Geometry Representations for Big Geometry Data with Unsupervised Feature Learning

Yeo-Jin Yoon*, Alexander Lelidis‡, A. Cengiz Öztireli†,
Jung-Min Hwang*, Markus Gross† and Soo-Mi Choi*
*Department of Computer Science and Engineering
Sejong University, Seoul, Republic of Korea
†Computer Graphics Laboratory, ETH Zürich
Zürich, Switzerland
‡TU Berlin, Berlin, Germany

***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 noises, outliers, and missing data. We present features learned on several patch databases, highlighting 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 RGBD sensors develops, acquiring various scanned data from the real world is becoming easy. This data consists of vertices and their connectivities, and it is called as 3D geometry data. Most of 3D geometry data from the real world are not easy to process because of its huge amounts of unstructured points and lower connectivity information. In computer graphics, many research works have been conducted to represent and analyze geometries. More recently, signal processing and machine learning techniques [1] have been applied for analyzing and reconstructing 3D geometries. In such approaches, it is often challenging to analyze the dominant components of 3D geometries to efficiently and robustly represent a big amount of geometry data. In addition, it is essential to reduce various types of noises and handle missing parts of 3D geometries. In the fields of 3D reconstruction and shape analysis, 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 features and analyze each feature's dominant components [2,3]. These methods have the advantages of fast processing speed and robustness against noises and outliers, but handling big missing parts is difficult. On the other hand, global methods [4] can handle big missing parts because they consider global shape characteristics, but the data processing time is much longer than local ones, especially for big geometry data.

To properly handle big geometry data composed of a huge amount 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 consisting of sparse approximation to extract the dominant components describing the overall patches from unsupervised data.

Sparse signal representation is used in various areas such as image processing, machine learning, neuroscience, and statistics [1]. Recently, this traditional technique is extending its application area to 3D geometry processing with mesh denoising, 3D surface reconstruction and mesh segmentation. In the geometry processing, input signals are usually vertices and normals with basis functions, and each signal is analyzed by signal processing algorithms to reproduce surfaces with reducing noises and outliers. For example, sparse regulalrization [5] for image smoothing is extended to 3D geometries. To preserve edges and smooth noises, several operators such as discrete differential operator [6] and Laplacian [7] are used to compute gradient terms in sparse regularization. Other works attempted to use projection operator [8,9] to reduce noises and outliers when reconstructing surfaces. Most of sparsity methods in geometric processing are increasing because they are strong for preserving features and reducing noises and outliers.

Dictionary learning is one of the sparse signal representation methods and it optimizes dictionary and coefficient information at the same time. This technique is mainly used in image compression, super-resolution, and denoising. However, its application area is increasing in 3D geometry reconstruction, compression of point clouds and rendering because it solves geometric problems directly. In the area of geometric deformation and animation, the dictionary learning method is used for pose decompose [10] and analyzing mesh sequence [11] and highly deformable models [12]. In 3D reconstruction, sparse dictionary is used for storing local geometry features from triangle meshes [4]. Another research [13] uses self-similarity and K-SVD [14] for geometry compression on point sampled surfaces.

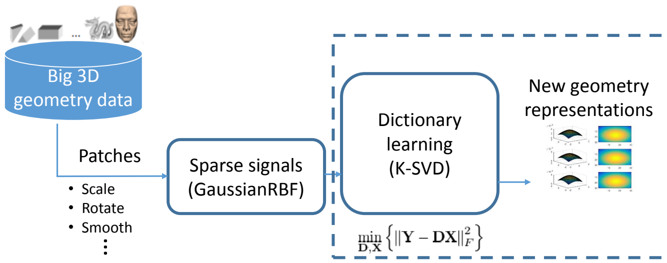

Fig. 1. Overall pipeline of geometry representations using unsupervised feature learning.

