Eurographics Workshop on 3D Object Retrieval (2018)
A. Telea, T. Theoharis, and R. C. Veltkamp (Editors)

# Geodesic-based 3D Shape Retrieval Using Sparse Autoencoders

Lorenzo Luciano and A. Ben Hamza

Concordia Institute for Information Systems Engineering, Concordia University, Montreal, Canada

**Abstract**

*In light of the increased processing power of graphics cards and the availability of large-scale datasets, deep neural networks have shown a remarkable performance in various visual computing applications. In this paper, we propose a geometric framework for unsupervised 3D shape retrieval using geodesic moments and stacked sparse autoencoders. The key idea is to learn deep shape representations in an unsupervised manner. Such discriminative shape descriptors can then be used to compute the pairwise dissimilarities between shapes in a dataset, and to find the retrieved set of the most relevant shapes to a given shape query. Experimental evaluation on three standard 3D shape benchmarks demonstrate the competitive performance of our approach in comparison with state-of-the-art techniques.*

## 1. Introduction

Shape retrieval is a fundamental problem in a wide range of fields, including computer vision, geometry processing, medical imaging, and computer graphics. Given a database of shapes, the goal of shape retrieval is to find the set of most relevant shapes to a query shape. The 3D shape retrieval problem, for instance, has been attracting much attention in recent years, fuelled primarily by increasing accessibility to large-scale 3D shape repositories that are freely available on the Internet [CFG*15].

Spectral geometry is at the core of several state-of-the-art techniques that effectively tackle the problem of nonrigid 3D shape retrieval, achieving excellent performance on the latest 3D shape retrieval contests [PSR*14, LBBC14, BCA*14, ZLCE*15, SYS*16]. Most of these approaches represent a 3D shape by a spectral signature, which is a concise and compact shape descriptor aimed at facilitating the retrieval tasks. Examples of spectral shape descriptors include global point signature [Rus07], heat kernel signature [SOG09], scale-invariant heat kernel signature [BK10], wave kernel signature [ASC11], spectral graph wavelet signature [LB13], improved wave kernel signature [LW15], and reduced biharmonic distance matrix signature [YY15].

The recent trend in shape analysis is geared towards using deep neural networks to learn features at various levels of abstraction. It is no secret that deep learning is the buzzword of the moment in both academic and industrial circles, and the performance of deep neural networks has been quite remarkable in a variety of areas such as speech recognition, image recognition, natural language processing, and geometry processing [Sch15, NYN*15, Ben09, BBL*16]. The trend toward deep neural networks has been driven, in part, by a combination of affordable computing hardware, open source software, and the availability of large-scale datasets.

Although applying deep neural networks to 3D shapes, particularly to mesh data, is not straightforward, several deep learning architectures have been recently proposed to tackle various 3D shape analysis problems in a bid to learn higher level representations of shapes [SMKLM15, WSK*15, QSN*16, BLH*14, BBZ*16]. Su *et al.* [SMKLM15] presented a convolutional neural network architecture that combines information from multiple views of a 3D shape into a single and compact shape descriptor. Wu *et al.* [WSK*15] proposed a deep learning framework for volumetric shapes via a convolutional deep belief network by representing a 3D shape as a probabilistic distribution of binary variables on a 3D voxel grid. Brock *et al.* [BLRW16] proposed a voxel-based approach to 3D object classification using variational autoencoders and deep convolutional neural networks, achieving improved classification performance on the ModelNet benchmark. Sedaghat *et al.* [SZB17] showed that forcing the convolutional neural network to produce the correct orientation during training yields improved classification accuracy. Bu *et al.* [BLH*14] introduced a deep learning approach to 3D shape classification and retrieval using a shape descriptor represented by a full matrix defined in terms of the geodesic distance and eigenfunctions of the Laplace-Beltrami operator [Ros97, BBK08]. Bai *et al.* [BBZ*16] introduced a real-time 3D shape search engine based on the projective images of 3D shapes. Xie *et al.* [XDF17] proposed a multi-metric deep neural network for 3D shape retrieval by learning non-linear distance metrics from multiple types of shape features, and by enforcing the outputs of different features to be as complementary as possible via the Hilbert-Schmidt independence criterion. A comprehensive review of deep learning advances in 3D shape recognition can be found in [ICNK17].

In this paper, we introduce a deep learning approach, dubbed



DOI: 10.2312/3dor.20181049

DeepGM, to 3D shape retrieval. The proposed technique leverages recent developments in machine learning and geometry processing to effectively represent and analyze 3D shapes at various levels of abstraction in an effort to design a compact yet discriminative shape representation in an unsupervised way. More specifically, we use stacked sparse autoencoders to learn deep shape descriptors from geodesic moments of 3D shapes. The geodesic moments are geometric feature vectors defined in terms of the geodesic distance on a 3D shape, while stacked sparse autoencoders are deep neural networks consisting of multiple layers of sparse autoencoders that attempt to enforce a constraint on the sparsity of the output from the hidden layer.

We show that our proposed framework unsupervisedly learns geometric features from shapes with the aim of designing a highly discriminative shape descriptor that yields better retrieval results compared to existing methods, including supervised learning techniques. The main contributions of this paper may be summarized as follows:

- We present a geometric framework for 3D shape retrieval using geodesic moments.
- We propose an unsupervised approach for learning deep shape descriptors using stacked sparse autoencoders.
- We show through extensive experiments the competitive performance of the proposed approach in comparison to existing shape retrieval techniques on several 3D shape benchmarks using various evaluation metrics.

The remainder of this paper is organized as follows. In Section 2, we introduce a deep learning framework with geodesic moments for 3D shape retrieval using stacked sparse autoencoders, and we discuss the main components of our proposed algorithm. Experimental results on both synthetic and real datasets are presented in Section 3 to demonstrate the efficiency of our approach. Finally, we conclude in Section 4.

## 2. Method

In this section, we present a deep learning approach to 3D shape retrieval using geodesic moments and stacked sparse autoencoders. We start by defining the geodesic moments, and then we describe in detail the key steps of our proposed algorithm.

