Geometry Representations with Unsupervised Feature Learning

Yeo-Jin Yoon[1], Alexander Lelidis[2], A. Cengiz Öztireli[3],
Jung-Min Hwang[1], Markus Gross[3] and Soo-Mi Choi[1*]
[1]Department of Computer Science and Engineering
Sejong University, Seoul, Republic of Korea
Email: {yjyoon, gwa89}@sju.ac.kr, smchoi@sejong.ac.kr
[2]TU Berlin, Berlin, Germany
Email: alexlelidis@gmx.de
[3]Computer Graphics Laboratory, ETH Zürich
Zürich, Switzerland
Email: {cengizo, grossm}@inf.ethz.ch

***Abstract*—Geometry data in massive amounts can be generated thanks to the modern capture devices and mature geometry modeling tools. It is essential to develop the tools to analyze and utilize this big data. In this paper, we present an exploration of analyzing geometries via learning local geometry features. After extracting local geometry patches, we parameterize each patch geometry by a radial basis function based interpolation. We use the resulting coefficients as discrete representations of the patches. These are then fed into feature learning algorithms to extract the dominant components explaining the overall patch database. This simple approach allows us to handle general representations such as point clouds or meshes with noise, outliers, and missing data. We present features learned on several patch databases to illustrate the utility of such an analysis for geometry processing applications.**

***Index Terms*—Geometry representations, dictionary learning, big geometry data.**

I. Introduction

As sensor technology such as depth cameras develops, acquiring 3D data from the real world is becoming easy. Most 3D geometry data from the real world is not easy to process because of its massive size and unstructured nature [1]. There exist several methods to handle manifold surfaces, and a few others that address more complex non-manifold geometries [2–4]. These methods typically rely on only the geometry data at hand without considering existing big 3D geometry data. Utilizing the existing data can lead to significant gains in accuracy, especially in the case of missing data, noise, and outliers, which are typical for real data.

In order to utilize such 3D data in a unified framework, we propose to represent local patches from the geometry data with unsupervised feature learning. We divide existing 3D models into local patches, convert the patches into a unified representation, and apply clustering on the resulting feature vectors to learn a dictionary of local geometry patches (Fig. 1). We also illustrate a preliminiary application of our framework to shape reconstruction.

* Corresponding author.

II. Related Work

In order to reconstruct accurate geometry representations, it is essential to handle noise, outliers, and missing parts of 3D geometries. Existing methods for solving this problem can be divided into two groups: local and global methods. Local methods divide a 3D model into local geometric patches and analyze each patch's dominant components [5,6]. These methods have the advantages of fast processing speed and robustness against noise and outliers, but handling large missing parts is difficult. On the other hand, global methods [7] can handle large holes since they consider global shape characteristics, but the computational complexity increases significantly, especially for big geometry data.

To properly handle big geometry data composed of a massive amounts of unstructured point data, we explore geometry representations using a local approach. We use geometry patches as local geometric features, and employ a dictionary learning scheme based on sparse coding to extract the dominant components describing the overall patches from unsupervised data.

Sparse signal representations are used in various areas such as image processing, machine learning, neuroscience, and statistics. Recently, signal processing and machine learning techniques have been applied for analyzing and reconstructing 3D geometries [8]. In 3D geometry processing, sparse coding is used for mesh denoising, 3D surface reconstruction, mesh segmentation, among others [9–12]. We also utilize sparse learning methods. In constrast to previous methods, we utilize sparse coding to generate compact representations of big geometry data.

