Spherical Superpixels: Benchmark and Evaluation

Liang Wan[1,2], Xiaorui Xu[1,2], Qiang Zhao[3] (✉), and Wei Feng[1,2]

[1] College of Intelligence and Computing, Tianjin University, China
{lwan,2016218015,wfeng}@tju.edu.cn
[2] Key Research Center for Surface Monitoring and Analysis of Cultural Relics (SMARC), SACH, China.
[3] Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China
zhaoqiang@ict.ac.cn

Abstract. Although a variety of superpixel algorithms have been developed and adopted as elementary tools in low-level computer vision and multimedia applications, most of them are designed for planar images. The quick growth of spherical panoramic images raises the urgent need of spherical superpixel algorithms and also a unifying benchmark of spherical image segmentation for the quantitative evaluation. In this paper, we present a general framework to establish spherical superpixel algorithms by extending planar counterparts, under which two spherical superpixel algorithms are developed. Furthermore, we propose the first segmentation benchmark of real-captured spherical images, which are manually annotated via a three-stage process.We use this benchmark to evaluate eight algorithms, including four spherical ones and the four corresponding planar ones, and discuss the results with respect to quantitative segmentation quality, runtime as well as visual quality.

Keywords: Superpixels · Spherical Image · Panorama · Benchmark · Evaluation.

1 Introduction

Superpixels group image pixels, which are perceptually similar in colors and close in space, to form visually meaningful entities. Since they heavily reduce the number of primitives to process, superpixels have gained great popularity as elementary tools in low-level computer vision and multimedia applications, such as image recognition [4], image parsing [24] [17], object localization [14], saliency detection [21] [12]. By today, a great number of superpixel algorithms have been developed for *planar* images [6][9][7][1][3]. As a key building block of many applications [8], the performance of superpixel algorithms may greatly affect the success of subsequent tasks. For the purpose of a fair performance comparison, several works [25] [1] have focused on evaluating planar superpixel algorithms quantitatively from multiple aspects, including the adhesion to object boundaries and shape regularity of superpixels. The evaluation is usually conducted on the BSD image segmentation dataset [2], which provides ground truth segmentations for multiple planar images.

Spherical superpixels, as a new type of superpixels, have received increasing attentions in recent years due to the quick growth of *spherical panoramas*, which cover 360 degree field of view and are widely used in recent applications [30]. The distinct difference between spherical superpixels from planar superpixels results from the fact that the underlying geometry of spherical images is a closed surface while that of planar images is an open domain. Directly applying planar superpixel algorithms on the unrolled spherical images or the transformed piecewise perspective images [16][4] would degrade the performance inevitably. Currently, there are only two spherical superpixel algorithms reported in the literature, namely spherical EGS (SphEGS) [28] and spherical SLIC (SphSLIC) [31]. Both extend the planar counterparts to the spherical domain.

Another dilemma that spherical superpixel algorithms are facing is the lack of spherical segmentation benchmark for the evaluation of their segmentation quality. To circumvent this situation, SphSLIC utilizes a transformed BSD segmentation dataset [31]. However, since the transformation from the planar to the spherical domain imposes assumptions on the field of view and the viewing direction for planar images, the transformed spherical images have large empty regions, and what's more, the evaluation is inevitably biased.

In this paper, we aim to bridge the planar superpixel algorithms with the spherical domain such that we can benefit from the rich development of planar superpixel algorithms, and obtain multiple spherical superpixel algorithms with as few costs as possible. Second, we plan to provide a high-quality segmentation dataset of real-captured spherical panoramas to fulfill the gap.

Specifically, inspired by the schemes of SphEGS [28] and SphSLIC [31], we first present a general framework (Section 3) on the establishment of spherical superpixel algorithms by extending planar counterparts, in which three main factors should be carefully concerned. Within the framework, we adapt two well-known planar superpixel algorithms, edge-augmented mean shift [6] and ETPS [29], to the spherical domain, and discuss the adaptions in detail. Next, we introduce the spherical segmentation dataset which is created by using an interactive image annotation tool (Section 4). With our tool, manual annotation for a spherical image can be conveniently obtained via a three-stage process. To our knowledge, this is the first benchmark for spherical image segmentation. Based on this dataset, we compare the performance of eight algorithms, including the two newly adapted ones, SphEGS [28], SphSLIC [31] and their planar counterparts, in terms of segmentation quality, runtime and visual quality (Section 5).

In the following, we first give a brief overview of the planar and spherical superpixel algorithms and existing superpixel evaluation works in Section 2. We then describe the general extension framework and the two new developed spherical superpixel algorithms in Section 3. Section 4 introduces the spherical segmentation dataset and the quantitative metrics, followed by experimental results reported in Section 5. Section 6 concludes the paper and gives the possible directions of future work.

2 Related Work

In this section, we introduce related works from three aspects: planar superpixel segmentation algorithms, current spherical superpixel algorithms and the research on superpixel evaluation.

