Egyptian Computer Science Journal Vol. 38 No. 1 January 2014 ISSN-1110-2586

3D Object Retrieval Using Compact Shape Descriptor

Hanan ElNaghy, Safwat Hamad & M. Essam Khalifa

Faculty of Computer and Information Sciences,
Ain Shams University, Cairo, Egypt
Hanan.Muhammad@fcis.asu.edu.eg, sHamad@cis.asu.edu.eg, esskhalifa@cis.asu.edu.eg

Abstract

The need for effective and efficient 3D object retrieval approaches is emerging in a broad range of applications in science and engineering. With the increasing understanding of shape geometry and topology in the context of shape similarity, workable solutions for 3D object retrieval are being produced. In this paper, we present a novel technique for 3D Object Retrieval. The key idea of the proposed approach is based on the synergy between Heat Kernel Signatures (HKS) and Bag of Features (BoF) paradigm. For a given 3D model, the proposed approach considers a set of feature points, defined by an innovative feature point detection algorithm, and associate them with a compact HKS feature descriptor. Then, a vocabulary is constructed and BoF vector describing each 3D model is computed. Finally, the challenging problem of matching two given 3D models sums up to measuring the distance between their corresponding BoF distributions. The proposed approach is not only computationally efficient but also highly discriminative. It achieves state of the art results on SHREC 2011 dataset of non-rigid models, confirming its invariance to different kinds of deformation and possible noise.

Keywords: *3D Object Retrieval, Shape Matching, Shape Descriptor, Heat Kernel Signature, Bag-of-Features*

1. Introduction

3D Object Retrieval plays an important role in many today's applications. It is now recognized as one of the most significant research areas in computer graphics and multimedia. As a result of emerging multimedia computing technologies, large databases of 3D models are distributed freely or commercially on the World Wide Web. Therefore, it is necessary to design 3D object retrieval methods which enable the users to efficiently and effectively search interested 3D models. The primary challenge to a content based 3D object retrieval system is how to extract the most representative features to discriminate the shapes of various 3D models. Generally speaking, there are some desirable properties for a typical 3D object retrieval method. It should represent important shape properties while being independent of 3D object representation. Additionally, it is anticipated to be compact to store, insensitive to noise and quick to compute and compare. Another important requirement is being invariant to different kinds of deformations and transformations.

Over the last years, the content based 3D object retrieval problem has been extensively investigated in the literature. A large number of competing techniques have been developed for the purpose of content based retrieval of 3D objects. Recently, Sun et al. [1] proposed Heat Kernel Signatures (HKS) as a deformation invariant descriptor based on diffusion geometry. Bronstein et al. [2] presents a scale invariant version of the HKS. Fang et al. [3,4] defined the Temperature Distribution (TD) of the Heat Mean Signature (HMS) as a shape descriptor for shape matching. Ohbuchi et al. [5] use Bag of features (BoF) approach as a part of their introduced method. The method, named Bag-of-Features SIFT (BF-SIFT), employs Scale Invariant Feature Transform (SIFT) by Lowe [6]. An improved algorithm was propped by Furuya and Ohbuchi [7] , which extracts much larger number of local visual features by sampling each depth image densely and randomly. The method applies GPU for SIFT feature extraction and an efficient randomized decision tree for encoding SIFT features into visual words. Shape Google Work by Bronstein et al. [8] introduce a shape retrieval method based on the HKS, where a shape is represented by a Bag of Features (BoF) vector.

The rest of this paper is organized as follows. Section 2 provides an overview of our proposed technique for 3D Object Retrieval, explaining the four consecutive phases constituting the proposed technique. Section 3 shows the experimental results. Finally, conclusions and suggestions for future works are given in Section 4.

2. Proposed Approach Overview

The process of our proposed 3D object retrieval technique proceeds through four main phases: HKS Computation, Feature Point Detection and Description, Bag of Features and finally the Matching phase. Basically, the initial phase of HKS computation computes HKS descriptor for each 3D model in order to be used successively by feature detection and description phase. During feature detection and description phase, an initial set of feature points are first detected from 3D models which then undergo an innovative filtering technique. The main intention of such filtering technique is to discard irrelevant feature points keeping only stable significant feature points entirely describing a given 3D model. At the end of this phase, each feature point emerging from the filtering technique is associated with a compact HKS-based feature descriptor vector in order to construct the final feature space. The next phase is the Bag of Features phase, which is subdivided into two main sub phases: vocabulary construction sub phase and 3D object representation sub phase. Throughout the vocabulary construction sub phase, a geometric vocabulary is built by clustering the feature space, such that the clusters' centroids represent the geometric words of the constructed dictionary. Then, 3D object representation sub phase performs vector quantization in the feature space in order to represent each 3D model by its BoF vector. The final matching phase determines the similarity or dissimilarity of given 3D models through comparing their corresponding BoF vectors. Each of these phases constituting the proposed 3D object retrieval approach will be described comprehensively through the following sub sections.

