DOI: 10.1111/cgf.13272

Pacific Graphics 2017
J. Barbic, W.-C. Lin, and O. Sorkine-Hornung
(Guest Editors)

Volume 36 (2017), Number 7

Data-Driven Sparse Priors of 3D Shapes

O. Remil[1,2] Q. Xie[1] X. Xie[1] K. Xu[3] J. Wang[†][1,4]

[1]Nanjing University of Aeronautics and Astronautics, China
[2]Ecole Militaire Polytechnique, Algeria
[3]National University of Defense Technology, China
[4]Chuzhou University, China

Abstract

We present a sparse optimization framework for extracting sparse shape priors from a collection of 3D models. Shape priors are defined as point-set neighborhoods sampled from shape surfaces which convey important information encompassing normals and local shape characterization. A 3D shape model can be considered to be formed with a set of 3D local shape priors, while most of them are likely to have similar geometry. Our key observation is that the local priors extracted from a family of 3D shapes lie in a very low-dimensional manifold. Consequently, a compact and informative subset of priors can be learned to efficiently encode all shapes of the same family. A comprehensive library of local shape priors is first built with the given collection of 3D models of the same family. We then formulate a global, sparse optimization problem which enforces selecting representative priors while minimizing the reconstruction error. To solve the optimization problem, we design an efficient solver based on the Augmented Lagrangian Multipliers method (ALM). Extensive experiments exhibit the power of our data-driven sparse priors in elegantly solving several high-level shape analysis applications and geometry processing tasks, such as shape retrieval, style analysis and symmetry detection.

Categories and Subject Descriptors (according to ACM CCS): I.3.5 [Computer Graphics]: Computational Geometry and Object Modeling—Geometric algorithms, languages, and systems

1. Introduction

As the scale of available 3D shapes on internet becomes surprisingly large, a growing number of data-driven methods have been proposed in computer graphics community over the past decade. By leveraging the prosperity of big data, these techniques are introduced with the goal of extracting high-level shape information and meaningful mappings from 3D model databases. They devote themselves to automatically mining latent patterns in geometry and structure of shapes, instead of relying on hard-coded rules or explicitly programmed instructions. With these learned patterns serving as strong priors, many geometry processing applications can be solved more accurately and efficiently [XKHK15].

Image priors have become a popular tool in image processing as they have been applied to different applications such as image denoising [EA06, XZZ*15], structural image editing [BSFG09] and more. Image priors are defined as patches formed by pixels within a fixed square area. One famous approach [EA06] is the use of sparse and redundant image patch representations over trained dictionaries for image denoising. With the success achieved in 2D image processing applications, prior-based methods also accomplish an outstanding progress in 3D shape processing by taking advantage of the self-similarity concept. These methods are mainly exploited in 3D surface denoising [YBS06], 3D shape reconstruction [PMG*05,GSH*07] and point cloud compression [HMHB08, DCV14]. Similar to image priors, 3D shape priors are point-set neighborhoods sampled from a 3D model within a bounding sphere. Suppose that any 3D model can be considered to be formed with a set of local patches, which we refer to as local shape priors, it is obvious that most of them are likely to have similar geometry. Consequently, the corresponding sets of local shape priors would possess a significant number of redundant elements sharing similar geometric properties. Thus, it is reasonable to leverage few of them to represent a 3D model, or even represent a whole family of 3D models. Meanwhile, we observe that the local shape priors extracted from a family of 3D models lie in a very low-dimensional manifold, Therefore, a compact and informative subset of priors can be learned to efficiently encode all shapes of that family, leading to a number of promising applications in computer graphics.

In this paper, we concentrate on learning techniques to distill an efficient set of local shape priors from existing 3D model repositories. The goal is then converted to select a compact dictionary of priors from the library to reconstruct all database models. These priors, referred to as "*sparse priors*", are able to represent the sur-

[†] Corresponding Author: junwang@outlook.com

face of 3D models within the same family, and can be regarded as a highly effective and promising tool in many high-level geometry processing applications, see Section 5.3. It is worth pointing out that a similar work is proposed in [RXX*17].Their framework is mainly based on the Affinity Propagation (AP) method [FD07], which is a naïve way to learn the representative priors. They tend to optimize the reconstruction error without any theoretical guarantee. Furthermore, AP clustering suffers from a few drawbacks: 1) a small perturbation of similarities may influence the choice of one or few exemplars, leading to a different partitioning of the data; 2) priors selection is restricted by a hard constraint, in which each prior is forced to be its own self-exemplar if it is decided to be an exemplar by other priors, hence, the change of one exemplar may result in a large avalanche of other changes and 3) when dealing with irregular multi-dimensional data, AP clustering strongly relies on cluster-shape regularity and it may force the division of single clusters into separate ones. In contrast, our method minimizes the error in a direct manner, in which the optimization is divided into several sub-problems that have closed form solutions with theoretical guarantees. In addition, their method is not able to handle big datasets. Given the variance of shapes within a single class of object, learning from a small number of shapes (as in their case) is unlikely to cover the variety needed for the whole class of object.

The aim of this paper is to formulate a global, sparse optimization to alleviate the problems caused by the hard constraints in the affinity propagation algorithm. It enforces selecting representative priors from the library while minimizing the reconstruction error. As the first advantage, our algorithm is adaptive to the complexity of the data and does not rely on the cluster-shape regularity. In addition, we introduce a balance parameter that puts a trade-off between the number of sparse priors and the reconstruction error. As a result, our method does not require to pre-define the number of clusters in advance. Considering the non-convexity of the optimization problem, we design an efficient solver by first converting the problem into a convex problem and then decomposing it into several sub-problems that have closed-form solutions.

By manipulating the sparse priors, we develop a number of promising high-level applications. Following the Bag-of-Words framework, our sparse priors can be used to build a compact and informative shape descriptor that is able to deal with different shape categories. We demonstrate through examples the robustness of our prior-based descriptor to several transformations and strengths, and its effectiveness for shape retrieval and style analysis. Furthermore, the sparse priors can be utilized to detect symmetries within input point clouds. The key idea is to assign labels to the priors according to their nearest sparse prior, and use these labels to find symmetries or repetitive structures. To the best of our knowledge, this is the first attempt on learning 3D local shape priors from a collection of models via sparse optimization, and using them to handle various advanced geometry processing operations.

Overall, our main contributions can be summarized as follows:

- We formulate a global, sparse optimization problem that enforces selecting representative shape priors from a collection of 3D shapes while minimizing the reconstruction error.
- We design an effective solver so that the closed-form solution can be achieved on this optimization problem. We show that these learned priors construct a compact and informative shape representation of the whole family of object.
- Based on the learned sparse priors, we develop a number of promising high-level applications including shape retrieval, style analysis and symmetry detection, which exhibit the flexibility and adaptivity of our approach to different application scenarios.