2.1 Planar Superpixels

In the literature, there are a variety of superpixel algorithms developed for planar images. According to their high-level approaches, those algorithms can be classified into several categories [25] [1]. Density-based algorithms, e.g. edge-augmented mean shift [6], perform mode-seeking in the density image, where each pixel is assigned to the corresponding mode it falls into. Patch-based approaches, such as TPS [9], partition an image into superpixels by connecting seed points through pixel paths. Graph-based methods like normalized cut [23], graph based image segmentation [7] and entropy rate superpixels [16] treat the image as an undirected graph and cut the graph based on the edge weights. Clustering-based approaches borrow the idea from classic k-means algorithm. SLIC [1], preSLIC [20] and linear spectral clustering [15] are the well-known algorithms belonging to this category. Energy optimization-based methods, e.g. ETPS [29], SEEDS [3] and CCS [26], iteratively optimize a formulated energy. There are also contour evolution based algorithms, which represent superpixels as evolving contours starting from initial seed pixels [13]. By observing the shape of superpixels, the superpixel algorithms can also be classified into those generating regular primitives and those producing irregular primitives. For example, SLIC [1] and ETPS [29] can produce regular and compact superpixels, while the outputs of mean shift [6] and graph based image segmentation [7] are irregular.

2.2 Spherical Superpixels

To generate superpixels for spherical images, many works directly apply planar superpixel algorithms on the unrolled spherical images or the transformed piecewise perspective images [19][5]. Although this strategy works well for some applications, it would degrade the performance of the algorithms and produce superpixels with open contours. To deal with this problem, Yang and Zhang [28] presented a panoramic over-segmentation algorithm based on the graph based image segmentation method [7]. To make the original algorithm applicable to spherical images, additional neighborhood relationship is introduced when constructing the graph from the pixels of spherical images. This method gives irregular superpixels with closed contours. To generate more regular spherical superpixels, Zhao et al. [31] proposed an algorithm in the spirit of SLIC [1]. Their method explicitly considers the geometry for the spherical images and makes modifications to the initialization, assignment and update steps of SLIC algorithm. Based on these two works, we present the general framework that establishes spherical superpixel algorithms by extending planar ones.

2.3 Superpixel Evaluation

To access the performance of superpixel algorithms, some works rely on application-dependent evaluation. For example, Koniusz and Mikolajczyk [11] proposed a segmentation-based method for interest point extraction, and they compared the quality of interest points that are extracted with different superpixel algorithms. Li et al. [14] presented a fast object localization method based on superpixel grid and gave the localization accuracy for different superpixel algorithms. Superpixel algorithms have also been evaluated in the application of image parsing [24].

As an over-segmentation technique, superpixel generation is also usually evaluated on the standard planar image segmentation dataset, i.e. BSD500 [2]. Achanta et al. [1] gave the first categorization of superpixel algorithms by comparing popular superpixel algorithms for the adherence to image boundaries, speed and memory efficiency. The compactness and regularity of superpixels produced by different algorithms are compared in [22] and [10]. Stutz et al. [25] presented an extensive evaluation of 28 state-of-the-art superpixel algorithms on 5 image segmentation datasets regarding visual quality, segmentation performance, runtime, robustness to noise, blur and affine transformations.

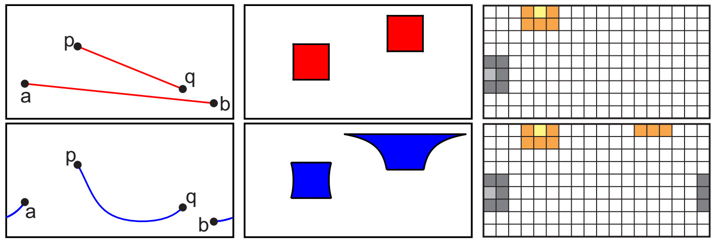
Zhao et al. [31] evaluated the performance of SphSLIC on a transformed BSD dataset, and presented the metrics for spherical superpixels. They assumed all the planar images in the BSD dataset have 90 degree field of view and the same viewing direction. Mapping the planar images to the spherical domain will lead to large empty regions in spherical panoramas. This indicates that the valid image information is defined on an open domain like a planar image. One real-captured spherical panorama, on the other hand, has image contents over the whole sphere, which means the image information is defined on a closed domain. Instead of relying on the transformed BSD dataset, we collect a new annotated segmentation dataset of real-captured spherical panoramas.

3 Spherical Superpixel Algorithms

Inspired by the schemes of SphEGS [28] and SphSLIC [31], we realize that one general way to establish spherical superpixel algorithms is to extend planar superpixel algorithms to make them applicable to spherical images. In this section, we first discuss the extension framework, and then give the extension details of two new spherical superpixel algorithms under this general framework.