2.1. HKS Computation

HKS computation phase is considered an essential preliminary stage, where its results of computed HKS point signatures will be intensively used in the following phase of feature detection and description. For discrete computation of the Heat Kernel Signature, first the Laplace-Beltrami operator is estimated on triangular meshes. Second, the smallest $\boldsymbol{k}$ eigenvalues and eigenfunctions of the Laplace-Beltrami operator are calculated. Then, HKS can be computed by uniformly sampling $\boldsymbol{m}$ points in the logarithmically scale over the time interval $[\boldsymbol{t_{min}}, \boldsymbol{t_{max}}]$ with:

$$\mathrm{t_{min}} = 4 \ln 10 / \lambda_{\mathrm{k}} \qquad (1)$$

and

$$\mathrm{t_{max}} = 4 \ln 10 / \lambda_2 \qquad (2)$$

where λ_2 and λ_k are the second and the $\mathrm{k^{th}}$ (last) eigenvalues, respectively.

Sun et al. [1] indicates that HKS remains unchanged for $t > t_{max}$ as it is mainly determined by the eigenvalue λ_2, while in order to estimate the HKS with $t < t_{min}$ one needs to compute more eigenvalues and eigenfunctions. At small time values, HKS captures local 3D model information. As time elapses the signature tends to capture more global 3D model details. However there is a limit on how small t can be for which the HKS can be estimated faithfully, which determined by the resolution of the mesh.

Finally, for each vertex x of the 3D model, HKS can be computed at each time scale over the interval $[t_{min}, t_{max}]$ using the following equation:

$$K_t(x,x) = \sum_{i=0}^{k} e^{-\lambda_i t} \ \phi_i(x)^2 \qquad (3)$$

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

In a real discrete environment, given a triangular mesh with $\boldsymbol{n}$ vertices, Laplcaian matrix and its corresponding $\boldsymbol{k}$ eigenvalues and eigenfunctions are first computed. Since the Laplacian is represented by a sparse matrix $(n \times n)$, calculating several dozen of eigenvalues and eigenfunctions remains a considerably fast procedure. Then, $\boldsymbol{m}$ time scales are defined over the interval $[\boldsymbol{t_{min}}, \boldsymbol{t_{max}}]$ for computing the HKS. Finally, HKS descriptor (feature vector of length $n \times m$)can be computed using these eigenvalues, eigenfunctions and time scales applying equation (3).

2.2. Feature Point Detection and Description

This phase is subdivided into three main sub phases: Feature Detection, Filtering and Feature Description. First, the feature detection sub phase considers the HKS critical points on the surface of a given 3D model to act as a handful of primary feature points. For a given 3D model in a real discrete environment, each vertex is currently represented by a feature vector of length $\boldsymbol{m}$ representing HKS values at $\boldsymbol{m}$ time scales along the interval $[\boldsymbol{t_{min}}, \boldsymbol{t_{max}}]$, computed from the preliminary phase of HKS computation. A point x is a critical point, i.e. local maxima, if $\boldsymbol{K_t(x,x)} > \boldsymbol{K_t(x_i,x_i)}$ for all $\boldsymbol{x_i}$ in the 2-ring neighbourhood of $\boldsymbol{x}$. Therefore, each point x is checked at the $\boldsymbol{m}$ time scales to compare its HKS value with its neighborhood's values in order to determine whether it is a critical point or not.

Second, the filtering sub phase minimizes the number of initially extracted feature points by eliminating the insignificant ones. The main objective of such filtering technique is to present a compact set of significant feature points capable of describing the 3D model's geometry efficiently, while remaining stable under small perturbations. Keeping the number of feature points as minimum as possible reduces the time and space required for clustering the feature space.

Finally, the feature description phase associates each of these significant feature points with a d-dimensional feature descriptor $(\boldsymbol{p(x)})$ that is: compact in size, efficient to compute, informative, discriminative and robust. Thus, a subset of m-dimensional HKS feature descriptor $K_t(x,x)$ is selected to encode each feature point at $\boldsymbol{d}$ significant time scales such that $d \in \mathbb{R}^+ and\ d < m$. This chosen subset acts as a window on the m-dimensional feature descriptor keeping the most discriminative and informative d HKS values, whereas ignoring those redundant and non-discriminative $m-d$ HKS values. Earlier to selecting such $\boldsymbol{d}$-dimensional feature vector ($\boldsymbol{p}(\boldsymbol{x})$), HKS feature descriptor should undergo an essential preliminary phase of normalization. Each feature descriptor ($\boldsymbol{K_t}(\boldsymbol{x},\boldsymbol{x})$) associated with feature points need to be normalized in order to facilitate the process of clustering the feature space. This normalization process produces a normalized $\boldsymbol{m}$-dimensional feature vector ($\boldsymbol{h}(\boldsymbol{x})$), such that:

$$h(x) = (h_1(x), \dots, h_m(x)); \quad h_i(x) = c(x)K_{t_i}(x,x) \qquad (4)$$

where the constant $c(x)$ is selected in such a way that $\|h(x)\|_2 = 1$ [9].

