Deep Spherical Superpixels

Rémi Giraud, Michaël Clément

Deep Spherical Superpixels

Rémi Giraud[1] and Michaël Clément[2]

[1] Univ. Bordeaux, Bordeaux INP, IMS, CNRS UMR 5218, France.
remi.giraud@ims-bordeaux.fr
[2] Univ. Bordeaux, Bordeaux INP, LaBRI, CNRS UMR 5800, France.
michael.clement@labri.fr

Abstract. Over the years, the use of superpixel segmentation has become very popular in various applications, serving as a preprocessing step to reduce data size by adapting to the content of the image, regardless of its semantic content. While the superpixel segmentation of standard planar images, captured with a 90° field of view, has been extensively studied, there has been limited focus on dedicated methods to omnidirectional or spherical images, captured with a 360° field of view. In this study, we introduce the first deep learning-based superpixel segmentation approach tailored for omnidirectional images called DSS (for Deep Spherical Superpixels). Our methodology leverages on spherical CNN architectures and the differentiable K-means clustering paradigm for superpixels, to generate superpixels that follow the spherical geometry. Additionally, we propose to use data augmentation techniques specifically designed for 360° images, enabling our model to efficiently learn from a limited set of annotated omnidirectional data. Our extensive validation across two datasets demonstrates that taking into account the inherent circular geometry of such images into our framework improves the segmentation performance over traditional and deep learning-based superpixel methods. Our code is available online[3].

Keywords: Superpixels · Omnidirectional Images · Spherical CNN

1 Introduction

The vast majority of computer vision methods are tailored for standard RGB images, *i.e.*, captured with a standard 90° field of view (FoV). However, acquisition devices with wider FoV have become more and more popular in the recent years. In particular, omnidirectional images with a 360°×180° FoV are very interesting to capture the entire environment of a scene. Over the literature, such imagery may be equally referred as omnidirectional, spherical, 360°, or even panoramic. Naturally, such acquisition introduces distortions when projecting the capture on a planar 2D image. Nevertheless, many dedicated methods have been successfully applied on these images, for example for scene reconstruction [21], semantic segmentation [27] for autonomous driving, or in the context of mixed or virtual reality [18].

[3] https://github.com/rgiraud/dss

To efficiently apply deep learning-based architectures to these images, a few adjustments must be made to consider their specific geometry. For instance, the input images are horizontally circular so the pixels of the first column should be considered spatially adjacent to the pixels of the last column. Some methods explicitly take into account these geometrical properties, for instance with spherical convolutional neural networks (SCNNs) that have demonstrated higher performance on 360° images than standard CNNs [6]. Nevertheless, as for any deep learning-based method, a significant amount of annotated data is necessary for an efficient training, especially when tackling segmentation applications.

For regular standard images, various segmentation datasets are available with different content, resolution or precision in the annotations. However, only a few spherical image datasets are available, such as SUN360 [25] or Matterport3D [4]. Moreover, due to the tediousness of a pixel-wise semantic segmentation process, they generally only provide layout, depth or camera pose information [19]. In the context of autonomous driving, many datasets contain pixel-wise semantic annotations but the FoV is generally limited to standard rectangular acquisition [8,7], or the images are captured by a fisheye lense introducing other distortions [29]. Hence, deep learning segmentation methods that are applied to 360° imagery may highly necessitate specific data augmentation strategies [21,27].

In a more general context of image segmentation methods, non-semantic decompositions into superpixels offer numerous benefits. These methods regularly group pixels into homogeneous and connected regions, respecting the image contours. They have mainly been popularized by SLIC [1], a simple method that uses a locally constrained iterative K-means clustering, computed on color and position features. Then, many derived methods have been proposed, such as the non-iterative SNIC method [2], LSC [5] which expands the feature space of SLIC, or SCALP [11] that computes a color consistency along the path between a pixel and the centroid of its superpixel. Other methods like GMMSP [3] propose different strategies, such as using a Gaussian Mixture Model.

The first superpixel method tailored for spherical images was proposed in [22], extending SLIC. The spherical geometry is considered in the clustering distance, that is computed using the 3D positions of pixels on the sphere. The produced superpixels are regular on the 3D sphere domain and are able adapt to the distortions of objects induced by the projection on the 2D planar image, leading to higher segmentation performance compared to planar methods. Following, many planar superpixel algorithms have had their omnidirectional counterparts, such as SSNIC [20], SphLSC and SphSPS (or SphSCALP) [9].

Nevertheless, over the years, all these traditional approaches have started to report saturated performance over the segmentation benchmarks. With the Superpixel Sampling Network (SSN) method [12], a first deep learning framework has been proposed to compute a segmentation into superpixels. SSN and following methods, *e.g.*, [26], enable to improve the segmentation accuracy by computing more advanced features, with the use of a CNN trained on higher-level annotated segmentations (for example from semantic segmentations). However, these deep learning methods have only been designed for standard planar images.

Contributions In this work, we propose the first deep learning-based method called Deep Spherical Superpixels (DSS), able to segment omnidirectional images into spherical superpixels. The contributions of this work are listed as follows:

i We introduce the first deep learning-based superpixel segmentation method tailored for omnidirectional images, leveraging spherical CNN architectures and the differentiable K-means superpixel algorithm ;
ii We make use of specific data augmentation strategies designed for 360° images, whose effectiveness is demonstrated through an ablation study ;
iii We comprehensively evaluate the proposed method against state-of-the-art approaches, including both traditional planar and spherical approaches as well as deep learning-based methods, evaluated for the first time on the spherical superpixel segmentation task ;
iv We propose a quantitative validation on the Panorama Segmentation Dataset (PSD) [22], the reference for spherical superpixels, on initial and noisy images, and also on a newly considered omnidirectional road dataset, Wild PAnoramic Semantic Segmentation (WildPASS) [28] ;
v The source code of our method is made available to the research community.