3.1 Extension Framework

Our extension framework aims to bridge spherical images and planar superpixel algorithms. Note that the visual information of a spherical image is defined



(a) (b) (c)

Fig. 1. The three terms that should be considered when extending planar superpixel algorithms to spherical ones: (a) spatial distance measure, (b) neighborhood range and (c) boundaries problem. The 1st and 2nd rows are the planar and spherical cases respectively, when the equirectangular map is used to represent the sphere. See text for more discussions.

on a viewing sphere, but it has to be parameterized into 2D rectangular domain for storage. As we know, spherical parameterization schemes may suffer from geometric distortions, and different schemes may represent the sphere with a different number of 2D rectangular base faces, which always have discontinuity problem at patch boundaries. Hence the process of grouping pixels into superpixels needs to take into account of both the spherical geometry and the parametrization scheme, focusing on three main factors including: 1) spatial distance measure, 2) neighborhood range and 3) boundaries problem (Fig. 1).

In our current work, we choose equirectangular map to represent the spherical image due to three reasons. First, equirectangular map is the most common way used by camera vendors and research communities to represent and store spherical panoramas. Second, it is used by both SphEGS [28] and SphSLIC [31]. Third, it has relatively simple neighborhood range and boundaries problem as compared to other spherical parameterization methods.

Spatial Distance Measure. In superpixel algorithms, perceptually similar pixels which are also close in space are grouped together. To measure the spatial distance between two pixels, the planar superpixel algorithms always use 2D Euclidean distance on spatial coordinates of pixels. For the spherical panorama, we need to reply on spherical distances, which measure the length of the great cycle connecting the two pixels over the sphere.

As shown in Fig. 1(a), the shortest path between two pixels in an unrolled spherical image with 2D Euclidean distance is the line segment connecting them, while the shortest path becomes a curve when spherical distance measure is used. Another distance measure that considers the geometry for spherical panoramas is cosine dissimilarity [31]. Supposing that two arbitrary pixels on the panorama have spatial coordinates of $[X_1, Y_1, Z_1]$ and $[X_2, Y_2, Z_2]$, respectively, the cosine dissimilarity between them is computed as

$$D_s = 1 - \langle [X_1, Y_1, Z_1], [X_2, Y_2, Z_2] \rangle \,, \tag{1}$$

where $\langle \mathbf{a}, \mathbf{b} \rangle$ is the dot product of vector $\mathbf{a}$ and $\mathbf{b}$.

Neighborhood Range. Some density/clustering-based superpixel methods locate neighboring pixels or weight them in a local region. In planar algorithms, the neighborhood of a pixel is always defined as a square patch centered at that pixel. For spherical panoramas, different regions on the sphere have different levels of distortions. Thus, when defining the neighborhood for a pixel of spherical images, we should take the pixel position into account to compensate the distortions.

Let us take the equirectangular map for example. It parameterizes ray directions by zenith and azimuth angles, and then a uniform grid in the parametric domain oversamples the sphere near the poles. As shown in Fig. 1(b), the neighborhoods that have the same size on the sphere would have different shapes in the unrolled map [31].

Boundaries Problem. In the planar image (Fig. 1(c)), if a pixel p lies at the left (top) boundary, the neighboring pixels at the left (on the top) of it would be undefined. However, a sphere is a closed surface. When a sphere is parameterized into a set of 2D rectangular base faces, special attentions should be paid when we access the neighbors of a pixel lying near the boundaries of base faces [28].

In the equirectangular representation, the left and right boundaries correspond to the same meridian, while the top and bottom boundaries correspond to the north and south poles of a sphere respectively. Therefore, as shown in Fig. 1(c), if a pixel p is at the left boundary of the spherical image, the pixels at the right boundary with similar vertical coordinates are also neighbors. If p is at the top boundary, the pixels at the top boundary with horizonal coordinates as $x + \frac{w}{2}$ are also neighbors, where x is the horizonal coordinate of p and w is the width of the spherical image.

3.2 Two New Spherical Superpixel Algorithms

Under the above extension framework, we develop two new spherical superpixel algorithms by extending the planar mean shift [6] and ETPS [29]. These two methods are ranked as the fifth and the first best planar superpixel algorithms in a recent quantitative comparison [25].

Spherical Mean Shift (SphMS). Mean shift (MS) algorithm [6] is an iterative procedure for locating local maxima of a density function. It can be used to find modes in the joint spatial-color feature domain of an image. Pixels that converge to the same mode thus define the superpixels. The algorithm successively computes the mean shift vectors according to the kernel-weighted averages of pixel features, and translates the kernel windows. To develop the spherical mean shift, all the three terms in the extension framework should be considered. Specifically, we do the following modifications:

1. The original 2D spatial coordinates of a pixel in its feature representation are converted to a 3D unit vector, according to the equirectangular map computation. The cosine similarity (Eq. 1) is then adopted as the distance measure between a pixel and the mode's center.
2. Since the spherical panorama has different levels of distortions in the equirectangular map, the search window defined by the kernel should be modulated by the zenith angle.
3. When the mode's center goes near the boundaries of the equirectangular map, the search window is set to cross the boundaries.