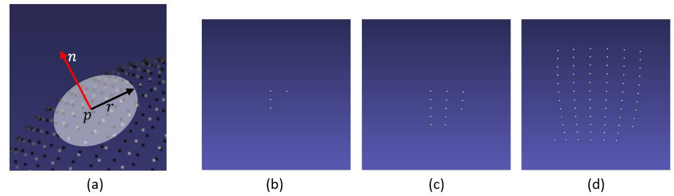

Fig. 2. Generating patches from a sphere model, which consists of 2,500 points. (a) shows determining a patch from a bounding sphere with a center point p, a radius r, and a patch normal n. (b) shows a patch with $r = 0.3$, (c) is a patch with $r = 0.5$, and (d) is a patch with $r = 0.7$.

In this paper, we explore representations for big geometry data using unsupervised feature learning with geometric patches. Fig. 1 shows our suggested overall pipeline. First, we define local geometric patch data from raw point clouds, and parameterize local features using a radial basis function. The dictionary learns dominant components of local features and represents overall patches. We perform experiments and analyze geometry by creating patch datasets from several shape models.

In section 2, creating method of geometric patch data and parameterized local geometric data is described with dictionary learning. We show and analyze the experimental results of dictionary learning in section 3, and discuss about future work and conclusion in section 4.

II. Method

To construct a local geometry representation for dictionary, choosing basis function is important. In this paper, we use Gaussian radial basis function to parameterize local geometry based on a geometric patch, and learn the dictionary using this representation. For the experiment with various local geometries, different patch sets are created with different sizes and rotation angles. The training data is learned using well-known dictionary method K-SVD [14].

A. Geometric Patch Generation

Raw geometry data has different sizes, number of vertices and density of points depending on creating methods. To get the coherent patches, we first unitize the size of a model as range of [-1,1]. Then seed points are selected to divide the model into patches, and a patch size is determined by the bounding sphere with the radius r from a seed point. To avoid the sampled points in a bounding sphere are shared with different bounding spheres, we measure the distances between points and the centers of sharing bounding spheres. The sampled points are included in the nearest bounding sphere, same as [13]. Then the vertex coordinates of each patch are stored to generate local geometric patches.

To get Gaussian height fields, local planes should be determined. Therefore, we set the center of a bounding sphere as a local plane's center and calculate a normal of the local plane by averaging vertex normals in a bounding sphere. This normal determination is useful since the vertex normals are oriented to outside of the surface. When 3D models are synthesized or reconstructed, the global shape can be easily constructed since the orientation problem of the local geometries is solved. The height field stores distances and heights of included vertices. Here, the distance is stored as a radial distance on a local plane from the center of the plane to the vertex, and the height is stored as an orthogonal distance from the vertex to the local plane. These information are stored as training data for dictionary learning. Fig. 2 shows generating method of patches from a sphere model. Depending on radius r, the patch size is determined differently. As decreasing of r, a patch includes less vertices, hence the number of local geometry representation is increased. This means the increment of signals of the dictionary.

B. Dictionary Learning

To learn the dictionary, we assume that the input signals can be sparsely represented. We used the Gaussian radial basis function for the height field in the previous section. The uniform grid is created by Gaussian function, and the size of grid is 16 × 16 for each signal [15]. The dictionary alternatively performs sparse coding and updating atoms in each iteration. Therefore, it finds the best coefficient matrix and dictionary to make better representations of training data. Basic algorithm is as follows:

$$\min_{D,X} ||F - DX||_F^2, s.t.||x_i||_0 \leq s, \forall i = 1, 2, ..., m. \quad (1)$$

In (1), X indicates coefficients of training data, D indicates dictionary and F indicates a signal of training data $F = [f_1, ..., f_n]$. The number of signals is same as the number of columns of the training matrix F.

The dictionary learning is performed with sparse coding and dictionary update. In sparse coding, the coefficient matrix X and sparse matrix D are determined by minimizing error, and a pursuit algorithm is used to compute $x_i \in X$. In the second phase, dictionary atoms $d_k \in D$ are updated. This process performs iteratively until the error is converged. In our experiments, the number of iterations is 25 to 30 depending on the training data size.