### 2.1. Geodesic Moments

A 3D shape is usually modeled as a triangle mesh $\mathbb{M}$ whose vertices are sampled from a Riemannian manifold. A triangle mesh $\mathbb{M}$ may be defined as a graph $\mathbb{G} = (\mathcal{V}, \mathcal{E})$ or $\mathbb{G} = (\mathcal{V}, \mathcal{T})$, where $\mathcal{V} = \{\mathbf{v}_1, \ldots, \mathbf{v}_m\}$ is the set of vertices, $\mathcal{E} = \{e_{ij}\}$ is the set of edges, and $\mathcal{T}$ is the set of triangles. Each edge $e_{ij} = [\mathbf{v}_i, \mathbf{v}_j]$ connects a pair of vertices $\{\mathbf{v}_i, \mathbf{v}_j\}$ (or simply $\{i, j\}$). We define the $k$th geodesic moment at a mesh vertex $j$ as

$$\mu_k(j) = k \sum_{i=1}^{m} d_{ij}^{k-1} a_i, \tag{1}$$

where $a_i$ is the area of the Voronoi cell at vertex $i$, and $d_{ij}$ is the geodesic distance between mesh vertices $i$ and $j$. Hence, we may represent the shape $\mathbb{M}$ by an $m \times p$ geodesic moment matrix $\mathbf{M} = (\mu_1, \ldots, \mu_m)^\mathsf{T}$, where $\mu_j = (\mu_1(j), \ldots, \mu_p(j))$ is a $p$-dimensional vector consisting of the first $p$ moments (i.e. arranged in increasing order of magnitude) at vertex $j$. Figure 1 illustrates a triangle mesh consisting of $m = 689$ vertices, as well as the graph geodesic distance matrix between all mesh vertices, and the normalized vertex area plot.

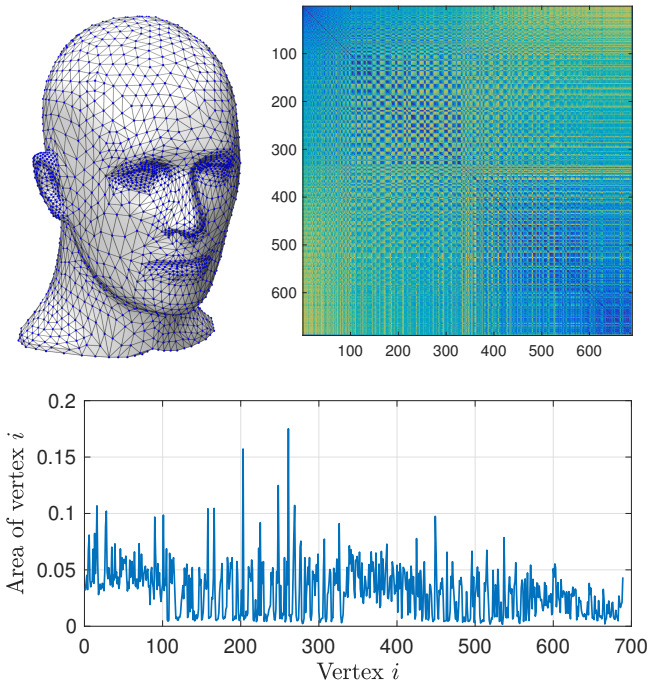


**Figure 1:** *Triangle mesh (top left); graph geodesic distance matrix (top right); and normalized vertex area plot (bottom).*

### 2.2. Proposed Algorithm

The objective of 3D shape retrieval is to search and extract the most relevant shapes to a query shape from a dataset of 3D shapes. The retrieval accuracy is usually evaluated by computing a pairwise dissimilarity measure between shapes in the dataset. A good retrieval algorithm should result in few dissimilar shapes. A commonly used dissimilarity measure for content-based retrieval is the $\ell_1$-distance, which quantifies the difference between each pair of 3D shapes.

Our proposed DeepGM approach to 3D shape retrieval consists of two major steps. In the first step, we compute the $p \times p$ matrix $\mathbf{S}_i = \mathbf{M}_i^\mathsf{T} \mathbf{M}_i$ for each shape $\mathbb{M}_i$ in the dataset $\mathcal{D} = \{\mathbb{M}_1, \ldots, \mathbb{M}_n\}$, where $\mathbf{M}_i$ is the geodesic moment matrix and $p$ is the number of geodesic moments. Then, each matrix $\mathbf{S}_i$ is reshaped into a $p^2$-dimensional feature vector $\mathbf{x}_i$ by stacking its columns one underneath the other. Subsequently, all feature vectors $\mathbf{x}_i$ of all $n$ shapes in the dataset are arranged into a $p^2 \times n$ data matrix $\mathbf{X} = (\mathbf{x}_1, \ldots, \mathbf{x}_n)$.

In the second step, we use stacked sparse auto-encoders to learn deep features by training the hidden layers of the network individ-

ually in an unsupervised way. An autoencoder is comprised of an encoder and a decoder, as depicted in Fig. 2.

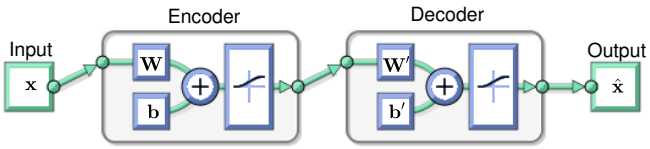


**Figure 2:** *Graphical diagram of an autoencoder.*

The encoder maps an input vector to a hidden representation and the decoder maps back the hidden representation to a reconstruction of the original input. More precisely, The encoder, denoted by $f_\theta$, maps an input vector $\mathbf{x} \in \mathbb{R}^q$ to a hidden representation (referred to as code, activations or features) $\mathbf{a} \in \mathbb{R}^r$ via a deterministic mapping

$$\mathbf{a} = f_\theta(\mathbf{x}) = \sigma(\mathbf{W}\mathbf{x} + \mathbf{b}), \tag{2}$$

parameterized by $\theta = \{\mathbf{W}, \mathbf{b}\}$, where $\mathbf{W} \in \mathbb{R}^{r \times q}$ and $\mathbf{b} \in \mathbb{R}^q$ are the encoder weight matrix and bias vector, and $\sigma$ is a nonlinear element-wise activation function such as the logistic sigmoid or hyperbolic tangent. The decoder, denoted by $g_{\theta'}$, maps back the hidden representation $\mathbf{h}$ to a reconstruction $\hat{\mathbf{x}}$ of the original input $\mathbf{x}$ via a reverse mapping