2 Deep Spherical Superpixels Method

In this Section, we introduce our proposed Deep Spherical Superpixels (DSS) method. First, we present the Superpixel Sampling Network (SNN) [12] framework that we use as basis for our method (Section 2.1). Then, we detail the 360° coordinates system (Section 2.2) and our modifications of SSN to generate spherical superpixels (Section 2.3). Finally, we present the 360°-specific data augmentation used to enable our model to efficiently learn from a limited set of annotated omnidirectional data (Section 2.4).

2.1 Superpixel Sampling Network

In the superpixel segmentation literature, the Simple Linear Iterative Clustering (SLIC) algorithm is one the most simple yet accurate method [1]. It performs a locally constrained K-means clustering starting from a regular sampling grid. This clustering relies on a spatial and a color distance between each pixel and a superpixel centroid. Although SLIC is interesting for its rapidity and ease to use, its clustering accuracy can be limited since it is only based on RGB or Lab image features.

In [12], an end-to-end framework is proposed using a convolutional neural network (CNN) trained to learn how to provide more advanced features as input to a differentiable SLIC clustering algorithm. The network is trained to produce superpixels that are contained into higher-level annotated segmentations (for example from semantic segmentations). In particular, the integration of SLIC into a deep learning framework is possible in a differentiable manner by considering *soft* mappings of pixels to superpixels. At inference time, the final *hard* mapping,

associating a pixel to a unique superpixel, is only computed to generate the final segmentation.

The SSN model takes as input images of size $N = h \times w$, represented with 5 channels corresponding to *Lab* color features (3 channels) and xy pixel coordinates (2 channels). The goal of the model is to learn deep features that are more suitable to perform a differentiable clustering into superpixels. To achieve this, the SSN model uses a CNN composed of three blocks, each with two convolutional layers, batch normalization and ReLU activation, with a max pooling layer applied after each block. For the output, feature maps of each block are upsampled to the original image size (if necessary, for the second and third blocks) and concatenated. The original *Lab* and xy features are also concatenated into the output feature maps, resulting in D-dimensional pixel features (*i.e.,* 5 channels from input features and $D-5$ learned deep features). In practice, the SSN model used $D = 20$ in their experiments. For more details about this architecture, the reader can refer to [12].

These learned features are then fed to the aforementioned differentiable clustering to compute soft assignments of pixels to superpixels. These soft assignments are in turn used to compute a loss function tailored for the desired superpixel properties. For example, to obtain superpixels matching semantic segmented objects, the loss is comprised of two terms: (i) a pixel-wise cross-entropy term between ground-truth semantic segmentation and predicted soft superpixels and (ii) a compactness term which encourages superpixels to have low spatial variance.

The method is therefore end-to-end trainable and can learn deep pixel features tailored for subsequent superpixels properties. In the following, we present how to adapt this approach to the specific case of generating spherical superpixels for omnidirectional images.

2.2 Spherical Geometry

The projection system between the planar equirectangular 2D space and the 3D spherical space is depicted in Fig. 1. This relationship can be understood through the projection of vertical and horizontal coordinates of the plane onto the sphere's meridians and latitude circles. This process creates a spherical image where the width w is double the height h. It implies a horizontal continuity in the planar image domain that characterizes omnidirectional images. Hence, each pixel $p = [j, i]$ in the 2D space matches a 3D point $X = [x, y, z]$ on the unit sphere following the equations:

$$p = \begin{bmatrix} j = \lfloor \frac{\theta w}{2\pi} \rfloor \\ \\ i = \lfloor \frac{\phi h}{\pi} \rfloor \end{bmatrix} \quad \leftrightarrow \quad X = \begin{bmatrix} x = \sin(\frac{y\pi}{h})\cos(\frac{2x\pi}{w}) \\ y = \sin(\frac{y\pi}{h})\sin(\frac{2x\pi}{w}) \\ z = \cos(\frac{y\pi}{h}) \end{bmatrix}, \tag{1}$$

where $\theta = \arctan2(y, x)$ is the azimuthal angle, and $\phi = \arccos(z)$ is the polar angle. Note that this mapping of coordinates considers that $j \in [-\frac{w}{2}, \frac{w}{2}]$ so to map x to $[0, w]$, we have $x \leftarrow x + w$ when $x \leq 0$.

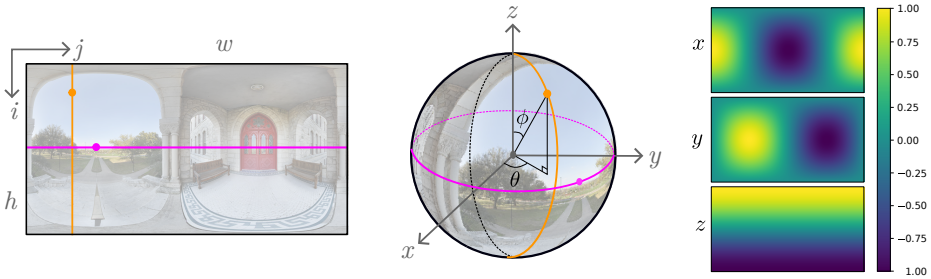


Fig. 1. 2D Planar and 3D spherical system coordinates. A pixel at position $[j, i]$ in the 2D space is mapped to a 3D point $[x, y, z]$ on the unit sphere following (1). This point can also be represented by its respective azimuthal and polar angles θ and ϕ.

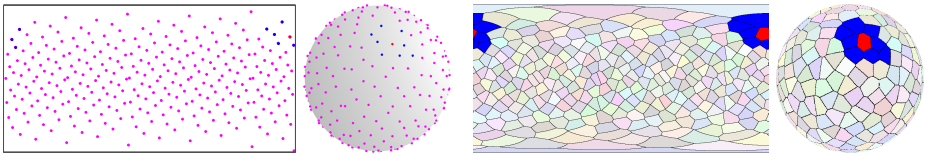
2.3 Spherical Superpixel Clustering Network

In this Section, we describe our adaptation of the K-means differentiable superpixel clustering network [12] to provide superpixels that are regular over the spherical domain. We use the same CNN architecture as basis for our method.

