symmetry

MDPI

Article

Superpixels with Content-Awareness via a Two-Stage Generation Framework

Cheng Li [1], Nannan Liao [2], Zhe Huang [3], He Bian [1], Zhe Zhang [1] and Long Ren [1,*]

[1] Xi'an Institute of Optics and Precision Mechanics, Chinese Academy of Sciences, Xi'an 710119, China; licheng@opt.ac.cn (C.L.); bianhe@opt.ac.cn (H.B.); zhangzhe@opt.ac.cn (Z.Z.)
[2] Institute of Intelligent Control and Image Engineering, Xidian University, Xi'an 710071, China; nnliao@stu.xidian.edu.cn
[3] Wuhan Second Ship Design and Research Institute, Wuhan 430025, China; huangzhe0507@foxmail.com
* Correspondence: renlong@opt.ac.cn

Abstract: The superpixel usually serves as a region-level feature in various image processing tasks, and is known for segmentation accuracy, spatial compactness and running efficiency. However, since these properties are intrinsically incompatible, there is still a compromise within the overall performance of existing superpixel algorithms. In this work, the property constraint in superpixels is relaxed by in-depth understanding of the image content, and a novel two-stage superpixel generation framework is proposed to produce content-aware superpixels. In the global processing stage, a diffusion-based online average clustering framework is introduced to efficiently aggregate image pixels into multiple superpixel candidates according to color and spatial information. During this process, a centroid relocation strategy is established to dynamically guide the region updating. According to the area feature in manifold space, several superpixel centroids are then split or merged to optimize the regional representation of image content. Subsequently, local updating is adopted on pixels in those superpixel regions to further improve the performance. As a result, the dynamic centroid relocating strategy offers online averaging clustering the property of content awareness through coarse-to-fine label updating. Extensive experiments verify that the produced superpixels achieve desirable and comprehensive performance on boundary adherence, visual satisfactory and time consumption. The quantitative results are on par with existing state-of-the-art algorithms in terms with several common property metrics.

Keywords: superpixels; content awareness; centroid relocation; online average clustering

check for updates

Citation: Li, C.; Liao, N.; Huang, Z.; Bian, H.; Zhang, Z.; Ren, L. Superpixels with Content-Awareness via a Two-Stage Generation Framework. *Symmetry* **2024**, *16*, 1011. https://doi.org/10.3390/sym16081011

Academic Editor: Sergei D. Odintsov

Received: 1 July 2024
Revised: 30 July 2024
Accepted: 5 August 2024
Published: 8 August 2024

Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

In the field of image processing, the concept of superpixels is generalized to semantically segment a visual image into several simple connected sub-regions [1]. A superpixel is intuitively a collection of spatially connected pixels with similar low-level visual features such as color and texture, which could represent a mass of image entities [2]. On this basis, superpixel segmentation has gradually become a popular preprocessing tool for various advanced computer vision tasks to boost running efficiency. The emerging superpixel algorithms have found applications in diverse fields, including image classification [3,4], semantic segmentation [5,6], and object detection and tracking [7,8].

Since their inception in 2003, the pursuit of superpixels with sensational performance has remained a focal point in this field. In recent years, an increasing number of superpixel algorithms have been proposed to improve on former achievements [9–12]. In general, a good superpixel generation algorithm has properties including boundary accuracy, visual satisfactory and running efficiency. These requirements also lead to an essential difference from conventional image over-segmentation or contour detection, whose practical significance can be reflected in the following three aspects [13]:

- Boundary accuracy. When employed as a region-level feature for image segmentation or edge detection, it evaluates the effectiveness of superpixels in delineating image object boundaries and their consistency with the ground truth;
- Feature quality. For some applications such as image reconstruction and data compression, it is important to preserve regional homogeneity, spatial relationships and other information pertaining to internal objects within an image;
- Running efficiency. As an advantaged tool for video and image preprocessing, superpixel segmentation plays a pivotal role in computer vision applications that necessitate real-time performance.

Nevertheless, these properties still suffer from a certain degree of incompatibility, resulting in a rigorous constraint on the properties observed in the generated superpixels. In practice, this can be delineated as two sub-problems based on their manifestations.

Accuracy and quality: Real-world scenarios often exhibit diverse objects and textured backgrounds, leading to a common compromise between segmentation accuracy and feature quality in most superpixel methods [14]. For example, it is usually difficult for superpixels with color-based measurement to maintain a balanced trade-off between boundary adherence and region homogeneity, especially in low-contrast regions [15]. In addition, a greater compactness for superpixels sometimes deteriorates the accuracy of region representation despite promoting the visual satisfactory and spatial topology [16]. Furthermore, since regular visual information is rarely observed, the criterion of size uniformity is hard to follow strictly. A typical instance of this incompatibility is twig and sinuous object representation, which becomes an intractable problem for most algorithms except for some specialized optimizations [17,18].

Performance and efficiency: Most methods mainly create superpixels by solving a cost function established by global and local correlations of several low-level features [19]. The generation process is then recast as a minimization problem. Since it is time-consuming to find the globally optimal solution, a sub-optimal substitute is usually utilized to find the best compromise. Among them are clustering-based approaches, which usually adopt an empirical iteration time as the expiration. Nevertheless, it might be insufficient for detailed regions to converge accurately. Moreover, some dispensable strategy in structured superpixel generation frameworks might introduce additional time consumption, such as initialization and post-processing optimization.

Given the diverse performance of various superpixel generation algorithms in the literature, it remains a major challenge to seek synergistic optimization rather than property compromise. On the other hand, several instructive works have achieved state-of-the-art performance with respect to specific aspects. In practical applications, the combination of their strengths might scale new heights. Building upon this premise, this paper proposes a novel approach termed superpixels with Content-Awareness via a Two-Stage generation framework (CATS). It combines the clustering pattern of [20] with the density evaluation of [21], which enhances both pixel classification precision and image content sensitivity. As a result, it can produce superpixels with better boundary adherence and visual satisfaction while maintaining operational efficiency.

In summary, the main contributions of this work are as follows:

- According to the area in manifold space, a region-density-based clustering centroid relocating strategy is proposed to adjust the spatial distribution of superpixels;
- By using redistributed clustering centers, a coarse-to-fine implementation of online average clustering framework is designed for improving the feature performance;
- The synergistic CATS superpixel feature demonstrates comparable performance in terms of segmentation accuracy, spatial compactness, and operational efficiency.

The subsequent sections are organized as follows. Section 2 reviews several state-of-the-art (SOTA) superpixel algorithms. In Section 3, the proposed CATS superpixel generation framework is elaborated on in detail. Section 4 presents both qualitative and quantitative analyses. Concluding remarks and future perspectives are included in Section 5.

2. Related Work

Over the past two decades, the field has witnessed the emergence of numerous approaches for generating superpixels. Several reviews have presented comprehensive overviews of the previous works and categorized them based on mathematical principles or low-level features [22]. In this paper, the most SOTA superpixel algorithms are simply classified into two categories, iteration-demand and iteration-free. In the following review, only the closely related literature is discussed in detail.

2.1. Iteration-Demand Superpixels

Iteration-demand superpixels usually establish an energy function by the sum of pixel-cluster correlation or intra-region heterogeneity. In this way, the superpixel generation process is recast as a minimization problem that can be iteratively optimized [23].

Simple Linear Iterative Clustering (SLIC) [24] is widely recognized as a seminal work in this field, and its heuristic framework has garnered significant acclaim from subsequent researchers. The generation of SLIC superpixels is essentially an unsupervised classification process based on pixel clustering. Specifically, it restricts the conventional k-means algorithm into local context to reduce the calculation of Centroidal Voronoi Tessellation (CVT) in a 5-dimensional fusion feature space. The correlation between a pixel and a cluster centroid is measured by a normalized distance in both 3-dimensional CIELAB color space and 2-dimensional Euclidean space. Based on the measurement with a spatial constraint term, SLIC superpixels exhibit satisfactory color homogeneity while maintaining pleasant spatial compactness. Due to the exceptional overall performance of SLIC, various enhancements encompassing feature substitution, structure optimization and clustering acceleration have been made to this framework in subsequent studies.

Manifold SLIC (MSLIC) [25] and Minimum Barrier Superpixel (MBS) [26] are two representative variants with emphasis on both feature and structure innovation. MSLIC maps the 5-dimensional fusion feature space to a 3-dimensional manifold space, and then computes a Restricted CVT on the manifold surface. By utilizing the property that area elements in this space can effectively measure image content density, the inverse projection could induce content-sensitive superpixel segmentation results. Compared with the conventional SLIC, MSLIC superpixels better preserve object boundaries in regions with detailed image content while maintaining regular smoothness with desirable compactness. MBS proposes a novel feature correlation measurement for pixel classification within the iterative clustering framework. The proposed Compact-Aware Minimum Barrier Distance (CA-MBD) can not only control the compactness continuously, but maintains a balanced trade-off with accuracy. In addition, an efficient raster-scanning approach is introduced to guide the computation of CA-MBD, which guarantees the overall running efficiency.

Boundary-Aware Superpixel Segmentation (BASS) [27] and Content-Adaptive Superpixel (CAS) [28] introduce additional image cues along with the most frequently used color and spatial features. BASS incorporates a grayscale-weighted geodesic distance term to measure the correlation between pixels and superpixel clusters. In this way, BASS enables smaller superpixels to focus on informative areas while larger ones cover evenly distributed regions. In comparison to the SLIC algorithm, it generates superpixels with higher regional homogeneity and boundary accuracy. CAS incorporates four low-level features to model global superpixels, including color, texture, contour gradient, and spatial distance. An objective function is formulated based on inter-superpixel and intra-superpixel variances, which is then iteratively optimized through adaptive weight updates. During this process, the feature weights with weaker discrimination capabilities are updated by clustering results that demonstrate stronger discrimination, so as to accurately classify pixels of different image objects. Experimental results report that it exhibits exceptional intra-superpixel homogeneity and compactness in distribution with adaptability towards variations in image content.

