# Spherical superpixel segmentation

**Conference Paper** · July 2016

DOI: 10.1109/ICME.2016.7552898



Qiang Zhao
Chinese Academy of Sciences


Jiawan Zhang
Tianjin University

# SPHERICAL SUPERPIXEL SEGMENTATION

*Qiang Zhao, Liang Wan*, Jiawan Zhang*

School of Computer Software, Tianjin University, Tianjin, China
{qiangzhao,lwan,jwzhang}@tju.edu.cn

**ABSTRACT**

In this paper, we present a superpixel generation method for spherical images, which cover 360° field-of-view. Unlike previous works that directly use existing superpixel algorithms on unrolled spherical images, our approach explicitly considers the geometry for spherical images and uses sphere as the underlying representation. For quantitative evaluation, we make a spherical image segmentation database by transforming Berkeley segmentation dataset to the spherical domain. Experimental results show that our method can get better performance in terms of adherence to image boundaries and spherical size variance. What's more, superpixels generated by our method all have closed contours.

***Index Terms***— Superpixel, clustering, spherical image

## 1. INTRODUCTION

Superpixel generation has become a standard preprocessing step in many computer vision applications, such as image parsing [1], depth estimation [2], segmentation [3] and object localization [4], etc. The key idea of superpixel generation algorithms is grouping similar pixels into perceptually meaningful atomic regions [5], which can dramatically reduce the complexity of subsequent computer vision tasks.

In the literature, many superpixel algorithms have been proposed, which can be categorized into graph based methods [6] and gradient descent based methods [5]. These methods are designed to generate superpixels for conventional planar images. Simply applying existing methods on spherical images [7, 8] or projected piecewise perspective images [9] has some problems. Because of the wide field-of-view of spherical images, these images are modeled by image sphere instead of image plane. As a result, unrolled spherical images inevitably have distortions, which may impact the performance of superpixel algorithms. Another problem comes from the fact that the image sphere is a closed surface. The superpixels of spherical image should have closed contours after we map them to the sphere, which is not guaranteed by the planar methods. In addition, although some planar methods are designed to generate compact and regular superpixels, their uniformities are lost when used for spherical images.

To deal with these problems, we propose a spherical superpixel generation algorithm. Our algorithm explicitly considers the geometry for spherical images. It resembles the idea of famous SLIC algorithm [5] and uses clustering to generate superpixels in spherical domain. In our approach, we use Hammersley points [10] sampled on sphere to initialize superpixel centers. In assignment and update steps, we use cosine dissimilarity and spherical distance as the spatial distance measure. For quantitative evaluation, we make spherical image segmentation database by transforming Berkeley segmentation dataset to the spherical domain. We also test our algorithm on real captured spherical images. Experimental results show that our method can give better performance.

### 1.1. Related Work

Superpixels correspond to homogeneous subregions in one image. This statement has two meanings: (1) pixels from one superpixel have similar appearance and depth, or they belong to the same object, (2) the contour of superpixels adheres well to object boundaries. These properties of superpixels are not only used to deal with conventional planar images, but are also used in applications involving spherical images. In [9], spherical images are converted into piecewise perspective images, from which superpixels are generated using graph based method [6]. Then a superpixel based multi-view stereo method has been proposed, which assigns one depth per superpixel. Cabral and Furukawa [7] directly used planar superpixel generation algorithm [6] on indoor spherical images, then exploited texture homogeneity of these images and employed structure classification to infer 3D cues for floorplan reconstruction. Sakurada and Okatani proposed a change detection method which uses features of convolutional neural network in combination with superpixel segmentation [8]. Given a low resolution change map estimated from CNN features, their method integrates this low resolution map with superpixels segmentation of spherical images generated by planar algorithm [6] to get precise boundaries of the changes. Despite the successes of the usage of superpixels, these works

*is the corresponding author. This work is supported by the National Natural Science Foundation of China (61572354), the National Science and Technology Support Project (2013BAK01B05, 2014BAK09B04), the National Social Science Foundation of China (12&ZD231), Philosophy and Social Sciences Research Project of Tianjin (TJSR15-008).

still have the three problems we talked about previously.

## 2. SPHERICAL SUPERPIXELS

When dealing with spherical images, we pay attention to the underlying geometry, which will be first described.