2. Related Work

Data-Driven Shape Analysis and Processing. The key point of data-driven methods is to improve the analysis and processing of individual shapes by analyzing and aggregating information from a set of shapes. We refer the reader to the survey on data-driven shape analysis and processing techniques [XKHK15]. The applications of such techniques include a variety of methods such as shape retrieval [TV08,LBBC14], shape reconstruction [PMG*05,GSH*07] and matching [VKZHCO11].However, these works focus only on some specific geometry processing tasks, while our work can be applied to solve a variety of application scenarios.

Self-similarity. Self-similarity detection has been a predominant issue that has gained interest over the past decade. Previous works in image processing [BCM05] attempt to tackle this problem, by introducing the notion of non-local means (NLM). Their idea is to denoise a pixel by exploiting pixels of the whole image that may entail the same information. Barnes et al. [BSFG09] presented an interactive image editing tool by using a randomized algorithm to find similar image patches. More recently, Xu et al. [XZZ*15] proposed a patch group (PG) based algorithm to learn image priors for denoising. By taking advantage of self-similarity, some outstanding progresses have also been accomplished in 3D shape processing. It has mainly exploited in surface denoising [YBS06, Dig12, GAB12], reconstruction [GSH*07, SDK09, ZSW*10], surface registration [Son15] and compression [HMHB08, DCV14]. Schnabel et al. [SDK09] considered a hole filling approach where the reconstruction is guided by a set of primitive shapes such as planes and cylinders. While in [ZSW*10], the authors introduced a scan-consolidation framework using repeated geometry in urban buildings to enhance imperfect scans of urban models. [Dig12] and [GAB12] proposed to exploit the surface self-similarity for surface denoising and meshless geometry processing respectively. Hubo et al. [HMHB08] and Dign et al. [DCV14] both presented a compression technique that encodes the point clouds based on shape self-similarity. In our work, we investigate the same non-local idea, by using the self-similarity patches learned from 3D collection models in various geometry processing tasks.

Subset Selection and Clustering. Finding a compact and informative subset of a large collection of data points has been the center of many problems in computer vision and computer graphics communities. An increasing number of algorithms have been proposed to handle this difficult and ill-posed problem. One class of these methods finds representatives from data points that lie in one or multiple low-dimensional subspaces and typically operate on the measurement data vectors directly. The Rank Revealing QR (RRQR) algorithm [BMD09] tries to select a subset of columns of the data matrix that gives the best conditioned submatrix. This algorithm relies on low-rankness assumption, and it is not guaranteed to find the globally optimal solution. In addition, randomized and

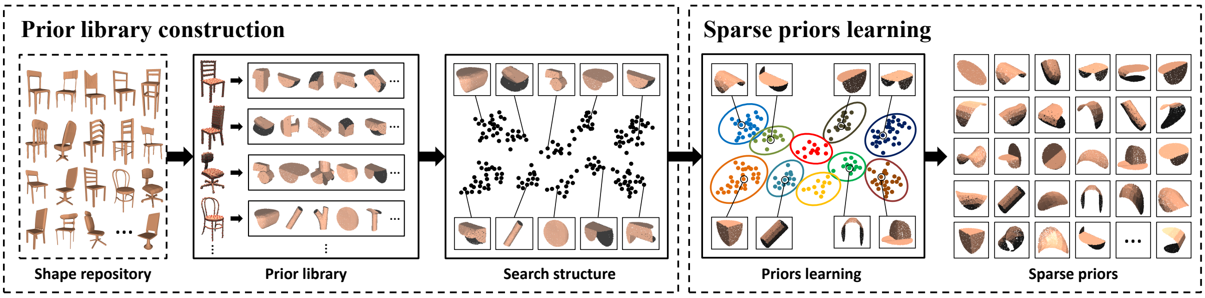

Figure 1: *Overview of our algorithm. Given a shape repository - chairs, we extract priors from each shape to construct prior library, which are inserted into a search structure based on their descriptors. An optimization model is then performed to obtain the most representative priors, referred to as* sparse priors. *Subsequently, these priors can be utilized for multiple high level applications in geometry processing.*

greedy algorithms [Tro09] for selecting few columns from a low-rank matrix have also been proposed. [ESV12] and [EMO*12] assumed that the data can be represented as a linear combination of the representatives, and formulated the problem of finding representatives as a joint-sparse recovery problem. The second class of algorithms finds representatives on the assumption that there is a natural grouping of the data collection. These algorithms typically operate on similarities/dissimilarities between data points. Kmedoids [KR87] algorithm, similar to Kmeans, tries to find K representatives from pairwise dissimilarities between data points. However, it depends on the initialization and their performances drop a lot as the number of representatives increases. The Affinity Propagation (AP) algorithm [FD07] takes as input measures of similarity between pairs of data points, and tries to find exemplars by passing messages between data points. However, a small perturbation of similarities may influence the choice of one or few exemplars. Other clustering methods such as hierarchical and spectral clustering assume that the data are distributed around cluster centers, and they can not handle big data because of the size of the similarity matrix. Compared to these subset selection and clustering methods, our method has several advantages. First, it is insensitive to the initialization and adaptive to the complexity of the data. Second, it does not assume that the data are distributed around some centers. Third, the number of representatives k is determined automatically.

3. Overview

Our algorithm, expecting a 3D model repository as input, aims at learning a set of 3D sparse priors. The pipeline of our data-driven sparse priors is outlined in Figure 1.

The first step is the construction of 3D prior library in Section 4.1. Given a collection of 3D models of the same family, we start by sampling point-set neighborhoods, which are regarded as local shape priors. Since the local priors are sampled from mesh models, their geometric information such as descriptors, normals and points belonging to sharp features can be easily computed. We extract such information and store them for later use. Therefore, we are able to obtain both the 3D priors and their corresponding shape descriptors, which comprise our initial prior library.

The second major phase is the sparse priors learning, where a small number of representative priors, namely sparse priors, are extracted in Section 4.2. The initial prior library usually contains a significant number of duplicated priors. Thus, we aim at extracting the most representative priors from the library. To this end, we consider a learning framework that enforces selecting representative priors from the library. More specifically, we seek the set of sparse priors by minimizing the reconstruction error defined as our objective function. Considering this non-convexity optimization problem, we design a relaxation strategy to convert it into a convex optimization problem, which is solved by the Augmented Lagrange Multipliers method. As a result, a small number of representative priors are obtained from the library.

Finally, we show the potential of our sparse priors in dealing with several geometry processing applications such as shape retrieval, shape style analysis and symmetry detection in Section 5.3.

4. Algorithm

In this section, we discuss the two successive steps of our algorithm. The first step selects a subset of points which will serve as center points, and then decomposes the input point clouds into 3D priors and computes a feature vector for each prior using the ensemble of shape function (ESF) descriptors. The second one exploits self-similarity among prior library to learn a set of representative sparse priors which is able to represent the input shape repository.

