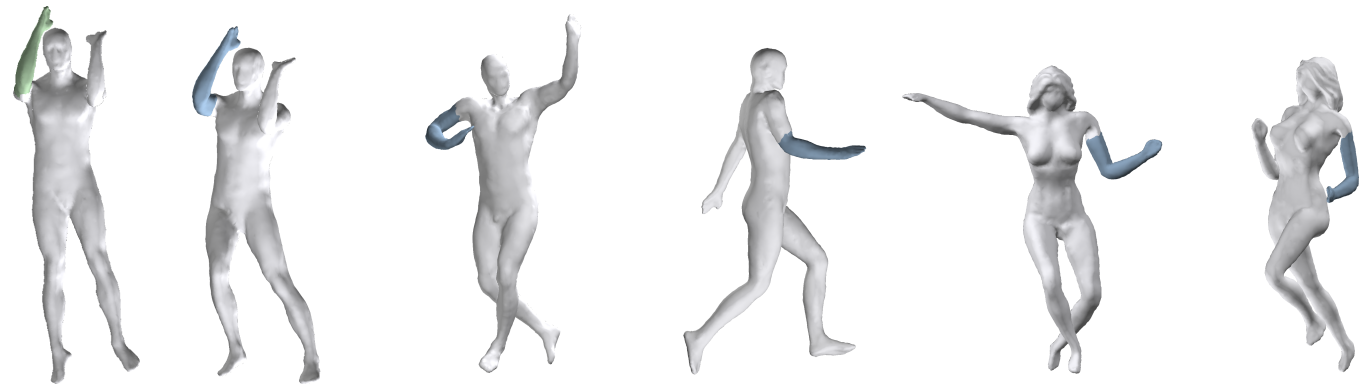
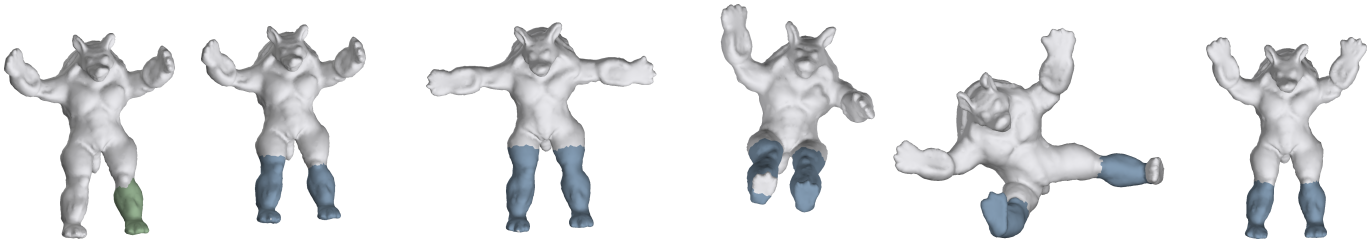


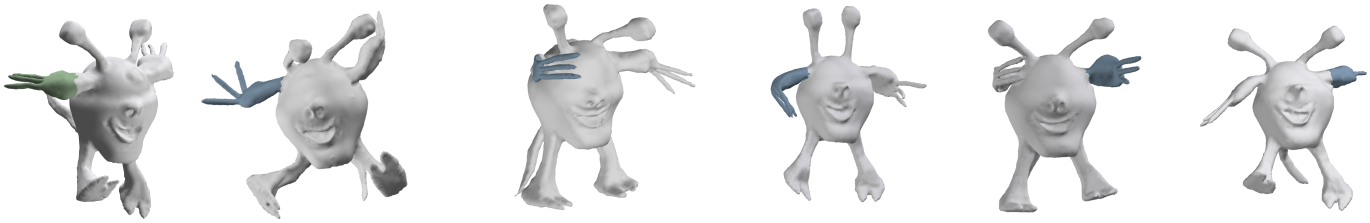
Fig. 7. Example of models present in SHREC'11 database.



(b) Human arm query



(d) Armadillo leg query



(f) Alien arm query

Fig. 8. Queries (first column) and their respective first retrieved parts.

In our tests, we extracted some fragments from shapes in order to find similar regions in the same shape and in other objects in the database. The similarity is given in terms of $S_{\max}$: we consider higher $S_{\max}$ values as most similar parts. In all experiments, we select a region from a target model, extracted from the database, run CRP matrix over all other 3D models, extract $S_{\max}$ value from each matrix, and rank the highest ones. Fig. 1 and Fig. 8 illustrate the problem of finding similar regions in different objects. From a selected region (green), our method finds the most similar regions in the same mesh (blue parts). For the human arm query, the results are visually humans, but the classes are different, as the database separates "men" from "women". In the armadillo leg query, we asked the method to return the top two regions.