### 2.1. The Geometry for Spherical Images

As shown in Figure 1, we denote the resolution of the spherical image to be $w \times h$ pixels, and represent it using Equirectangular projection. Because spherical images cover 360 degree field of view horizontally, we can know that the radius $r$ of the sphere is $r = \frac{w}{2\pi}$. Therefore the surface area $A$ of the sphere is computed as $A = 4\pi r^2 = \frac{w^2}{\pi}$.

Given a point $(X, Y, Z)$ on the unit sphere, its spherical coordinates can be represented as $(\theta, \phi)$, where $\theta = \arctan 2(Y, X)$ is the azimuthal angle, and $\phi = \arccos(Z)$ is the polar angle. After mapping the range of $\theta$ from $(-\pi, \pi]$ to $[0, 2\pi)$, the corresponding pixel position $(x, y)$ on the equirectangular image plane is $x = \frac{\theta w}{2\pi}$ and $y = \frac{\phi h}{\pi}$.

### 2.2. Our Approach

Our approach is based on SLIC algorithm and extends it to the spherical domain. SLIC [5] is an adaptation of k-means clustering algorithm for superpixel generation, which limits the search space in assignment step to a region proportional to the superpixel size and combines color and spatial proximity as distance measure. In initialization, we use Hammersley points to uniformly sample the sphere. Specifically, the $k$ initial cluster centers $C_i = [X_i\ Y_i\ Z_i\ l_i\ a_i\ b_i]$ are assigned by sampling the unit sphere with Hammersley points [11], where $[X_i\ Y_i\ Z_i]$ is the unit vector describing the sampling position and $[l_i\ a_i\ b_i]$ is the LAB color of spherical image pixel corresponding to the sampling point. Because superpixels should adhere well to image boundaries, the initial centers are moved to the lowest gradient position in a 3×3 neighborhood. The gradient of the spherical image is given by

$$\nabla I = [\frac{1}{\sin\phi}\frac{\partial I}{\partial \theta},\ \frac{\partial I}{\partial \phi}]. \tag{1}$$

After initialization, the algorithm iterates between the assignment step, which associates each pixel $p = [X\ Y\ Z\ l\ a\ b]$ to its nearest cluster center according to distance measure $D(C_i, p)$ discussed in Section 2.3,

$$L(p) = \underset{i|(x,y)\in R_i}{\arg\min}\ D(C_i, p), \tag{2}$$

and the update step, which adjusts cluster centers,

$$C_i = \underset{C_i}{\arg\min} \sum_{L(p)=i} D(C_i, p). \tag{3}$$

Compared to standard k-means clustering, the key to speeding up SLIC algorithm is only considering the cluster centers falling in a neighborhood of each pixel. This is equivalent to limiting the search space to a $2S \times 2S$ region around each cluster center, where $S$ is the superpixel size. In the spherical case, the superpixel size can be computed from surface area of the sphere, and gives

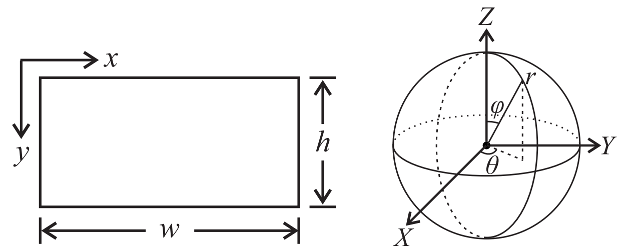



**Fig. 1**. The geometry for spherical images

$$S = \sqrt{\frac{A}{k}} = \frac{w}{\sqrt{k\pi}}. \tag{4}$$

Considering the geometry of the spherical image, we define the local search region $R_i$ for cluster $i$ as

$$R_i = \{(x, y)|x_i - \frac{S}{\sin\phi} \le x \le x_i + \frac{S}{\sin\phi},\ y_i - S \le y \le y_i + S\}, \tag{5}$$

where $(x_i, y_i)$ are the 2D image coordinates for the center of $i$-th cluster and $\phi = \frac{y\pi}{h}$ is the polar angle corresponding to $y$-th row of the image. Note that when $y$ coordinate of the search region falls outside the valid range, mirror texture address mode should be used. That is to say the search region for $y$ will become

$$\begin{cases} 0 \le y \le y_i + S, & \text{if } y_i - S < 0 \\ y_i - S \le y \le h - 1, & \text{if } y_i + S \ge h \\ y_i - S \le y \le y_i + S, & \text{otherwise} \end{cases} \tag{6}$$