The experimental results section will show our conducted experiments for choosing the most optimum d-dimensional feature descriptor ($\boldsymbol{p}(\boldsymbol{x})$) that could achieve successful retrieval results.

2.3. Bag of Features

Feature detection and description phase yields a set of succinct feature points each described with d-dimensional feature vector $p(x)$. These detected feature points together with their associated HKS based feature descriptor (d-dimensional feature vector) acts as a distinctive signature for a given 3D model describing its most significant features. Unfortunately, the complexity of using such distinctive signature directly for the purpose of 3D matching and retrieval is extremely high and unfeasible. Therefore, Bag-of-Features (BoF) paradigm [9] is applied. BoF paradigm is mainly divided into two main sub phases: Vocabulary Construction and 3D Object Representation.

The key idea if the first sub phase of vocabulary construction is to cluster the descriptor space using an adequate clustering technique so as to construct the required geometric vocabulary $\mathcal{P} = \{p_1, \dots, p_v\}$. It is important to highlight that the number of clusters (K) is equivalent to the vocabulary size (v). The v centroids of the clusters represent the visual geometric words of the vocabulary $\{p_1, \dots, p_v\}$, such that each visual word p_i is a d-dimensional HKS based feature vector.

Then, 3D object representation sub phase represents a given 3D model by a compact shape descriptor $F(X)$. Such shape descriptor represents the 3D model's relation with the geometric vocabulary through computing the distribution of geometric words. Given a geometric vocabulary $\mathcal{P} = \{p_1, \dots, p_v\}$ and a 3D model X with n vertices $\{x_1, \dots, x_n\}$, where each vertex x is associated with a d-dimensional HKS-based descriptor $p(x)$, it is feasible to compute a feature distribution $\theta(x)$ for each vertex x. Such feature distribution $\theta(x) = \{\theta_1(x), \dots, \theta_v(x)\}^T$ is a $v \times 1$ vector whose elements are defined as follows:

$$\theta_i(x) = c(x)\, e^{-\frac{\|p(x)-p_i\|_2^2}{2\sigma^2}} \qquad (5)$$

where $c(x)$ is selected in such a way that $\|\,\theta(x)\| = 1$ [8].

$\theta_i(x)$ can be interpreted as the probability of the vertex x to be associated with the geometric word or descriptor p_i from the constructed geometric vocabulary $\mathcal{P}$. Equation (5) represents the "soft" version of vector quantization. By choosing the parameter $\sigma \approx 0$, the process boils down to the standard "hard" vector quantization. Applying hard vector quantization is analogous to signing each descriptor $p(x)$ the index of its nearest neighbour p_i in the geometric vocabulary. In case of hard vector quantization, elements of feature distribution $\theta(x) = \{\theta_1(x), \dots, \theta_v(x)\}^T$ can be defined as follows:

$$\theta_i(x) = 1 \;\; if \;\; i = argmin_{i \in [1 \dots v]} \|p(x) - p_i\|; \qquad \qquad (6)$$
$$\theta_i(x) = 0 \; otherwise$$

where v is the number of geometric words in the vocabulary.

After vector Quantization, the feature distribution $\theta(x)$ is finally integrated over the entire 3D model, yielding a $v \times 1$ vector *F(X)*:

$$F(X) = \int_X \theta(x) \, d\mu(x) \qquad (7)$$

which is referred as a Bag-of-Features (BoF) or Bag-of-Words (BOW).

The feature vector $F(X)$ is considered a compact and significant descriptor for a given 3D model X [9]. It should be pointed out that in a real discrete environment, BoF is computed as follows:

$$F(X) = \sum_{i=1}^{n} \theta(x_i) \qquad (8)$$

where n is the number of vertices for a given 3D model X.

2.4. Matching

The feature vector representing the bag of features computed for a given 3D model through our proposed 3D object retrieval technique, entirely reflects its geometrical features and is highly discriminative compared to other feature descriptors. Thereupon, two given 3D models can be compared by comparing their corresponding bags of features. For comparing two 3D objects, their corresponding bags of features are treated as feature vectors (histograms), such that their similarity or dissimilarity can be determined through calculating the distance between their bags of features. Similar 3D models tend to have similar bags of features, whereas different 3D models tend to have different bags of features.

In order to compare two given 3D models X and Y, the distance d_{BoF} between the two bags of features $F(X)$ and $F(Y)$ is defined as follows:

$$d_{BoF} = \|F(X) - F(Y)\| \qquad (9)$$

On comparing two given 3D models, smaller values of distance (d_{BoF}) indicates that the two 3D models are highly similar to each other, while on the other hand higher values of d_{BoF} indicates that the two 3D models are dissimilar to each other. Several retrieval experiments will be carried out through the following section proving the correctness of the proposed 3D objet retrieval approach.