III. Experiments

In our experiments, we compare and analyze learning results depending on geometric patch groups with different sizes, orientations and mixed components. We compare the error convergences during iterative learning, and take the biggest four eigenvalues from the learned dictionary to show the learning results on different patch groups.

A. Scales of Patches

When the geometric patch size is too small, all local geometries seem to be small and same, thus we cannot get a proper dictionary. On the contrary, when the geometry patch size is too big, the complexity of a local geometry is too high and this is similar to consider a global geometry. In this experiment, we create sphere and bumpy sphere models. Each model has 5,500 vertices, and the bounding volume is [-1,1]. From these shape models, we get three different patch sets by using bounding spheres, which have radius 0.3, 0.5 and 0.7, respectively. The total dataset for this experiment is six patch sets from two different models. Then we make a discrete representation on each patch with the Gaussian basis function and put all signals into the training matrix. The created models and K-SVD error convergences are shown in Fig. 3. The first column shows error plots of a sphere model, and the size of patches are increased from (a) to (c) as 0.3, 0.5 and 0.7, respectively. We can notice that the error convergences are better when the patch size is bigger in the sphere model case. However, the graphs of the bumpy sphere on the right column show unstable convergence results in less than 15 iterations when the patch size is increased.

Atoms of the learned dictionary are shown in Fig. 4. Each plot shows three atoms related to the three biggest eigenvalues in a learned dictionary with 30 $\times$ 30 Gaussian grid.

B. Angles of Patches

To see the results depending on different angles of patches, we created patch sets with different angles to make training sets. For a patch, we generated 36 rotated patches by rotating 10 degrees from 10 to 360, so that each patch has its own 36 rotated patches. The axis of the rotation is a normal of the patch by averaging included vertex normals. With the sphere model, we used 0.5 size of a patch set, and the total generated patches are 74 patches $\times$ 36 angles = 2,664. For comparison, we also created a patch set from the bumpy sphere with the patch size of 0.5. In this case, the total rotated patches are 79 patches $\times$ 36 angles = 2,844. Fig. 5 shows how we created patches by rotating angles.

When the training data is created, the data should be converted into training signals. We created training matrices from the generated patches. The matrix consists of 1,368 columns with the sphere model and 1,620 columns with the bumpy sphere model. Also, we added one more rotating patch set from open 3D geometry archive [16]. We chose rockerArm composed of 64 patches $\times$ 36 angles = 2,304 patches with 0.3 for the patch size. Fig. 6 shows the graphs of the singular values and atoms of the learned dictionary by showing indicating patches with Gaussian basis. As the number of training data increases, we can notice that the atoms in the dictionary look similar.