There are others in this category that focus on speeding up the clustering efficiency. In the work of [29], a new approach termed Iterative Boundaries implicit Identification

Superpixels (IBIS) is proposed to further reduce the complexity of SLIC-like methods. This can efficiently identify the coherence of the local region via a coarse-to-fine block-wise subdivision, and then update only a fraction of the pixels in the heterogeneous region in subsequent iterations. Consequently, the amount of computation for global updating in each iteration enters a steep decline. Compared with the conventional SLIC, it runs over 8 times faster than the conventional SLIC while performing with a matched segmentation accuracy. Fast Linear Iterative Clustering (FLIC) [30] superpixels incorporate feature continuity between neighboring pixels in real-world scenarios, facilitating consistent label assignment during the clustering process. By introducing an active search strategy based on adjacent pixel correlations, FLIC relaxes the constraints on k-means clustering search range in SLIC. This modification enables FLIC superpixels to simultaneously perform label assignment and cluster updates while significantly reducing computation time required for pixel-cluster similarity. In practical applications, this framework achieves rapid convergence of all superpixel clusters with just 2–3 global iterations.

2.2. Iteration-Free Superpixels

Several algorithms can generate superpixels without clustering, which primarily relies on a one-pass framework capable of achieving global optimal or suboptimal results [31]. They merely (potentially) necessitate making multiple adjustments to regional pixels or candidate superpixels and obviate multiple global updates.

Simple Non-Iterative Clustering (SNIC) [20] represents a significant improvement over SLIC, offering a more comprehensive approach. It employs a priority queue to efficiently inspect neighboring elements for pixel-cluster matching, which is manifested as a diffusion-based region growing process. Once a pixel is assigned with a definite label, it would be excluded in the subsequent operations. Compared with SLIC, SNIC effectively reduces the redundant computation during label updating in overlapping regions. Moreover, all superpixels are generated by absorbing neighboring pixels from the corresponding seed via a region growing pattern. Therefore, they can maintain spatial connectivity with uniformly distributed labels without the need for merging isolated partitions.

In the work of [32], Density-Based Spatial Clustering of Applications with Noise (DBSCAN) is introduced to optimize the clustering framework. In contrast to the k-means method, DBSCAN inspects all pixels only once without iterative execution. The proposed method adopts a split-and-merge scheme to generate superpixels. It first aggregates pixels into candidate superpixels by DBSCAN, and then merges the small clusters into adjacent homogenous superpixels to strengthen the spatial connectivity.

Ultra-fast Superpixel Extraction via Quantization (USEQ) [33] has been proposed for the rapid extraction of superpixels. Initially, spatial and color quantization are employed to approximately reconstruct the input image. Subsequently, the misclassified pixels are adjusted to their most visually and spatially similar superpixels using maximum a posteriori (MAP) estimation. In follow-up work, an adaptive sampling strategy was introduced to further enhance the content sensitivity of USEQ superpixels to the distribution of image content [34]. This strategy selects one or more superpixel candidates from each spatial quantization region based on the color variations, which enables the compactness to adapt to the differences in uniform or cluttered regions

The Dynamic Random Walk (DRW) [18] method maps the input image to an undirected graph model using graph theory and the Random Walk (RW) [35] model introduces a novel dynamic node to the conventional RW framework, effectively reducing redundant calculations by constraining the walking range. To address the issue of seed-lacking, the energy function of the random walk is redefined. Moreover, it incorporates first arrival probabilities between node pairs to mitigate interference across partitions. Consequently, DRW superpixels exhibit enhanced boundary adherence with linear time complexity.

Watershed Transform (WT) [36] is an efficient and instructive over-segmentation approach rooted in topological theory that can be described as a flooding process. It first identifies the minima of gradient images and subsequently submerges the corresponding pixels.

To further control the amount and regularity of watershed partitions, WaterPixels (WP) [37] proposes a spatially regularized gradient to achieve a tunable trade-off between superpixel regularity and adherence to object boundaries. Essentially, it forms a joint gradient and spatial distance measurement similar to the feature correlation calculation in SLIC. More recently, in the work of [38], a two-phase implementation has been introduced in WT to further improve the performance. It designs different measurement for boundary pixels by two separate criteria that focus on the color homogeneity and shape regularity, respectively. As reported in the experiments, this approach could ameliorate the inter-inhibition between boundary adherence and spatial compactness.

3. Proposed Method

In this section, the proposed CATS framework is introduced in detail; the workflow is visualized in Figure 1 in advance for a better illustration. To improve the segmentation quality while maintaining the computational efficiency, the proposed approach performs superpixel extraction in a coarse-to-fine manner. A global online averaging clustering is first performed on the image plane to produce superpixel candidates and quantify the local content density in Section 3.1. From this foundation, an adaptive centroid relocation strategy is presented in Section 3.2 to dynamically split several candidates with dense content, as well as merge partitions with small areas. Subsequently, Section 3.3 shows the pseudocode of synthetic CATS implementation.

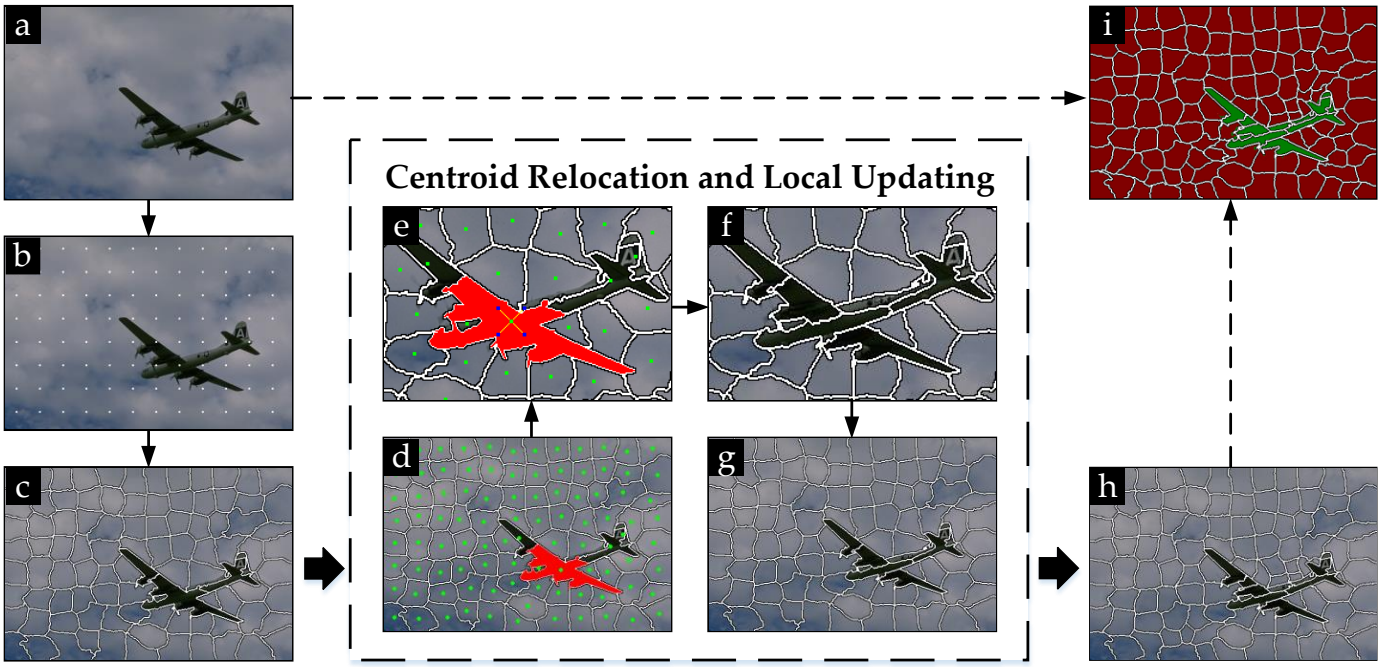


Figure 1. Workflow of the proposed CATS superpixel generation framework. (**a**) Input image; (**b**) Grid-level seeding initialization, each white dot indicates an initial cluster centroid; (**c**) Superpixel candidates produced by conventional SNIC method, the white outlines represent the boundaries of different superpixel regions; (**d**–**g**) Centroid relocation and local updating, note that the series of processes are performed iteratively until all superpixel candidates maintain a moderate content density. (**d**) Centroid distribution of (**c**), the green dots are relocated centroids and the region marked in red has the most dense content; (**e**) Zoom-in performance of (**d**), the blue dots are split centroids based on centroid relocation strategy in CATS; (**f**) Local updating results of (**e**); (**g**) Overall performance of (**f**); (**h**) Result of CATS superpixels; (**i**) Ground truth of (**a**) covered by CATS superpixels of (**h**).

3.1. Superpixel Candidate Generation

The online averaging clustering efficiently aggregates pixels in a non-iterative manner. Substantially, it employs a greedy algorithm and utilizes a priority queue to assign unique labels to all pixels. In this part, the implementation of online averaging clustering in SNIC is followed to generate superpixel candidates.