4.1. Prior Library Construction

Given a 3D shape repository, we first sample points on each shape and signify points belonging to sharp features. We then segment each shape into priors with a bounding sphere of radius R and compute their descriptors. Finally, we construct a library of such local priors. Algorithm 1 provides the implementation details for the prior library construction.

Prior creation. The creation of our prior library works under the following statements: 1) the database models are represented as triangle meshes and they belong to the same family of objects; 2) the sampled points for each shape are supposed to be dense enough so that meaningful priors can be created, which can be done by setting a sufficiently large radius R with respect to the point resolution; 3) the prior centers selected from the sampled point cloud are used to create the local priors within a radius R, in such a way the point

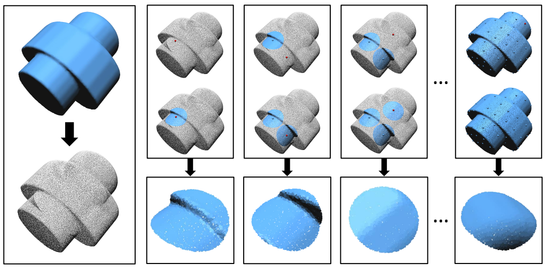

Figure 2: *Given a database model, a set of points are uniformly sampled. To create prior library, we first pick a point, find its neighbors within a predefined radius and label them as covered (in blue). We select a non covered point and continue in the same manner until all points are covered and all priors are created.*

cloud is totally covered. Note that we set the prior radius R as a relative value to each model in the database to make it scale-invariant.

Given a database model $\mathcal{M}$, we uniformly sample a set of points P on the model. To satisfy the third statement, we select a subset of points $S \in P$ as prior centers. The center selection is done in a dart-throwing fashion same as [DCV14]. We first pick a point from P as the first center, find its neighboring points Pr that lie within a sphere of radius R and label them as covered. We traverse the point cloud P until a non-covered point is found and a new center is added to S. The procedure repeats until all points of P are covered and all priors Pr are created from P, see Figure 2.

Feature vector. Before the learning stage, we require to measure how well the priors are similar to each other. In this section, we describe the descriptors used in our method and their computation. The simple global descriptor, Ensemble of Shape Functions (ESF) [WV11], is used as our shape descriptor. It is efficient and very unique because: 1) it is transformation and scale invariant; 2) it does not need any preprocessing such as normal estimation and 3) it can handle data imperfectness such as outliers, noise and incomplete surfaces. Notice that our algorithm is independent of the descriptor choice, and other shape descriptors can also be exploited. The ESF is a global shape descriptor and it consists of an ensemble of ten 64-bin-sized histograms, concatenated in a single 640 value histogram describing the properties of the prior. It combines three different shape functions-distance, angle and area [OFCD01]. See [WV11] for more details on this descriptor. Figure 2 shows some example priors with their corresponding ESF histograms.

Finally, the priors are inserted into an efficient data structure Kd-Tree, where the similarity between two priors is computed using the ℓ^1 distance between their corresponding feature vectors. By this means, we are able to obtain both the 3D priors and their corresponding descriptors, which comprise our initial prior library.

4.2. Sparse Priors Learning

The prior library consists of a large amount of priors extracted from the section above, which contains a significant number of duplicated priors. In this section, we propose an effective algorithm to extract a small number of representative priors from the library via convex optimization.

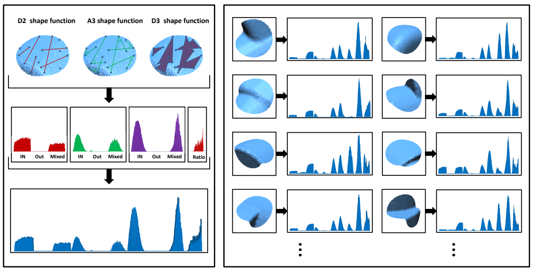

Figure 3: *The ESF descriptor calculated on a prior from Figure 2 with its ten 64-bin sub-histograms of shape functions (D2, A3 and D3) in left, and the ESF descriptors for some priors extracted from the same shape are illustrated in the right.*

Problem formulation. Given a library of priors $\mathcal{P}$, we are to extract a subset of $\mathcal{P}$, denoted by $\mathcal{P}^\star$, which only consists of representative priors. Under the framework of dictionary learning, the learned elements of the dictionary almost never coincide with the original data. Hence, they can not be considered as good representatives for the collection of data points. In order to find representative points that coincide with some of the actual data points, a learning framework is proposed to select representatives from the real data. More specifically, we learn a compact dictionary of priors that can efficiently represent the input database models (i.e. prior library). The geometric distances between priors are represented with the Euclidean distances between their corresponding descriptors.

Suppose there are N priors in the library, each of which has the ESF descriptor expressed by a 640-dimension vector. We arrange them in a "priors matrix" $Y \in \mathbb{R}^{640\times N}$ in a column-wise fashion $Y \triangleq [y_1, y_2 \dots, y_N] \in \mathbb{R}^{640\times N}$. The best set of representative priors is typically obtained by minimizing the reconstruction error defined as our objective function:

$$\sum_{i=1}^{N} \|y_i - Yc_i\|_2^2 = \|Y - YC\|_F^2 \tag{1}$$

with respect to the "selection matrix" $C \triangleq [c_1, c_2 \dots, c_N] \in \mathbb{R}^{N\times N}$, subject to the following constraints:

- To select $k \ll N$ representative priors from Y, we enforce:

  $$\|C\|_{0,q} \leq k. \tag{2}$$

  where the mixed norm $\|C\|_{0,q}$ counts the number of nonzero rows of C that correspond to the indices of the columns of the representative priors in Y.
- We tend to select exactly one representative for each prior, and not as a combination of the representative priors. Therefore, we enforce c_i to be the columns of the identity matrix:

  $$1^T C = 1^T, \|c_i\|_0 = 1. \tag{3}$$

  where $\|\cdot\|_0$ is the number of nonzero element of the column.