Features and superpixels initialization As input for the CNN, we use the *Lab* color features of the $N = h \times w$ pixels, denoted as $F_c \in \mathbb{R}^{N \times 3}$. The pixel coordinates are also given as input, but instead of the 2D pixel positions, we provide the 3D spherical coordinates $F_s \in [-1, 1]^{N \times 3}$. To match the coordinates domain, we normalize the Lab features F_c to also lie in $[-1, 1]$.

With classical 2D images, superpixel clusters are usually initialized by a regular sampling on the 2D grid. However, this strategy is not ideal with omnidirectional images as it does not respect the underlying 3D geometry. To overcome this issue, many spherical sampling strategies have been compared for superpixel clusters initialization [20,9]. In our proposed DSS method, as in [9], we use Hammersley sampling [24] to rapidly provide an appropriate set of K 3D points that are uniformly distributed on the unit sphere (see Fig. 2(a)). From this set of 3D points, we define an initial label map by a nearest neighbor computation on the 3D pixel position X (see Fig. 2(b)). Such spherically uniform sampling implies a sparser 2D sampling on the planar image near vertical borders. Classical planar methods that consider an initial regular grid would produce very irregular oversegmentation around the sphere's poles, as shown later in Section 3.3. From this label map, we extract the initial superpixel features with an average pooling.

Neighborhood-based distance In the original SLIC method [1], the K-means clustering is locally constrained so each superpixel can only aggregate a pixel in a fixed sized square window centered on the superpixel barycenter. For efficient implementation purposes, the K-means-based differentiable clustering of SSN [12] slightly differs by iteratively computing the pixel association within the 9-th superpixel neighborhood of the initialization map. Therefore, the core of the clustering distance computation is geometry-agnostic, once the superpixel



(a) Hammersley spherical sampling (b) Initial label map

Fig. 2. Spherical label map initialization. (a) A Hammersley sampling with $K = 300$ centroids points is computed on the unit sphere. Note the lower sampling density at the vertical borders, corresponding to the sphere's poles. (b) Corresponding label map, where each pixel is associated to the closest Hammersley barycenters, producing regular regions on the sphere. The 8 neighbors of the red superpixel (closest in the spherical space) are represented in blue.

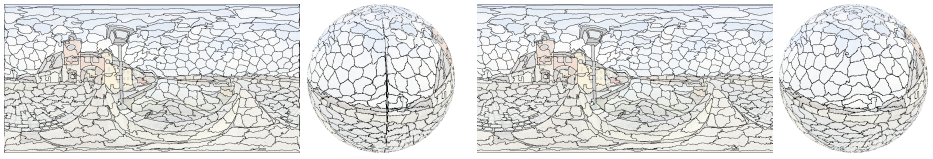
neighbors are identified. In our context, we can compute for each superpixel a n-th neighborhood with a nearest neighbors distance on their 3D barycenters in the spherical space. Such neighborhood is represented in Fig. 2.

Therefore, contrary to the planar square sampling, our method can define without ambiguity a $n \in [\![0, N]\!]$-th neighborhood. In practice, we use a $n = 9$ neighborhood, as in SSN.

Horizontally circular clustering 360° images are particularly characterized by their horizontally circular nature. This aspect is not considered in standard CNNs, which typically use zero padding strategies for convolutions and where the final receptive field may be also lower than the image dimension. In the context of spherical superpixel clustering, without any semantic aggregation of clustered regions, using standard convolutions is highly irrelevant since we would observe a discontinuity in the segmentation at the image borders.

For example, Fig. 3(a) shows the result obtained by using a standard zero padding strategy in the CNN layers. With 3×3 convolution kernels, the features extracted for pixels at $j = 0$ and $j = w - 1$ are not consistent with the ones of their neighborhood, which disrupts the selection of their closest superpixel among the 9 closest. When computing the hard clustering association, border pixels are generally associated to a disconnected region resulting in the appearance of an artificial vertical border in the spherical space, as for planar methods.

To take into account this horizontally circular geometry into our model, we propose to use a *spherical CNN* with a more natural circular padding strategy, as in [23,16]. Our spherical CNN uses a horizontal circular (or periodic) padding of half size of the kernel at each step requiring padding (convolutional or max pooling layers). A replicate padding strategy is used for vertical padding. Hence, the spherical CNN is fully able to preserve the 360° geometry in the final clustering and to compute relevant features at the borders. Note that other strategies may be possible, such as applying a large input circular padding as a preprocessing [17], but with many successive convolutions, this leads to handle significantly larger images, and thus to higher memory and time consumption.



(a) With zero padding (b) With circular padding

Fig. 3. Impact of the circular padding on the superpixel segmentation. (a) With standard zero padding, the CNN features of pixels at $j = 0$ and $j = w-1$ are not consistent with neighborhood, leading to a vertical border in the spherical space, as for planar methods. (b) With circular padding, the features remain consistent on the borders and the method is able to fully consider the geometry of the omnidirectional images.

Loss function Deep pixel features from our spherical CNN are fed to the differentiable clustering method to produce soft assignments $\mathcal{S}_{\text{soft}}$ of spherical superpixels. As in SSN, the model is trained with a loss comprised of a pixel-wise cross-entropy with ground-truth segmentation $\mathcal{G}$ denoted L_{seg}, and a compactness term L_{compact} to enforce superpixels with low spatial variance:

$$L = L_{\text{seg}}(\mathcal{G}, \mathcal{S}_{\text{soft}}) + \lambda L_{\text{compact}}(F_s, \mathcal{S}_{\text{soft}}). \tag{2}$$

Region connectivity After training, to compute the final superpixel segmentation of an image, a last step ensures the connectivity of the produced regions as for most superpixel clustering methods [1,12]. This is simply done by aggregating the smallest disconnected regions to the largest and nearest one but taking into account the circular aspect.