3.1.1. Initialization and Seeding

Given an image $\mathbf{I} = \{p_i\}_{i=1}^{N}$ with N pixels, let us denote by $\mathbf{P}(p_i) = [x(p_i), y(p_i)]$ and $\mathbf{C}(p_i) = [l(p_i), a(p_i), b(p_i)]$ the 2-dimensional Euclidean coordinate and 3-channel CIELAB color of each pixel $p_i \in \mathbf{I}$, respectively. In SNIC, the correlation of a pair of pixels p_i and p_j is measured by a normalized Euclidean distance $D(p_i, p_j)$ in a joint color and spatial space $\mathbb{R}^5$ as follows

$$D(p_i, p_j) = \sqrt{\left(\frac{D_{color}(p_i, p_j)}{N_c}\right)^2 + \left(\frac{D_{spatial}(p_i, p_j)}{N_s}\right)^2} \tag{1}$$

where N_c and N_s are two pre-defined factors to control the influence of color and position on $D(p_i, p_j)$, and

$$D_{color}(p_i, p_j) = \sqrt{\left(l(p_i) - l(p_j)\right)^2 + \left(a(p_i) - a(p_j)\right)^2 + \left(b(p_i) - b(p_j)\right)^2} \tag{2}$$

$$D_{spatial}(p_i, p_j) = \sqrt{\left(x(p_i) - x(p_j)\right)^2 + \left(y(p_i) - y(p_j)\right)^2} \tag{3}$$

First, a zeroed label map $\mathbf{L} = \{L(p_i)\}_{i=1}^{N}$ is introduced to record the clustering result, whose size is identical to the input image $\mathbf{I}$. Meanwhile, a priority queue $\mathbf{Q}$ is implemented by a small-root binary tree, which still pops up the top-most element with the minimal key value when it is not empty. Next, a set of seeds $\{p_k^{\mathrm{s}}\}_{k=1}^{K}$ is acquired by uniformly sampling K pixels on $\mathbf{I}$; each seed is centered at a grid cell with the step of $S = \sqrt{N/K}$. The n-th evenly distributed seed p_n^{s} is assigned with a unique label (i.e., $L(p_n^{\mathrm{s}}) = n$), which serves as the incipient centroid of the n-th cluster region $\Re_n$. All seeds are then pushed onto $\mathbf{Q}$.

3.1.2. Clustering and Updating

Each unlabeled adjacent pixel $p^{\mathrm{a|g}}$ to the growth point p^{g} of the current sprouting cluster $\Re^{\mathrm{g}}$ is inspected to calculate the pixel-cluster correlation $D\left(p^{\mathrm{a|g}}, c^{\mathrm{g}}\right)$, which is measured by Equation (1). c^{g} represents the centroid of $\Re^{\mathrm{g}}$. Then $p^{\mathrm{a|g}}$ is pushed onto $\mathbf{Q}$, and $D\left(p^{\mathrm{a|g}}, c^{\mathrm{g}}\right)$ acts as the key value for heap sorting. It is worth noting that all seeds are initialized with a zero key value. In this case, they could become the first growth points within the corresponding cluster regions, respectively.

The top-most element $p^{\mathrm{t|g}}$ (one of $p^{\mathrm{a|g}}$ that holds the minimal key value in $\mathbf{Q}$) is assigned with an identical label to its corresponding growth point p^{g} after it is popped, i.e., $L\left(p^{\mathrm{t|g}}\right) = L(p^{\mathrm{g}})$. The current sprouting cluster $\Re^{\mathrm{g}}$ that absorbs $p^{\mathrm{t|g}}$ as a new member is then updated by

$$\mathbf{C}(\Re^{\mathrm{g}}) = \frac{\mathbf{C}(c^{\mathrm{g}}) + \mathbf{C}\left(p^{\mathrm{t|g}}\right)}{|\Re^{\mathrm{g}}| + 1}, \mathbf{P}(\Re^{\mathrm{g}}) = \frac{\mathbf{P}(c^{\mathrm{g}}) + \mathbf{P}\left(p^{\mathrm{t|g}}\right)}{|\Re^{\mathrm{g}}| + 1} \tag{4}$$

where $|\Re^{\mathrm{g}}|$ indicates the member amount in $\Re^{\mathrm{g}}$. Thereafter, $p^{\mathrm{t|g}}$ replaces p^{g} as the latest growth point of $\Re^{\mathrm{g}}$ to inspect its unlabeled neighbor pixels in the next loop.

The clustering and updating are repeated until the priority queue $\mathbf{Q}$ is empty and there are no pixels in the image with unassigned labels. Finally, it outputs the updated label map as the clustering result. Notice that an unlabeled pixel does not necessarily result

in immediate labeling after being inspected. In practice, it might be inspected by different growth points, and the final label is determined by the cluster that is most relevant to it.

3.2. Centroid Relocation Strategy

The aforementioned implementation of superpixel candidate generation employs a grid-level seeding initialization, which partially ensures the size regularity and uniform distribution of superpixels. Nevertheless, due to the inconsistent density distribution of information in real-world images, superpixel generation relying solely on this process inevitably leads to mistaken label classification in several candidate regions. In other words, the performance of online averaging clustering is sensitive to the placement of initial seeds. To address this issue, a centroid relocation strategy is proposed based on the content density. This can evaluate and adjust the cluster centroid (position and amount) to guide the re-generation of some candidate superpixels.

3.2.1. Area of Manifold Surface

As illustrated in Figure 2, the key to perceiving the image content density in MSLIC is to transform the image plane into a 2-dimensional manifold space $\mathcal{M}$ through a stretching map $\Phi : I \to \mathbb{R}^5$. In this way, the feature of a pixel $p_i \in \mathcal{M}$ is described as $\mathbf{\Phi}(p_i) = (\lambda_s \mathbf{P}(p_i), \lambda_c \mathbf{C}(p_i))$, where $\lambda_s = 1/N_s$ and $\lambda_c = 1/N_c$.

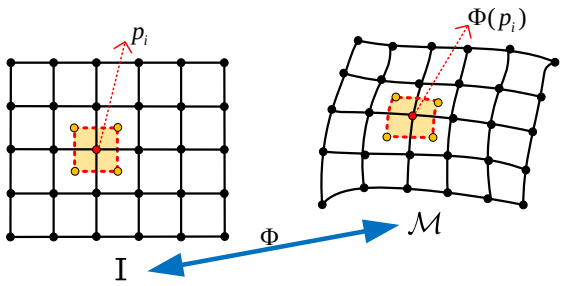


Figure 2. Image transformation from 2-dimensional plane to 2-dimensional manifold.

The area feature is very suitable for indicating the image content density on $\mathcal{M}$, which is further verified by the follow-up research of MSLIC [21]. Specifically, as shown in Figure 2, given a pixel $p_i \in \mathbf{I}$ on the image plane, let us denote by $\square_p$ a unit square centered at $\mathbf{P}(p_i)$ (red box filled by yellow), whose vertexes are notated as p_i^1, p_i^2, p_i^3 and p_i^4, respectively. Meanwhile, each vertex point holds the mean value of color and spatial feature of p_i and its three 8-neighboring pixels. In this case, the area of $\square_p$ can be computed by

$$Area(\square_p) = Area\left(\Delta p_i^1 p_i^3 p_i^4\right) + Area\left(\Delta p_i^1 p_i^2 p_i^4\right) \tag{5}$$

where

$$Area\left(\Delta p_i^1 p_i^3 p_i^4\right) \approx \frac{1}{2}\left\|\overrightarrow{\Phi\left(p_i^3\right)\Phi\left(p_i^1\right)}\right\| \cdot \left\|\overrightarrow{\Phi\left(p_i^3\right)\Phi\left(p_i^4\right)}\right\| \cdot \sin\theta \tag{6}$$

In Equation (6), $\sin\theta$ measures the sine value between vector $\overrightarrow{\Phi\left(p_i^3\right)\Phi\left(p_i^1\right)}$ and $\overrightarrow{\Phi\left(p_i^3\right)\Phi\left(p_i^4\right)}$. $\left\|\overrightarrow{\Phi\left(p_i^3\right)\Phi\left(p_i^1\right)}\right\|$ represents the distance between $\Phi\left(p_i^3\right)$ and $\Phi\left(p_i^1\right)$ on $\mathcal{M}$, which is also calculated by Equation (1). Similar calculations can be done for $\left\|\overrightarrow{\Phi\left(p_i^3\right)\Phi\left(p_i^4\right)}\right\|$.

3.2.2. Splitting and Merging

Since the conventional SNIC lacks the ability of global updating, the generated superpixels are sensitive to the placement of initial centroids (seeds). To tackle this issue, a straightforward idea is to relocate the centroids, including perturbing the position of

existing centroids and emerging new seeds for clustering. As shown in Figure 1e, the refined distribution of cluster centroids becomes more relevant to the local density, i.e., detailed regions acquire more centroids while sparse regions maintain the convergence.

In Figure 3, the area of each pixel on $\mathcal{M}$ can be calculated by Equation (5). Consequently, the total area of the manifold surface $Area(\mathcal{M})$ is obtained by summing up all pixels

$$Area(\mathcal{M}) = \sum_{i=1}^{N} Area(\Phi(\square p_i)) \tag{7}$$

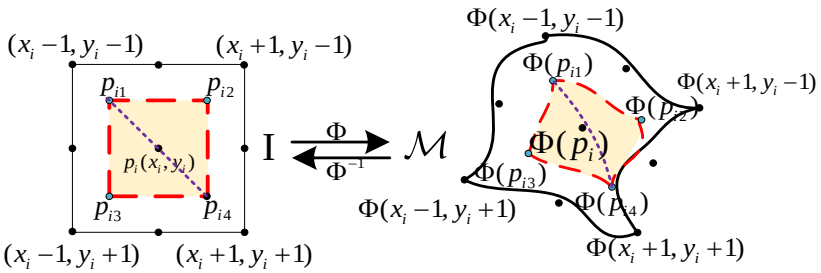


Figure 3. Schematic diagram of a pixel area on the 2-dimensional manifold space.