To sum up, the selection problem of k sparse priors can then be formulated as:

$$\begin{aligned} &\min_{C} \|Y - YC\|_F^2 \\ &\textbf{s.t.} \;\; \|C\|_{0,q} \leq k, \;\; 1^T C = 1^T, \;\; \|c_i\|_0 = 1. \end{aligned} \tag{4}$$

Algorithm 1 : Prior Library Construction

```
Input: Shape repository $\mathcal{S} = \{\mathcal{M}_i\}$, $i = \{1,\dots,m\}$, number of sampled points $N$, radius $R$.
Output: Prior library $\mathcal{L}$.
  function GENERATE PRIORS($\mathcal{S}, N, R$)
    for $i = 1$ to $m$ do
      $P$ = Sample points ($\mathcal{M}_i, N$);
      $d$ = Get bounding box diagonal;
      $R = R * d$;                                  ▷ relative radius value.
      kdtree ← $P$;                                  ▷ kdtree research structure.
      for $j = 1$ to $N$ do
        if ($P_j$.covered == true) then              ▷ covered point.
          continue;
        else                                         ▷ non covered point.
          $P_j$.covered = true;                      ▷ marked as covered.
          $Pr$ ← knn ($P_j$, kdtree, $R$);
          for all $p \in Pr$ do
            $p$.covered = true;
          end for
          $V(Pr)$= Compute Descriptor ($Pr$);
          $\mathcal{L} \leftarrow \{Pr, V(Pr)\}$;
        end if
      end for
    end for
    return ($\mathcal{L}$);
  end function
```

This is an NP-hard problem as it requires searching over every subset of the k columns of Y. In addition, the constraint $\|\cdot\|_0$ is hard to be optimized. Hence, the problem needs to be relaxed to:

$$\begin{aligned} &\min_C \|Y - YC\|_F^2 \\ &\textbf{s.t.}\ \ \|C\|_{1,q} \leq k,\ \ 1^T C = 1^T,\ \ C \geq 0. \end{aligned} \tag{5}$$

where $\|C\|_{1,q} \triangleq \sum_{i=1}^N \|c^i\|_q$ is the sum of the l_q norms of the rows in C. We let $q > 1$, making the optimization problem in (5) convex.

Optimization Procedure. The optimization problem in (5) is convex and can be solved by various methods. For efficiency, we adopt the Augmented Lagrange Multipliers (ALM) [LCWM10] method. We first convert the problem in (5) to the following equivalent problem:

$$\begin{aligned} &\min_C \|Y - YD\|_F^2 \\ &\textbf{s.t.}\ \ \|C\|_{1,q} \leq k,\ \ 1^T D = 1^T,\ \ J \geq 0, D = C,\ \ D = J. \end{aligned} \tag{6}$$

We then solve it by minimizing the following augmented Lagrangian function:

$$\begin{aligned} \mathcal{L} =& \|C\|_{1,q} + \lambda \|Y - YD\|_F^2 \\ &+ \langle Z_1, D - C\rangle + \langle Z_2, D - J\rangle + \delta^T(D^T 1 - 1) \\ &+ \frac{\mu}{2}(\|D - C\|_F^2 + \|D - J\|_F^2 + \|D^T 1 - 1\|_2^2) \\ &\textbf{s.t.}\ \ J \geq 0. \end{aligned} \tag{7}$$

where $\langle \cdot, \cdot \rangle$ is the matrix inner product, $\mu > 0$ is a penalty parameter and λ is a balance parameter. Our optimization does not need the number of sparse priors to be specified, we just need to set λ and let the algorithm determine the number of priors k automatically.

Algorithm 2 : Sparse Priors Learning

```
Input: Priors matrix $Y$.
Output: Sparse priors matrix $C$.
  Initialization:
  $D = C = J = 0, \delta = 0, Z_1 = Z_2 = 0, \mu = 10^{-6}$, $\mu_{max} = 10^6, \rho = 1.1$, and $\varepsilon = 10^{-8}$.
  while not converged do
    update matrices $D, C, J$:
    $D \leftarrow$ Equation (8);
    $C \leftarrow$ Equation (9);
    $J \leftarrow$ Equation (10);
    update $\delta$ and lagrange multipliers $Z_1, Z_2$:
    $Z_1, Z_2, \delta \leftarrow$ Equation (11);
    update parameter $\mu$:
    $\mu \leftarrow min(\rho\mu, \mu_{max})$
    check convergence conditions:
    $\|D^T 1 - 1\|_\infty < \varepsilon$;   $\|D - J\|_\infty < \varepsilon$;   $\|D - C\|_\infty < \varepsilon$.
  end while
```

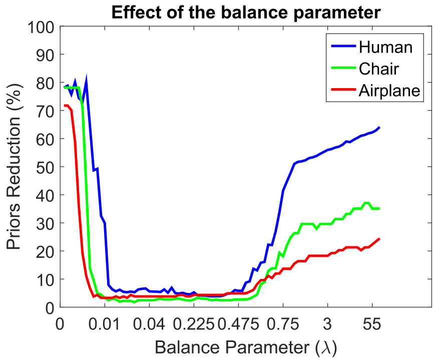

Figure 4: *Effect of the balance parameter on the number of sparse priors for the human, Chair and Airplane shape databases.*

The above problem can be minimized with respect to C, D, and J respectively, by fixing the other variables and updating the Lagrange multipliers Z_1 and Z_2. The pseudo-code for the optimization procedure is outlined in Algorithm 2.

More specifically, we initialize the parameters of our algorithm, and iterate the following steps:

Update D: We then minimize the lagrangian $\mathcal{L}$ with respect to D by setting the first derivative of $\mathcal{L}$ with respect to D equal to zero. As a result, D is obtained from:

$$\begin{aligned} &(\lambda Y^T Y + \mu I + \mu 1 1^T) D = \\ &(\lambda Y^T Y + \mu C + \mu J + \mu 1 1^T - Z_1 - Z_2 + 1\delta^T). \end{aligned} \tag{8}$$

Update C: We fix the values of D, J, δ, Z_1 and Z_2, and compute C by minimizing $\mathcal{L}$ with respect to C. As a result, we obtain:

$$\begin{aligned} C = \arg\min_C &\|C\|_{1,q} + \langle Z_1, D - C\rangle \\ &+ \langle Z_2, C - J\rangle + \frac{\mu}{2}\|D - C\|_F^2. \end{aligned} \tag{9}$$

Update J: With fixed D, C, δ, Z_1 and Z_2, J is obtained by mini-

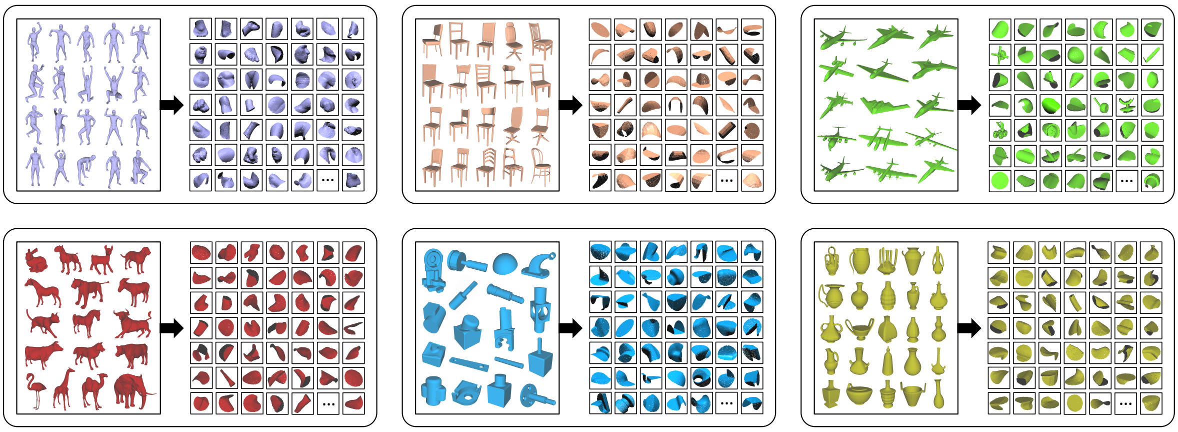

Figure 5: *Six categories of sparse priors learned by our algorithm. For each database, a set of sparse priors is obtained from a collection of 3D models. Considering the size of figure and explicitness of priors, we only choose to show* 48 *sparse priors among the learned priors.*