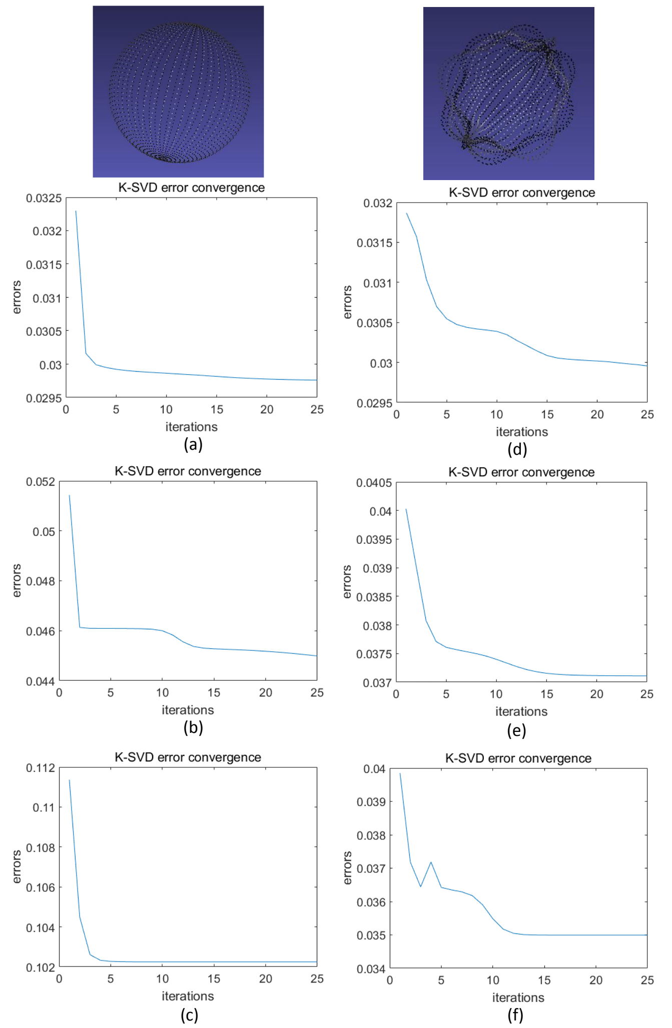

Fig. 3. K-SVD error convergences with simple shape models. The first column shows the convergence errors of patch sets from the sphere model, and the second column shows the convergence errors in the case of the bumpy sphere. From (a) to (c) show the error convergences when r = 0.3, 0.5 and 0.7, and the number of patches are 196, 74 and 38, respectively. From (d) to (f) show the error convergences with r=0.3, 0.5 and 0.7 and the number of patches are 227, 79 and 45, respectively. The iteration time is 25.

C. Mixing Patches

In this experiment, we constructed mixing patch sets with different shape models and different sizes of patch sets as training data. First, we compared the learned dictionary with different sizes of patch sets from the same model. We chose 0.5 and 0.7 size of patch sets and mixed two sets from the sphere model and the bumpy sphere model, respectively. Second, we mixed the same size patch sets from different models. For this, we chose 0.3 size of patch sets. The training results are shown on Fig. 7. In Fig 7 (a), (b) and (c), the numbers of patches are 112, 124 and 119, respectively. As shown on the right plots, the ranges of eigenvalues are different, but the shapes of the patches look similar.

IV. Conclusions

In this paper, we explored and presented local geometry features from dictionary learning for big geometry data using parameterized patches. We created several simple geometries to make training sets by changing patch sizes, rotation angles and mixing different patches from different models. In our experiments, we have shown that the local geometric representations can be handled properly with dictionary learning by showing the atoms of the dictionaries. In addition, the radial basis parameterized local patches and learning results have shown the potential utilization of geometric dominant components in geometry processing. In the near future, we are going to use the parameterized local features and dictionaries to reconstruct high resolution geometries or complex geometries from big geometry data.

Acknowledgment

This work was supported in part by Korean NRF and Swiss SER under the Korean-Swiss Cooperative Program (2013K1A3A1A14055180) and in part by the NRF grant (2014R1A2A1A11053135).

References

[1] 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.

[2] 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.

[3] 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.

[4] 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.

[5] L. Xu, C. Lu, Y. Xu, and J. Jia, "Image smoothing via l_0 gradient minimization," *ACM Trans. Graph.*, vol. 30, no. 6, pp. 174:1–174:12, Dec. 2011.

[6] 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.

[7] 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.

[8] 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.

[9] 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.

[10] 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.

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

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

[13] 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.