3. Experimental Results

3.1. Dataset and Evaluation Metrics

The shape retrieval dataset SHREC 2011 [10] is selected to evaluate the proposed approach and test its performance. It is the latest published online dataset dedicated for non-rigid 3D models. It is a large scale dataset consisting of 600 non-rigid 3D models that are equally classified into 30 categories. The 3D models are represented as watertight triangular meshes that are derived from 30 original models, represented in different deformed versions. The database contains a set of models which have similar overall appearances but belong to various categories because they are different in the details of local regions or/and topological structures. SHREC 2011 is the most diverse non-rigid 3D shape database available today in terms of object classes and deformations.

For the sake of quantitative assessment of the proposed approach performance, the following evaluation metrics are considered:

- **Nearest Neighbour (NN):** It is the percentage of queries for which the closest match belongs to the query's class.
- **First Tier (FT):** It is the recall for the $(C-1)$ closest matches, where C is the cardinality of the query's class
- **Second Tier (ST):** It is the recall for the $2(C-1)$ closest matches. It is a little less stringent than the First Tier statistic, since the considered closest matches are twice as big in case of First Tier computation.
- **E-Measure:** This evaluation measure considers only the first 32 retrieved models for every query and calculates the precision and recall over those results. The maximum score is 1.0, and higher values indicate better results.
- **Discounted Cumulative Gain (DCG**): This statistic gives more importance to correct detections near the front of the list more than correct results later in the ranked list; the objective is to reflect how well the overall retrieval would be viewed by a human.
- **Mean Average Precision (mAP):** It is also known in information retrieval as "average precision at seen relevant documents". It determines precision at each element when a new relevant 3D model gets retrieved; such that $\boldsymbol{mAP}$ favours systems which return relevant 3D models fast.

3.2. Parameters Settings

We conducted various experimental results to choose the optimum value or optimum way of computation for different influential parameters included within our proposed technique for 3D object retrieval. Since it will be hard to exhaustively show in detail all these experiments, we are going to focus only on the most influential parameter within our approach, which is the d-dimensional feature descriptor $(p(x))$ used for feature description . Several experiments are conducted to select the optimum way for selecting such vital d-dimensional feature descriptor $(p(x))$ achieving the best retrieval results. This is achieved by directly taking a subset of the normalized m-dimensional HKS-based feature vector $(h(x))$ in different ways, where m is equal to 100. One could think that choosing the feature descriptor $p(x)$ incorporating the values existing at the end of the feature vector $h(x)$ could achieve best results, since the HKS values at the end of the feature vector $K_t(x,x)$ are more stable than that at its beginning, where they remain almost unchanged at t values approaching the t_{max}. Therefore, the first experiment (Run 1) is applied by selecting p(x) to include the last six values $[h_{95}, h_{96}, h_{97}, h_{98}, h_{99}\ and\ h_{100}]$ in the feature vector $h(x)$, producing a sixth dimensional feature descriptor.

Another intuitive idea may arise from treating the 100 HKS values in the feature vector $h(x)$ as a signal which needs to be described or encoded by the most significant values or statistical measurements that could best describe its behavior. Thus the second experiment (Run 2) is conducted by selecting the minimum, maximum, mean, median and standard deviation of the 100 HKS values producing a fifth dimensional feature descriptor $p(x)$. On the other hand, the third, fourth and fifth experiments are conducted by sampling the HKS values in different ways using different intervals. In the third experiment (Run 3), $p(x)$ is chosen to incorporate a sequence of HKS values with interval equals to 5 starting with h_5 and ending with h_{95} producing a feature vector of dimensionality equals to 19. The fourth experiment (Run 4) is conducted by increasing the interval to 10; such that the feature vector $p(x)$ is chosen to incorporate a sequence of HKS values beginning with h_{10} and ending with h_{90} producing a ninth dimensional feature descriptor $p(x)$. Finally, the fifth experiment (Run 5) is conducted by increasing the interval to 20, such that the feature vector $p(x)$ is chosen to incorporate a sequence of HKS values starting with h_{20} and ending with h_{80} producing the shortest feature descriptor with fourth dimensionality. **Table 1** summarizes how p(x) is selected during our five conducted experiments, where the rightmost column indicates the dimensionality of the $p(x)$ feature vector.