It is also noted that the distance measure in MS is a combination of spatial distance and color distance. Since the color distance is kept intact, we need to weight the spherical spatial distance such that it has a similar magnitude as the planar spatial distance.

Spherical ETPS (SphETPS). Starting from an initial grid partitioning, ETPS [29] iteratively exchanges pixel blocks between neighboring superpixels, and uses a coarse-to-fine scheme. This algorithm encourages the color homogeneity, the shape regularization and the small boundary length within superpixels, and it enforces the superpixels to form a connected component. As the constraints of color coherence, boundary length and component connection can also work for spherical panoramas, we only need to consider the shape regularization and the boundaries problem. In more detail, three revisions are made to planar ETPS algorithm.

1. The shape regularization is formulated as the Euclidean distance between one pixel and the superpixel centroid. Given the equirectangular map, we map the 2D image coordinates of each pixel to the spherical domain, forming a 3D unit vector, and adopt the cosine dissimilarity (Eq. 1) as the spatial distance measurement.
2. To estimate the superpixel centroid over the sphere, we use an approximation scheme. We first calculate the centroid of pixels within a superpixel in 3D space, and then normalize it to a unit vector.
3. The equirectangular map has four boundaries. If a pixel block lies at the left boundary of the spherical image, the rightmost pixel block with the same vertical coordinate should be considered for potential exchange, and vice versa. If a pixel block lies at the top (bottom) boundary of the spherical image, the pixel block at the top (bottom) boundary with horizontal coordinate $x + \frac{w}{2}$ should be considered for potential exchange.

4 Dataset and Evaluation Metrics

In this section, we introduce the spherical segmentation dataset that we collected and the metrics used to evaluate the spherical superpixel algorithms.

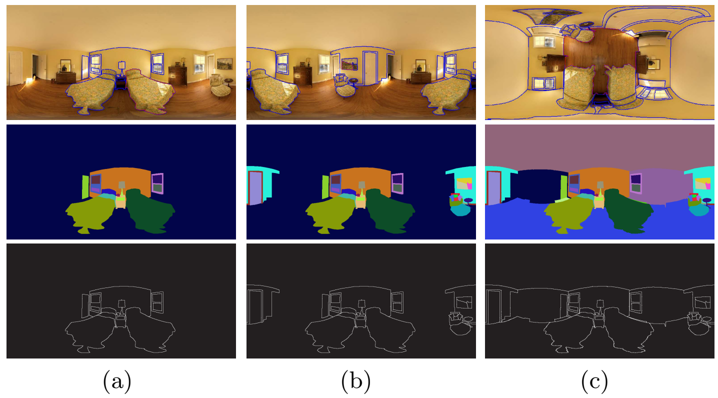


Fig. 2. Dataset preparation: manual annotation process (top), intermediate ground truth segmentation (middle) and boundaries (bottom). Our manual annotation process has three stages: (a) annotating the objects at the central part, (b) rotating the underlying sphere horizontally to annotate the objects that cross the left and right boundaries, (c) rotating the sphere vertically to annotate the objects that lie near the polar regions.

4.1 Dataset and Annotation Process

The authors of SphSLIC [31] utilized a transformed BSD segmentation dataset to evaluate the performance of spherical superpixel algorithms. Although this strategy is relatively easier than collecting a spherical segmentation dataset, it has some disadvantages. First, each spherical image in the transformed BSD dataset has only a limited meaningful region, and the remaining regions are all empty. This indicates that the transformation scheme can not emulate all the different levels of distortion in true spherical images. Second, when transforming the BSD dataset, all the planar images are assumed to have 90 degree field of view and the same viewing direction. Changing these two parameters will result in different transformed BSD dataset, and affect the evaluation performance. Therefore it is necessary to collect a dedicated dataset for spherical superpixel algorithm evaluation.

To construct the dataset, we choose 75 images from SUN360 spherical panorama database [27]. The selected images contain 50 indoor images and 25 outdoor images covering different categories. Each image is resized to 1024×512 pixels for manual annotation. The tool we used to annotate the spherical images is developed on the basis of the segmentation tool provided by Martin et al. [18], which is applied to acquire BSD image segmentation dataset. We made the modifications that allow the user to rotate the underlying image sphere in 3D space.

Our manual annotation process contains three basic stages, as illustrated in Fig. 2:

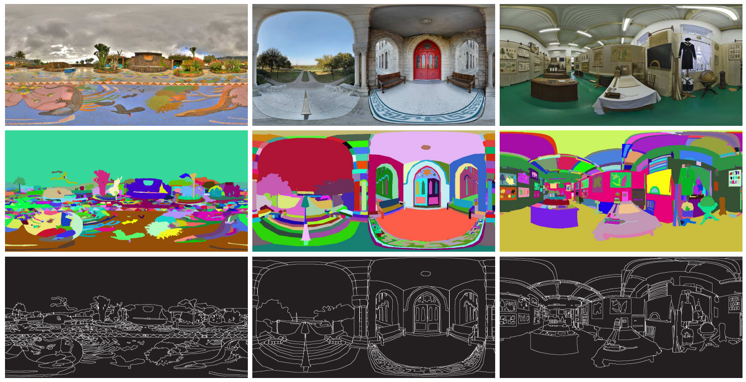


Fig. 3. Sample images and ground truth annotations from spherical segmentation dataset: (top) spherical images, (middle) ground truth segmentations and (bottom) ground truth boundaries.