mizing $\mathcal{L}$ with respect to J:

$$J = D + Z_2/\mu. \; and \; J = 0 \text{ if } (J_{ij} < 0). \tag{10}$$

Update δ and lagrange multipliers Z_1 and Z_2 : We fix D, C and J, and perform gradient ascent with step size of μ to update Z_1, Z_2 and δ:

$$\begin{aligned} \delta &= \delta + \mu(D-1). \\ Z_1 &= Z_1 + \mu(D-C). \\ Z_2 &= Z_2 + \mu(D-J). \end{aligned} \tag{11}$$

Convergence. The convergence of our optimization is monitored by imposing terminal conditions. During each iteration, we check whether the changes in these conditions are below a threshold.

Effect of the balance parameter. Figure 4 shows the percentage of selected sparse priors from the library as a function of the balance parameter λ. The range of values in the interval $[0.01, 0.5]$ produces less than 6% reduction of the prior library size, and provides a good time/quality compromise.

5. Results and Applications

In this section, we present the performance and versatility of our learning method. We show experimental results on a variety of shape databases. The effectiveness of our learning method is demonstrated through comparisons with several related clustering methods. To further assess our algorithm, we show some potential applications using our data-driven sparse priors.

5.1. Diverse Datasets

To evaluate our method, we test it over datasets across six shape categories: humans (71), chairs (531), airplanes (284), animals (62), mechanical parts (93) and vases (328) from Watertight Track of the SHREC 2007, Princeton Shape Benchmark [SMKF04], SCAPE [ASK*05], Geotopo [ALX*14], Co-segmentation [WAvK*12] and the benchmarks provided by [KLM*13] and [SP04]. For each one of these 6 categories, we run our algorithm to generate the prior library and learn the sparse priors. Figure 5 illustrates some of the sparse priors. Running time statistics and parameters for each dataset are shown in Table 1.

5.2. Comparison to Clustering methods

We compare our method to several related methods:

- **K-means++**: K-means algorithm divides the input data into several clusters with equal variance by minimizing the inertia where each cluster is described by the mean of its points. By using a technique to seed initial centers, k-means++ algorithm assigns the input points to their nearest center. It creates the new centers by taking the mean value of all points in each cluster and repeats until the value is smaller than a threshold.
- **Hierarchical clustering**: it seeks a hierarchy of clusters from the input data by progressively merging clusters based on their distance (points are likely to be connected to nearby points than to farther ones).
- **Spectral clustering**: a similarity matrix is built based on the relative similarity of each point pair in the input data and the Laplacian matrix is computed on it. The algorithm then maps the input points based on the eigenvectors of the Laplacian matrix.

All these methods require to pre-define the number of clusters. However, in our method, the number of sparse priors k can be monitored by the parameter λ. To evaluate the accuracy of the learned sparse priors, we define an error measure as:

$$Err_k = \sum_{i=1}^{N} d(V(Pr) - V(Pr^{\star})). \tag{12}$$

where N is the number of priors presented in the library, k is the number of sparse priors, $V(*)$ is the descriptor of prior $*$, $Pr^{\star}$ is the nearest sparse prior to the prior Pr and $d(V(Pr) - V(Pr^{\star}))$ is the Euclidean distance between the descriptors of the two priors Pr and $Pr^{\star}$. Note that the centers of the clusters given by k-means++ algorithm may not belong to the original input descriptors as required for comparison with other methods. To keep comparison fair, we

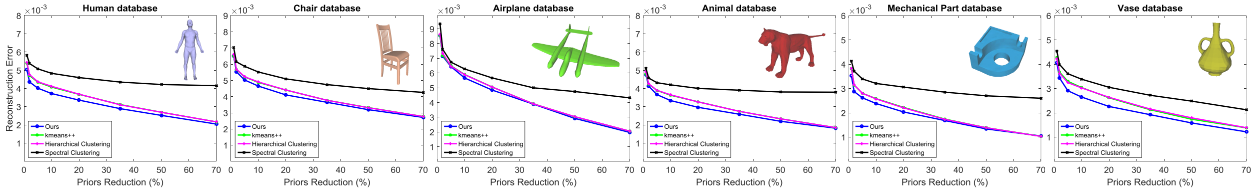

Figure 6: *Comparison to clustering methods. Performances of various method on six model databases show that our algorithm outperforms other clustering approaches. According to the plots, our algorithm (blue) can achieve a smaller reconstruction error with the same number of sparse priors than the other three methods.*

compute the nearest descriptor from the original input data to each cluster center given by k-means++.

In Figure 6, we generate plots to examine the distributions of the errors for all the methods, where the x-axis represents the reduction percentage of the prior library, and the y-axis shows the distributions of errors. For each shape category, our method learns k sparse priors where $k \in \{1\%, 2\%, 5\%, 10\%, 20\%, 30\%, 50\%, 70\%\}$ from the prior library size and for each k we compute the error Err_k. The plots reveal that our method outperforms other clustering methods and yields smaller reconstruction errors for all databases, indicating that our method learns the sparse priors more effectively.

5.3. Applications

The sparse priors learned by our method are a powerful new tool to solve a variety of geometry processing tasks. To name a few, in this section, we demonstrate how they can be exploited for shape retrieval, shape style analysis and symmetry detection.

Prior-based Descriptor. Inspired by the Bag-of-Words (BoW) framework, we apply our sparse priors to construct a compact and informative shape descriptor, which is able to deal with non-rigid shape deformations to some extent. Given a 3D model, we learn the set of k sparse priors $\mathcal{P}^\star$ as explained in sections 4.1 and 4.2, where each sparse prior is associated with its ESF descriptor. Given a new 3D model, we sample points and generate its priors using algorithm 1. For each prior, we apply approximate nearest neighbors (ANN) to find the corresponding nearest sparse prior from $\mathcal{P}^\star$. Indeed, the prior-based representation is obtained by counting the number of the priors matched to each sparse prior in $\mathcal{P}^\star$, where the resulting signature is a histogram of occurrences. Note that the dimension of the prior-based descriptor is equal to the number of the sparse priors k. Finally, we take advantage of these histograms as our prior-based descriptors. Figure 7 illustrates the steps of our descriptor extraction process.