Assuming that the expected number of superpixels is K, the average area of all candidate superpixels on $\mathcal{M}$ is

$$\overline{Area(\Phi(\Re))} = \frac{Area(\mathcal{M})}{K},\ i = 1, 2, 3, \cdots, K \tag{8}$$

In fact, the area of k-th candidate superpixel $\Re_k$ on $\mathcal{M}$ is

$$Area(\Phi(\Re_k)) = \sum_{p_i \in \Re_k} Area(\Phi(\square p_i)) \tag{9}$$

The ratio of area on $\mathcal{M}$ can be notated as a scaling factor λ_k to judge the splitting or merging operation, which is mandatory in the subsequent process.

$$\lambda_k = \frac{Area(\mathcal{M})}{Area(\Phi(\Re_k))} \tag{10}$$

Specifically, when the area of a candidate superpixel exceeds twice that of an average superpixel (i.e., $K \geq 2\lambda_k$), its regional content density is identified as high. Therefore, the region corresponding to $\Re_k$ requires more superpixels with smaller sizes to adhere to complex boundaries while maintain spatial compactness. To this end, a primary step is to resample more seeds in this region.

The splitting implementation in [25] is partly learnt in this work. As for the centroid c_k of superpixel candidate $\Re_k$, it is replaced by four symmetrically sampled seeds, c_k^1, c_k^2, c_k^3 and c_k^1 positioned at

$$\begin{aligned}
\mathbf{P}(c_k^1) &= \left(x(c_k) - \tfrac{\lambda_k}{K}S, y(c_k) + \tfrac{\lambda_k}{K}S\right) \\
\mathbf{P}(c_k^2) &= \left(x(c_k) - \tfrac{\lambda_k}{K}S, y(c_k) - \tfrac{\lambda_k}{K}S\right) \\
\mathbf{P}(c_k^3) &= \left(x(c_k) - \tfrac{\lambda_k}{K}S, y(c_k) - \tfrac{\lambda_k}{K}S\right) \\
\mathbf{P}(c_k^4) &= \left(x(c_k) + \tfrac{\lambda_k}{K}S, y(c_k) + \tfrac{\lambda_k}{K}S\right)
\end{aligned} \tag{11}$$

In addition, the initial color information of these seeds is the same as the corresponding pixels on $\mathbf{I}$.

There is a potential increase in the number of candidate superpixels after the aforementioned splitting operation, which might lead to a higher complexity for subsequent region-based information processing. Consequently, it becomes imperative to consider merging neighboring regions that exhibit low content density after global region splitting.

In contrast to the conventional MSLIC, which incorporates intricate merging conditions and outputs only one seed for subsequent region iteration, a region-level merging approach is employed to directly update candidate superpixels. The procedures are outlined as follows.

First, a region-based undirected graph $G = (V, E)$ is established to depict the spatial relationship of all superpixel candidates [39]. A node vertex $v_k \in V$ represents a cluster centroid c_k, and the weight ω_{ij} of an edge $e_k \in E$ that connects a node pair (v_i, v_j) depicts the inter-centroid color correlation

$$\omega_{ij} = \begin{cases} D_{color}(v_i, v_j) \text{ if } \Re_i | \Re_j \\ 0 \quad \text{otherwise} \end{cases} \tag{12}$$

where $\Re_i | \Re_j$ means that the corresponding clusters of c_i and c_j are spatially adjacent. In this way, the value of ω_{ij} can effectively indicate the color homogeneity between cluster $\Re_i$ and $\Re_j$, which is calculated by Equation (2).

Next, the merging operation commences from the global minimum weight, if the corresponding clusters ω_m and ω_n differ greatly in areas (i.e., the ratio θ of the larger area compared to the smaller one is set to 4) the smaller superpixel is merged into the larger counterpart. Otherwise ($\theta < 4$), the subsequent weight is selected in a descending order for processing.

$$\theta = \frac{Area(\Phi(\Re_m))}{Area(\Phi(\Re_n))}, Area(\Phi(\Re_m)) > Area(\Phi(\Re_n)) \tag{13}$$

Obviously, this merging process effectively mitigates the size disparities among global superpixels while preserving regional homogeneity. Meanwhile, it can be ceased at any time, which allows for precise control over the final count of superpixels.

3.3. Integrated CATS Framework

As shown in Algorithm 1, the integrated CATS superpixel generation framework is raised by incorporating the area feature indicator and splitting/merging strategy into online average clustering. The modular structure is concise for realization based on the implementation of standard SNIC and MSLIC.

Algorithm 1 CATS superpixel generation framework

Input: Source image $\mathbf{I} = \{p_i\}_{i=1}^{N}$, the preset superpixel number K.
Output: Label map $\mathbf{L} = \{L(p_i)\}_{i=1}^{N}$ of $\mathbf{I}$.
Initialize cluster seeds $\{p_k^{\mathrm{s}}\}_{k=1}^{K}$ by grid sampling with step $S = \sqrt{N/K}$.
Initialize $L(p_i) = 0$ for each pixel $p_i \in \mathbf{I}$.
Initialize a priority queue $\mathbf{Q}$ with a small root.
for each seed p_k^{s} in $\{p_k^{\mathrm{s}}\}_{k=1}^{K}$ **do**
  Create and push an element $\mathbf{e}(p_k^{\mathrm{s}}) = [\mathbf{P}(p_k^{\mathrm{s}}), \mathbf{C}(p_k^{\mathrm{s}}), k, 0]$ onto $\mathbf{Q}$.
end for
while $\mathbf{Q}$ is not empty **do**
  Pop the top-most element $\hat{\mathbf{e}} = \mathbf{e}\left(p^{\mathrm{t}|\mathrm{g}}\right)$ from $\mathbf{Q}$.
  if $L\left(p^{\mathrm{t}|\mathrm{g}}\right) = 0$ **then**
  Set $L\left(p^{\mathrm{t}|\mathrm{g}}\right) = L(p^{\mathrm{g}})$.
  Update $\Re^{\mathrm{g}}$ by Equation (4).
  Compute λ_k by Equation (10).
  if $K \geq 2\lambda_k$ **then**
   Compute seed set Ω by Algorithm 2.
   Create and push the elements in Ω onto $\mathbf{Q}$.
  else
   for each 4-neighboring pixel $p^{\mathrm{a}|\mathrm{g}}$ of p^{g} **do**
   if $L\left(p^{\mathrm{a}|\mathrm{g}}\right) = 0$ **then**
    Create and push an element $\mathbf{e}\left(p^{\mathrm{a}|\mathrm{g}}\right) = \left[\mathbf{P}\left(p^{\mathrm{a}|\mathrm{g}}\right), \mathbf{C}\left(p^{\mathrm{a}|\mathrm{g}}\right), L(p^{\mathrm{g}}), D\left(p^{\mathrm{a}|\mathrm{g}}, c^{\mathrm{g}}\right)\right]$ onto $\mathbf{Q}$.

Algorithm 1 *Cont.*

end if
end for
end if
end if
end while
Create a RAG G to depict the spatial relationship of all candidate superpixels $\{\Re_i\}_{i=1}^{M}$.
Refine the label map $\mathbf{L}$ by Algorithm 3.
Return $\mathbf{L}$.

Furthermore, there are several predefined operational details to ensure stability and efficiency. As shown in Algorithm 2, during the process of centroid splitting in a content-dense cluster, the newly generated centroids might fall on an already labeled pixel (i.e., located in other cluster regions). In this case, only the centroids located in regions that are identified as high content density by Equation (9) can be considered as new cluster seeds.

Algorithm 2 splitting operation within CATS

Input: Current cluster centroid c_k, current label map $\mathbf{L}$
Output: Set of new cluster centroids Ω.
Compute $\left\{c_k^i\right\}_{i=1}^{4}$ by Equation (11).
for each candidate seed c_k^i in $\left\{c_k^i\right\}_{i=1}^{4}$ **do**
if $L\left(c_k^i\right) = 0$ **then**
Set $\Omega = \Omega \cup \left\{c_k^i\right\}$.
end if
end for
Return Ω.

In the merging process of two content-sparse clusters in Algorithm 3, if the area of the merged superpixel exceeds twice that of the average, it will no longer split. This is because the merging occurs after the splitting, and the newly generated centroid must fall on a labeled pixel.

Algorithm 3 merging operation within CATS

Input: Current Label map $\mathbf{L}$ of candidate superpixels with the corresponding RAG G.
Output: Refined label map $\mathbf{L}'$.
while the number of nodes in G is greater than K **do**
Compute the area ratio θ corresponding to the global minimum weight ω_{mn}.
if $\theta \geq 4$ **then**
for each pixel p_i with label $L(p_i) = L(c_n)$ **do**
Set $L(p_i) = L(c_m)$.
end for
else
Set $\omega_{mn} = \infty$.
end if
Update G.
end while
Return $\mathbf{L}'$.

4. Experiments and Analysis

In this work, all experiments are implemented on an Intel i7-12700 processor with 2.1 GHz. This conducts comparative experiments on the Berkeley Segmentation Data Set 500 (BSDS500) [40], which is currently the most widely used dataset for superpixel generation. BSDS500 consists of 500 natural images with the size of 481 × 321 or 321 × 481,

including animals, buildings, people, and other objects. Each image in the dataset contains at least five human-annotated ground truths, which enable convenient and accurate quantitative evaluation of superpixel performance. Meanwhile, MSLIC and SNIC are utilized as baselines, as well as USEQ, BASS, IBIS and DBSCAN are introduced as up-to-date competitors. To ensure fairness, all algorithms are configured with default parameters recommended by the original works.