1. Since the central region of a spherical panorama in the format of equirectangular map is least distorted, we first label the objects in this region (see Fig. 2(a)).
2. Next, we rotate the underlying sphere horizontally due to the fact that some objects may be split by the left and right boundaries of the image. To make this rotation operation simple and efficient, we do the horizontal rotation by 180 degrees such that the left and right boundaries of the original spherical image meet at the central region (see the top image in Fig. 2(b)). Then the two parts of split objects can be easily assigned to a common segment. The last two images in Fig. 2(b) show the possible segments for the input image.
3. Finally, the sphere is rotated vertically by 90 degrees. The objects at the polar regions of the original spherical image now appear at the central region, which can be conveniently annotated (the top image in Fig. 2(c)).

In the experiment, we trained three students to get familiar with the annotation tool, and each of them annotated 25 images. The final result is the spherical segmentation dataset [1]. The number of segments for each image ranges from 250 to 1085, with the average as 510. Some annotation examples are shown in Fig. 3.

4.2 Metrics

The most important properties of superpixels are adherence to image boundaries and structural regularity. In this paper, we use three standard metrics to evaluate boundary adherence, i.e. boundary recall, under segmentation error and achievable segmentation accuracy, and two metrics to evaluate structural regularity, i.e. compactness and size variation. The spherical metrics resemble their

[1] The annotated dataset are available at `http://scs.tju.edu.cn/~lwan/data/sspdataset/sspdataset.rar`.

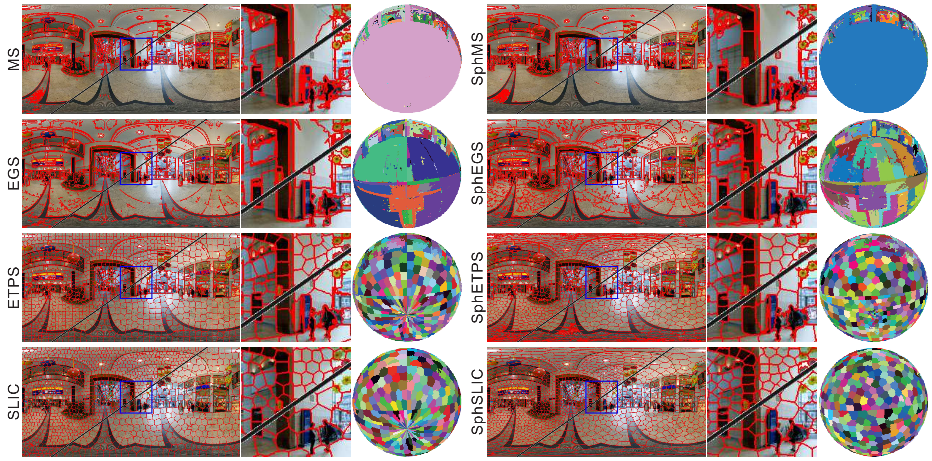


Fig. 4. Visual comparison of superpixels from different methods. For each method, we show the segmentation results with approximately 2000 and 670 superpixels, one zoom-in region, and the color-coded sphere for fewer superpxiels.

planar counterparts except that the pixel distortions are considered, usually by means of counting the solid angle of each pixel and the spherical distance. The detailed description of these metrics can be found in [31]. For the completeness, we give the brief introduction here.

Boundary recall measures the percentage of borders from the ground truth that coincide with those of the superpixels, given by

$$\frac{\sum_{p\in B(G)} A(p)I(\min_{q\in B(S)} d(p,q) < \varepsilon)}{A(B(G))}, \tag{2}$$

where $B(G)$ and $B(S)$ are the boundaries of ground truth $G = \{g_1, \cdots, g_m\}$ and superpixel segmentation $S = \{s_1, \cdots, s_n\}$, $d(p, q)$ is the spherical distance between p and q, $I(\cdot)$ is the indicator function and $A(\cdot)$ is the surface area of pixel or pixels set.