2.4 360°-Specific Data Augmentation

In the context of 360° imagery, the lack of extensive image datasets with segmentations makes it hard to train neural networks efficiently. To mitigate this data limitation, the use of data augmentation strategies is crucial. While simple augmentation techniques such as flips, blurs, and noise addition are applicable, they may be insufficient to provide enough diversity to the training process. However, many other conventional data augmentation strategies may alter the intrinsic 360° geometry and should not be used for such images. For instance, rotations or crops, as used in SSN [12], compromise the spherical geometry, leading to the loss of the horizontal mirror effect and the spatial distortion of the 2D label map. Using such augmentation techniques would lead the model to learn to provide irregular superpixels in the spherical space with artificial vertical borders at the edges, as for planar methods (see Fig. 3(a)).

To overcome these challenges, we propose to use data augmentation techniques tailored explicitly for 360° images. A straightforward augmentation technique would consist in horizontally rolling the 360° image and its ground-truth [14]. As stated in [16], such data augmentation strategy does not bring any

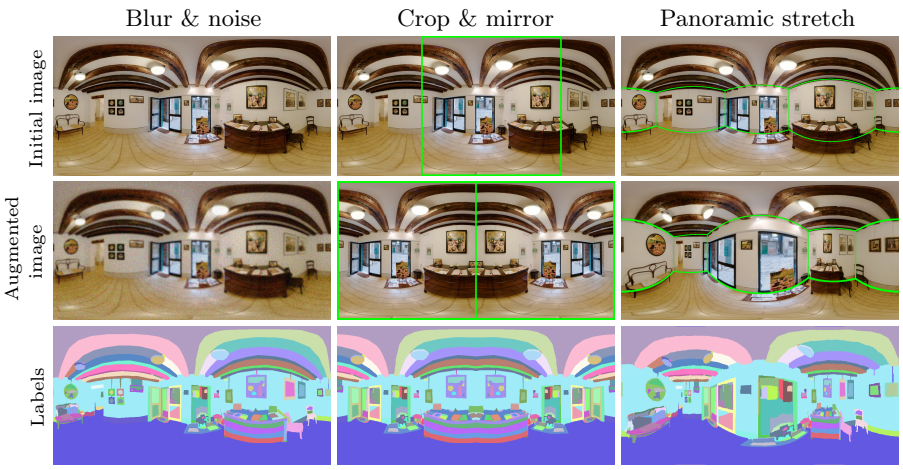


Fig. 4. Example of data augmentation used during training. **Left:** Standard Gaussian blur and noise (here with respective maximal variance $\sigma = 20$ and $\sigma = 2$). The ground-truth labels are not impacted. **Middle:** Crop & mirror strategy. A random crop of half-width is selected (represented by the green square) and mirrored to form a new 360° image. This method combines horizontal rolling, flipping and also creates information at the mirror border. **Right:** Panoramic stretch [21] to introduce distortions in the 360° image (here with parameters $k_x = 0.5$, $k_y = 1.25$ that correspond to a respective enlargement and a shrinking of the areas where $|x| \approx 1$ and $|y| \approx 1$). The layout of the scene is represented by the green lines to more easily apprehend the distortion.

diversity in a pure CNN network. Nevertheless, in our context, since average superpixel features are extracted according to an spherical initialization label map, a roll of the image may have a different impact on the produced segmentation. To go further, we also propose to combine random half-width cropping and horizontal mirroring of the input image and ground-truth (see Fig. 4(middle)). This way, in a single transformation, we combine rolling and flipping while creating information at the mirror border.

Finally, we use the *panoramic stretch* approach of [21] to introduce spatial distortions. To stretch a 360° image, $[x, y, z]$ coordinates are simply multiplied by a respective factor $[k_x, k_y, k_z]$ and projected back to the sphere. Pixel values are then computed using bilinear interpolation. Since setting k_z would affect the projection of x and y values the same way, authors propose to only set k_x and k_y parameters. The 3D coordinates maps in Fig. 1 represent the image area that would be affected by increasing one of the parameters. For instance, setting $k_y < 1$ would zoom on the region where y values are close to -1 and 1 (see Fig. 4(right)). We refer the reader to [21] to more details on the stretching algorithm and to our supp. mat. for additional examples.

With such data augmentation, we are able to greatly enrich the training dataset while preserving the spherical geometry of 360° images. We demonstrate the improvement of performance obtained using these techniques during training in Section 3.2.

3 Results

3.1 Validation Framework

Datasets In our experiments, we considered two relevant spherical segmentation datasets containing various accurately segmented objects (see examples in supp. mat.). The first dataset called Panorama Segmentation Dataset (PSD) [22] is the reference one and contains 75 images of 512×1024 pixels from the SUN360 dataset [25]. The ground-truth manual segmentations from [22], contain an average number of 510 objects with an average size of 1334 pixels. To fairly compare deep learning methods, we respectively consider 55, 5 and 15 images for the train, validation and test sets. In Section 3.3, we also compare the performance on PSD images affected by an additive white Gaussian noise of variance 20.

To further demonstrate the performance of DSS, we choose to consider for the first time in spherical superpixel methods evaluation, the Wild PAnoramic Semantic Segmentation (WildPASS) dataset [28], containing 500 omnidirectional natural road images. We resize the images to 512×1024 and split the dataset into respectively 300, 100 and 100 images for train, validation and test sets.

Parameter settings Our data augmentation is applied on-the-fly during training. It includes (i) applying a random Gaussian blur with a variance $\sigma \in [0, 2]$, (ii) adding Gaussian noise of variance $\sigma \in [0, 20]$, (iii) random flipping, horizontal rolling and half-width random crop and mirror with a 0.5 probability, and (iv) panoramic stretching with random parameters k_x and k_y between 0.5 and 2. During training, $\lambda = 1$ in (2) and images are downsized to 256×512 pixels, so our model can understand the whole scene's geometry, contrary to the 201×201 crops used in [12]. We refer the reader to the supp. mat. for training details.