The required number of regions is given as input to the method. Fig. 9 shows an example when we asked for the eight most similar segments in an ant model.

As mentioned before in Sec. VI-B, the lack of semantical information makes regions structurally similar be considered false positives. In the example of Fig. 10, a human arm selected as a query is mistakenly given as similar to a cat tail.

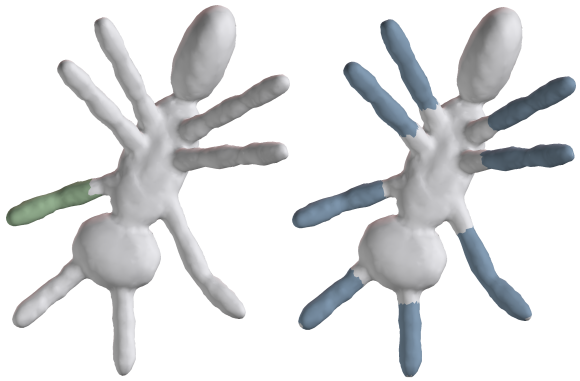
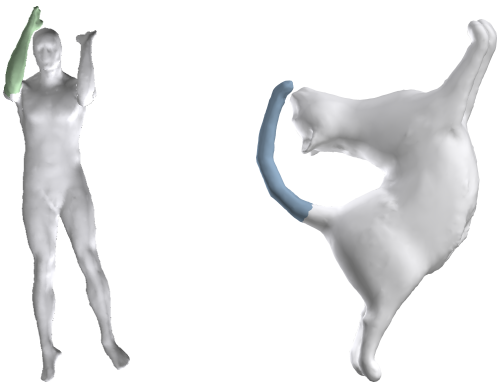


Fig. 9. Querying an ant leg, it is possible to find the most similar regions in the same model.

Even though they have same structure — almost cylindrical shape — the two parts are from different semantical classes.

VIII. Conclusion and Future Work

In this paper, we have proposed a method to find similar parts between shapes using Cross Recurrence Matrix with



(a) Query. (b) 6th most similar model.

Fig. 10. Example of limitation of our method: without semantic information, the method find regions in other models that may not be visually similar.

Recurrence Quantity Analysis measures. Our goal is to find models from a collection whose parts look similar to parts of another (or even the same) model. The proposed methodology performed well, producing good results in the accomplished tests. We also showed that it is possible to recover multiple regions with only one CRP, which resulted in a reduced processing time for the queries.

The next challenge is to investigate the use of alternative features besides HKS, as for example gaussian curvature or other isometry-invariant features. We also want to extend the proposed method to handle incomplete models, possibly with boundary, as well as point-set surface models using mesh-free differential operators [25].

ACKNOWLEDGMENT

We would like to thank the anonymous reviewers and Rodrigo Mello for their valuable comments. The authors are supported by CNPq (fellowships #302643/2013-3, #305796/2013-5 and #132340/2014-3) and São Paulo Research Foundation (FAPESP) under grants #11/22749-8, #12/24801-0 and #14/09546-9.

REFERENCES

[1] Z. B. Liu, S. H. Bu, K. Zhou, S. M. Gao, J. W. Han, and J. Wu, "A survey on partial retrieval of 3D shapes," *Journal of Computer Science and Technology*, vol. 28, no. 5, pp. 836–851, 2013.

[2] Z. Lian, A. Godil, B. Bustos, M. Daoudi, J. Hermans, S. Kawamura, Y. Kurita, G. Lavoué, H. Van Nguyen, R. Ohbuchi, Y. Ohkita, Y. Ohishi, F. Porikli, M. Reuter, I. Sipiran, D. Smeets, P. Suetens, H. Tabia, and D. Vandermeulen, "A comparison of methods for non-rigid 3D shape retrieval," *Pattern Recognition*, vol. 46, no. 1, pp. 449–461, 2013.

[3] A. M. Bronstein, M. M. Bronstein, A. M. Bruckstein, and R. Kimmel, "Partial similarity of objects, or how to compare a centaur to a horse," *International Journal of Computer Vision*, vol. 84, no. 2, pp. 163–183, 2009.