Table 1. Different selections of the d-Dimensional feature descriptor ($p(x)$) in our five conducted experiments

Run	$\mathbf{p(x)}$	Dimensionality (d)
Run 1	$[h_{95}(x), h_{96}(x), h_{97}(x), h_{98}(x), h_{99}(x), h_{100}(x)]$	6
Run 2	$[min(h(x)), max(h(x)), mean(h(x)), median(h(x)), stdev(h(x))]$	5
Run 3	$[h_5(x), h_{10}(x), \ldots, h_{90}(x), h_{95}(x)]$	19
Run 4	$[h_{10}(x), h_{20}(x), \ldots, h_{80}(x), h_{90}(x)]$	9
Run 5	$[h_{20}(x), h_{40}(x), h_{60}(x), h_{80}(x)]$	4

Table 2 shows the effect of varying the selection of the $\boldsymbol{d}$-dimensional feature descriptor ($\boldsymbol{p(x)}$) on the overall retrieval performance through our five conducted experiments. The six evaluation metrics used here are NN, FT, ST, E, DCG and MAP. All these experiments are evaluated on the whole database of SHREC 2011.

It is obvious from **Table 2** that the first experiment (Run 1) attains the worst results compared with the other four experiments. This Run was expected to outperform other experiments since it represents the $p(x)$ with the most expectedly stable HKS values (last 6 values in the $h(x)$ feature vector), but conversely in practice it is found that such approximately similar values describing feature points fail in providing a feature space with sufficient variability. Such variability is essentially required to achieve a successful clustering that is capable of producing distinct significant geometric words during vocabulary construction phase. Moreover, it can be easily detected that the fifth experiment (Run 5) outperforms all other experiments, particularly upon considering the FT, ST, E and MAP evaluation metrics, which is fortunately the one owning the shortest and most compact feature descriptor $p(x)$ with dimensionality equals to four. It is worth mentioning, that being compact in size is one of the most important requirement that should be achieved in a feature descriptor. Using a feature descriptor of fourth dimensionality will have an influential effect on reducing both the time and space complexities required for clustering the feature space during the vocabulary construction phase. It is also observed that the last four runs achieve the ideal score of NN evaluation measure (equals to 1). Thus, if the feature descriptor $p(x)$ is selected in a way similar to any of those four compared experiments then the first retrieved 3D model in the ranked list will be correctly retrieved for all query models in the database.

Considering the values of NN, ST, E, DCG and MAP evaluation metrics, the third and the fifth runs achieve approximate performance results outperforming other experiments. Even if we assume that they have exactly similar high performance results, it is more sensible to choose the more compact feature descriptor which makes $p(x)$ tested by the fifth run a more favourable choice than that tested by the third run. It should be pointed out that the length (dimensionality) of the feature descriptor $p(x)$ tested by the third run ($d = 19$) is almost five times the dimensionality of the feature descriptor tested by the fifth run ($d = 4$).

Table 2. Performance of proposed approach using different choices of d-dimensional feature descriptor ($p(x)$)

Run	NN	FT	ST	E	DCG	MAP (%)	Dimensionality(d)
Run 1	0.943	0.786	0.865	0.632	0.922	83.47	6
Run 2	1	0.977	0.99	0.736	0.997	98.8	5
Run 3	1	0.985	0.994	0.739	0.998	99.3	19
Run 4	1	0.984	0.994	0.738	0.997	99.19	9
Run 5	**1**	**0.988**	**0.995**	**0.741**	**0.998**	**99.44**	**4**

For the HKS computation of a 3D model with n vertices, first the cotangent weight scheme [11] is used for discretizing the Laplace-Beltrami operator (Laplacian matrix) on triangular meshes. Through experimental results, this discretization proves to perform better than other discretization techniques such as mesh Laplace operator [12] and preserves many important properties of the continuous Laplace-Beltrami operator, such as positive semi-definiteness, symmetry and locality. Then, the sparse eigensolver implemented in Matlab computes the smallest 300 eigenvalues and eigenvectors ($k = 300$) of the Laplace-Beltrami operator. Finally, HKS ($\mathrm{K_t(x, x)}$) is computed for each vertex by uniformly sampling 100 points ($m = 100$) in the logarithmically scale over time interval $[t_{min}, t_{max}]$ with $t_{min} = 4\, ln\, 10 / \lambda_{300}$ and $t_{max} = 4\, ln\, 10 / \lambda_2$nd.

For vocabulary construction, K-means algorithm is chosen for clustering the feature space. The K-means method is implemented several times with different values of K(number of clusters), in order to determine the optimum value for the parameter K which is equivalent to the parameter v (vocabulary size). The best retrieval results are achieved when K is set to 64. Therefore, the constructed geometric vocabulary ($\mathcal{P} = \{\mathrm{p_1}, \dots, \mathrm{p_{64}}\}$) comprises 64 geometric words.

For 3D object representation, hard version of vector quantization scheme is adopted since it achieves retrieval results outperforming that while using “soft” version of vector quantization. After vector Quantization, the feature distribution $\theta(\mathrm{x})$ is integrated over the entire 3D model yielding the final Bag-of-Features vector ($\mathrm{F(X)}$) which is a 64-dimensional feature vector. Finally, $L1 - norm$ (Manhattan distance) is used for comparing two different bags of features during the final phase of matching.