Shape retrieval. Nowadays, the fast growth of 3D shape repositories makes the non-rigid shape retrieval one of the big challenges in search and classification. In this part, we bring the spirit of our prior-based descriptor to the problem of non-rigid shape retrieval. Each 3D model has to be associated with its prior-based descriptor. Given a query shape, the goal is to retrieve similar shapes from a collection of 3D models, where the similarity between two shapes is quantitatively measured by the Euclidean distance based on their respective descriptors.

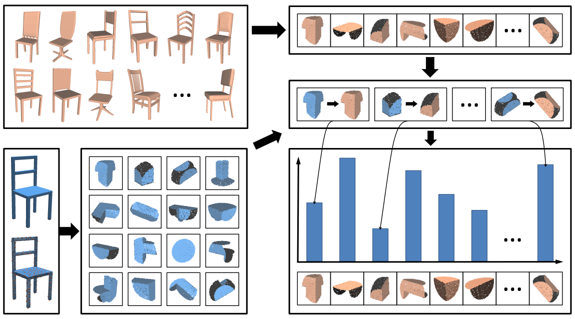

Figure 7: *Prior based descriptor. Given a chair model, we learn the set of sparse priors (Orange). Given a new chair model, we generate its priors (blue), and find their nearest sparse priors. By accumulating the number of matched priors to each sparse prior, a histogram as a compact description for the chair model is obtained.*

To evaluate the robustness of our shape descriptor to different classes of transformations, we conduct a set of experiments on the SHREC 2010 robust large-scale shape retrieval benchmark as it is the only dataset that includes multiple modifications and transformations with different strength of each shape [BBC*10]. It consists of 1184 shapes, out of which 456 unrelated shapes are used as negatives, whereas 715 transformed shapes obtained from 13 shape classes are used as queries. The queries are obtained from the 13 shapes by applying 11 classes of transformations with five different strengths (55 per shape). Notice that each query shape has only one corresponding shape in the dataset. The transformations consist of isometry, topology, sampling, local and global scale, big holes and micro holes, noise, shot noise, partial occlusion, and mixed transformations. For each query shape, we run our algorithm to obtain the first r top-ranked retrieved shapes from the null shapes (456 negatives and 13 positives). Some of these retrieved shapes are illustrated in Figure 8. The retrieval quality is quantitatively evaluated using the mean Average Precision (mAP). Table 2 presents the retrieval performance of our method. We refer the reader to [BBC*10] for the performance of other retrieval methods. Comparisons to tables 1 to 10 from [BBC*10] demonstrate the ability of our method to achieve accurate results. As shown, our method can efficiently deal with isometry, topology, global scale, micro and big holes classes and it shows a performance of 83.74% mAP on the full query set.

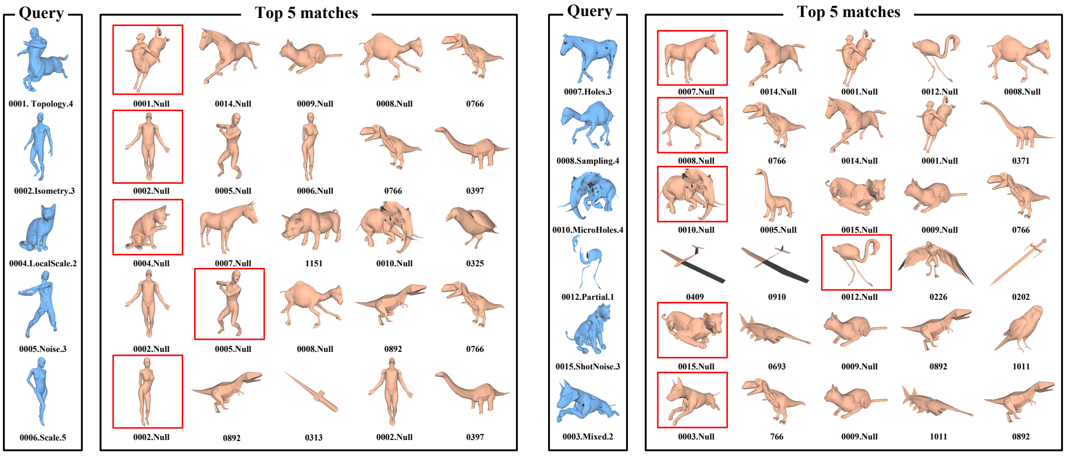

Figure 8: *Shape retrieval results on the SHREC 2010 robust large-scale shape retrieval benchmark. Our algorithm is able to retrieve the relevant shapes (red boxes) for different transformations and strengths. The results demonstrate the effectiveness of our prior-based descriptors in providing a compact and informative description of 3D shapes.*

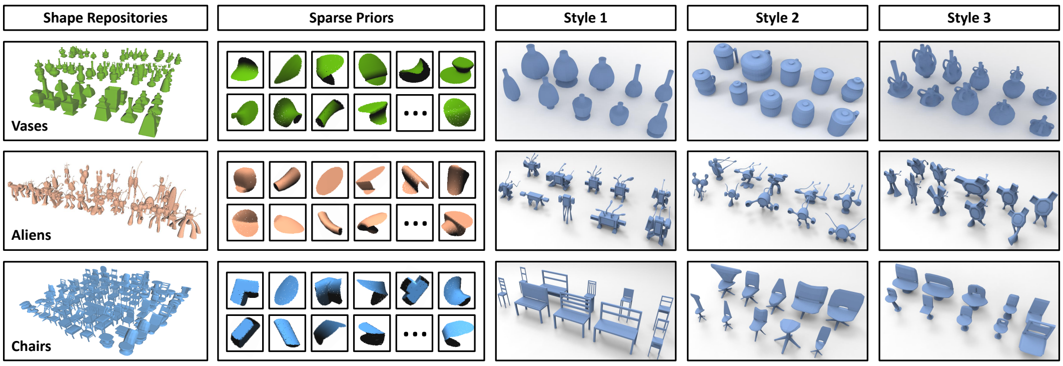

Figure 9: *Shape style analysis using our learned sparse priors. Given three model repositories (vases, tele-aliens and chairs), our algorithm automatically learns their sparse prior sets (first column). We then utilize the corresponding spare priors to generate prior-based descriptors for each shape. Finally, a clustering method is employed to classify the models into several clusters with distinct styles.*

Shape style analysis. In recent years, a growing number of online 3D shape repositories have caused an increasing demand to effectively organize and explore these large 3D shape collections. Man-made objects, like chairs, usually convey the design intent, and style is often a significant component of the intent. Distinct visual styles are omnipresent in art and design. The main challenge in this context is to distinguish the style from the structure, which can be regarded as a high-level ill-posed problem. While understanding style is crucial to shape understanding, very little research in computer graphics has explored shape styles.