4.1. Visual Comparison

The qualitative performance is visually compared in this part. Figure 4 shows four images and the corresponding ground-truths in the BSDS500 dataset, and Figure 5 exhibits the segmentation results of seven superpixel algorithms, wherein white outlines are boundaries of different labels. Obviously, all algorithms could produce superpixels with visually appealing shapes except for BASS, which generates irregularly shaped superpixels with uneven sizes. Furthermore, USEQ superpixels are more sensitive to textured content, resulting in unstable boundaries with sinuous and disordered shapes. On the other hand, most algorithms fail to accurately capture boundaries of small objects and strip areas (such as the bird and twig). For example, both MSLIC and IBIS superpixels could nearly adhere to the boundary of the bird through several iterations, but the accuracy is still low. As for other iteration-free methods, SNIC and DBSCAN, they also assume a uniform content complexity on the image plane, which neglect the local difference of information distribution. In contrast, the proposed CATS eases the constraints of the local optimum in the non-iterative clustering though online updating of seed placement (amount and position). Despite adopting a SNIC-like framework, CATS achieves better boundary fitting and uniform shape representation in these areas.

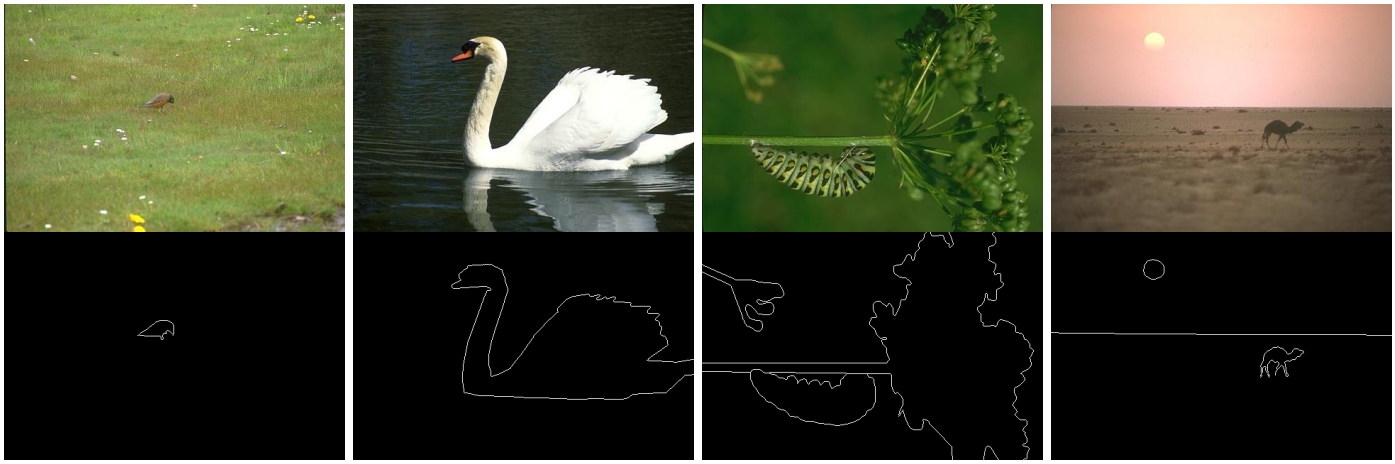


Figure 4. Four images and the corresponding ground-truths in the BSDS500 dataset.

It is also worth noting that several algorithms exhibit limited accuracy on fitting boundaries within regions containing intricate structures or low contrast information (like the worm and sun). Owing to the adequate resampled seeds, small distinctions within these regions could be depicted by the CATS algorithm. Consequently, it demonstrated superior boundary adherence and exhibited heightened sensitivity towards areas encompassing complex structures, resulting in regular-shaped generated superpixels.

Apart from the aforementioned performance, another outstanding advantage of CATS is the segmentation under a limited amount of superpixels. Figure 6 demonstrates the consistency between the ground truth and the outlines of CATS superpixels for more intuitive comparison of boundary precision. It is obvious that almost all boundary pixels of image objects can be detected by superpixels with moderate over-segmentation. Specifically, CATS superpixels effectively preserve intricate structures in accordance with the actual outlines of different objects. This is done by reallocating the grid-level sampled seeds, which can both consider global image property and local content awareness. Therefore, it

successfully reconciles the inherent contradiction between retaining accurate details and maintaining an optimal number of superpixels simultaneously.

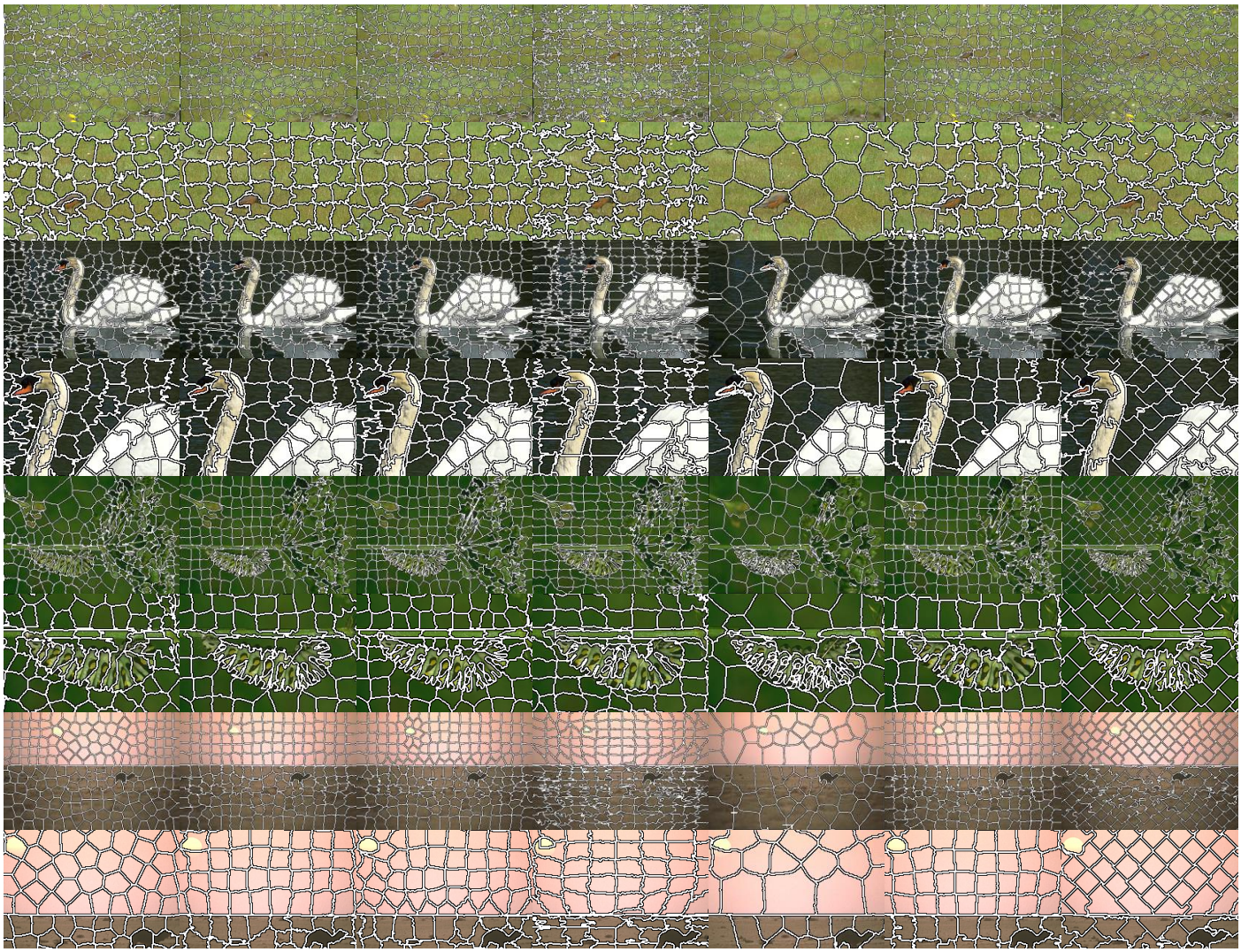


Figure 5. Visual comparison with approximately 200 superpixels. Alternating rows show each segmented image followed by the zoom-in performance. (**a**) MSLIC. (**b**) SNIC. (**c**) CATS. (**d**) USEQ. (**e**) BASS. (**f**) IBIS. (**g**) DBSCAN.

4.2. Metrical Evaluation

As mentioned in the introduction, a desirable superpixel generation method should effectively mitigate the property incompatibility while achieving synergistic optimization. To quantitatively evaluate the results, four popular metrics are introduced from the extended Berkeley segmentation benchmark [41], including Boundary Recall (BR), Under-Segmentation Error (USE), Achievable Segmentation Accuracy (ASA) and COMpactness (COM). Specifically, 200 images in the test subset are selected to calculate the curves in Figure 7, which represent the average results ranging from 50 to 500 superpixels. It should be noted that the source code of BASS lacks the controllability of superpixel number; as a substitute, the expected superpixel number is adjusted to approximate the range. In addition, the source code of DBSCAN cannot batch the dataset when the expected superpixel number is less than 150.