Evaluation metrics The main challenge in superpixel segmentation is the ability to produce superpixels that are contained into the image objects, with respect to a ground-truth segmentation. Regularity is also an important aspect to interactive applications or to later extract significant neighborhoods [10]. Since these criteria are generally contradictory, efficiently maximizing both is usually the bottleneck of superpixel methods. These aspects can be relevantly evaluated with state-of-the-art dedicated metrics [10]. In the following, we denote superpixel segmentation as $\mathcal{S} = \{S_i\}$ and ground-truth segmentation as $\mathcal{G} = \{G_j\}$ with their respective borders $\mathcal{B}(\mathcal{S})$ and $\mathcal{B}(\mathcal{G})$.

The mainly reported measure is the segmentation accuracy, with the Achievable Segmentation Accuracy (ASA) [13] such that:

$$\mathrm{ASA}(\mathcal{S}, \mathcal{G}) = \frac{1}{\sum\limits_{S_i \in \mathcal{S}} |S_i|} \sum_{S_i} \max_{G_j \in \mathcal{G}} |S_i \cap G_j|. \tag{3}$$

This aspect can also be evaluated by focusing on the contour adherence of superpixels to the object borders, using the Boundary-Recall (BR) such that:

Table 1. Ablation study of the proposed DSS method on PSD and noisy PSD images on ASA (↑), CD/BR (↓) and GGR (↑). CD is given for BR=0.8. Best and second best results are respectively in bold and underlined font.

Data augmentation				PSD			Noisy PSD		
Gaussian blur&noise	Horizontal crop&mirror	Panoramic strecth	Circular padding	ASA	CD/BR	GGR	ASA	CD/BR	GGR
-	-	-	✓	0.862	0.134	0.385	0.858	0.139	0.386
✓	-	-	✓	0.877	<u>0.119</u>	**0.444**	0.868	0.132	**0.461**
✓	✓	-	✓	<u>0.888</u>	**0.117**	<u>0.413</u>	0.883	**0.124**	<u>0.423</u>
✓	✓	✓	-	0.887	0.124	0.387	<u>0.884</u>	0.134	0.390
✓	✓	✓	✓	**0.890**	0.122	0.388	**0.886**	<u>0.132</u>	0.392

$$\mathrm{BR}(\mathcal{S},\mathcal{G}) = \frac{1}{|\mathcal{B}(\mathcal{G})|} \sum_{p\in\mathcal{B}(\mathcal{G})} \delta[\min_{q\in\mathcal{B}(\mathcal{S})} \|p-q\| < \epsilon], \tag{4}$$

with ϵ a distance threshold set to 2 pixels [10], and $\delta[a] = 1$ when a is true and 0 otherwise. Since it only measures recall, BR should be compared to the Contour Density (CD), *i.e.*, the proportion of border pixels of the generated superpixels.

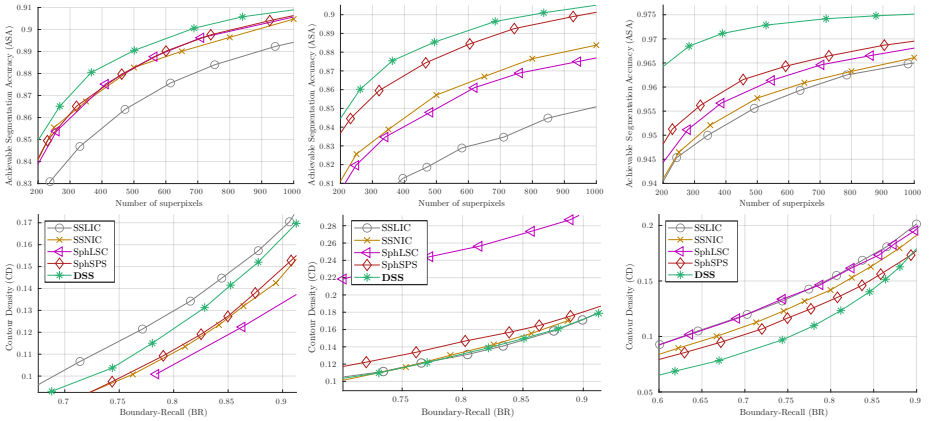
Finally, to evaluate the regularity aspect, we use the Generalized Global Regularity (GGR) metric that adapts the metric proposed in [10] to 360° images [9]. This metric evaluates the convexity, balanced pixel distribution, contour smoothness of each shape and also how homogeneous the shape distribution is within the segmentation. We refer the reader to [9] to more details on the GGR metric.

3.2 Ablation Study

In Table 1, we report the impact of each data augmentation strategy and the spherical CNN architecture, *i.e.*, using circular padding instead of zero padding [12] on the PSD and noisy PSD images for an average number of $K = 500$ superpixels. Each augmentation strategy increases the training efficiency in terms of segmentation accuracy, while the circular padding logically improves the spherical regularity by cancelling the artificial horizontal border of the segmentation. This confirms the interest of improving the original SSN method with spherical CNN architecture and specific augmentation strategies for 360° images.

3.3 Comparison to State-of-the-Art Methods

Compared methods In our experiments, we compare DSS to the spherical methods: SSLIC [30], SSNIC [20], SphLSC and SphSPS [9]. We also compare to some recent planar methods: LSC [5], SNIC [2], GMMSP [3], and SSN [12] All methods are used with the default regularity parameters. For the SSLIC method [30], that does not have one, we use a color weight of 20 to try to optimize its segmentation accuracy. For SSN [12], we compare to both the initial network trained on the BSD dataset [15] containing planar natural images (SSN-BSD) and to a retrained network on the targeted dataset (SSN-PSD, SSN-WP).