$$\hat{\mathbf{x}} = g_{\theta'}(\mathbf{a}) = \sigma(\mathbf{W}'\mathbf{a} + \mathbf{b}'), \tag{3}$$

parameterized by $\theta' = \{\mathbf{W}', \mathbf{b}'\}$, where $\mathbf{W}' \in \mathbb{R}^{q \times r}$ and $\mathbf{b}' \in \mathbb{R}^q$ are the decoder weight matrix and bias vector, respectively. The encoding and decoding weight matrices $\mathbf{W}$ and $\mathbf{W}'$ are usually constrained to be of the form $\mathbf{W}' = \mathbf{W}^\mathsf{T}$, which are referred to as tied weights. Assuming the tied weights case for simplicity, the parameters $\{\mathbf{W}, \mathbf{b}, \mathbf{b}'\}$ of the network are often optimized by minimizing the squared error $\sum_{i=1}^N \|\mathbf{x}_i - \hat{\mathbf{x}}_i\|_2^2$, where $N$ is the number of samples in the training set, $\mathbf{x}_i$ is the $i$th input sample and $\hat{\mathbf{x}}_i$ is its reconstruction.

To penalize large weight coefficients in an effort to avoid overfitting the training data and also to encourage sparsity of the output from the hidden layer, the following objective function is minimized instead

$$\mathcal{L}(\mathbf{W}, \mathbf{b}, \mathbf{b}') = \frac{1}{2}\sum_{i=1}^N \|\mathbf{x}_i - \hat{\mathbf{x}}_i\|_2^2 + \frac{\lambda}{2}\|\mathbf{W}\|_F^2 + \beta \sum_{j=1}^N \mathrm{KL}(\rho \| \hat{\rho}_j), \tag{4}$$

where $\lambda$ is a regularization parameter that determines the relative importance of the sum-of-squares error term and the weight decay term, and $\beta$ is the weight of the sparsity regularization term. This sparsity regularizer is the Kullback-Leibler divergence $\mathrm{KL}(\rho \| \hat{\rho}_j)$, which is a dissimilarity measure between $\rho$ and $\hat{\rho}_j$, and it is defined as

$$\mathrm{KL}(\rho \| \hat{\rho}_j) = \rho \log \frac{\rho}{\hat{\rho}_j} + (1-\rho) \log \frac{1-\rho}{1-\hat{\rho}_j}, \tag{5}$$

where $\hat{\rho}_j$ is the average activation value of the hidden unit $j$ and $\rho$ is its desired value which is typically small.

A stacked sparse autoencoder is a deep neural network consisting of multiple layers of stacked encoders from several sparse autoencoders. This stacked network is pre-trained layer by layer in a unsupervised fashion, where the output from the encoder of the first autoencoder is the input of the second autoencoder, the output from the encoder of the second autoencoder is the input to the third autoencoder, and so on. After pre-training, the entire stacked sparse autoencoder can be trained using backpropagation to fine-tune all the parameters of the network.

The geodesic vectors $\mathbf{x}_i$ of all $n$ shapes in the dataset are arranged into a $\kappa \times n$ data matrix $\mathbf{X} = (\mathbf{x}_1, \ldots, \mathbf{x}_n)$ on which a deep auto-encoder is performed, resulting in an $r_L \times n$ matrix $\mathbf{A} = (\mathbf{a}_L^{(1)}, \ldots, \mathbf{a}_L^{(n)})$ whose columns are deep learned shape representations (referred to as DeepGM descriptors), where $r_L$ is the total number of units in the last hidden layer of the network. The geodesic feature vector of a 3D table model is displayed in Figure 3(top), while Figure 3(bottom) shows the DeepGM descriptor of a 3D table model.

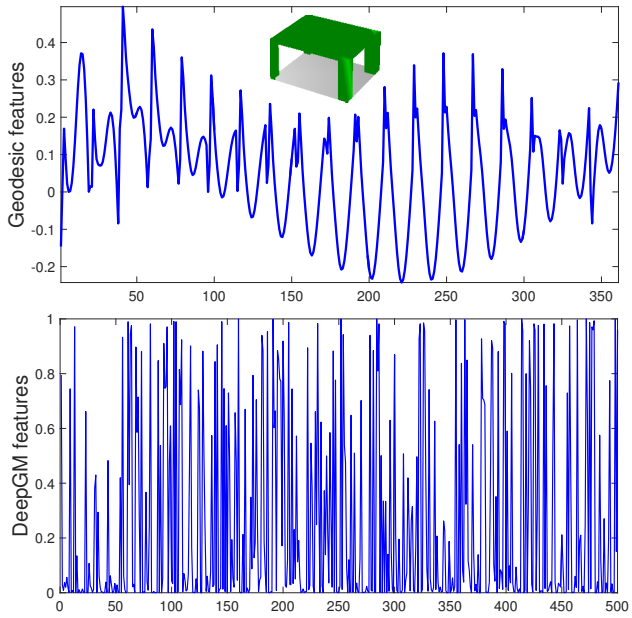


**Figure 3:** *Geodesic features (top) and DeepGM features (bottom) of a 3D table model.*

Finally, we compare a query shape to all shapes in the dataset using the $\ell_1$-distance between the DeepGM descriptors to measure the dissimilarity between each pair for 3D shape retrieval. Algorithm 1 summarizes the main algorithm steps of our DeepGM approach to 3D shape retrieval.