Under segmentation error evaluates how many pixels from superpixels leak across the boundary of ground truth segment. It is expressed as

$$\frac{\sum_i(\sum_{j|s_j\cap g_i\neq\emptyset} A(s_j)) - 4\pi}{4\pi}. \tag{3}$$

Achievable segmentation accuracy gives the best performance when taking superpixels as units for object segmentation. It is computed as

$$\frac{\sum_j \max_i A(s_j \cap g_i)}{4\pi}. \tag{4}$$

Compactness is a numerical quantity representing the degree to which a shape is compact. It is based on the isoperimetric quotient $Q(s_j) = \frac{4\pi A(s_j) - A^2(s_j)}{L^2(s_j)}$,

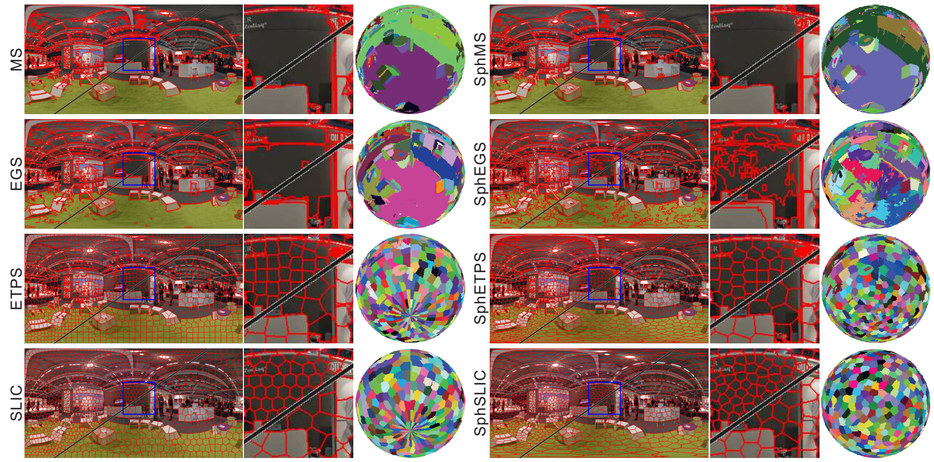


Fig. 5. Visual comparison of superpixels from different methods. For each method, we show the segmentation results with approximately 2000 and 670 superpixels, one zoom-in region, and the color-coded sphere for fewer superpxiels.

where $L(s_j)$ is the perimeter of superpixel s_j. Then the compactness is the weighted mean of isoperimetric quotient of each superpixel, where weights are the areas of the superpixels, i.e.

$$\frac{\sum_j Q(s_j)A(s_j)}{4\pi}. \tag{5}$$

Size variation describes the uniformity of superpixel size. It can be defined as the variance of superpixel areas

$$\frac{n\sum_j A^2(s_j) - 16\pi^2}{n^2}. \tag{6}$$

5 Experimental Results

In the experiments, we evaluate eight superpixel algorithms based on our spherical segmentation dataset, including the four spherical superpixel algorithms, i.e. SphMS, SphETPS, SphEGS [28], SphSLIC [31], and the four planar counterparts. In order to make a fair comparison, we have determined the optimal parameter values for each algorithm separately.

5.1 Visual Comparison

Fig. 4 and Fig.5 shows the segmentation results for two spherical panoramas at two different numbers of superpixels. We can see that EGS and SphEGS generally produce superpixels that are more tightly adhesive to object boundaries,

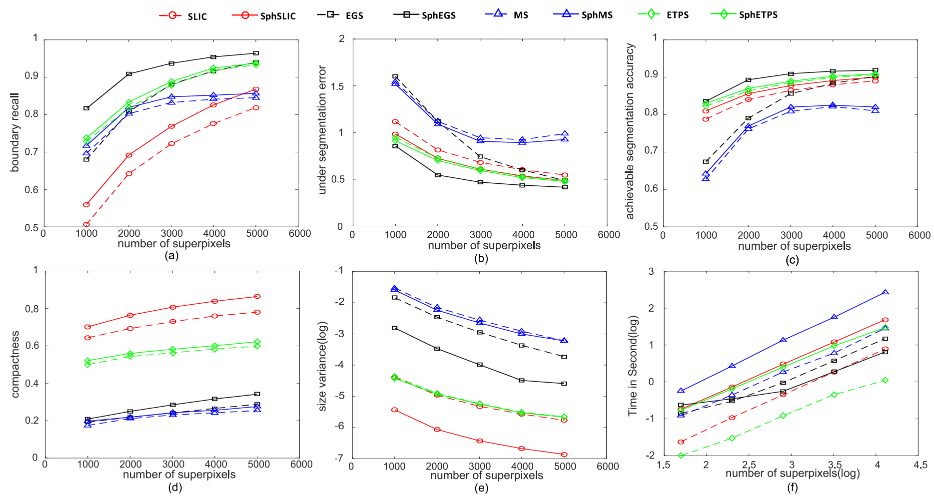


Fig. 6. The comparisons of different methods on the spherical segmentation dataset: (a) boundary recall; (b) under segmentation error; (c) achievable segmentation accuracy; (d) compactness; (e) size variation; and (f) running time.

while SphSLIC ans SphETPS generate spherical superpixels with most regular shapes. It is also obvious that for all the superpixel algorithms, the more superpixels are created, the more object boundaries can be captured. In comparison, SphEGS captures more object boundaries than the other spherical superpixel algorithms. As shown in the blowups in Fig. 4, SphEGS grasps the lights and the edges of walls accurately, while SphETPS partitions those lights into separate segments, which is better than SphSLIC.