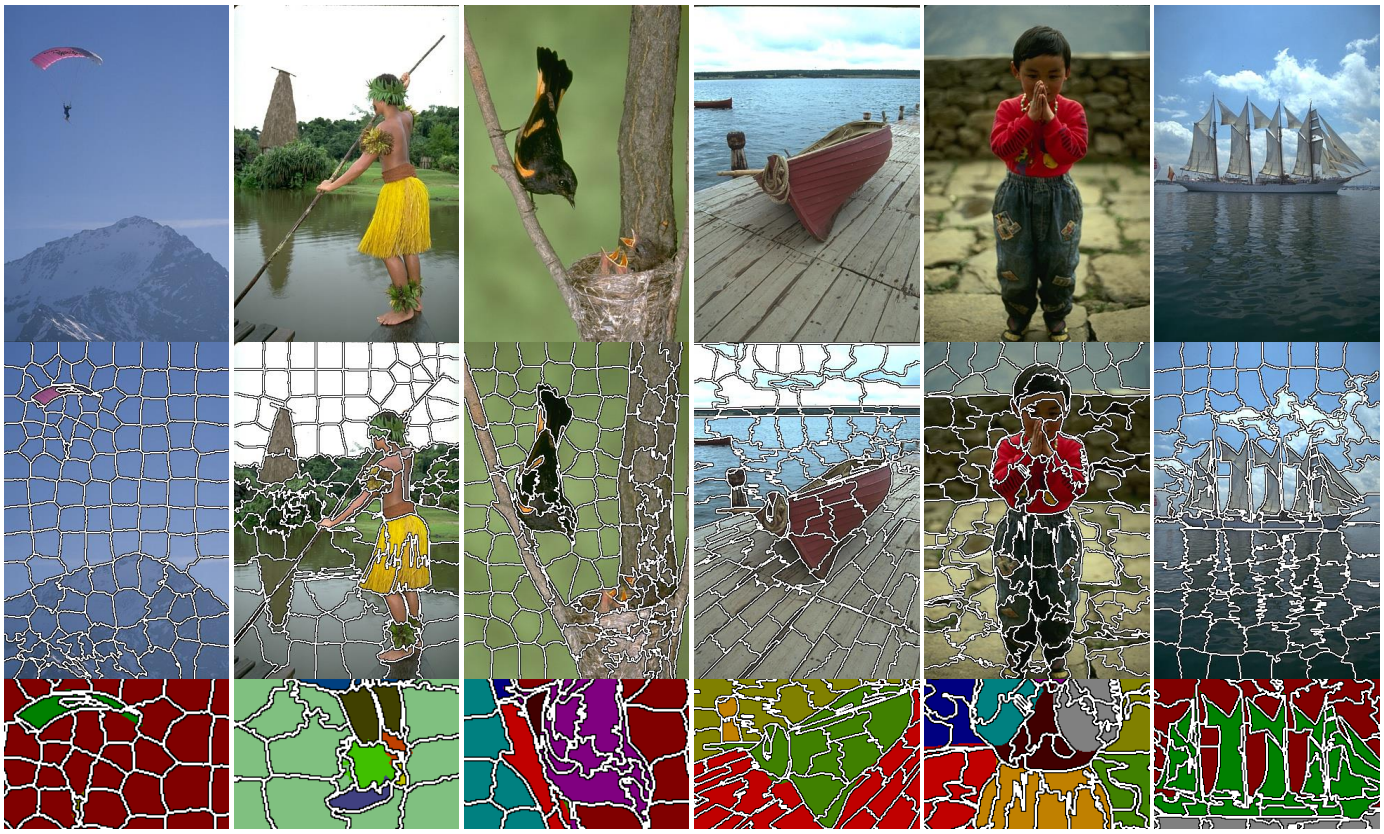


Figure 6. Visual results with approximate 100 CATS superpixels. Alternating rows show the zoom-in performance of ground truth covered by CATS superpixels.

Formally, let $T = \{T_m\}_{m=1}^{M}$ and $\Re = \{\Re_k\}_{k=1}^{K}$ be the ground truth and the generated superpixels of $\mathbf{I}$, respectively; the abovementioned metrics can be described as follows.

BR is versatile in evaluating the object segmentation and boundary detection. Mathematically, it is the ratio of ground truth boundaries covered by superpixel outlines [42].

$$\mathrm{BR}(T, \Re) = \frac{\sum_{p_i \in T_b} \Pi\left(\min_{p_j \in \Re_b} \left\|\mathbf{P}(p_i) - \mathbf{P}(p_j)\right\|_2 < r\right)}{T_b} \tag{14}$$

where T_b and $\Re_b$ represent boundary pixels in T and outline pixels in $\Re$, respectively. The indicator $\Pi(\cdot)$ returns the logic value whether the expression is true. The threshold r is set to 2 in this paper, which means that an outline pixel can be identified as an object boundary if there is a ground truth boundary pixel near it with the distance of 2 by Equation (3).

USE indicates the overall degree of one object overlapped with a superpixel [43]. Different from BR, it utilizes segmentation regions instead of boundaries for evaluation.

$$\mathrm{USE}(T, \Re) = \frac{\sum_{m=1}^{M}\left(\sum_{\Re_k | \Re_k \cap \Re_m \neq \phi} |\Re_k|\right) - N}{N} \tag{15}$$

As a result of the region over-segmentation, a superpixel region $\Re_k$ is theoretically able and only able to cover one object. Therefore, the smaller the USE, the fewer boundary pixels of objects that are mistakenly included in $\Re_k$.

ASA quantifies the achievable accuracy of subsequent steps such as image segmentation and target recognition [44]. It is the proportion of valid segmentation regions generated by superpixels in the image.

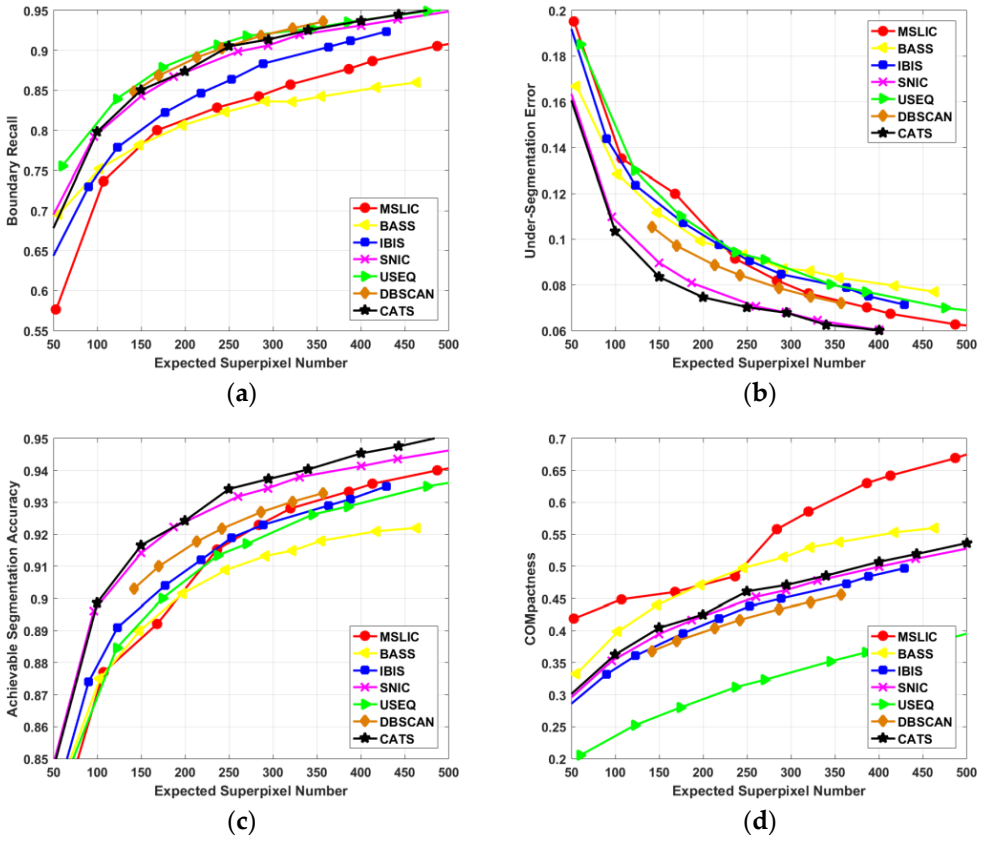


Figure 7. Quantitative evaluation of eight algorithms on the test subset of BSDS500. A higher metrical value corresponds to better outcomes with the exception of USE. (**a**) BR. (**b**) USE. (**c**) ASA. (**d**) COM.

$$\mathrm{ASA}(T,\Re)=\frac{\sum_{k=1}^{K}\mathrm{argmax}_m|\Re_k\cap T_m|}{\sum_{m=1}^{M}|T_m|} \tag{16}$$

If a superpixel only intersects with one ground truth region, the entire superpixel is considered as a valid segmentation region; conversely, if a superpixel intersects with multiple ground truth regions, the largest intersection is taken as the valid segmentation region. Therefore, when using superpixels as the basic units for image segmentation instead of individual pixels, the ASA value represents the upper limit of achievable accuracy.

COM can effectively describe the regularity of superpixel shapes [45]. Based on the isoperimetric theorem, the value of superpixel compactness is defined as the weighted isoperimetric quotient.

$$\mathrm{CO}(\Re)=\sum_{\Re_k}\frac{|\Re_k|}{N}\cdot Q(\Re_k), \tag{17}$$

Within k-th superpixel $\Re_k$, let $\boxed{\Re_k}$ and $\left|\boxed{\Re_k}\right|$ denote the collection of boundary pixels and the corresponding perimeter. The isoperimetric quotient is calculated by

$$\mathrm{Q}(\Re_k)=\frac{|\Re_k|}{\pi\left(\frac{\left|\boxed{\Re_k}\right|}{2\pi}\right)^2}, \tag{18}$$

The denominator of the right term is the area of a perfect circle with an equal perimeter, and its weight in global compactness represents the proportion of its own size to the overall image.