## 3. Experiments

In this section, we conduct extensive experiments to assess the performance of the proposed DeepGM approach in 3D shape retrieval. The effectiveness of our approach is validated by performing a comprehensive comparison with several shape retrieval methods

**Algorithm 1** DeepGM Retrieval

**Input:** Dataset $\mathcal{D} = \{\mathbb{M}_1, \ldots, \mathbb{M}_n\}$ of $n$ shapes, and $p$ geodesic moments.

1: **for** $i = 1$ to $n$ **do**
2: Compute the $m \times p$ geodesic moment matrix $\mathbf{M}_i$ for each 3D shape $\mathbb{M}_i$, where $m$ is the number of vertices.
3: Compute the $p \times p$ matrix $\mathbf{S}_i = \mathbf{M}_i^{\mathsf{T}} \mathbf{M}_i$, and reshape it into a $p^2$-dimensional vector $\mathbf{x}_i$
4: **end for**
5: Arrange all the feature vectors $\mathbf{x}_i$ into a $p^2 \times n$ data matrix $\mathbf{X} = (\mathbf{x}_1, \ldots, \mathbf{x}_n)$
6: Apply a stacked sparse auto-encoder on $\mathbf{X}$ to find the $r_L \times n$ matrix $\mathbf{A} = (\mathbf{a}_L^{(1)}, \ldots, \mathbf{a}_L^{(n)})$ of deepGM descriptors, where $r_L$ is the number of units in the last hidden layer.
7: Compute the $\ell_1$-distance between the DeepGM vector of the query and all DeepGM vectors in the dataset, and find the closest shape(s).

**Output:** Retrieved set of most relevant shapes to the query.

using standard performance evaluation metrics that are widely used in retrieval tasks.

**Datasets:** The effectiveness of the proposed shape retrieval framework is evaluated on three standard and publicly available 3D shape benchmarks [PSR*14, CFG*15]: synthetic SHREC-2014, real SHREC-2014, and SHREC-2016 [CFG*15]. Sample shapes from these widely-used datasets are displayed in Fig. 4.

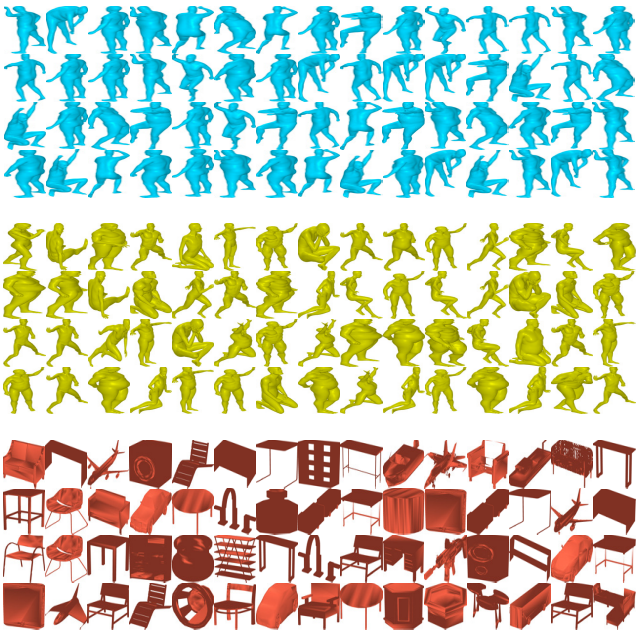

**Figure 4:** *Sample shapes from real SHREC-2014 (top), synthetic SHREC-2014 (middle), and SHREC-2016 (bottom).*

**Implementation details:** All the experiments were carried out on a desktop computer with a CPU Core i7 processor running at 3.4 GHz and 32 GB RAM; and all the algorithms were implemented in MATLAB. For feature extraction, we employed a stacked sparse autoencoder with two layers, as illustrated in Figure 5. We used the logistic sigmoid function as an activation function for both autoencoders. The sizes of the hidden layers for the first and second autoencoders are set to $h_1 = 1000$ and $h_2 = 500$, respectively. In the objective function of the stacked sparse autoencoder, we set the regularization parameter to $\lambda = 0.0001$, and the weight of the sparsity regularization term to $\beta = 3$. We also set the number of geodesic moments to $p = 20$ for all datasets. In other words, each shape in the synthetic SHREC-2014, real SHREC-2014 and SHREC-2016 datasets is represented by an input geodesic feature vector of dimension $p^2 = 400$.

In our DeepGM approach, we used the features learned by the second autoencoder to perform 3D shape retrieval. That is, we used the 500-dimensional deep feature vectors of the second hidden layer to compute the $\ell_1$-distance matrix.

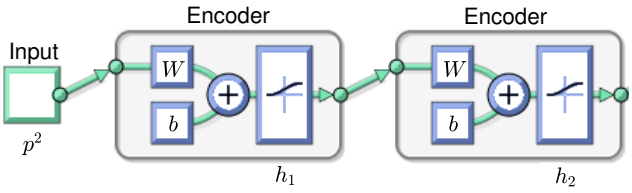


**Figure 5:** *Architecture of a two-layer stacked autoencoder.*

### 3.1. Results

In this section, we report the retrieval results of our approach and the baseline techniques on the synthetic SHREC-2014, real SHREC-2014 and SHREC-2016 datasets.

#### 3.1.1. SHREC-2014 dataset

The SHREC-2014 benchmark [PSR*14] consists of two datasets: real and synthetic. The real SHREC-2014 dataset is composed of 400 shapes made from 40 human subjects in 10 different poses. Half of the human subjects are male, and half are female. The poses of each subject are built using a data-driven deformation technique, which can produce realistic deformations of articulated meshes [CLC*13].

The synthetic SHREC-2014 dataset, on the other hand, consists of 15 different human models, each of which has its own unique body shape. Five human models are male, five are female, and five are child body shapes. Each of these models exists in 20 different poses, resulting in a total of 300 shapes. The same poses are used for each body shape, and objects are considered from the same class if they share the same body shape.

**Evaluation metrics:** The proposed approach is evaluated in comparison to existing state-of-the-art methods using several standard evaluation metrics [SMKF04], including Nearest Neighbor (NN), First-tier (FT) and Second-tier (ST), E-Measure (E), and Discounted Cumulative Gain (DCG).

**Baseline methods:** We carried out a comprehensive comparison between the proposed DeepGM framework and several state-of-the-art methods, including histograms of area projection transform (HAPT) [GL12], heat kernel signature based on time serial (HKS-TS) [PSR*14], Euclidean distance based canonical forms (EDBCF) [PSRM15], supervised dictionary learning (supDLtrain) [LBBC14], reduced biharmonic distance matrix (R-BiHDM) [YY15], and high-level feature learning using deep belief networks (3D-DL) [BLH*14]. These baselines are the best performing methods on the SHREC-2014 datasets.