By extending the adjacency relationship for boundary pixels, SphETPS, SphEGS, and SphSLIC overcome the discontinuity problem that can be obviously observed in the color-coded spheres for ETPS, EGS and SLIC. For better illustration, the equirectangular maps are rotated horizontally by 45 degrees. The left and right image boundaries now meet together as the vertical red lines in the right parts of panoramas for EGS, ETPS and SLIC. Their spherical versions, on the other hand, are not subjected to this problem.

To observe the shape regularity on the spherical domain, we use color-coded spheres to illustrate the segments. By adopting the spherical distance, SphETPS is able to assign less superpixels near the polar regions, while ETPS has many small superpixels crowded together in the polar regions. As for SphMS, it improves the performance with respect to MS. For example, in Fig. 4, SphMS captures some salient object boundaries in the southern polar region.

5.2 Quantitative Evaluation

Fig. 6 gives the performance of different algorithms in terms of boundary recall, under segmentation error, achievable segmentation accuracy, compactness and size variation. In comparison, we find that almost all the spherical superpixel algorithms get much better performance than their planar counterparts for the metrics under evaluation, while SphSLIC and SphEGS have much larger improvements than SphMS and SphETPS.

The boundary recall curves are plotted in Fig. 6(a). We can see that SphEGS achieves the best recall values for all the numbers of superpixels, followed by SphETPS and EGS, while SLIC reports the lowest values. For the under segmentation error as shown in Fig. 6(b), among the eight methods, SphEGS has the minimum under segmentation errors. SphETPS and SphSLIC achieve similar performance when the number of superpixels increases. In the term of the achievable segmentation accuracy, SphMS has the lowest values while SphETPS reports the second best performance (Fig. 6 (c)). We notice that the increase of the superpixel number has relatively small impact on SphMS and MS, as they quickly converge after using 2000 superpixels. This may be because they are inclined to reach local minimum in the clustering.

As for compactness and size variation, which contribute to the structural regularity, SphSLIC and SphETPS significantly outperform the other spherical superpixel algorithms as shown in Fig. 6(d) and 6(e), because they start from uniformly distributed seeds or grids and explicitly consider the geometry for spherical image and superpixel regularity. On the other hand, EGS, MS and their spherical extensions do not constrain the compactness, and hence they generate superpixels with irregular shapes and less compactness.

5.3 Running Time

We evaluate the timing performance on a computer installed with Intel(R) Core (TM) i5-4590 CPU @ 3.30GHz. Here, we use the authors' implementations for SphSLIC, SphEGS, and all the planar algorithms. Both SphMS and SphETPS are implemented by adapting the original codes of the planar versions. It is also noted that these algorithms cannot generate exactly the same number of superpixels as required, usually with a small variation. For a fair comparison, given each expected superpixel number, we run each algorithm for 10 times and count the average of the running times.

Fig. 6(f) plots the running time for different superpixel algorithms. For most algorithms, their running time increases linearly with respect to the number of superpixels. However, SphEGS even spends less time than EGS for a larger superpixel number. Among all the methods, ETPS achieves the fastest performance, SphMS is the slowest algorithm, and SphETPS spends less time than SphSLIC when the number of superpixels increases.

6 Conclusion

In this paper, we have presented a general framework for establishing spherical superpixel algorithms, in which three main factors are considered to extend planar counterparts. By applying this framework, two well-known planar superpixel algorithms, i.e. mean shift and ETPS, are adapted to the spherical domain. We then collected a high-quality segmentation dataset of spherical images, and evaluate the performance of eight different algorithms, including the two newly adapted, two existing spherical superpixel algorithms, and their planar counterparts, in terms of quantitative performance, visual quality and running time.

Currently, our dataset contains 75 spherical panoramas, one groundtruth segmentation for each panorama. In the future, we would like improve the quality of the dataset by collecting multiple groudtruth segmentations for each individual panorama. Another direction is adapting more planar superpixel algorithms to the spherical domain, and making a more thorough comparison of spherical superpixel algorithms.

Acknowledgments

This work is supported by National Natural Science Foundation of China (61572354, 61671325, 61702479).