All these settings are experimentally found to give optimal retrieval performance on SHREC 2011 database. Matlab is used for implementing most of the code where some parts are written in C with Mex interface. Some other parts of the code are completely written in C++ using Microsoft Visual Studio 2010. All the implementation and the conducted experiments are completed on a laptop with 1.73 GHz Intel Core i7 CPU, 6 MB cache and 6 GB of RAM.

3.3. Retrieval Results

This section is devoted for visualizing samples of our retrieval results, besides showing the robustness of the proposed technique against various deformation, transformations and noise. Some examples of retrieved 3D models returned by the proposed technique are visualized in **Figure 1**. The leftmost column of 3D models with green squares represents the queries belonging to different categories. The ten following columns represent the closest ten retrieved 3D models for each query, ordered from left to right based on their relevance, which is inversely proportional to the distance from the query 3D model. It is obvious that the retrieved 3D models of all given queries belong to the same query's category, indicating that they are relevant matches.

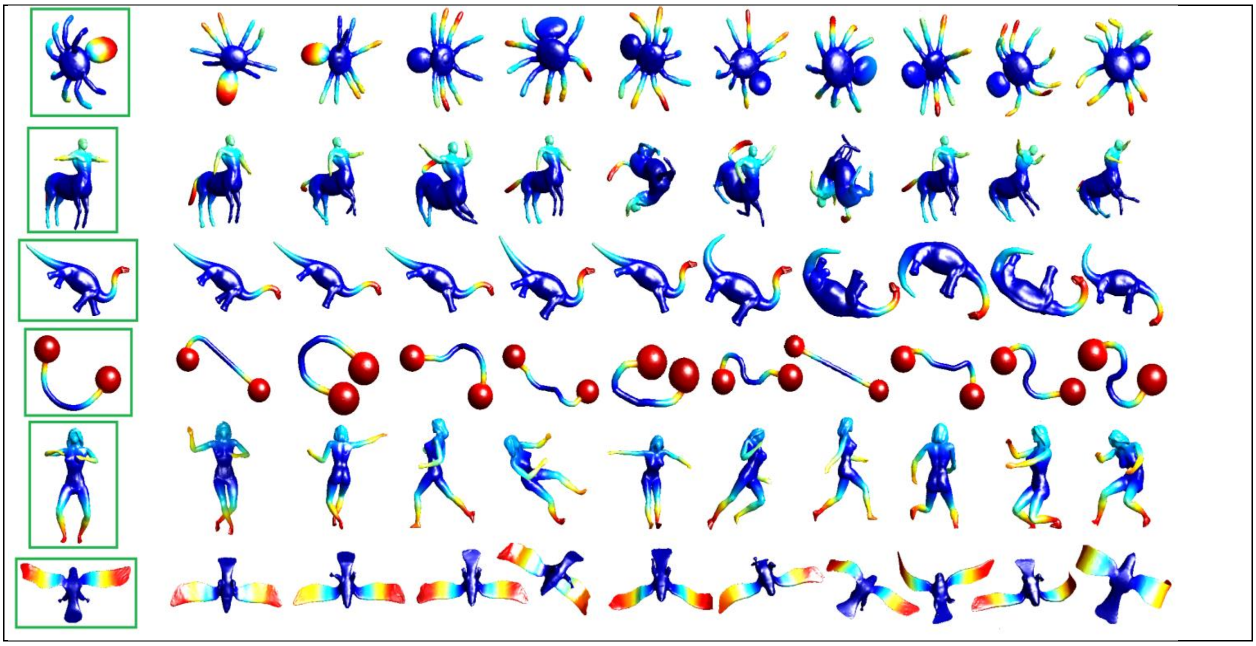

Figure 1. Sample of 3D object retrieval results.

It is essential to highlight that SHREC 2011 dataset contains 3D models arbitrarily scaled, rotated and translated. Therefore, such outstanding retrieval results indicate that the proposed technique is invariant to different kinds of transformations. This implies that a preliminary stage of pose normalization is not required before applying the proposed algorithm. Moreover, **Figure 2** shows how the proposed descriptor is intrinsically invariant to different isometric deformations. The retrieval results of three query models with different deformations, belonging to the same category are visualized. It is obvious that the first 10 retrieved 3D models of all the given queries are relevant matches despite of the existence of different classes of deformations applied on their corresponding query models.