In particular, we utilize dictionary learning to get a few representative features that summerize local patches from large scale geometry data. Dictionary learning is a common technique in image compression, super-resolution and denoising. It has also generated state-of-the-art results

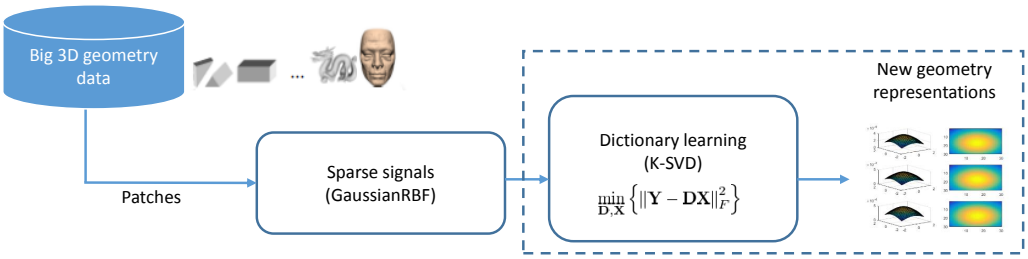


Fig. 1. Overall pipeline of geometry representations using unsupervised feature learning.

in geometry processing in recent years. For deformation and animation, dictionary learning is used for pose decomposition [13], analyzing mesh sequences [14], or highly deformable models [15]. In 3D reconstruction, sparse dictionaries are used for storing local geometry features and reconstructing triangle meshes [7]. Another work [16] exploits self-similarity using K-SVD [17] for geometry compression on point sampled surfaces. This method stores local geometry patches to construct a geometry dictionary, which are then used for high quality reconstructions. We also utilize a similar approach, but propose a new unified way of encoding local patches that is applicable to handle massive amounts of existing data.

III. Method

Given a set of high quality 3D geometries, we first extract local patches. Then, we represent each of these patches with a feature vector of coefficients for radial basis function based interpolation. Finally, we run K-SVD [17] to cluster the feature vectors and learn a representative set of features. This compact set can then be used for various processing tasks.

A. Feature Vectors for Local Patches

Raw geometry data can be some with different number of geometry entities such as points or triangles. To make consistent patches for different models, a random set of seed points is selected to divide the model into local patches with radius r, similar to existing works [16]. Then the coordinates of points within each patch are stored as a local height field. To get a local frame for the height field, we first fit a local plane. We set the center of the bounding sphere as the center and calculate the normal of the local plane by averaging normals that belong to the points in the bounding sphere. Each point within the bounding volume is then stored in this local frame (Fig. 2). Finally, we fit a smooth function to this sampled height field via radial basis based interpolation. We use Gaussians with centers residing on a regular grid of size 16×16 on the local plane as the radial basis functions. The resulting coefficients form our feature vector. As illustrated in Fig. 2, we take bounding spheres of multiple sizes, to capture the geometric structures at different scales.

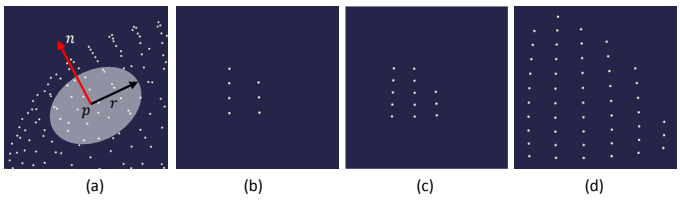


Fig. 2. Generating patches from a 3D model. (a) A patch from a bounding sphere with the center point p, radius r, and patch normal n. (b) A patch with $r = 0.05$, (c) $r = 0.15$, and (d) $r = 0.25$.

B. Dictionary Learning

Once we get the feature vectors from the local patches as described in the previous section, we learn a compact representation for these feature vectors by the dictionary learning method K-SVD [17]. The K-SVD algorithm solves the following system.

$$\min_{\mathbf{D},\mathbf{X}} ||\mathbf{F} - \mathbf{D}\mathbf{X}||_F^2 \quad ||\mathbf{x}_i||_0 \leq s \tag{1}$$

Here, $\mathbf{X}$ stores the coefficients of training data with columns denoted with $\mathbf{x}_i$, $\mathbf{D}$ is the learned dictionary, and $\mathbf{F}$ denotes a matrix where each column stores a feature vector $\mathbf{F} = [\mathbf{f}_1, ..., \mathbf{f}_n]$.