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

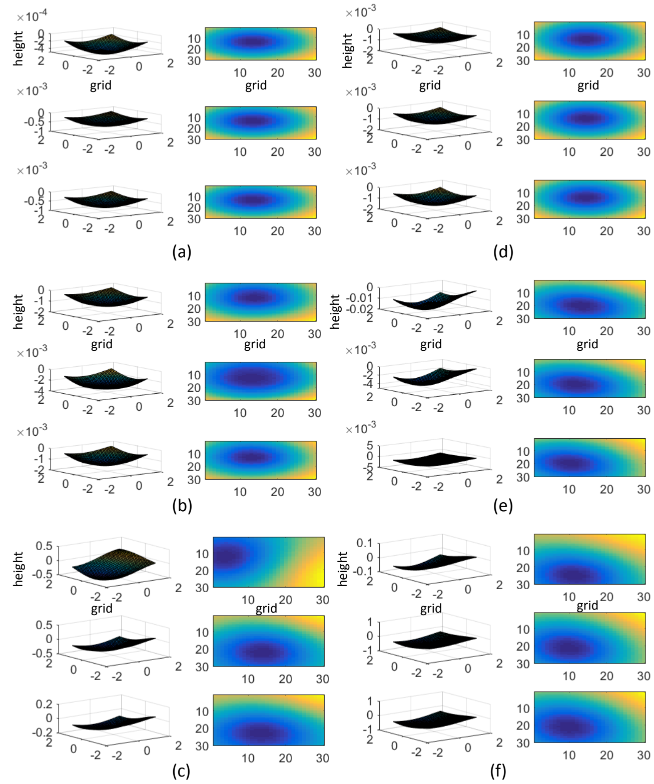

Fig. 4. Atoms from the learned dictionaries. The first column shows the dictionary atoms of the sphere model, and the second column shows the dictionary atoms of the bumpy sphere model. From (a) to (c) show three atoms of the dictionary with 196, 74 and 38 patches and 0.3, 0.5 and 0.7 size of patch sets, respectively. From (d) to (f) show three atoms of the dictionary with 227, 79 and 45 patches and 0.3, 0.5 and 0.7 size of patch sets, respectively. The right side in a plot shows color coding for the left patches. Blue to yellow indicates low to high value.

[15] A. Lelidis, "Structure-aware surface reconstruction with sparse moving least squares," bachelor Thesis, TU Berlin, 2015.

[16] G. Lavoue, E. D. Gelasca, F. Dupont, A. Baskurt, and T. Ebrahimi, "Perceptually driven 3d distance metrics with application to watermarking," in *IN: PROCEEDINGS OF SPIE'06*, 2006, pp. 63 120L–63 120L.

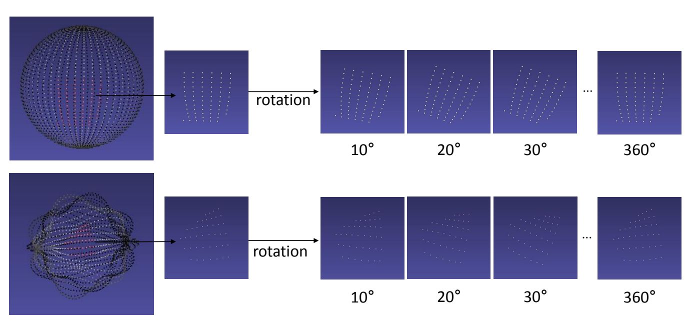

Fig. 5. Generation of rotating patches from a sphere and a bumpy sphere model. The first row shows example patches from the sphere, and the second row shows patch samples from the bumpy sphere.