References

1. Achanta, R., Shaji, A., Smith, K., Lucchi, A., Fua, P., Süsstrunk, S.: SLIC superpixels compared to state-of-the-art superpixel methods. IEEE TPAMI **34**(11), 2274–2282 (2012)
2. Arbelaez, P., Maire, M., Fowlkes, C., Malik, J.: Contour detection and hierarchical image segmentation. IEEE TPAMI **33**(5), 898–916 (2011)
3. Van den Bergh, M., Boix, X., Roig, G., Van Gool, L.: Seeds: Superpixels extracted via energy-driven sampling. IJCV **111**(3), 298–314 (2015)
4. Bu, S., Liu, Z., Han, J., Wu, J.: Superpixel segmentation based structural scene recognition. In: ACM MM. pp. 681–684 (2013)
5. Cabral, R., Furukawa, Y.: Piecewise planar and compact floorplan reconstruction from images. In: IEEE CVPR. pp. 628–635 (2014)
6. Comaniciu, D., Meer, P.: Mean shift: a robust approach toward feature space analysis. IEEE TPAMI **24**(5), 603–619 (2002)
7. Felzenszwalb, P., Huttenlocher, D.: Efficient graph-based image segmentation. IJCV **59**(2), 167–181 (2004)
8. Feng, W., Jia, J., Liu, Z.Q.: Self-validated labeling of markov random fields for image segmentation. IEEE TPAMI **32**(10), 1871–1887 (2010)
9. Fu, H., Cao, X., Tang, D., Han, Y., Xu, D.: Regularity preserved superpixels and supervoxels. IEEE TMM **16**(4), 1165–1175 (2014)
10. Giraud, R., Ta, V.T., Papadakis, N.: Robust shape regularity criteria for superpixel evaluation. In: IEEE ICIP (2017)
11. Koniusz, P., Mikolajczyk, K.: Segmentation based interest points and evaluation of unsupervised image segmentation methods. In: BMVC. pp. 1–11 (2009)
12. Lei, J., Wang, B., Fang, Y., Lin, W., Callet, P.L., Ling, N., Hou, C.: A universal framework for salient object detection. IEEE TMM **18**(9), 1783–1795 (2016)
13. Levinshtein, A., Stere, A., Kutulakos, K.N., Fleet, D.J., Dickinson, S.J., Siddiqi, K.: Turbopixels: Fast superpixels using geometric flows. IEEE TPAMI **31**(12), 2290–2297 (2009)

14. Li, L., Feng, W., Wan, L., Zhang, J.: Maximum cohesive grid of superpixels for fast object localization. In: IEEE CVPR. pp. 3174–3181 (2013)
15. Li, Z., Chen, J.: Superpixel segmentation using linear spectral clustering. In: IEEE CVPR. pp. 1356–1363 (2015)
16. Liu, M.Y., Tuzel, O., Ramalingam, S., Chellappa, R.: Entropy rate superpixel segmentation. In: IEEE CVPR. pp. 2097–2104 (2011)
17. Liu, S., Feng, J., Domokos, C., Xu, H., Huang, J., Hu, Z., Yan, S.: Fashion parsing with weak color-category labels. IEEE TMM **16**(1), 253–265 (2014)
18. Martin, D., Fowlkes, C., Tal, D., Malik, J.: A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In: IEEE ICCV. pp. 416–423 (2001)
19. Micusik, B., Kosecka, J.: Piecewise planar city 3d modeling from street view panoramic sequences. In: IEEE CVPR. pp. 2906–2912 (2009)
20. Neubert, P., Protzel, P.: Compact watershed and preemptive slic: on improving trade-offs of superpixel segmentation algorithms. In: IEEE ICPR. pp. 996–1001 (2014)
21. Ren, Z., Hu, Y., Chia, L.T., Rajan, D.: Improved saliency detection based on superpixel clustering and saliency propagation. In: ACM MM. pp. 1099–1102 (2010)
22. Schick, A., Fischer, M., Stiefelhagen, R.: An evaluation of the compactness of superpixels. PRL **43**, 71–80 (2014)
23. Shi, J., Malik, J.: Normalized cuts and image segmentation. IEEE TPAMI **22**(8), 888–905 (2000)
24. Strassburg, J., Grzeszick, R., Rothacker, L., Fink, G.A.: On the influence of superpixel methods for image parsing. In: VISAPP (2). pp. 518–527 (2015)
25. Stutz, D., Hermans, A., Leibe, B.: Superpixels: An evaluation of the state-of-the-art. CVIU **166**, 1–27 (2017)
26. Tasli, H.E., Cigla, C., Gevers, T., Alatan, A.A.: Super pixel extraction via convexity induced boundary adaptation. In: IEEE ICME. pp. 1–6 (2013)
27. Xiao, J., Ehinger, K.A., Oliva, A., Torralba, A.: Recognizing scene viewpoint using panoramic place representation. In: IEEE CVPR. pp. 2695–2702 (2012)
28. Yang, H., Zhang, H.: Efficient 3d room shape recovery from a single panorama. In: CVPR. pp. 5422–5430 (2016)
29. Yao, J., Boben, M., Fidler, S., Urtasun, R.: Real-time coarse-to-fine topologically preserving segmentation. In: IEEE CVPR. pp. 2947–2955 (2015)
30. Zhao, Q., Wan, L., Feng, W., Zhang, J., Wong, T.: Cube2video: Navigate between cubic panoramas in real-time. IEEE TMM **15**(8), 1745–1754 (2013)
31. Zhao, Q., Dai, F., Ma, Y., Wan, L., Zhang, J., Zhang, Y.: Spherical superpixel segmentation. IEEE TMM **20**(6), 1406–1417 (2018)