The dictionary learning is performed in two alternating steps: sparse coding of feature vectors, and dictionary updates. In the sparse coding step, the coefficient matrix $\mathbf{X}$ is computed by minimizing the above error. For this step, we used orthogonal matching pursuit, as it leads to better results than the standard matching pursuit algorithm. In the dictionary update step, the columns of $\mathbf{D}$, which are the learned dictionaries, are computed again by minimizing the above error, while keeping the coefficients in $\mathbf{X}$ fixed. These two steps are repeated until convergence. Here, s determines sparsity of the signals. For our case, we have observed that number of iterations ranging from 25 to 30 are sufficient for convergence.

IV. Experiments

In our experiments, we compare and analyze learning results for different geometry patch sets. We analyze the convergence of the learning algorithm, and illustrate a few of the learned dictionaries to illustrate the final representation.

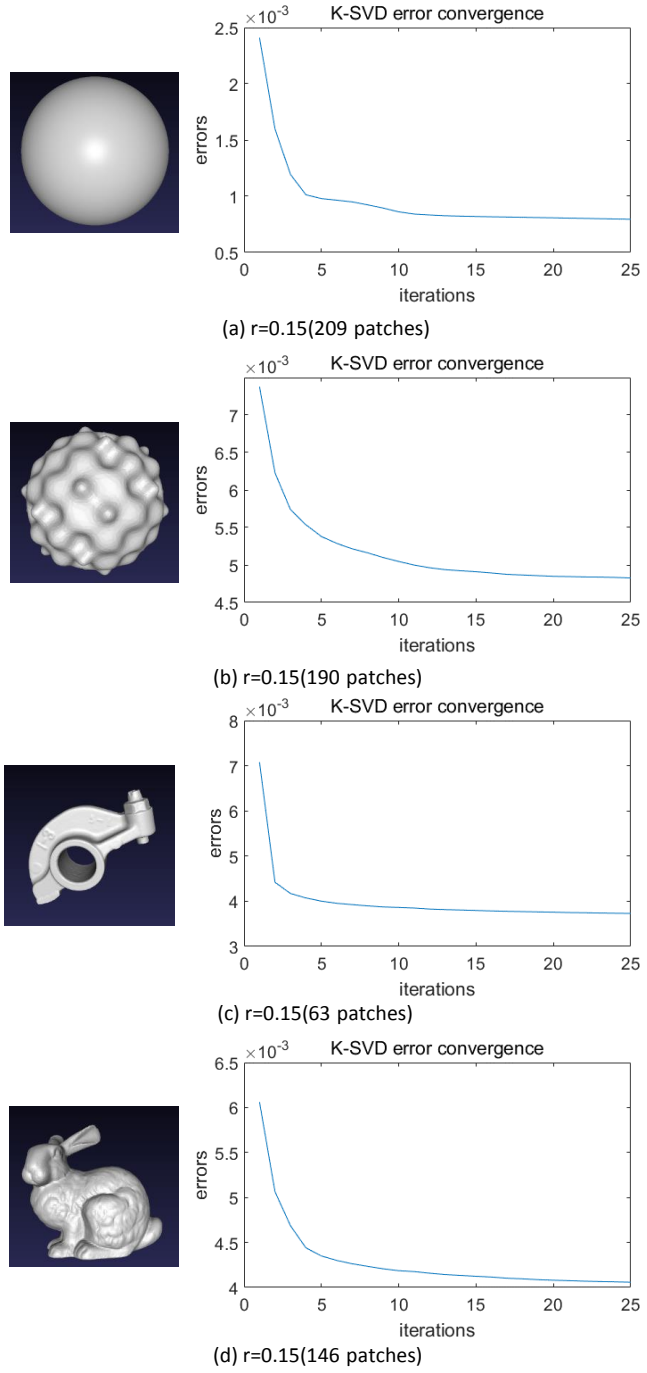


Fig. 3. K-SVD error convergences with simple shape models. The error plots are with the radius $r = 0.15$. The numbers of vertices of the sphere, bumpy sphere, rocker-arm and bunny models are 41k, 68k, 40k and 34k, respectively. The iteration count is 25.