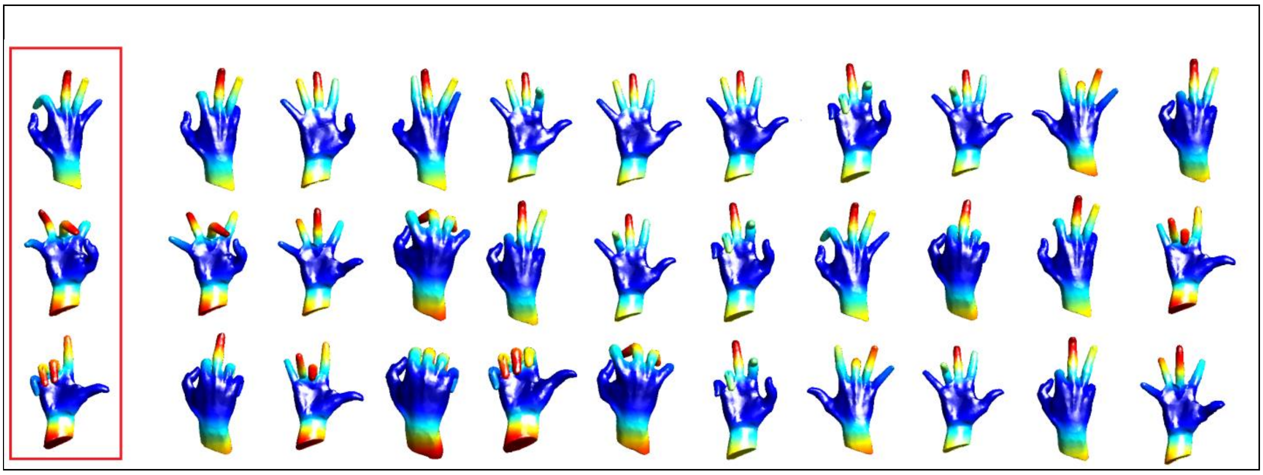

Figure 2. Sample of 3D object retrieval results given queries with different deformations.

In order to emphasize the robustness of the proposed technique with respect to noise, **Figure 3** illustrates some of the retrieval results for different query models corrupted with a noise level equals to 0.01. Here, the noise level is analogous to the noise-to-signal ratio (NSR) between the variance of noise and the variance of the original signal (coordinates of the vertices). Green-colored models indicate irrelevant matches belonging to different category other than the query's category. It is clear from the figure that our technique attains reasonable retrieval rate in case of corrupted queries, having insignificant number of false matches versus the number of correct matches confirming the stability of our technique against noise.

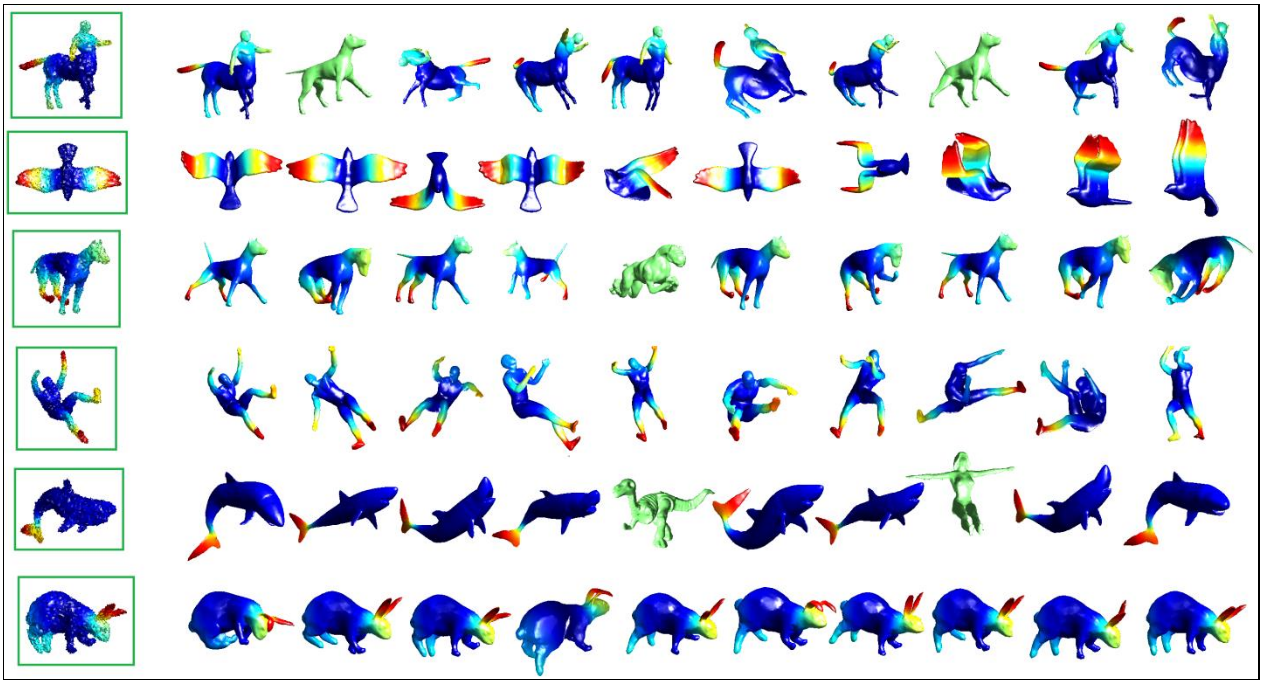

Figure 3. Sample of retrieval results given corrupted models with noise level = 0.01.