To investigate the problem of style analysis, we rely on our prior-based description to truly capture how the styles differ from the input shapes. Here, we are faced with an input set already belonging to the same family, and our goal is to further refine the classification. For instance, we first extract descriptors for a given set of 3D models within the same family, and then group the stylistically similar shapes using the spectral clustering technique to form what we call style clusters. Figure 9 shows our style clustering results on several classes of 3D models, where shape classes are arranged into rows and styles into columns. As illustrated, our prior-based descriptor is a key indicator of stylistic similarity between 3D shapes.

Symmetry detection. It aims at discovering redundancy in the form of reoccurring structures in geometric objects. For the graph-based symmetry detection [BBW*08], one may construct the graph based on extracted feature points and then detect symmetric geometry by graph matching. Such graph construction can be readily achieved by our sparse priors.

In figure 10, given an input raw scan, we first generate the priors P and learn the sparse priors $\mathcal{P}^{\star}$ (presented in different colors). We then search the n nearest priors for each sparse prior from the input scan (here n=5), and we label their center points C_i with the same color of each sparse prior. From these centers, we build a k-NN graph $G(C,E)$ that describes the similarity structure of the object, with the labeled center points C_i as nodes and the line segments

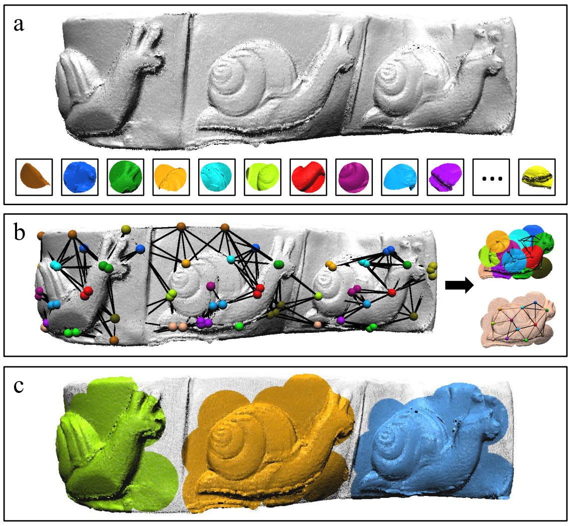

Figure 10: *Symmetry detection. (a) Given an input point data, we learn the set of sparse priors (in different colors). (b) For each sparse prior, we find n nearest local priors and label them with the same colors. A k-NN graph (in black) is then constructed from the local prior centers. Finally, a subgraph matching technique is employed to detect the reoccurring pattern. (c) Repetitive structures.*

$E(i,j)$ between two centers C_i and C_j as edges (we use k=10). Given the graph G, we employ a randomized subgraph search algorithm [BBW*08] to identify the reoccurring patterns S_i in this graph, where the label information is incorporated to refine edge matching. There are two remarkable advantages of our prior-based graph construction. On one hand, the graph nodes have the label information corresponding to the matched sparse priors, which significantly accelerates the graph matching process. On the other hand, our prior matching strategy is performed on point-set neighborhoods, which makes the extraction of the graph nodes more robust to noise than those based on feature point extraction.

5.4. Performance and Parameters

We have implemented our algorithm in C++ and all experiments are performed on a PC with a 2.4GHz CPU and 6.0 GB of RAM. Table 1 lists the running time performance and statistics for the learned sparse priors in Figure 5. The timing of our algorithm includes prior library generation and sparse priors learning. As can be seen in the table, the prior generation stage is computationally the most involved of our pipeline. It actually depends on the number of shapes presented in the database.

Our method has several parameters: 1) the prior radius R and the number N of sampled points per shape; 2) the penalty and the balance parameters. The radius R is chosen relatively to the value of the diagonal of the bounding box of each model. Through all experiments, values between 0.05 and 0.1 produce pleasing results (see Table 1). The number N of sampled points does not affect the quality of the results ($N = 500k$ points for all experiments).

Table 1: *Running time statistics and parameters for different shape datasets. N: number of 3D models, R: relative radius value, Lib: size of prior library, k: number of sparse priors, PG-time: priors generation time and L-time: learning time.*

Datasets	N	R	Lib	k	PG-time	L-time
Humans	71	0.1	10153	391	6m51s	5m41s
Chairs	531	0.05	87615	2158	32m48s	25m25s
Airplanes	284	0.05	36636	1212	21m10s	14m51s
Animals	62	0.1	12152	652	4m27s	7m16s
Mech parts	93	0.05	26412	839	8m58s	11m43s
Vases	328	0.05	85608	1423	25m59s	25m37s

Table 2: *Retrieval performances of our prior-based descriptors to different classes of transformations (mAP in %).*

	Strength				
Transform.	1	≤ 2	≤ 3	≤ 4	≤ 5
Isometry	100.00	100.00	100.00	100.00	100.00
Topology	100.00	100.00	100.00	100.00	100.00
Holes	100.00	100.00	100.00	100.00	100.00
Micro holes	100.00	100.00	100.00	100.00	100.00
Scale	100.00	100.00	100.00	100.00	100.00
Local scale	100.00	100.00	97.69	95.35	94.87
Sampling	100.00	96.15	93.55	85.66	81.76
Noise	100.00	93.27	87.96	74.72	61.46
Shot noise	100.00	95.49	93.95	92.05	92.56
Partial	13.49	12.47	12.31	11.97	12.85
Mixed	85.06	83.28	84.73	79.69	77.61
Average	90.77	89.15	88.19	85.40	83.74

5.5. Limitations

Our method is expected to behave well with different shape categories. However, there are still a few limitations that have to be discussed. Our framework is sufficiently general to work with different kind of shape categories; however, it may fail with repositories with a small number of redundant elements. It remains to be seen whether it works effectively with more challenging shape categories such as complex organic shapes.

6. Conclusions and Future Work

In this paper, we propose a sparse optimization framework for learning data-driven sparse priors from a collection of 3D shapes. A variety of experimental results on synthetic and real data demonstrate the effectiveness and validity of our method. Our sparse priors can also be regarded as a highly effective and promising tool to elegantly solve high-level shape analysis applications and geometry processing tasks. We believe that our method is the first attempt on learning 3D local shape priors from a collection of 3D models via sparse optimization, which opens the doors for many high-level shape analysis and understanding applications.

An interesting future research direction is to use the global information from the input database by using prior graphs that have high-level information within the shape structures (e.g., symmetries and repetitions) and then integrate the local information of the sparse priors with them.

Acknowledgements

We thank the anonymous reviewers for their valuable comments and suggestions. The work was supported in part by National Natural Science Foundation of China (61402224), the Fundamental Research Funds for the Central Universities (NE2014402, NE2016004), the Natural Science Foundation of Jiangsu Province (BK2014833), the NUAA Fundamental Research Funds (NS2015053), and Jiangsu Specially-Appointed Professorship.