A. Patches of Different Scales

When the patch size is too small, all local geometries become very similar, thus we cannot get a meaningful dictionary. On the contrary, when the patch size is too large, we capture global geometric properties, which can vary vastly from one object to the other, again avoiding to produce a meaningful dictionary. Hence, the patch size should be set carefully.

To provide insight into this behavior, we use a sphere, bumpy sphere, rocker-arm, and Stanford bunny. The models have from $35,000$ to $68,000$ vertices, and the bounding box is normalized to be of unit volume. From these models, we get different patch sets by using bounding spheres of radius 0.05, 0.15 and 0.25. The total dataset for this experiment is 12 patch sets from four different models. The used models and convergence of the K-SVD error are shown in Fig. 3. The error converges to near zero in all shape models with different patch sizes.

Atoms of the learned dictionary are shown in Fig. 4. Each of them shows six dictionary atoms capturing the largest variations. The dictionary is learned with the size of 30 and sparsity of 9. The atoms from the patch set with the radius of 0.05 are not presented here since the patch size is too small and almost the same. In the Fig. 4 (a) and (b), the dictionary atoms are learned from the rocker-arm model with the patch radii of 0.15 and 0.25, respectively. We can easily notice that the atoms of the Fig. 4 (a) describe local features better than (b). In the Fig. 4 (c) and (d), the atoms of the dictionary are learned from the bunny model with the patch radii of 0.15 and 0.25, respectively. In the case of the bunny model, the local features are well represented in both (c) and (d).

B. Patches of Different Orientations

Although the computed normal determines how the locally fit plane is oriented, there is one more degree of freedom, the rotation of the local coordinate system in the plane. We tested how different rotations of patches affect the learning results. For this test, we create patch sets with different rotation angles in the locally fit plane. For a patch, we generate 36 versions of that patch by rotating the patch in steps of 10 degrees from 10 to 360. The axis of the rotation is the normal of the locally fitted plane in the patch. Fig. 5 illusrates how to generate rotated patches in the locally fit plane. We create new databases of patches consisting of 190×36 for the bumpy sphere, 63×36 for the rocker-arm, and 146×36 for the bunny models. We use 0.15 as the patch radius. We illustrate the resulting graphs of the four singular values as given by the K-SVD algorithm and the corresponding dictionary atoms in Fig. 6. When we compare the learned atoms of the rocker-arm and the bunny using the rotated patches with the original ones in Fig. 4, the shapes of the first four atoms are different each other.

C. Mixing Patches

In this experiment, we construct mixed patch sets from different shape models and of different sizes. First, we compare dictionaries for patches of different sizes from the same model. We choose 0.15 and 0.25 as the radii of the patches, and mix the resulting two sets for the sphere model (Fig. 7 (a)), and the bumpy sphere model (Fig. 7 (b)). Finally in Fig. 7 (c) we also show the dictionary atoms when mixing patches from the two models for the patch radius 0.15.