The $x$ coordinate is different, and wrap texture address mode should be used. When $x_i - \frac{S}{\sin\phi} < 0$, which means the search region is close to the left boundary of the unrolled spherical image, the range of $x$ will become

$$x \in [0, x_i + \frac{S}{\sin\phi}] \cup [x_i - \frac{S}{\sin\phi} + w, w - 1], \tag{7}$$

and if $x_i + \frac{S}{\sin\phi} \ge w$, the range of $x$ will become

$$x \in [0, x_i + \frac{S}{\sin\phi} - w] \cup [x_i - \frac{S}{\sin\phi}, w - 1] \tag{8}$$

Figure 2 gives exemplar search regions when $x$ and $y$ coordinates fall outside valid ranges. The shapes of these regions agree with the fact that the polar region of the spherical images has more distortions than the central regions.

**Fig. 2**. Local search region (in red) when image coordinates fall outside valid range: the yellow points are superpixel centers. For this $512 \times 256$ spherical image, the search region size is set as 50 pixels for illustration purpose.

### 2.3. Distance Measure

When assigning pixel $p$ to clusters, the distance measure $D(C_i, p)$ is defined as

$$D(C_i, p) = \sqrt{(\frac{d_s}{N_s})^2 + (\frac{d_c}{N_c})^2}, \quad (9)$$

where $N_s$ and $N_c$ are maximum spatial and color distances of pixel to its cluster center after each iteration. The definition of color distance $d_c$ is the same as that of SLIC algorithm, and is given by the LAB Euclidian distance between the surrounding pixel and the superpixel $C_i$

$$d_c = \sqrt{(l - l_i)^2 + (a - a_i)^2 + (b - b_i)^2}. \quad (10)$$

The spatial distance $d_s$ is different. Because the cluster centers and all the pixels are located on the sphere, we use cosine dissimilarity to evaluate $d_s$, which is given by

$$d_s = 1 - \cos([X, Y, Z], [X_i, Y_i, Z_i]), \quad (11)$$

and we term this method as *Cos-SphSLIC*. This gives a hybird clustering algorithm, where color component is the standard k-means clustering and spatial component is the spherical k-means clustering [12]. The new cluster centers can be computed as the mean $[X\ Y\ Z\ l\ a\ b]$ vector of all the pixels belonging to the cluster, followed by a normalization procedure that makes $[X_i\ Y_i\ Z_i]$ have unit length.

Besides the cosine dissimilarity distance, we may choose $d_s$ to be the widely used spherical distance (or known as great circle distance), which is given by

$$d_s = \arccos(X \times X_i + Y \times Y_i + Z \times Z_i). \quad (12)$$

This method is termed as *Avg-SphSLIC*. Different from Cos-SphSLIC, the update step for the spatial part $[X_i\ Y_i\ Z_i]$ becomes finding a point on the sphere that the sum of spherical distance between this point and the clustered points is minimum. Unfortunately, there is no analytic form for this calculation. In this paper, we adopt an iterative spherical average algorithm [13] to compute the updated cluster center.

## 3. EXPERIMENTAL RESULTS

In this section, we first qualitatively compare our method with SLIC and efficient graph based segmentation (EGS) [6], then we give quantitative performance and running time of different methods. Finally, we give discussion about two variants of our spherical superpixel algorithm.

### 3.1. Qualitative Comparison

We select spherical images from SUN360 dataset [14] for qualitative comparison. For each image we generate different number of superpixels with various methods. Examples of superpixel segmentation produced by each method are shown in Figures 5 and 6. We can see that the superpixels generated by our method adapt well with the distortion of spherical images. That is to say the superpixels near the polar region appear to be larger than those near the central region. As shown in the zoom-in images, our method can get similar performance to SLIC with less superpixels for the polar region, and achieve better image boundary adherence than SLIC with more superpixels for the central region.

To show another advantage of our method, we map the superpixel segmentation result to the image sphere as shown in Figure 5 (d) and Figure 6 (d). The sphere is rotated so that we facing the vertical boundary of the spherical image and one pole of the sphere can be observed. We can see that there are noticeable seams between superpixels generated by planar algorithms, which is marked by red box. Another fact is that the superpixels of these methods near the polar region are stuck together. In contrast, superpixels generated by our method are more uniformly distributed on the sphere.