4. Conclusion and Future Works

In this paper, an efficient and robust approach relying on the synergy between Heat Kernel Signatures (HKS) and Bag of Features (BoF) paradigm has been designed for the purpose of 3D object retrieval. The experimental results lead to the conclusion that the proposed technique is quite effective for the purpose of 3D Object Retrieval, showing very high retrieval accuracy and descriptive power. It achieves state of the art results on SHREC 2011 dataset; a public well known benchmark of non-rigid 3D models. The results have indeed confirmed that the proposed descriptor is invariant against different kinds of deformations and transformations on one hand and can handle 3D models under noise (distortion) on the other hand. Moreover, it is significant that the proposed technique is computationally efficient. This is certainly due to the fact that the applied innovative filtering technique together with the choice of compact 4^{th} dimensional HKS based feature vector for point feature description dramatically reduces time and space complexity required for applying the whole BoF model.

In Future work, it is intended to explore in depth the following topics:

- Adapting the proposed 3D Object Retrieval technique to handle partial matching problem.
- Testing the proposed technique on domain-specific benchmarks.
- Boosting retrieval efficiency by using parallel or distributed computing methods.
- Exploring other applications possibly taking advantage of the proposed approach such as dense correspondence between non-rigid 3D models, segmentation and pose estimation.
- Evaluating the performance of the proposed innovative filtering technique separately on a public well-known benchmark dedicated for feature detection and description.
- Extending the application of the proposed 3D Object Retrieval technique on volumetric 3D models.

References:

[1] Jian Sun, Maks Ovsjanikov, and Leonidas Guibas, "A Concise and Provably Informative Multi-Scale Signature Based On Heat Diffusion," *Proceedings of the Seventh Eurographics Symposium on Geometry Processing (SGP '09)*, pp. 1383 - 1392, 2009.

[2] Michael M. Bronstein, "Scale-Invariant Heat Kernel Signatures for Non-Rigid Shape Recognition," *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '10)*, pp. 1704 - 1711, June 2010.

[3] Yi Fang, Mengtian Sun, and Karthik Ramani, "Temperature Distribution Descriptor for Robust 3D Shape Retrieval," *Proceedings of the IEEE Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 9 - 16, June 2011.

[4] Yi Fang, Mengtian Sun, Minhyong Kim, and Karthik Ramani, "Heat-Mapping: A Robust Approach Toward Perceptually Consistent Mesh Segmentation," *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '11)*, pp. 2145 - 2152, June 2011.

[5] Ryutarou Ohbuchi, Kunio Osada, Takahiko Furuya, and Tomohisa Banno, "Salient Local Visual Features for Shape-Based 3D Model Retrieval," *Proceedings of the International Conference on Shape Modeling and Applications (SMI '08)*, pp. 93 - 102, June 2008.

[6] David G. Lowe, "Distinctive Image Features from Scale-Invariant Keypoints," *International Journal of Computer Vision (IJCV)*, vol. 60, no. 2, November 2004.

[7] Takahiko Furuya and Ryutarou Ohbuchi, "Dense Sampling and Fast Encoding For 3D Model Retrieval Using Bag-Of-Visual Features," *Proceedings of the ACM International Conference on Image and Video Retrieval (CIVR '09)*, no. 26, pp. 1 - 8, July 2009.

[8] Alexander M. Bronstein, Michael M. Bronstein, Leonidas J. Guibas, and Maks Ovsjanikov, "Shape Google: Geometric Words and Expressions for Invariant Shape Retrieval," *ACM Transactions on Graphics (TOG)*, vol. 30, no. 1, January 2011.

[9] Alexander M. Bronstein, Michael M. Bronstein, Leonidas J. Guibas, and Maks Ovsjanikov, "Shape Google: A Computer Vision Approach to Isometry Invariant Shape Retrieval," *Proceedings of the 12th IEEE International Conference on Computer Vision Workshops (ICCVW '09)*, pp. 320 - 327, 2009.

[10] Z. Lian et al., "SHREC'11 Track: Shape Retrieval on Non-rigid 3D Watertight Meshes," *Proceedings of the 4th Eurographics conference on 3D Object Retrieval (3DOR'11)*, pp. 79 - 88, 2011.

[11] Ulrich Pinkall and Konrad Polthier, "Computing Discrete Minimal Surfaces and Their Conjugates," *Experimental Mathematics (EXP MATH)*, vol. 2, no. 1, pp. 15 - 36, 1993.

[12] Mikhail Belkin, Jian Sun, and Yusu Wang, "Discrete Laplace Operator on Meshed Surfaces," *Proceedings of the twenty-fourth annual Symposium on Computational Geometry (SCG '08)*, pp. 278 - 287, 2008.