(a) PSD images (b) Noisy PSD images (c) WP images

Fig. 5. Comparison of DSS to state-of-the-art methods. **Top:** Segmentation accuracy evaluated with ASA (3). **Bottom:** Contour adherence in terms of CD vs BR (4).

Evaluation of performance We compare the proposed DSS to the spherical methods in terms of segmentation accuracy (ASA) and also contour adherence (CD/BR) for several superpixel numbers required K, on the PSD images Fig. 5(a), noisy PSD images Fig. 5(b) and WP images Fig. 5(c).

We observe that DSS obtains the highest segmentation accuracy (ASA) on all type of images. We can also see that our method is robust to noise contrary to most state-of-the-art methods that present a significant loss of performance on such slightly altered images. Finally, we can note that DSS superpixels also have the highest contour adherence (lowest CD/BR) compared to other methods, only except on noise-free PSD images. This can be simply explained by the fact that our method, as SSN, does not explicitly integrate a contour adherence loss and that the ground-truth segmentations in the PSD dataset contain many annotations of very thin objects that impact such metric.

In Table 2, we report results for $K = 500$, also including the regularity metric (GGR), and performance obtained with planar methods. We observe that GGR discriminates well the planar and spherical methods. DSS is among the spherical methods, having higher spherical regularity than planar methods, and it also preserves its regularity in the presence of noise.

Compared to SSN, we can first notice that SSN trained on the BSD does not generalize very well when applied on PSD or WP images. It demonstrates the capacity of CNNs to extract semantic information and that performance of generalization may highly depend on the similarity of annotations. We also observe that DSS slightly outperforms SSN retrained on the PSD and WP datasets, in terms of segmentation accuracy. SSN is able to train its network by providing image crops, which is a much more efficient learning strategy than to provide the whole image, as we have to do in DSS. Nevertheless, with our data augmentation strategy, we can maintain the same level of accuracy while generating spherical superpixels that may follow the deformed objects.

Table 2. Quantitative comparison of DSS to state-of-the-art methods for an average number of $K = 500$ superpixels on ASA (↑), CD/BR (↓) and GGR (↑). CD is given for BR=0.8. Best and second best results are respectively in bold and underlined font.

		PSD			Noisy PSD			WP		
		ASA	CD/BR	GGR	ASA	CD/BR	GGR	ASA	CD/BR	GGR
Planar	LSC [5]	0.877	0.138	0.347	0.844	0.303	0.334	0.962	0.153	0.313
	SNIC [2]	0.864	0.129	0.361	0.852	0.139	0.357	0.958	0.146	0.322
	GMMSP [3]	0.877	0.136	0.339	0.849	0.329	0.328	0.963	0.157	0.306
	SSN-BSD [12]	0.879	0.119	0.328	0.863	0.147	0.321	0.967	0.134	0.296
	SSN-PSD/WP [12]	<u>0.887</u>	0.114	0.334	0.873	0.141	0.328	<u>0.972</u>	<u>0.120</u>	0.303
Spherical	SSLIC [30]	0.866	0.130	0.421	0.821	**0.130**	0.383	0.956	0.152	0.399
	SSNIC [20]	0.883	<u>0.110</u>	**0.462**	0.857	0.134	**0.399**	0.958	0.142	<u>0.410</u>
	SphLSC [9]	0.882	**0.105**	0.397	0.850	0.252	0.357	0.960	0.152	0.360
	SphSPS [9]	0.883	0.112	<u>0.452</u>	<u>0.877</u>	0.146	0.389	0.962	0.133	**0.411**
	DSS	**0.890**	0.122	0.388	**0.886**	<u>0.132</u>	<u>0.392</u>	**0.973**	**0.118**	0.356

Finally, qualitative results are respectively shown on PSD, noisy PSD and WP images in Fig. 6, 7, 8. For planar methods, we can note the projection irregularity around the sphere's poles. DSS produces spherically regular superpixels that well capture the image objects.

4 Conclusion

In this work, we proposed DSS, the first deep learning-based spherical superpixel segmentation method. The proposed approach leverages on spherical CNN architectures dedicated to omnidirectional images having a circular geometry. We demonstrated that combining a deep learning strategy that respects the spherical geometry along with appropriate data augmentation enables to achieve higher and more robust segmentation performance than both traditional and deep learning-based methods.

We firmly believe that the presented work holds significant value for the community, given the importance of achieving both accurate segmentation and high regularity in the acquisition space, here spherical, for an effective display and processing of adjacent relationships in computer vision preprocessing tasks.

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 Transactions on Pattern Analysis and Machine Intelligence **34**, 2274–2282 (2012)
2. Achanta, R., Süsstrunk, S.: Superpixels and polygons using simple non-iterative clustering. In: Conference on Computer Vision and Pattern Recognition (2017)
3. Ban, Z., Liu, J., Cao, L.: Superpixel segmentation using gaussian mixture model. IEEE Transactions on Image Processing **27**(8), 4105–4117 (2018)
4. Chang, A., Dai, A., Funkhouser, T., Halber, M., Niessner, M., Savva, M., Song, S., Zeng, A., Zhang, Y.: Matterport3D: Learning from RGB-D data in indoor environments. In: International Conference on 3D Vision (2017)
5. Chen, J., Li, Z., Huang, B.: Linear spectral clustering superpixel. IEEE Transactions on Image Processing **26**, 3317–3330 (2017)