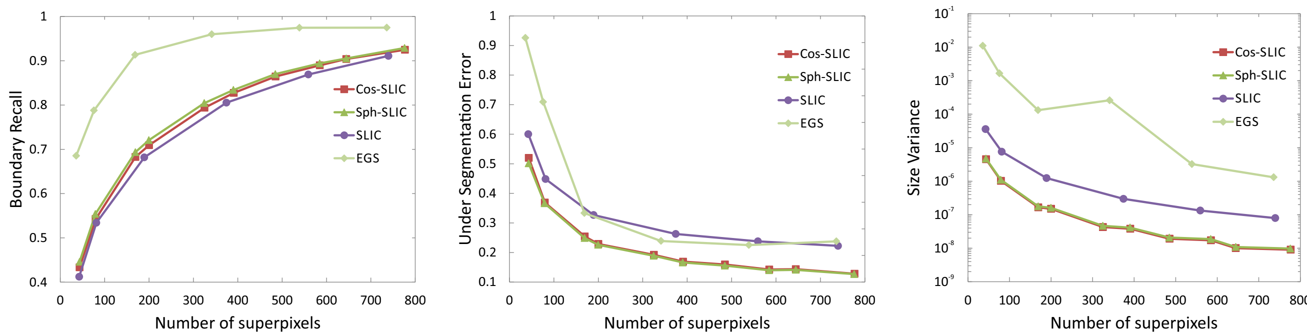
### 3.2. Quantitative Evaluation

The most commonly used benchmark to evaluate the performance of superpixel algorithms is Berkeley Segmentation Dataset [15]. It contains 300 natural images that have been segmented by different human subjects. For spherical images, although an annotated dataset [16] is used to compute semantic segmentation accuracy, it is inappropriate for superpixel algorithm evaluation as only a coarse 3D bounding box is marked for each object, which is not tightly aligned with image boundaries. In this paper, we leverage the *transformed* Berkeley dataset to evaluate different algorithms before a satisfactory spherical image segmentation benchmark is available. Specifically, we convert each planar image of Berkeley dataset to a spherical one, with the assumption that the planar images have 90° field of view. To simulate the image distortion of the spherical images, the view direction is set so that its intersection point with image sphere having spherical coordinate $(\frac{5\pi}{4}, \frac{\pi}{4})$. The segmentation and boundary ground truth of the original dataset are transformed accordingly. One exemplar transformed image and corresponding ground truth segmentation and boundary are shown in Figure 3.

The most important property of superpixels is adherence to image boundaries [5]. The standard metrics used to evaluate boundary adherence are boundary recall and under segmentation error. Boundary recall measures what fraction of

(a) Image (b) Ground truth segmentation (c) Ground truth boundary

**Fig. 3**. Image, segmentation and boundary examples of transformed Berkeley dataset (Note that only the meaningful part of the spherical image and the annotation is shown for the sake of space.)




(a) Boundary Recall (b) Under Segmentation Error (c) Size Variance (Log Scale)

**Fig. 4**. Quantitative performance of different methods on transformed Berkeley dataset

the ground truth edges fall within two pixels apart from one superpixel boundary. In our case, each edge pixel $(x, y)$ is given a weight $\sin(y\pi/h)$ to make this metric applicable to spherical images. The under segmentation error measures how many pixels from superpixels overlapping a ground truth segment leak across the boundaries. The area of each pixel on the sphere is used to account for the geometry of spherical images. These two measures for different methods are shown in Figures 4(a) and 4(b). In comparison, our method can obtain consistently better performance than SLIC, while EGS defines a predicate for boundaries and can achieve the best performance for the boundary recall. Our method has minimum under segmentation errors and EGS has worse performance than SLIC for this metric, which is the same as in the planar case. Note that the boundary recall of our method can be improved by increasing the weight of spatial distance, but this will also decrease the performance with respect to under-segmentation error.

Another important property of superpixels recently realized is structural regularities. It can be measured by size variation, which describes uniformity of superpixel size. In this paper, the superpixel size is computed as the ratio of superpixel area to the sphere area. Then size variation is defined as the variance of superpixel sizes. Figure 4(c) gives the superpixel size variation of different methods. Our method can generate more uniformly sized superpixels.