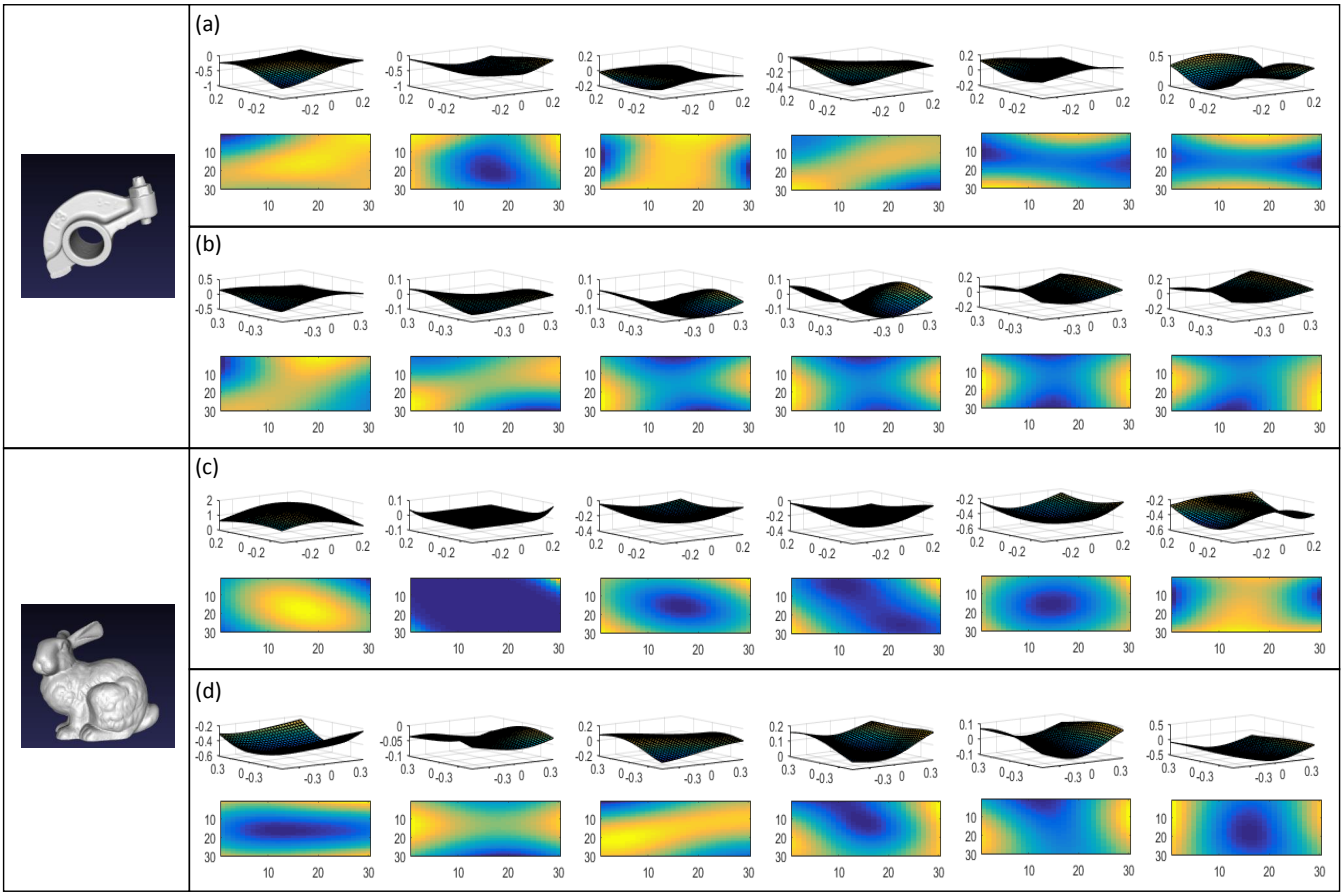


Fig. 4. Atoms from the learned dictionaries from the rocker-arm and the bunny models. In (a) to (d), atoms are arranged in importance order from left to right. (a) and (c) indicate the dictionary atoms of the patch set where the radius $r = 0.15$. (b) and (d) indicate the dictionary atoms of the patch set where the radius $r = 0.25$. The color gradients of blue to yellow indicate low to high value in each color map. Please note that every plot has its own coordinate system.