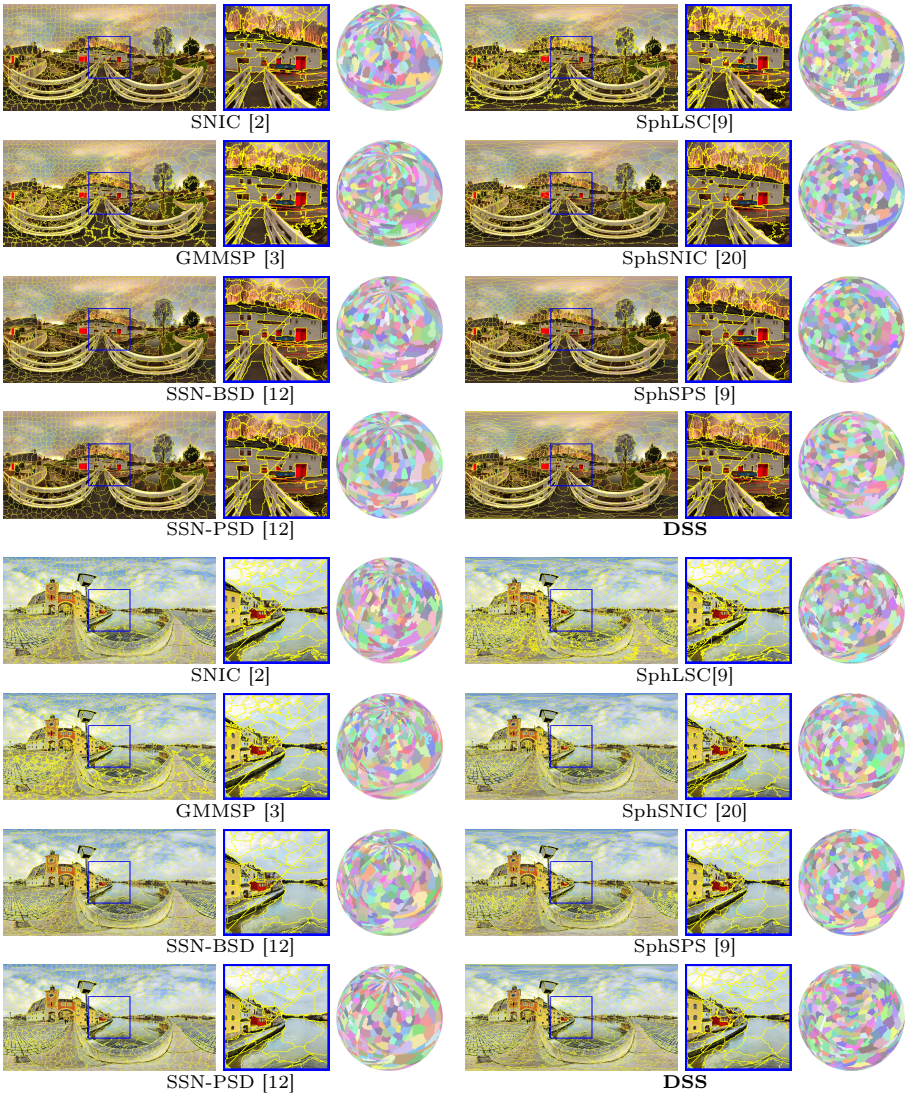


Fig. 6. Qualitative comparison on PSD images, for planar (left) and spherical methods (right) for two superpixel numbers $K = 1200$ (top-left) and $K = 400$ (bottom right).

6. Cohen, T.S., Geiger, M., Köhler, J., Welling, M.: Spherical CNNs. In: International Conference on Learning Representations (2018)
7. Cordts, M., Omran, M., Ramos, S., Rehfeld, T., Enzweiler, M., Benenson, R., Franke, U., Roth, S., Schiele, B.: The Cityscapes dataset for semantic urban scene understanding. In: Conference on Computer Vision and Pattern Recognition (2016)
8. Geiger, A., Lenz, P., Urtasun, R.: Are we ready for Autonomous Driving? The KITTI Vision Benchmark Suite. In: Conference on Computer Vision and Pattern Recognition (2012)

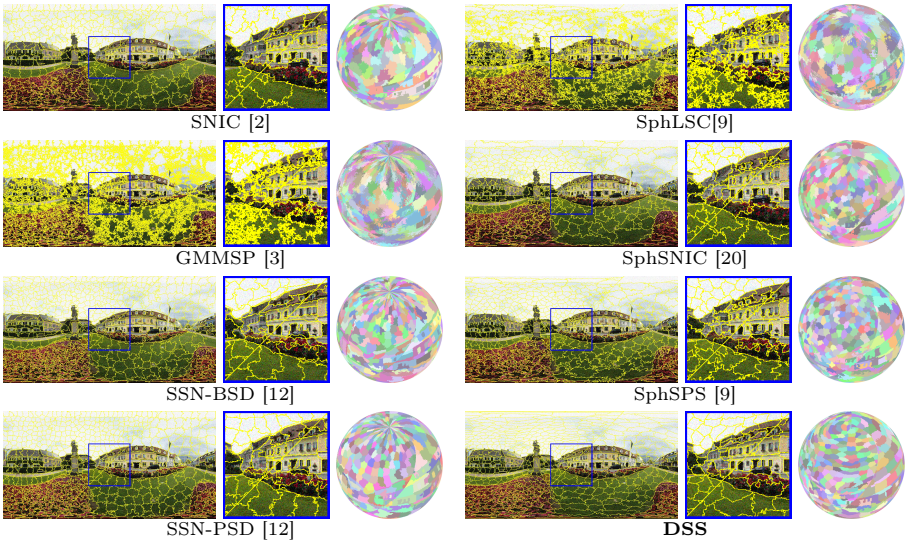


Fig. 7. Qualitative comparison on a noisy PSD image for planar (left) and spherical methods (right) for two superpixel numbers $K = 1200$ (top-left) and $K = 400$ (bottom right). DSS is able to preserve its regularity and accuracy compared to most methods.