To compare the speed of various algorithms, we generate approximately the same number of superpixels on a computer with Intel 3.40GHz CPU and 8GB RAM. For $512 \times 256$, $1024 \times 512$ and $2048 \times 1024$ spherical images, about 800, 3200, 12800 superpixels are generated respectively using different algorithms and Table 1 gives the running time. Our two methods take more time to generate superpixels, mainly because of time-consuming trigonometric functions in the assignment and update steps.

**Table 1**. Running time comparison

| Method | 512×256 | 1024×512 | 2048×1024 |
|---|---|---|---|
| EGS | 0.2s | 0.8s | 3.4s |
| SLIC | 0.1s | 0.5s | 2.2s |
| Cos-SphSLIC | 0.7s | 2.8s | 11.3s |
| Avg-SphSLIC | 1.5s | 10.9s | 126.3s |

### 3.3. Discussion

From Figure 4, we can see that our two distance measures have almost the same performance. However Avg-SphSLIC needs an iterative procedure to adjust the cluster center and takes much more time than Cos-SphSLIC as shown in Table 1. Therefore, although the spherical distance is a more natural way to measure the distance between two points on the sphere surface, we always use Cos-SphSLIC in practice.

## 4. CONCLUSION

In this paper, we propose a superpixel generation algorithm for spherical images. Unlike previous works, our method ex-