The BR curves are all depicted in Figure 7a. A higher BR metrical value indicates a closer approximation between the superpixel edge and the actual image boundaries. It is obvious that CATS performs on par with USEQ and DBSCAN, which are both top-rank in terms of BR. Meanwhile, with further increases in the number of superpixels, the performance may potentially improve. On the other hand, four iteration-demand algorithms exhibit relative lower performances, which is partly because of the incompatibility between segmentation accuracy and feature quality. The constraint of spatial compactness term might induce misclassification of unlabeled pixels during clustering, which is then accumulated gradually within the subsequent iterations.

Figure 7b displays how USE varies within each algorithm. In general, a lower value indicates a higher proportion of each superpixel intersecting with only a single object. It adopts region-level results to evaluate the segmentation accuracy. Apparently, CATS achieves the best performance among all algorithms. Owing to the splitting-based re-clustering, the under-segmented regions can be easily updated with new superpixels that avoid overlapping with multiple objects. USEQ acquires the best BR but performs backward on USE. Since BR evaluates the degree of ground truth boundaries detected by the superpixels that do not consider false detection, both USEQ and DBSCAN generate dense but not precise boundaries.

The difference of ASA is plotted in Figure 7c. A greater ASA value indicates a higher proportion of correctly segmented parts. The closer the segmentation result of an algorithm is to the ground truth, the better the segmentation effect. Compared with other algorithms, CATS established an early performance advantage even under a limited superpixel number. Intrinsically, it benefits from the advantages of online averaging clustering that is also utilized in SNIC; therefore, they are similar in the trend ASA curve. When the number increases, the variance of image content can be captured by CATS more easily; as a result, it gradually exceeds SNIC and becomes the top-most one.

Figure 7d compares the compactness among different algorithms as the number of superpixels increases. A higher COM value signifies greater compactness within each superpixel region. It can be observed that CATS lags behind MSLIC and BASS, whose performance on BR, USE and ASA are inferior among all algorithms. On the other side, by re-clustering the irregular superpixel candidates with complex image content, CATS optimizes the COM of SNIC to a certain extent. As shown in Figure 5, compared with the other two iteration-free methods (USEQ and DBSCAN), CATS superpixels are trimmer and clearer, providing a better visual quality. It is also worth noting that, as the most related work to CATS, SNIC makes a balanced and considerable trade-off among these metrics. Nevertheless, the grid-level seeding strategy ignores the uneven distribution of image information, resulting in the downside of clustering small objects with rich details. Conversely, CATS could overcome this inherent shortcoming through the centroid relocation strategy. Moreover, it shows stronger controllability on superpixel number (i.e., the production is approximate or identical to the expectation) since the merging process can dynamically adjust the amount of superpixel candidates.

Execution Time (ET) is measured as the average runtime of 200 selected images in the test subset of BSDS500. Table 1 illustrates the comparison of all algorithms on the same platform. For fair comparison, no multithread or parallel programming is utilized in each implementation (the multithreading in USEQ is manually disabled).

In theory, both MSLIC and BASS work in the conventional SLIC-like manner, whose time consumption comes into two aspects, the computation complexity in a single loop and the total iteration times. Despite baseline work on SLIC showing $O(N)$ complexity, the practical calculations of pixel-centroid correlation within the clustering process of MSLIC and BASS are more sophisticated than the linear Euclidean distance measurement. Accordingly, there is a heavy computation burden that lags an order of magnitude behind the running efficiency.

As an exception in the iteration-demand superpixel algorithm, IBIS benefits from the reduction of pixels participating in the calculation of distance from neighboring pixels

within each iteration. Consequently, it becomes the fastest method that even outperforms three iteration-free approaches. On the other hand, as shown in Figure 5, the improvement of IBIS in efficiency comes at the cost of mediocre segmentation accuracy and feature quality. In fact, IBIS merely maintains an approximate quality level as SLIC, which is proved to be significantly inferior than SNIC.

Table 1. Comparison of execution times among different algorithms (milliseconds). The italic numbers represent the lowest values.

Method	Expected Superpixel Number									
	50	**100**	**150**	**200**	**250**	**300**	**350**	**400**	**450**	**500**
MSLIC	166	171	178	180	180	178	186	185	189	190
BASS	190	213	235	244	260	272	279	289	295	299
IBIS	*21*	*20*	*18*	*18*	*18*	*18*	*18*	*18*	*18*	*18*
SNIC	33	33	33	33	34	34	34	34	34	34
USEQ	25	25	25	25	25	25	25	25	25	25
DBSCAN	-	-	-	31	31	30	30	29	29	29
CATS	40	37	37	37	37	37	37	36	37	38

It can be also observed that USEQ exhibits a stable and favorable ET in a wide range of expected superpixel numbers. Since all pixels are processed in USEQ through a one-pass pattern in the MAP estimation, its complexity is $O(N)$, which works strictly independent with the number of superpixels. Nevertheless, the extreme performance is not applicable to some high-precision requirements. Similar situations may also occur in DBSCAN, which is corroborated by Figure 5.

Notably, despite SNIC concatenating the process of assigning and updating the results in faster convergence than SLIC, it showcases inferior efficiency compared to USEQ and DBSCAN. This is partly because of the existence of redundant computations of correlation measurements on pixels between two adjacent superpixel regions. To maintain the simplicity, CATS follows the conventional online averaging clustering framework in SNIC even if it can be accelerated through several effective strategies [43]. As listed in Table 1, it costs an additional 10% runtime to further adjust the SNIC superpixel candidates, which is still 4 times faster than the other baseline MSLIC on average.

In summary, the proposed CATS superpixels exhibit desirable outperformance compared with other up-to-date algorithms, particularly in terms of under-segmentation error and compactness. To acquire content awareness, CATS performs fine-grained segmentation on regions with detailed context while merging smaller partitions with low region complexity. Consequently, as the number of superpixels increases, both interior homogeneity and boundary adherence improve and consistently remain at the forefront. Unlike the conventional grid-level seeding strategy, CATS superpixels achieve a desired number of superpixels without parameter adjustment. Additionally, it outperforms SNIC, IBIS, and DBSCAN and USEQ in terms of spatial compactness. Therefore, it can be employed as an advantageous pre-processing tool for various tasks, offering a comprehensive performance between feature performance and running efficiency.

5. Conclusions

This work proposes an optimized superpixel algorithm termed CATS to generate compact and regular superpixels with content adaptation. By leveraging the property of area elements in manifold feature space, a novel centroid splitting and merging operator is devised in CATS. A majority of seeds are then resampled based on image content density during the updating process. Accordingly, more seeds are allocated in regions with higher content density to generate smaller superpixels, while relatively fewer seeds are assigned in areas with lower content density to generate larger superpixels. In addition, with the advantage of label expansion patterns in the online averaging clustering framework, redundant pixel inspecting and correlation measurement operations can be effectively

avoided. Experimental results demonstrate that the proposed CATS achieves structure sensitivity with visual satisfaction, and delivers superior segmentation accuracy compared to other SOTA algorithms.

Future research will primarily focus on exploring strategies to further enhance the efficiency of CATS and apply it to several advanced computer vision tasks.

Author Contributions: Conceptualization and methodology, L.R. and N.L.; software and validation, N.L. and H.B.; formal analysis and investigation, L.R. and Z.Z.; resources, H.B. and Z.H.; data curation, Z.Z. and Z.H.; writing—original draft preparation, C.L.; writing—review and editing, L.R.; visualization, Z.Z., Z.H. and N.L.; supervision, L.R. All authors have read and agreed to the published version of the manuscript.

Funding: This research is supported financially by Photon Plan in Xi'an Institute of Optics and Precision Mechanics of Chinese Academy of Sciences (Grant No. S24-025-III) and Natural Science Basic Research Plan in Shaanxi province of China (Grant No. 2023-JC-QC0714).

Data Availability Statement: The BSDS500 dataset and the reference codes in this work are available at https://github.com/davidstutz/superpixel-benchmark. The source code of MSLIC is available at https://github.com/opencv/opencv_contrib/pull/923. The source code of BASS is available at https://github.com/arubior/bass-superpixels.git. The source code of DBSCAN is available at https://github.com/shenjianbing/Real-time-Superpixel-Segmentation-by-DBSCAN-Clustering-Algorithm-.git. The source code of SNIC is available at https://www.epfl.ch/labs/ivrl/research/snic-superpixels. The source code of USEQ is available at http://cvml.cs.nchu.edu.tw/USEQ.html. The source code of IBIS is available at https://github.com/xapha/IBIS.git. All URLs are accessed on 29 June 2024. Data underlying the results presented in this paper are not publicly available but can be obtained from the corresponding author upon reasonable request.

Acknowledgments: Thanks to the editor and anonymous reviewers for their suggestions and comments that ameliorate our work.