References

[ALX*14] Alhashim I., Li H., Xu K., Cao J., Ma R., Zhang H.: Topology-varying 3d shape creation via structural blending. *ACM Transactions on Graphics* (2014). 6

[ASK*05] Anguelov D., Srinivasan P., Koller D., Thrun S., Rodgers J., Davis J.: Scape: shape completion and animation of people. *ACM Transactions on Graphics* (2005). 6

[BBC*10] Bronstein A. M., Bronstein M. M., Castellani U., Falcidieno B., Fusiello A., Godil A., Guibas L. J., Kokkinos I., Lian Z., Ovsjanikov M., Patané G., Spagnuolo M., Toldo R., imati Genova C.: Shrec 2010: robust large-scale shape retrieval benchmark. In *Proceedings of the 3D Object Retrieval Conference* (2010). 7

[BBW*08] Berner A., Bokeloh M., Wand M., Schilling A., Seidel H.-P.: A graph-based approach to symmetry detection. In *IEEE/ EG Symposium on Volume and Point-Based Graphics* (2008). 8, 9

[BCM05] Buades A., Coll B., Morel J.-M.: A non-local algorithm for image denoising. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (2005). 2

[BMD09] Boutsidis C., Mahoney M. W., Drineas P.: An improved approximation algorithm for the column subset selection problem. In *Proceedings of the Twentieth Annual ACM-SIAM Symposium on Discrete Algorithms* (2009). 2

[BSFG09] Barnes C., Shechtman E., Finkelstein A., Goldman D. B.: Patchmatch: A randomized correspondence algorithm for structural image editing. *ACM Transactions on Graphics* (2009). 1, 2

[DCV14] Digne J., Chaine R., Valette S.: Self-similarity for accurate compression of point sampled surfaces. *Computer Graphics Forum* (2014). 1, 2, 4

[Dig12] Digne J.: Similarity based filtering of point clouds. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (2012). 2

[EA06] Elad M., Aharon M.: Image denoising via sparse and redundant representations over learned dictionaries. *IEEE Transactions on Image Processing* (2006). 1

[EMO*12] Esser E., Moller M., Osher S., Sapiro G., Xin J.: A convex model for nonnegative matrix factorization and dimensionality reduction on physical space. *IEEE Transactions on Image Processing* (2012). 3

[ESV12] Elhamifar E., Sapiro G., Vidal R.: See all by looking at a few: Sparse modeling for finding representative objects. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (2012). 3

[FD07] Frey B. J. J., Dueck D.: Clustering by passing messages between data points. *Science* (2007). 2, 3

[GAB12] Guillemot T., Almansa A., Boubekeur T.: Non local point set surfaces. In *Proceedings of the International Conference on 3D Imaging, Moleding, Processing, Visualization and Transmission* (2012). 2

[GSH*07] Gal R., Shamir A., Hassner T., Pauly M., Cohen-Or D.: Surface reconstruction using local shape priors. In *Symposium on Geometry Processing* (2007). 1, 2

[HMHB08] Hubo E., Mertens T., Haber T., Bekaert P.: Self-similarity based compression of point set surfaces with application to ray tracing. *Computers & Graphics* (2008). 1, 2

[KLM*13] Kim V. G., Li W., Mitra N. J., Chaudhuri S., DiVerdi S., Funkhouser T.: Learning part-based templates from large collections of 3d shapes. *ACM Transactions on Graphics* (2013). 6

[KR87] Kaufman L., Rousseeuw P.: *Clustering by Means of Medoids*. Reports of the Faculty of Mathematics and Informatics. 1987. 3

[LBBC14] Litman R., Bronstein A., Bronstein M., Castellani U.: Supervised learning of bag-of-features shape descriptors using sparse coding. *Computer Graphics Forum* (2014). 2

[LCWM10] Lin Z., Chen M., Wu L., Ma Y.: The augmented lagrange multiplier method for exact recovery of corrupted low-rank matrices. *arXiv preprint* (2010). 5

[OFCD01] Osada R., Funkhouser T. A., Chazelle B., Dobkin D. P.: Matching 3d models with shape distributions. In *Shape Modeling International* (2001). 4

[PMG*05] Pauly M., Mitra N. J., Giesen J., Gross M. H., Guibas L. J.: Example-based 3d scan completion. In *Symposium on Geometry Processing* (2005). 1, 2

[RXX*17] Remil O., Xie Q., Xie X., Xu k., Wang j.: Surface reconstruction with data-driven exemplar priors. *Computer-Aided Design* (2017). 2

[SDK09] Schnabel R., Degener P., Klein R.: Completion and reconstruction with primitive shapes. *Computer Graphics Forum* (2009). 2

[SMKF04] Shilane P., Min P., Kazhdan M., Funkhouser T.: The princeton shape benchmark. In *Proceedings of the Shape Modeling International* (2004). 6

[Son15] Song P.: Local voxelizer: A shape descriptor for surface registration. *Computational Visual Media 1* (2015), 279–289. 2

[SP04] Sumner R. W., Popović J.: Deformation transfer for triangle meshes. In *ACM Transactions on Graphics* (2004). 6

[Tro09] Tropp J. A.: Column subset selection, matrix factorization, and eigenvalue optimization. In *Proceedings of the Twentieth Annual ACM-SIAM Symposium on Discrete Algorithms* (2009). 3

[TV08] Tangelder J. W., Veltkamp R. C.: A survey of content based 3d shape retrieval methods. *Multimedia tools and applications* (2008). 2

[VKZHCO11] Van Kaick O., Zhang H., Hamarneh G., Cohen-Or D.: A survey on shape correspondence. In *Computer Graphics Forum* (2011). 2

[WAvK*12] Wang Y., Asafi S., van Kaick O., Zhang H., Cohen-Or D., Chen B.: Active co-analysis of a set of shapes. *ACM Transactions on Graphics* (2012). 6

[WV11] Wohlkinger W., Vincze M.: Ensemble of shape functions for 3d object classification. In *IEEE International Conference on Robotics and Biomimetics* (2011). 4

[XKHK15] Xu K., Kim V. G., Huang Q., Kalogerakis E.: Data-driven shape analysis and processing. *Computer Graphics Forum* (2015). 1, 2

[XZZ*15] Xu J., Zhang L., Zuo W., Zhang D., Feng X.: Patch group based nonlocal self-similarity prior learning for image denoising. In *IEEE International Conference on Computer Vision* (2015). 1, 2

[YBS06] Yoshizawa S., Belyaev A., Seidel H.-P.: Smoothing by example: Mesh denoising by averaging with similarity-based weights. In *IEEE International Conference on Shape Modeling and Applications* (2006). 1, 2

[ZSW*10] Zheng Q., Sharf A., Wan G., Li Y., Mitra N. J., Cohen-Or D., Chen B.: Non-local scan consolidation for 3d urban scenes. *ACM Transactions on Graphics* (2010). 2