**Performance evaluation:** To compute the pairwise distance matrix between all pairs of shapes in the real and synthetic SHREC-2014 datasets, we represent each shape by a 500-dimensional deep feature vector that is learned by the second autoencoder. More specifically, a 1000-dimensional feature representation is learned from the 400-dimensional geodesic feature vector using the first autoencoder. Then, the second autoencoder is employed to learn a reduced shape representation of 500 dimensions.

**Results:** In the first step of our DeepGM approach, each shape in the real and synthetic SHREC-2014 datasets is represented by a 400-dimensional geodesic feature vector (i.e. $p = 20$). Hence, the data matrix $\mathbf{X}$ for the real SHREC-2014 dataset is of size $400 \times 400$, while the data matrix for the synthetic SHREC-2014 datasetand is of size $400 \times 300$. Training the stacked sparse auto-encoder yields a DeepGM matrix $\mathbf{A}$ of size $500 \times 400$ for real SHREC-2014, and $500 \times 300$ for synthetic SHREC-2014.

Table 1 shows the retrieval rates for all methods on the real SHREC-2014 dataset, which consists of 400 shapes. A distance matrix of size $400 \times 400$ is constructed by computing the $\ell_1$-distance between each pair of the 500-dimensional deep feature vectors. Finally, a retrieval test on this distance matrix is conducted and the scores for the evaluation metrics are computed. As can be seen, comparing with the state-of-the-art supervised approach supDLtrain, our unsupervised DeepGM approach performs relatively well and gives the second best results for all the evaluation metrics except for NN and DCG.

**Table 1:** *Performance comparison results on the real SHREC-2014 dataset. Boldface numbers indicate the best retrieval performance.*

| | Retrieval Evaluation Measures (%) | | | | |
|---|---|---|---|---|---|
| Method | NN | FT | ST | E | DCG |
| HAPT [GL12] | **84.5** | 53.4 | 68.1 | 35.5 | 79.5 |
| HKS-TS [ZLCE*15] | 24.5 | 25.9 | 46.1 | 31.4 | 54.8 |
| EDBCF [PSRM15] | 1.0 | 1.2 | 4.0 | 4.3 | 27.9 |
| supDLtrain [LBBC14] | 79.3 | **72.7** | **91.4** | **43.2** | **89.1** |
| R-BiHDM [YY15] | 68.5 | 54.1 | 74.2 | 38.7 | 78.1 |
| 3D-DL [BLH*14] | 22.5 | 19.3 | 37.4 | 26.2 | 50.4 |
| DeepGM | 72.5 | 53.6 | 82.7 | 41.2 | 78.2 |

Table 2 summarizes the retrieval rates for all methods on the synthetic SHREC-2014 dataset, which consists of 300 shapes. A distance matrix of size $300 \times 300$ is obtained by computing the $\ell_1$-distance between each pair of the 400-dimensional deep feature vectors. Finally, a retrieval test on this distance matrix is conducted and the scores for the evaluation metrics are computed. As can be seen, DeepGM is the top performing method in terms of the NN measure at 99.3%, with a performance improvement of 3.3% over supDLtrain. Again, although our approach is unsupervised, it still outperforms supDLtrain in terms of NN and E measures, and gives the second best results in terms of the other evaluation metrics. Even in the case of ST and DCG, we are very close to the best reported performance with a thin 0.8% margin. A key advantage of unsupervised approaches is the possibility to learn larger and more complex models than with supervised methods. Supervised learning may be susceptible to over-fitting the training data and requires a large body of labeled data. Another advantage of unsupervised approaches is the ability to discover meaningful structure in the data.

**Table 2:** *Performance comparison results on the synthetic SHREC-2014 dataset. Boldface numbers indicate the best retrieval performance.*

| | Retrieval Evaluation Measures (%) | | | | |
|---|---|---|---|---|---|
| Method | NN | FT | ST | E | DCG |
| HAPT [GL12] | 97.0 | 73.3 | 92.7 | 65.5 | 93.6 |
| HKS-TS [ZLCE*15] | 46.7 | 47.6 | 74.3 | 50.4 | 72.9 |
| EDBCF [PSRM15] | 11.3 | 18.2 | 33.3 | 21.7 | 50.7 |
| supDLtrain [LBBC14] | 96.0 | **88.7** | **99.1** | 72.1 | **97.5** |
| R-BiHDM [YY15] | 79.3 | 57.2 | 76.0 | 53.3 | 83.6 |
| 3D-DL [BLH*14] | 92.3 | 76.0 | 91.1 | 64.1 | 92.1 |
| DeepGM | **99.3** | 81.4 | 98.3 | **72.3** | 96.7 |

### 3.1.2. SHREC-2016 dataset

The ShapeNet Core55 (SHREC-2016) is a subset of the ShapeNet dataset [CFG*15]. The ShapeNetCore contains about 51,300 models of over 55 common categories. Each of these common categories may be subdivided into several further subcategories. The SHREC-2016 dataset is split into a 70% training set, a 10% validation set, and a 20% test set.

**Baseline methods:** Using the SHREC-2016 shape benchmark, we carried out an extensive comparison between the proposed DeepGM framework and several state-of-the-art methods, including Multi-view Convolutional Neural Networks (MVCNN) [SMKLM15], Graphics Processing Unit acceleration and Inverted File Twice (GIFT) [BBZ*16], View Aggregation (VA) [SYS*16], Channel-wise CNN for Multitask Learning by Triplet (CCMLT) [SYS*16], and DB-FMCD-FUL-LCDR which is an appearance-based 3D shape feature extraction approach using pre-trained convolutional neural networks [SYS*16]. These baselines are the best performing approaches on the SHREC-2016 dataset.