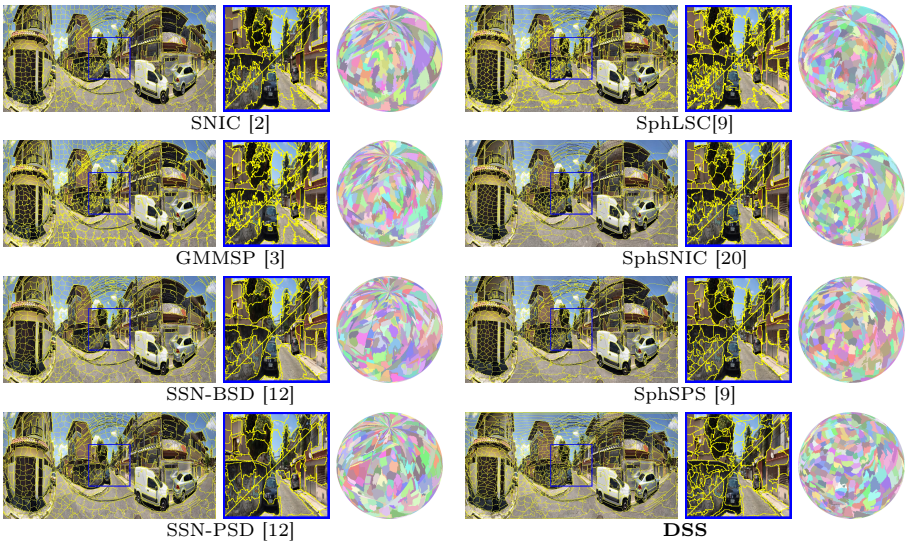


Fig. 8. Qualitative comparison on a WP image for planar (left) and spherical methods (right) for two superpixel numbers $K = 1200$ (top-left) and $K = 400$ (bottom right). Note how DSS is able to capture the car in the image center.

9. Giraud, R., Pinheiro, R.B., Berthoumieu, Y.: Generalization of the shortest path approach for superpixel segmentation of omnidirectional images. Pattern Recognition **142**, 109673 (2023)

10. Giraud, R., Ta, V.T., Papadakis, N.: Evaluation framework of superpixel methods with a global regularity measure. Journal of Electronic Imaging **26**(6) (2017)
11. Giraud, R., Ta, V.T., Papadakis, N.: Robust superpixels using color and contour features along linear path. Computer Vision and Image Understanding **170**, 1–13 (2018)
12. Jampani, V., Sun, D., Liu, M.Y., Yang, M.H., Kautz, J.: Superpixel sampling networks. In: European Conference on Computer Vision (2018)
13. Liu, M.Y., Tuzel, O., Ramalingam, S., Chellappa, R.: Entropy rate superpixel segmentation. In: Conference on Computer Vision and Pattern Recognition (2011)
14. Lo, S.C.B., Li, H., Wang, Y., Kinnard, L., Freedman, M.T.: A multiple circular path convolution neural network system for detection of mammographic masses. IEEE Transactions on Medical Imaging **21**(2), 150–158 (2002)
15. 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: International Conference on Computer Vision (2001)
16. Schubert, S., Neubert, P., Pöschmann, J., Protzel, P.: Circular convolutional neural networks for panoramic images and laser data. In: IEEE Intelligent Vehicles Symposium (2019)
17. Shi, B., Bai, S., Zhou, Z., Bai, X.: DeepPano: Deep panoramic representation for 3-d shape recognition. IEEE Signal Processing Letters **22**(12), 2339–2343 (2015)
18. da Silveira, T.L.T., Jung, C.R.: Dense 3D scene reconstruction from multiple spherical images for 3-DoF+ VR applications. In: IEEE Conference on Virtual Reality and 3D User Interfaces (2019)
19. da Silveira, T.L.T., Pinto, P.G.L., Murrugarra-Llerena, J., Jung, C.R.: 3D scene geometry estimation from 360 imagery: A survey. ACM Computing Surveys **55**(4), 1–39 (2022)
20. da Silveira, T.L., de Oliveira, A.Q., Walter, M., Jung, C.R.: Fast and accurate superpixel algorithms for 360° images. Signal Processing **189** (2021)
21. Sun, C., Hsiao, C.W., Sun, M., Chen, H.T.: HorizonNet: Learning room layout with 1D representation and Pano Stretch data augmentation. In: Conference on Computer Vision and Pattern Recognition (2019)
22. Wan, L., Xu, X., Zhao, Q., Feng, W.: Spherical Superpixels: Benchmark and evaluation. In: Asian Conference on Computer Vision (2018)
23. Wang, T.H., Huang, H.J., Lin, J.T., Hu, C.W., Zeng, K.H., Sun, M.: Omnidirectional CNN for visual place recognition and navigation. In: International Conference on Robotics and Automation (2018)
24. Wong, T.T., Luk, W.S., Heng, P.A.: Sampling with Hammersley and Halton points. Journal of Graphics Tools **2**(2), 9–24 (1997)
25. Xiao, J., Ehinger, K.A., Oliva, A., Torralba, A.: Recognizing scene viewpoint using panoramic place representation. In: Conference on Computer Vision and Pattern Recognition (2012)
26. Yang, F., Sun, Q., Jin, H., Zhou, Z.: Superpixel segmentation with fully convolutional networks. In: Conference on Computer Vision and Pattern Recognition (2020)
27. Yang, K., Hu, X., Fang, Y., Wang, K., Stiefelhagen, R.: Omnisupervised omnidirectional semantic segmentation. IEEE Transactions on Intelligent Transportation Systems (2020)
28. Yang, K., Zhang, J., Reiß, S., Hu, X., Stiefelhagen, R.: Capturing omni-range context for omnidirectional segmentation. In: Conference on Computer Vision and Pattern Recognition (2021)

29. Yogamani, S., Hughes, C., Horgan, J., Sistu, G., Varley, P., O'Dea, D., Uricár, M., Milz, S., Simon, M., Amende, K., et al.: Woodscape: A multi-task, multi-camera fisheye dataset for autonomous driving. In: International Conference on Computer Vision (2019)
30. Zhao, Q., Dai, F., Ma, Y., Wan, L., Zhang, J., Zhang, Y.: Spherical superpixel segmentation. IEEE Trans. on Multimedia **20**(6), 1406–1417 (2018)