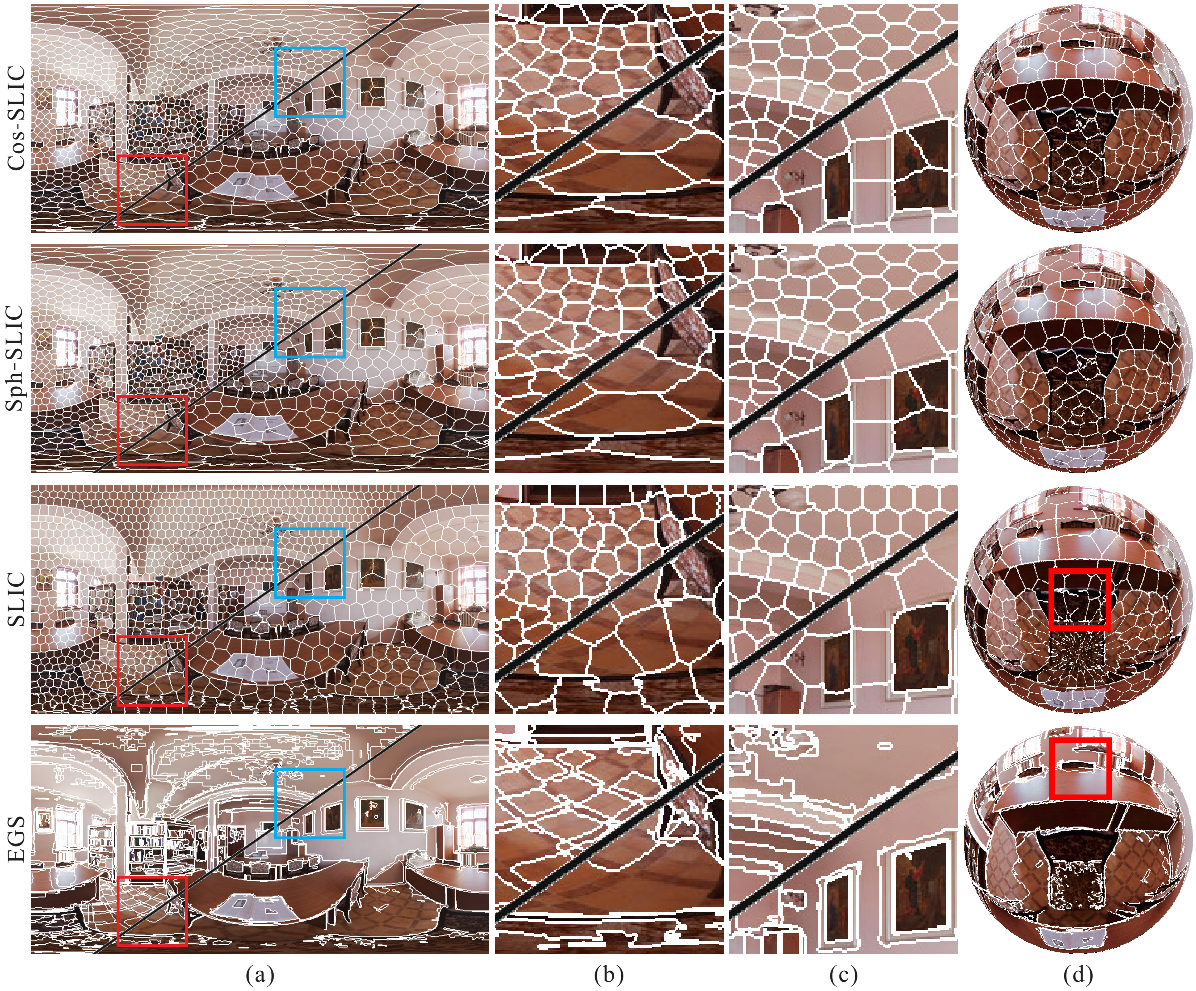



**Fig. 5**. Visual comparison of superpixels produced by various methods: (a) spherical image segmentation results of approximately 1860 and 500 superpixels, (b) and (c) zoom-in results, (d) results mapped to the sphere.

plicitly considers the geometry for the spherical images. In our approach, we use Hammersley points to initialize the superpixel centers and take cosine dissimilarity as the distance measure for the clustering. We also transform the widely used Berkeley segmentation dataset for quantitative evaluations. Experimental results show that our method can get better performance and can generate closed superpixel segments. In the future, we will be interested in investigating the application of spherical superpixels, such as image based rendering (IBR) involving spherical data and spherical images based 3D reconstruction. Another direction involves collecting a fine annotated spherical image dataset for quantitative evaluation of spherical superpixel generation algorithms.

## 5. REFERENCES

[1] J. Tighe and S. Lazebnik, “Superparsing: Scalable nonparametric image parsing with superpixels,” in *ECCV*, 2010, pp. 352–365.

[2] C. L. Zitnick and S. B. Kang, “Stereo for image-based rendering using image over-segmentation,” *IJCV*, vol. 75, no. 1, pp. 49–65, 2007.

[3] A. Vazquez-Reina, S. Avidan, H. Pfister, and E. Miller, “Multiple hypothesis video segmentation from superpixel flows,” in *ECCV*, 2010, pp. 268–281.

[4] B. Fulkerson, A. Vedaldi, and S. Soatto, “Class segmentation and object localization with superpixel neighborhoods,” in *ICCV*, 2009, pp. 670–677.

[5] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Süsstrunk, “SLIC superpixels compared to state-of-the-art superpixel methods,” *IEEE TPMAI*, vol. 34, no. 11, pp. 2274–2282, 2012.

[6] P. Felzenszwalb and D. Huttenlocher, “Efficient graph-based image segmentation,” *IJCV*, vol. 59, no. 2, pp. 167–181, 2004.