**Evaluation metrics:** The DeepGM approach is evaluated on the SHREC-2016 dataset using several standard evaluation metrics [SYS*16], including Precision and Recall (P@N and R@N),

F-score (F1@N), Mean Average Precision (mAP), and Normalized Discounted Cumulative Gain (NDCG). Precision is the fraction of the models retrieved that are relevant to the query, while recall is the fraction of the models that are relevant to the query that are actually retrieved. The F-score is the weighted mean of precision and recall. The mean average precision for a set of queries is the mean of the average precision scores for each of these queries. The normalized discounted cumulative gain is a measure of the rankings quality of the retrieval results.

These evaluation metrics are used in macro and micro averaged versions. The macro version gives an unweighed average over the entire dataset (all models are averaged with equal weight). In the micro version, the query and retrieval results are treated equally across categories.

**Performance evaluation:** The SHREC-2016 dataset is divided into three distinct sets: a training set containing $36,147$ models, a validation set containing $5,165$ models, and a test set composed of $10,366$ models. For each of these three sets, we used a two-layer stacked sparse autoencoder to learn high-level feature descriptors for each shape. We first compute a 400-dimensional geodesic feature representation for each shape and then use it as input to the proposed DeepGM neural network model, resulting in a 500-dimensional deep shape descriptor for each shape.

**Results:** Following the same setup as in the previous experiments, the data matrices of geodesic feature vectors for the SHREC-2016 training, validation, and test datasets are of size $400 \times 36,147$, $400 \times 5,165$ and $400 \times 10,366$, respectively. The results on the SHREC-2016 dataset are summarized in Tables 3, 4 and 5. As can be seen, DeepGM outperforms the best performing method on the SHREC-2016 training dataset by a margin of 5.4% (resp. 8.6%) in terms of P@N using microALL (resp. macroALL). DeepGM also performs better than MVCNN on both the SHREC-2016 validation and test datasets in terms of P@N and NDCG. On the SHREC-2016 validation dataset, DeepGM has an NDCG score of 97.2 with microALL compared to just 93.8 for MVCNN. In terms of NDCG, DeepGM comes out way ahead with a score of 95.8 with macroALL versus 88.0 for MVCNN on the SHREC-2016 test dataset. It is also worth pointing out that DeepGM performs consistently better than the baseline methods using three evaluation metrics, namely P@N, mAP and NDCG. Overall, DeepGM delivers robust retrieval performance.

**Feature visualization:** The high-level features learned by our proposed DeepGM can be visualized using the t-Distributed Stochastic Neighbor Embedding (t-SNE) [vdMH08], which is a dimensionality reduction technique that is particularly well-suited for embedding high-dimensional data into a space of two or three dimensions. Figure 6 displays the t-SNE embeddings of the shapes in the SHREC-2016 dataset using the 400-dimensional geodesic feature vectors (top) and the 500-dimensional deep features (bottom) generated by our DeepGM approach. As can be seen, the two-dimensional embeddings corresponding to DeepGM are more separable than the ones corresponding to geodesic feature vectors. With geodesic features, the points are not discriminated very well, while with DeepGM features, the points are discriminated much better. In other words, DeepGM learns more discriminative features for 3D shape retrieval tasks, indicating the superior performance of deep features over shallow ones. Moreover, Figure 6 shows that the unsupervised DeepGM approach is exploratory in nature in the sense it can discover patterns and meaningful sub-groups in a dataset.

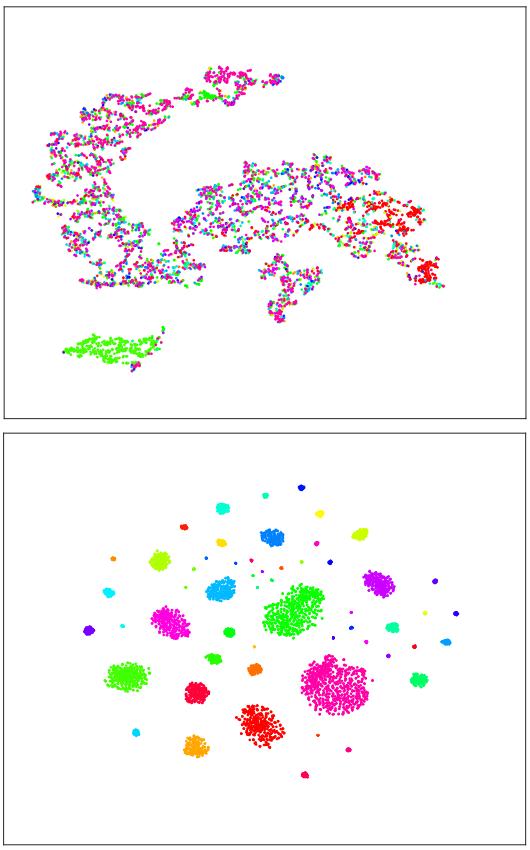

**Figure 6:** *Two-dimensional t-SNE feature visualization of geodesic moments (top) and DeepGM features (bottom) on the SHREC-2016 dataset.*

## 4. Conclusion

In this paper, we introduced an efficient geometric approach to 3D shape retrieval using geodesic moments and stacked sparse autoencoders. The proposed approach learns deep shape descriptors in an unsupervised way by leveraging the hierarchical representations in a discriminatively trained deep learning model. We showed that our DeepGM approach provides a comparable performance on the real and synthetic SHREC-2014 datasets, even against supervised techniques. Although our approach is unsupervised, it still outperforms supDLtrain in terms of several measures on synthetic SHREC-2014. In addition, DeepGM outperforms the state of the art on the more recent SHREC-2016 dataset by a comfortable margin of 7.8% on the test dataset using the NDCG metric. The two-dimensional visualization of shape representations demonstrates the discriminative power of deep features compared to the shallow ones. It is important to point out that the retrieval performance of DeepGM