V. Conclusions

In this paper, we presented an exploration of learning local geometric structures for big geometry data. We have presented several examples with patches extracted from different models, of different sizes, orientations, and mixing. Our local approach allows us to have controllable generalization properties in contrast to previous structure-aware methods in geometry processing. It also leads to compact and local geometry representations that capture the dominant structures in the data. We believe such ideas can be used to inject prior information into the techniques used for various geometry processing applications such as smoothing, abstraction, or reconstruction in future works.

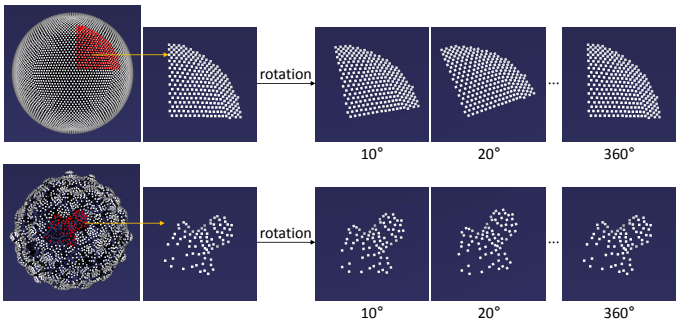


Fig. 5. Generation of rotating patches from the sphere and the bumpy sphere models.

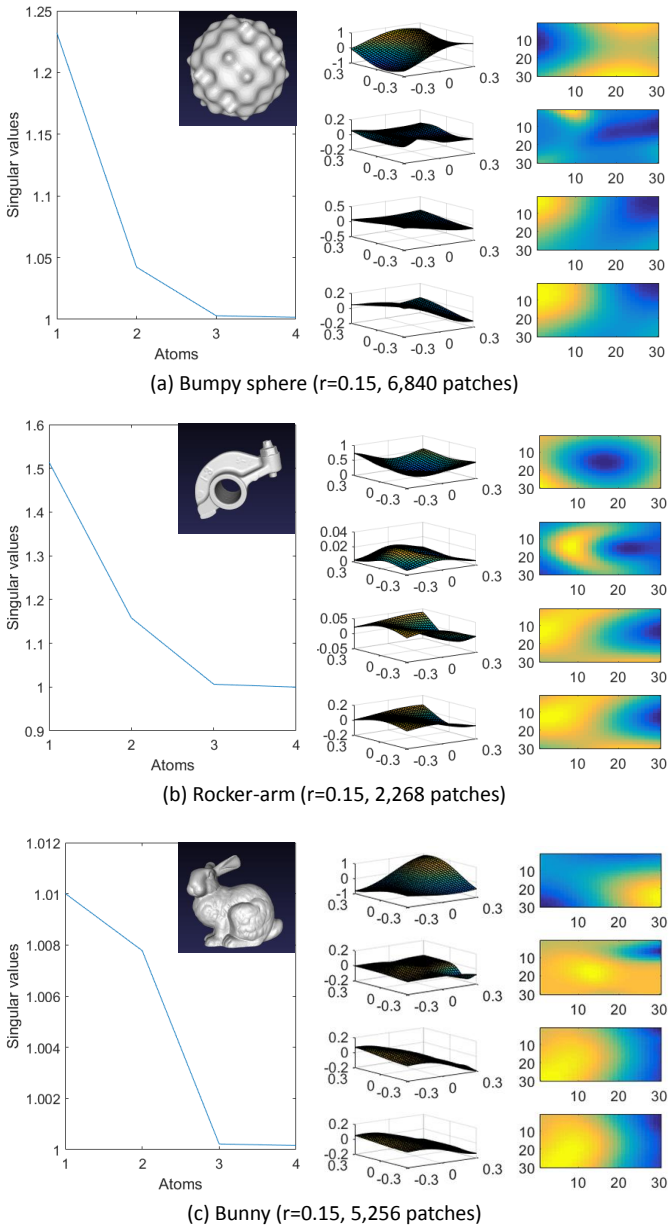
Acknowledgment

This research was supported in part by the Bio & Medical Technology Development Program of the NRF funded by the Korean government, MSIP(NRF-2015M3A9A7029725) and in part by (NRF-2014R1A2A1A11053135).