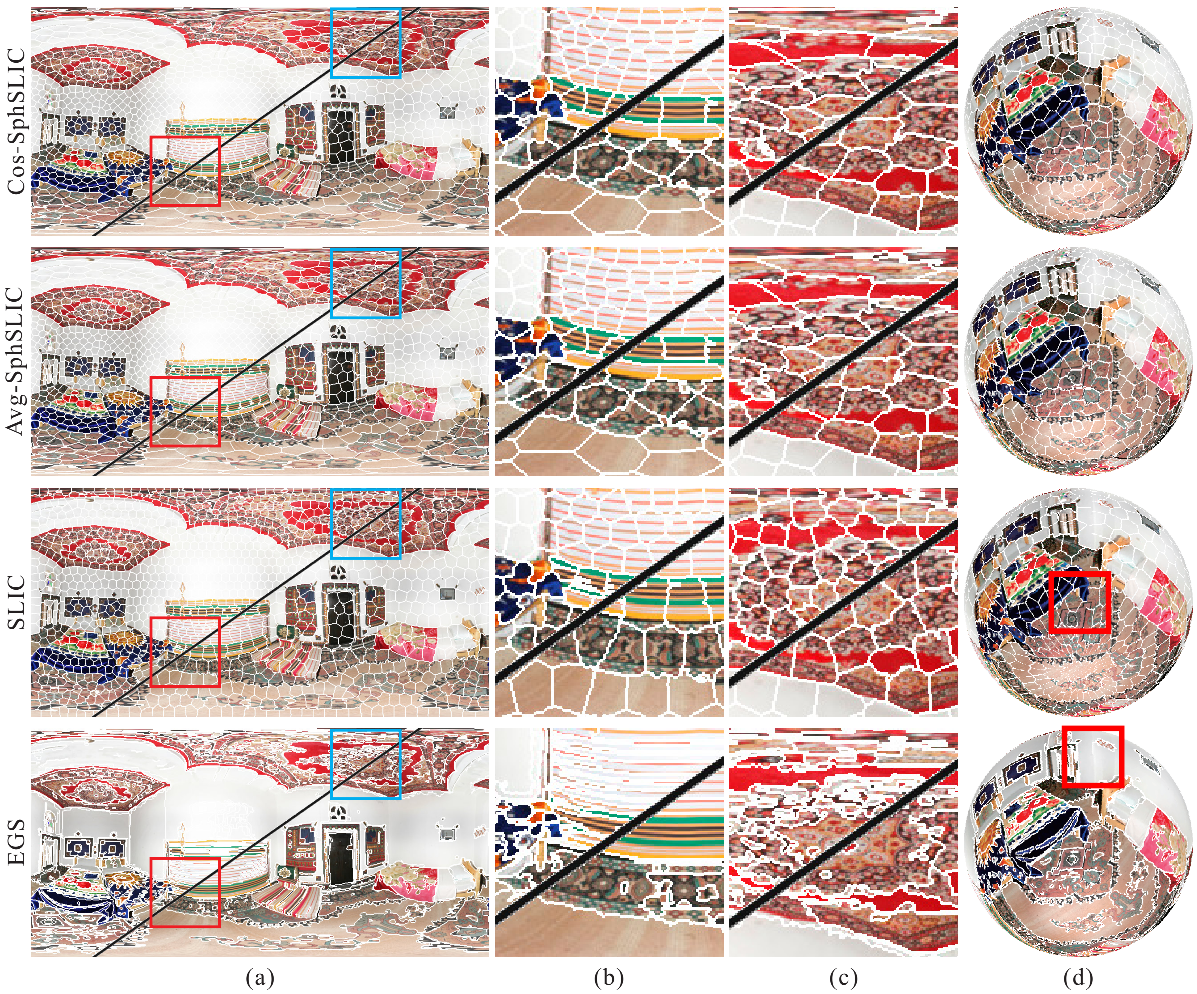



**Fig. 6**. Visual comparison of superpixels produced by various methods: (a) spherical image segmentation results of approximately 1718 and 786 superpixels, (b) and (c) zoom-in results, (d) results mapped to the sphere.

[7] R. Cabral and Y. Furukawa, “Piecewise planar and compact floorplan reconstruction from images,” in *CVPR*, 2014, pp. 628–635.

[8] K. Sakurada and T. Okatani, “Change detection from a street image pair using cnn features and superpixel segmentation,” in *BMVC*, 2015.

[9] B. Micusik and J. Kosecka, “Piecewise planar city 3d modeling from street view panoramic sequences,” in *CVPR*, 2009, pp. 2906–2912.

[10] J. J. Cui and W. Freeden, “Equidistribution on the sphere,” *SIAM Journal on Scientific Computing*, vol. 18, no. 2, pp. 595–609, 1997.

[11] T. T. Wong, W. S. Luk, and P. A. Heng, “Sampling with Hammersley and Halton points,” *Journal of graphics tools*, vol. 2, no. 2, pp. 9–24, 1997.

[12] K. Hornik, I. Feinerer, M. Kober, and C. Buchta, “Spherical k-means clustering,” *Journal of Statistical Software*, vol. 50, no. 1, pp. 1–22, 2012.

[13] S. R. Buss and J. P. Fillmore, “Spherical averages and applications to spherical splines and interpolation,” *ACM ToG*, vol. 20, no. 2, pp. 95–126, 2001.

[14] J. X. Xiao, K. A. Ehinger, A. Oliva, and A. Torralba, “Recognizing scene viewpoint using panoramic place representation,” in *CVPR*, 2012, pp. 2695–2702.

[15] D. Martin, C. Fowlkes, D. Tal, and J. Malik, “A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics,” in *ICCV*, 2001, pp. 416–423.

[16] Y. D. Zhang, S. R. Song, P. Tan, and J. X. Xiao, “Panocontext: A whole-room 3d context model for panoramic scene understanding,” in *ECCV*, 2014, pp. 668–686.