**Table 3:** *Performance comparison results on the SHREC-2016 training dataset. Boldface numbers indicate the best retrieval performance.*

| | Retrieval Evaluation Measures (%) | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | microALL | | | | | macroALL | | | | |
| Method | P@N | R@N | F1@N | mAP | NDCG | P@N | R@N | F1@N | mAP | NDCG |
| MVCNN [SMKLM15] | 93.9 | 94.4 | **94.1** | 96.4 | 92.3 | 90.9 | 93.5 | **92.1** | 96.4 | 94.7 |
| GIFT [BBZ*16] | 84.1 | 57.1 | 62.0 | 90.7 | 91.2 | 63.4 | 45.2 | 47.2 | 81.5 | 89.1 |
| VA [SYS*16] | 82.7 | **99.6** | 86.4 | 99.0 | 97.8 | 37.4 | **99.7** | 46.0 | 98.2 | **98.6** |
| CCMLT [SYS*16] | 88.4 | 26.0 | 36.3 | 91.7 | 89.1 | 58.6 | 49.7 | 42.8 | 77.5 | 86.3 |
| DeepGM | **99.3** | 60.0 | 67.6 | **99.7** | **98.1** | **99.5** | 88.4 | 91.1 | **99.9** | **98.6** |

**Table 4:** *Performance comparison results on the SHREC-2016 validation dataset. Boldface numbers indicate the best retrieval performance.*

| | Retrieval Evaluation Measures (%) | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | microALL | | | | | macroALL | | | | |
| Method | P@N | R@N | F1@N | mAP | NDCG | P@N | R@N | F1@N | mAP | NDCG |
| MVCNN [SMKLM15] | 80.5 | 80.0 | **79.8** | 91.0 | 93.8 | 64.1 | 67.1 | **64.2** | 87.9 | 92.0 |
| GIFT [BBZ*16] | 74.7 | 74.3 | 73.6 | 87.2 | 92.9 | 50.4 | 57.1 | 51.6 | 81.7 | 88.9 |
| VA [SYS*16] | 34.3 | **92.4** | 44.3 | 86.1 | 93.0 | 8.70 | **87.3** | 13.2 | 74.2 | 85.4 |
| CCMLT [SYS*16] | 68.2 | 52.7 | 48.8 | 81.2 | 88.1 | 24.7 | 64.3 | 26.6 | 57.5 | 71.2 |
| DB-FMCD-FUL-LCDR [SYS*16] | 30.6 | 76.3 | 37.8 | 72.2 | 88.6 | 9.60 | 82.8 | 14.0 | 60.1 | 80.1 |
| DeepGM | **83.3** | 77.2 | 74.5 | **95.6** | **97.2** | **88.6** | 48.7 | 55.6 | **94.0** | **96.4** |

**Table 5:** *Performance comparison results on the SHREC-2016 test dataset. Boldface numbers indicate the best retrieval performance.*

| | Retrieval Evaluation Measures (%) | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | microALL | | | | | macroALL | | | | |
| Method | P@N | R@N | F1@N | mAP | NDCG | P@N | R@N | F1@N | mAP | NDCG |
| MVCNN [SMKLM15] | 77.0 | 77.0 | **76.4** | 87.3 | 89.9 | 57.1 | 62.5 | **57.5** | 81.7 | 88.0 |
| GIFT [BBZ*16] | 70.6 | 69.5 | 68.9 | 82.5 | 89.6 | 44.4 | 53.1 | 45.4 | 74.0 | 85.0 |
| VA [SYS*16] | 50.8 | **86.8** | 58.2 | 82.9 | 90.4 | 14.7 | **81.3** | 20.1 | 71.1 | 84.6 |
| CCMLT [SYS*16] | 71.8 | 35.0 | 39.1 | 82.3 | 88.6 | 31.3 | 53.6 | 28.6 | 66.1 | 82.0 |
| DB-FMCD-FUL-LCDR [SYS*16] | 42.7 | 68.9 | 47.2 | 72.8 | 87.5 | 15.4 | 73.0 | 20.3 | 59.6 | 80.6 |
| DeepGM | **78.4** | 73.2 | 69.6 | **93.6** | **96.5** | **85.4** | 45.9 | 52.3 | **92.2** | **95.8** |

yields consistent retrieval results across all datasets used for experimentation, while baselines perform less coherently from one dataset to another. This consistent performance is largely attributed to the fact that features learned via deep learning are transferable to other learning tasks, and even to other modalities and datasets.

## References


[ASC11] AUBRY M., SCHLICKEWEI U., CREMERS D.: The wave kernel signature: A quantum mechanical approach to shape analysis. In *Proc. Computational Methods for the Innovative Design of Electrical Devices* (2011), pp. 1626–1633. 1

[BBK08] BRONSTEIN A., BRONSTEIN M., KIMMEL R.: *Numerical Geometry of Non-rigid Shapes*. Springer, 2008. 1

[BBL*16] BRONSTEIN M., BRUNA J., LECUN Y., SZLAM A., VANDERGHEYNST P.: Geometric deep learning: going beyond Euclidean data. *arXiv:1611.08097* (2016). 1

[BBZ*16] BAI S., BAI X., ZHOU Z., ZHANG Z., LATECKI L. J.: Gift: A real-time and scalable 3d shape search engine. In *Proc. CVPR* (2016), pp. 5023–5032. 1, 5, 7

[BCA*14] BIASOTTI S., CERRI A., ABDELRAHMAN M., AONO M., BEN HAMZA A., EL-MELEGY M., FARAG A., GARRO V., GIACHETTI A., GIORGI D., GODIL A., LI C., LIU Y.-J., MARTONO H., SANADA C., TATSUMA A., VELASCO-FORERO S., XU C.-X.: SHREC'14 track: Retrieval and classification on textured 3D models. In *Proc. Eurographics Workshop on 3D Object Retrieval* (2014), pp. 111–120. 1

[Ben09] BENGIO Y.: Learning deep architectures for AI. *Foundations and Trends in Machine Learning 2*, 1 (2009), 1–127. 1

[BK10] BRONSTEIN M., KOKKINOS I.: Scale-invariant heat kernel signatures for non-rigid shape recognition. In *Proc. CVPR* (2010), pp. 1704–1711. 1

[BLH*14] BU S., LIU Z., HAN J., WU J., JI R.: Learning high-level

feature by deep belief networks for 3-D model retrieval and recognition. *IEEE Trans. Multimedia 24*, 16 (2014), 2154–2167. 1, 5

[BLRW16] BROCK A., LIM T., RITCHIE J., WESTON N.: Generative and discriminative voxel modeling with convolutional neural networks. *arXiv:1608.04236* (2016). 1