References

[1] M. Berger, A. Tagliasacchi, L. M. Seversky, P. Alliez, J. A. Levine, A. Sharf, and C. Silva, "State of the art in surface reconstruction from point clouds," *Eurographics STAR (Proc. of EG'14)*, 2014.

[2] A. B. Goldberg, X. Zhu, A. Singh, Z. Xu, and R. Nowak, "Multi-manifold semi-supervised learning," in *Proc. of 12th International Conference on Artificial Intelligence and Statistics*, vol. 5, 2009, pp. 169–176.

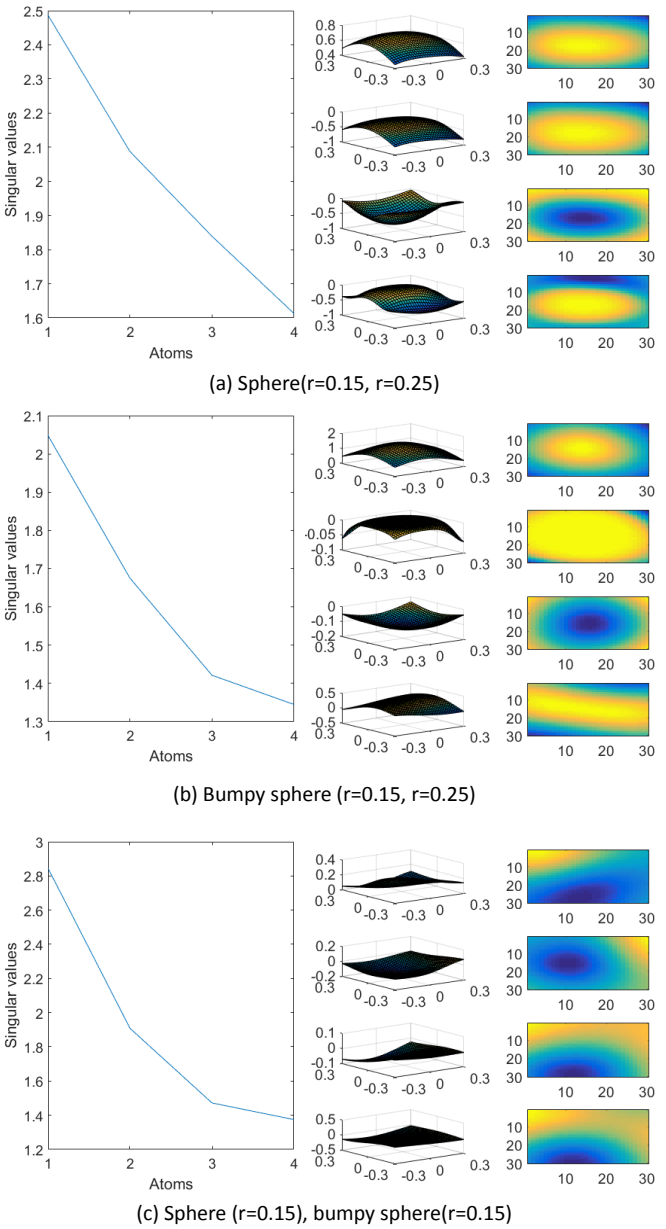


(a) Bumpy sphere (r=0.15, 6,840 patches)

(b) Rocker-arm (r=0.15, 2,268 patches)

(c) Bunny (r=0.15, 5,256 patches)

Fig. 6. Atoms from the learned dictionary when patches with rotations are used to construct the patch database. For each inset, we plot the largest singular values and the corresponding atoms. The inset (a) is for the bumpy sphere, (b) for the rocker-arm, and (c) for the bunny model. The importance order is from top to down.