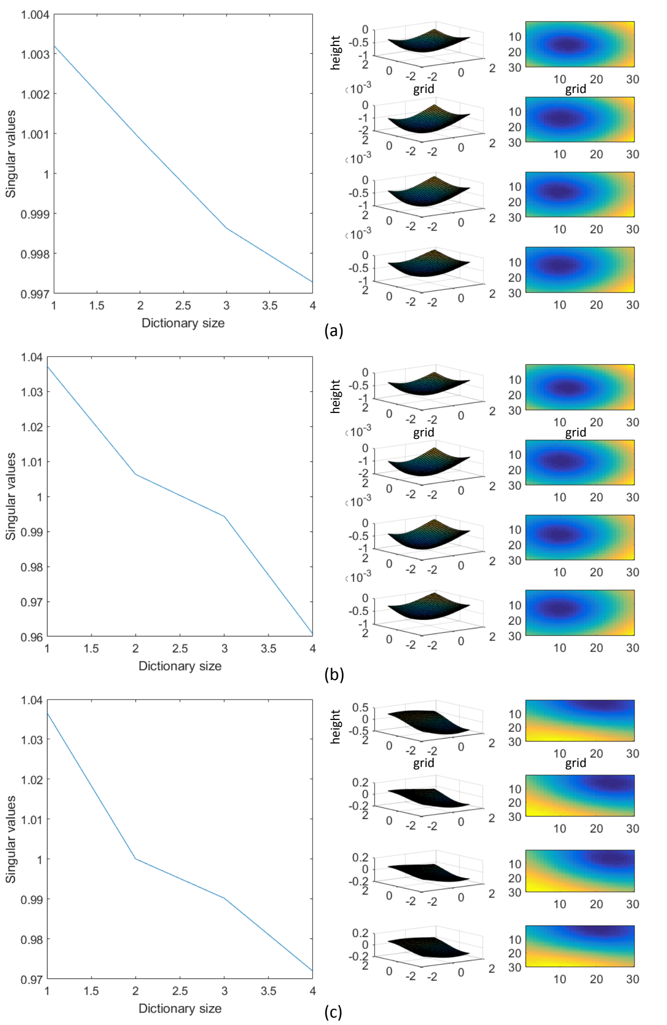

Fig. 6. Atoms of a dictionary with rotated patch sets. In (a), the left plot shows the four biggest singular values in the learned dictionary, and the right plot shows patches, which indicate the four biggest eigenvalues from the sphere model. (b) and (c) show the singular values and patches from the bumpy sphere model and the rockerArm model, respectively.

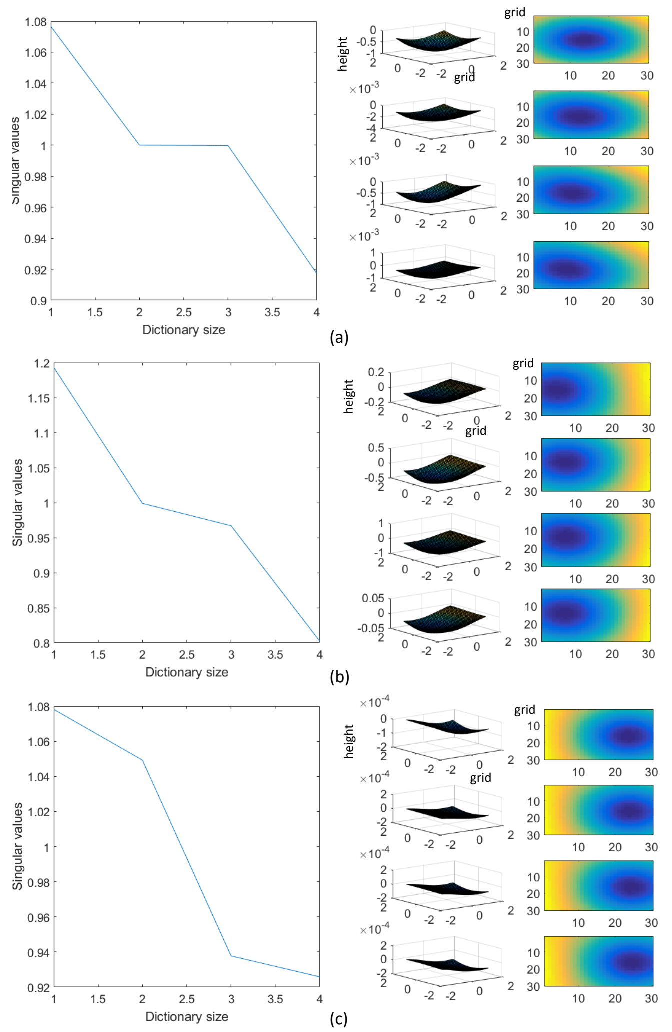

Fig. 7. Atoms of a dictionary with mixed patch sets. In (a), the four biggest singular values and patches, which indicate the four biggest eigenvalues are shown. A patch set is created by mixing 0.5 and 0.7 size of patch sets from the sphere model. (b) shows the atoms of a learned dictionary with mixing two patch sets from the bumpy sphere model. We mixed 0.5 and 0.7 size of patch sets. And (c) shows the atoms of a learned dictionary with mixing 0.3 size of patches from the sphere and bumpy sphere models.