[CFG*15] CHANG A., FUNKHOUSER T., GUIBAS L., HANRAHAN P., HUANG Q., LI Z., SAVARESE S., SAVVA M., SONG S., SU H., XIAO J., YI L., YU F.: ShapeNet: An information-rich 3D model repository. *arXiv:1512.03012* (2015). 1, 4, 5

[CLC*13] CHEN Y., LAI Y.-K., CHENG Z.-Q., MARTIN R. R., JIN S.-Y.: A data-driven approach to efficient character articulation. In *Proc. Computer-Aided Design and Computer Graphics* (2013), pp. 32–37. 4

[GL12] GIACHETTI A., LOVATO C.: Radial symmetry detection and shape characterization with the multiscale area projection transform. *Computer Graphics Forum 31*, 5 (2012), 1669–1678. 4, 5

[ICNK17] IOANNIDOU A., CHATZILARI E., NIKOLOPOULOS S., KOMPATSIARIS I.: Deep learning advances in computer vision with 3D data: A survey. *ACM Computing Surveys 50* (2017), 1–38. 1

[LB13] LI C., BEN HAMZA A.: A multiresolution descriptor for deformable 3D shape retrieval. *The Visual Computer 29* (2013), 513–524. 1

[LBBC14] LITMAN R., BRONSTEIN A., BRONSTEIN M., CASTELLANI U.: Supervised learning of bag-of-features shape descriptors using sparse coding. *Computer Graphics Forum 33*, 5 (2014), 127–136. 1, 4, 5

[LW15] LIMBERGER F., WILSON R.: Feature encoding of spectral signatures for 3D non-rigid shape retrieval. In *Proc. BMVC* (2015). 1

[NYN*15] NODA K., YAMAGUCHI Y., NAKADAI K., OKUNO H., OGATA T.: Audio-visual speech recognition using deep learning. *Applied Intelligence 42*, 4 (2015), 722–737. 1

[PSR*14] PICKUP D., SUN X., ROSIN P., MARTIN R., CHENG Z., LIAN Z., AONO M., BEN HAMZA A., BRONSTEIN A., BRONSTEIN M., BU S., CASTELLANI U., CHENG S., GARRO V., GIACHETTI A., GODIL A., HAN J., JOHAN H., LAI L., LI B., LI C., LI H., LITMAN R., LIU X., LIU Z., LU Y., TATSUMA A., YE J.: SHREC'14 track: Shape retrieval of non-rigid 3D human models. In *Proc. Eurographics Workshop on 3D Object Retrieval* (2014), pp. 1–10. 1, 4

[PSRM15] PICKUP D., SUN X., ROSIN P., MARTIN R.: Geometry and context for semantic correspondences and functionality recognition in manmade 3D shapes. *Pattern Recognition 48*, 8 (2015), 2500–2512. 4, 5

[QSN*16] QI C., SU H., NIESSNER M., DAI A., YAN M., GUIBAS L.: Volumetric and multi-view CNNs for object classification on 3D data. In *Proc. CVPR* (2016). 1

[Ros97] ROSENBERG S.: *The Laplacian on a Riemannian Manifold*. Cambridge University Press, 1997. 1

[Rus07] RUSTAMOV R.: Laplace-Beltrami eigenfunctions for deformation invariant shape representation. In *Proc. Symp. Geometry Processing* (2007), pp. 225–233. 1

[Sch15] SCHMIDHUBER J.: Deep learning in neural networks: An overview. *Neural Networks 61* (2015), 85–117. 1

[SMKF04] SHILANE P., MIN P., KAZHDAN M., FUNKHOUSER T.: The Princeton shape benchmark. In *Proc. SMI* (2004), pp. 167–178. 4

[SMKLM15] SU H., MAJI S., KALOGERAKIS E., LEARNED-MILLER E.: Multi-view convolutional neural networks for 3D shape recognition. In *Proc. ICCV* (2015), pp. 945–953. 1, 5, 7

[SOG09] SUN J., OVSJANIKOV M., GUIBAS L.: A concise and provably informative multi-scale signature based on heat diffusion. *Computer Graphics Forum 28*, 5 (2009), 1383–1392. 1

[SYS*16] SAVVA M., YU F., SU H., AONO M., CHEN B., COHEN-OR D., DENG W., SU H., BAI S., BAI X., N. FISH J. H., KALOGERAKIS E., LEARNED-MILLER E., LI Y., LIAO M., MAJI S., WANG Y., ZHANG N., ZHOU Z.: SHREC'16 track: Large-scale 3D shape retrieval from ShapeNet Core55. In *Proc. Eurographics Workshop on 3D Object Retrieval* (2016). 1, 5, 7

[SZB17] SEDAGHAT N., ZOLFAGHARI M., BROXN T.: Orientation-boosted voxel nets for 3d object recognition. 1

[vdMH08] VAN DER MAATEN L., HINTON G.: Visualizing data using t-SNE. *Journal of Machine Learning Research 9* (2008), 2579–2605. 6

[WSK*15] WU Z., SONG S., KHOSLA A., YU F., ZHANG L., TANG X., XIAO J.: 3D ShapeNets: A deep representation for volumetric shapes. In *Proc. CVPR* (2015), pp. 1912–1920. 1

[XDF17] XIE J., DAI G., FANG Y.: Deep multi-metric learning for shape-based 3D model retrieval. *IEEE Trans. Multimedia 19* (2017), 32463–2474. 1

[YY15] YE J., YU Y.: A fast modal space transform for robust nonrigid shape retrieval. *The Visual Computer 32*, 5 (2015), 553–568. 1, 4, 5

[ZLCE*15] Z. LIAN J. Z., CHOI S., ELNAGHY H., EL-SANA J., FURUYA T., GIACHETTI A., ISAIA R. G. L., LAI L., LI C., LI H., LIMBERGER F., MARTIN R., NAKANISHI R., NONATO A. N. L., OHBUCHI R., PEVZNER K., PICKUP D., ROSIN P., SHARF A., SUN L., SUN X., TARI S., UNAL G., WILSON R.: SHREC'15 track: Non-rigid 3D shape retrieval. In *Proc. Eurographics Workshop on 3D Object Retrieval* (2015), pp. 1–14. 1, 5