(a) Sphere(r=0.15, r=0.25)

(b) Bumpy sphere (r=0.15, r=0.25)

(c) Sphere (r=0.15), bumpy sphere(r=0.15)

Fig. 7. Atoms from the learned dictionary when (a) patches of radiuses 0.15 and 0.25 are used in the same patch database for the (a) sphere and (b) bumpy sphere models. In (c), we further show the atoms when patches from the two models are mixed in a database for radius 0.15.

[3] T. Zhang, A. Szlam, and G. Lerman, "Median k-flats for hybrid linear modeling with many outliers," in *Computer Vision Workshops (ICCV Workshops), 2009 IEEE 12th International Conference on*, 2009, pp. 234–241.

[4] Y. W. Mikhail Belkin, Qichao Que and X. Zhou, "Toward understanding complex spaces: Graph laplacians on manifolds with singularities and boundaries," in *25th Conference on Learning Theory (COLT 2012), JMLR W&CP(COLT)*, vol. 23, 2012, pp. 36:1–36:26.

[5] G. Mustafa, H. Li, J. Zhang, and J. Deng, "ℓ1-regression based subdivision schemes for noisy data," *Computer-Aided Design*, vol. 58, pp. 189–199, 2015.

[6] R. Hu, L. Fan, and L. Liu, "Co-segmentation of 3d shapes via subspace clustering," *Computer Graphics Forum*, vol. 31, no. 5, pp. 1703–1713, 2012.

[7] S. Xiong, J. Zhang, J. Zheng, J. Cai, and L. Liu, "Robust surface reconstruction via dictionary learning," *ACM Trans. Graph.*, vol. 33, no. 6, pp. 201:1–201:12, Nov. 2014.

[8] L. Xu, R. Wang, J. Zhang, Z. Yang, J. Deng, F. Chen, and L. Liu, "Survey on sparsity in geometric modeling and processing," *Graphical Models*, 2015, in Press.

[9] Y. Lipman, D. Cohen-Or, D. Levin, and H. Tal-Ezer, "Parameterization-free projection for geometry reconstruction," *ACM Trans. Graph.*, vol. 26, no. 3, Jul. 2007.

[10] H. Huang, D. Li, H. Zhang, U. Ascher, and D. Cohen-Or, "Consolidation of unorganized point clouds for surface reconstruction," *ACM Trans. Graph.*, vol. 28, no. 5, pp. 176:1–176:7, Dec. 2009.

[11] L. He and S. Schaefer, "Mesh denoising via l_0 minimization," *ACM*

Trans. Graph., vol. 32, no. 4, pp. 64:1–64:8, Jul. 2013.

[12] H. Zhang, C. Wu, J. Zhang, and J. Deng, "Variational mesh denoising using total variation and piecewise constant function space," *Visualization and Computer Graphics, IEEE Transactions on*, vol. 21, no. 7, pp. 873–886, Jul. 2015.

[13] B. H. Le and Z. Deng, "Smooth skinning decomposition with rigid bones," *ACM Trans. Graph.*, vol. 31, no. 6, pp. 199:1–199:10, Nov. 2012.

[14] ——, "Two-layer sparse compression of dense-weight blend skinning," *ACM Trans. Graph.*, vol. 32, no. 4, pp. 124:1–124:10, Jul. 2013.

[15] ——, "Robust and accurate skeletal rigging from mesh sequences," *ACM Trans. Graph.*, vol. 33, no. 4, pp. 84:1–84:10, Jul. 2014.

[16] J. Digne, R. Chaine, and S. Valette, "Self-similarity for accurate compression of point sampled surfaces," *Computer Graphics Forum*, vol. 33, no. 2, pp. 155–164, 2014.

[17] M. Aharon, M. Elad, and A. Bruckstein, "K-svd: Design of dictionaries for sparse representation," in *IN: PROCEEDINGS OF SPARS05*, 2005, pp. 9–12.