[4] F. Mémoli, "On the use of Gromov-Hausdorff distances for shape comparison," in *Eurographics Symposium on Point-based Graphics*, 2007, pp. 81–90.

[5] M. Ovsjanikov, Q. Mérigot, F. Mémoli, and L. Guibas, "One point isometric matching with the heat kernel," *Computer Graphics Forum*, vol. 29, no. 5, pp. 1555–1564, 2010.

[6] J. Sun, M. Ovsjanikov, and L. Guibas, "A concise and provably informative multi-scale signature based on heat diffusion," *Computer Graphics Forum*, vol. 28, no. 5, pp. 1383–1392, 2009.

[7] A. M. Bronstein, M. M. Bronstein, L. Guibas, and M. Ovsjanikov, "Shape google: geometric words and expressions for invariant shape retrieval," *ACM Transactions on Graphics*, vol. 30, no. 1, pp. 1:1–1:20, 2011.

[8] G. Lavoué, "Combination of bag-of-words descriptors for robust partial shape retrieval," *The Visual Computer*, vol. 28, no. 9, pp. 931–942, 2012.

[9] H. Tabia, H. Laga, D. Picard, and P.-H. Gosselin, "Covariance descriptors for 3d shape matching and retrieval," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014, pp. 4185–4192.

[10] L. H. Harper, "Optimal assignments of numbers to vertices," *Journal of the SIAM*, vol. 12, no. 1, pp. 131–135, 1964.

[11] M. Reuter, S. Biasotti, D. Giorgi, G. Patanè, and M. Spagnuolo, "Discrete Laplace-Beltrami operators for shape analysis and segmentation," *Computers & Graphics*, vol. 33, no. 3, pp. 381– 390, 2009.

[12] F. R. K. Chung, *Spectral Graph Theory*. AMS, 1997, vol. 92.

[13] U. Pinkall and K. Polthier, "Computing discrete minimal surfaces and their conjugates," *Experimental Mathematics*, vol. 2, no. 1, pp. 15–36, 1993.

[14] M. Meyer, M. Desbrun, P. Schröder, and A. H. Barr, *Visualization and Mathematics III*. Springer, 2003, ch. Discrete Differential-Geometry Operators for Triangulated 2-Manifolds, pp. 35–57.

[15] E. P. Hsu, *Stochastic Analysis on Manifolds*. AMS, 2002.

[16] M. Abdelrahman, M. El-Melegy, and A. Farag, "Heat kernels for non-rigid shape retrieval: Sparse representation and efficient classification," in *Ninth Conference on Computer and Robot Vision (CRV)*, 2012, pp. 153–160.

[17] N. Marwan and J. Kurths, "Cross recurrence plots and their applications," *Mathematical Physics Research at the Cutting Edge*, pp. 101–139, 2004.

[18] K. T. Alligood, T. D. Sauer, and J. A. Yorke, *Chaos: An Introduction to Dynamical Systems*. Springer.

[19] F. Takens, *Detecting Strange Attractors in Turbulence*. Springer, 1981.

[20] H. Kantz and T. Schreiber, *Nonlinear Time Series Analysis*, 2nd ed. Cambridge University Press, 2003.

[21] J. Serrà, X. Serra, and R. G. Andrzejak, "Cross recurrence quantification for cover song identification," *New Journal of Physics*, vol. 11, no. 9, pp. 1–20, 2009.

[22] M. Berger, L. G. Nonato, V. Pascucci, and C. T. Silva, "Fiedler trees for multiscale surface analysis," *Computers & Graphics*, vol. 34, no. 3, pp. 272–281, 2010.

[23] E. Rodola, S. R. Bulò, and D. Cremers, "Robust region detection via consensus segmentation of deformable shapes," *Computer Graphics Forum*, vol. 33, no. 5, pp. 97–106, 2014.

[24] Z. Lian, a. Godil, B. Bustos, M. Daoudi, J. Hermans, S. Kawamura, Y. Kurita, and G. Lavoué, "SHREC 11 Track : Shape Retrieval on Non-rigid 3D Watertight Meshes," *Eurographics Workshop on 3D Object Retrieval*, 2011.

[25] F. Petronetto, A. Paiva, E. S. Helou, D. E. Stewart, and L. G. Nonato, "Mesh-free discrete Laplace-Beltrami operator," *Computer Graphics Forum*, vol. 32, no. 6, pp. 214–226, 2013.