Conflicts of Interest: Author Zhe Huang was employed by the company Wuhan Second Ship Design and Research Institute. The remaining authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

References

1. Wang, S.; Lan, J.; Lin, J.; Liu, Y.; Wang, L.; Sun, Y.; Yin, B. Adaptive hypergraph superpixels. *Displays* **2023**, *76*, 102369. [CrossRef]
2. Ren, X.; Malik, J. Learning a Classification Model for Segmentation. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), Nice, France, 13–16 October 2003; pp. 10–17.
3. Diao, Q.; Dai, Y.; Wang, J.; Feng, X.; Pan, F.; Zhang, C. Spatial-pooling-based graph attention U-Net for hyperspectral image classification. *Remote Sens.* **2024**, *16*, 937. [CrossRef]
4. Huang, S.; Liu, Z.; Jin, W.; Mu, Y. Superpixel-based multi-scale multi-instance learning for hyperspectral image classification. *Pattern Recognit.* **2024**, *149*, 110257. [CrossRef]
5. Mu, Y.; Ou, L.; Chen, W.; Liu, T.; Gao, D. Superpixel-based graph convolutional network for UAV forest fire image segmentation. *Drones* **2024**, *8*, 142. [CrossRef]
6. Chen, G.; He, C.; Wang, T.; Zhu, K.; Liao, P.; Zhang, X. A superpixel-guided unsupervised fast semantic segmentation method of remote sensing images. *IEEE Geosci. Remote Sens. Lett.* **2022**, *19*, 2506605. [CrossRef]
7. Hu, K.; He, W.; Ye, J.; Zhao, L.; Peng, H.; Pi, J. Online visual tracking of weighted multiple instance learning via neutrosophic similarity-based objectness estimation. *Symmetry* **2019**, *11*, 832. [CrossRef]
8. Qiu, Y.; Mei, J.; Xu, J. Superpixel-wise contrast exploration for salient object detection. *Knowl. Based Syst.* **2024**, *292*, 111617. [CrossRef]
9. Zhang, D.; Xie, G.; Ren, J.; Zhang, Z.; Bao, W.; Xu, X. Content-sensitive superpixel generation with boundary adjustment. *Appl. Sci.* **2020**, *10*, 3150. [CrossRef]
10. Chuchvara, A.; Gotchev, A. Efficient image-warping framework for content-adaptive superpixels generation. *IEEE Signal Process. Lett.* **2021**, *28*, 1948–1952. [CrossRef]
11. Liao, N.; Guo, B.; Li, C.; Liu, H.; Zhang, C. BACA: Superpixel segmentation with boundary awareness and content adaptation. *Remote Sens.* **2022**, *14*, 4572. [CrossRef]
12. Sun, L.; Ma, D.; Pan, X.; Zhou, Y. Weak-boundary sensitive superpixel segmentation based on local adaptive distance. *IEEE Trans. Circuits Syst. Video Technol.* **2022**, *33*, 2302–2316. [CrossRef]
13. Li, C.; He, W.; Liao, N.; Gong, J.; Hou, S.; Guo, B. Superpixels with contour adherence via label expansion for image decomposition. *Neural Comput. Appl.* **2022**, *34*, 16223–16237. [CrossRef]

14. Uziel, R.; Ronen, M.; Freifeld, O. Bayesian Adaptive Superpixel Segmentation. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), Seoul, Republic of Korea, 27 October–2 November 2019; pp. 8470–8479.
15. Achanta, R.; Marquez, P.; Fua, P.; Susstrunk, S. Scale-Adaptive Superpixels. In Proceedings of the IS&T Color and Imaging Conference (CIC), Vancouver, BC, Canada, 12–16 November 2018; pp. 1–6.
16. Pan, X.; Zhou, Y.; Zhang, Y.; Zhang, C. Fast generation of superpixels with lattice topology. *IEEE Trans. Image Process.* **2022**, *31*, 4828–4841. [CrossRef] [PubMed]
17. Zhou, P.; Kang, X.; Ming, A. Vine spread for superpixel segmentation. *IEEE Trans. Image Process.* **2023**, *32*, 878–891. [CrossRef] [PubMed]
18. Kang, X.; Zhu, L.; Ming, A. Dynamic random walk for superpixel segmentation. *IEEE Trans. Image Process.* **2020**, *29*, 3871–3884. [CrossRef] [PubMed]
19. Giraud, R.; Ta, V.; Papadakis, N. Robust superpixels using color and contour features along linear path. *Comput. Vis. Image Underst.* **2018**, *170*, 1–13. [CrossRef]
20. Achanta, R.; Susstrunk, S. Superpixels and Polygons Using Simple Non-Iterative Clustering. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 4895–4904.
21. Liu, Y.; Yu, M.; Li, B.; He, Y. Intrinsic manifold SLIC: A simple and efficient method for computing content-sensitive superpixels. *IEEE Trans. Pattern Anal. Mach. Intell.* **2018**, *40*, 653–666. [CrossRef] [PubMed]
22. J, P.; Kumar, B.V. An extensive survey on superpixel segmentation: A research perspective. *Arch. Comput. Method Eng.* **2023**, *30*, 3749–3767. [CrossRef]
23. Xu, Y.; Gao, X.; Zhang, C.; Tan, J.; Li, X. High quality superpixel generation through regional decomposition. *IEEE Trans. Circuits Syst. Video Technol.* **2023**, *33*, 1802–1815. [CrossRef]
24. Achanta, R.; Shaji, A.; Smith, K.; Lucchi, A.; Fua, P.; Susstrunk, S. SLIC superpixels compared to state-of-the-art superpixel methods. *IEEE Trans. Pattern Anal. Mach. Intell.* **2012**, *34*, 2274–2282. [CrossRef]
25. Liu, Y.; Yu, C.; Yu, M.; He, Y. Manifold SLIC: A Fast Method to Compute Content-Sensitive Superpixels. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 651–659.
26. Hu, Y.; Li, Y.; Song, R.; Rao, P.; Wang, Y. Minimum barrier superpixel segmentation. *Image Vis. Comput.* **2018**, *70*, 1–10. [CrossRef]
27. Rubio, A.; Yu, L.; Simo-Serra, E.; Moreno-Noguer, F. BASS: Boundary-Aware Superpixel Segmentation. In Proceedings of the International Conference on Pattern Recognition (ICPR), Cancun, Mexico, 4–8 December 2016; pp. 2824–2829.
28. Xiao, X.; Zhou, Y.; Gong, Y. Content-adaptive superpixel segmentation. *IEEE Trans. Image Process.* **2018**, *27*, 2883–2896. [CrossRef] [PubMed]
29. Bobbia, S.; Macwan, R.; Benezeth, Y.; Nakamura, K.; Gomez, R.; Dubois, J. Iterative boundaries implicit identification for superpixels segmentation: A real-time approach. *IEEE Access* **2021**, *9*, 77250–77263. [CrossRef]
30. Zhao, J.; Hou, Q.; Ren, B.; Cheng, M.; Rosin, P. FLIC: Fast Linear Iterative Clustering with Active Search. In Proceedings of the AAAI Conference on Artificial Intelligence (AAAI), New Orleans, LA, USA, 2–7 February 2018; pp. 7574–7581.
31. Kesavan, Y.; Ramanan, A. One-Pass Clustering Superpixels. In Proceedings of the Conference on Information and Automation for Sustainability, Colombo, Sri Lanka, 22–24 December 2014; pp. 1–5.
32. Shen, J.; Hao, X.; Liang, Z.; Liu, Y.; Wang, W.; Shao, L. Real-time superpixel segmentation by DBSCAN clustering algorithm. *IEEE Trans. Image Process.* **2016**, *25*, 5933–5942. [CrossRef] [PubMed]
33. Huang, C.; Wang, W.; Lin, S.; Lin, Y. USEQ: Ultra-Fast Superpixel Extraction via Quantization. In Proceedings of the IEEE International Conference on Pattern Recognition (ICPR), Cancun, Mexico, 4–8 December 2016; pp. 1965–1970.
34. Huang, C.; Wang, W.; Wang, W.; Lin, S.; Lin, Y. USEAQ: Ultra-fast superpixel extraction via adaptive sampling from quantized regions. *IEEE Trans. Image Process.* **2018**, *27*, 4916–4931. [CrossRef] [PubMed]
35. Grady, L. Random walks for image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.* **2006**, *28*, 1768–1783. [CrossRef] [PubMed]
36. Vincent, L.; Soille, P. Watersheds in digital spaces: An efficient algorithm based on immersion simulations. *IEEE Trans. Pattern Anal. Mach. Intell.* **1991**, *13*, 583–598. [CrossRef]
37. Machairas, V.; Faessel, M.; Cardenas, D.; Chabardes, T.; Walter, T.; Decencière, E. Waterpixels. *IEEE Trans. Image Process.* **2015**, *24*, 3707–3716. [CrossRef] [PubMed]
38. Yuan, Y.; Zhu, Z.; Yu, H.; Zhang, W. Watershed-based superpixels with global and local boundary marching. *IEEE Trans. Image Process.* **2020**, *29*, 7375–7388. [CrossRef]
39. Zhong, D.; Li, T.; Dong, Y. An efficient hybrid linear clustering superpixel decomposition framework for traffic scene semantic segmentation. *Sensors* **2023**, *23*, 1002. [CrossRef]
40. Arbelaez, P.; Maire, M.; Fowlkes, C.; Malik, J. Contour detection and hierarchical image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.* **2011**, *33*, 898–916. [CrossRef] [PubMed]
41. Stutz, D.; Hermans, A.; Leibe, B. Superpixels: An evaluation of the state-of-the-art. *Comput. Vis. Image Underst.* **2018**, *166*, 1–27. [CrossRef]
42. Li, C.; Guo, B.; Wang, G.; Zheng, Y.; Liu, Y.; He, W. NICE: Superpixel segmentation using non-iterative clustering with efficiency. *Appl. Sci.* **2020**, *10*, 4415. [CrossRef]
43. Martin, D.R.; Fowlkes, C.C.; Malik, J. Learning to detect natural image boundaries using local brightness, color, and texture cues. *IEEE Trans. Pattern Anal. Mach. Intell.* **2004**, *26*, 530–549. [CrossRef]

44. Xu, L.; Luo, B.; Pei, Z.; Qin, K. PFS: Particle-filter-based superpixel segmentation. *Symmetry* **2018**, *10*, 143. [CrossRef]
45. Liu, M.; Tuzel, O.; Ramalingam, S.; Chellappa, R. Entropy rate superpixel segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Colorado Springs, CO, USA, 20–25 June 2011; pp. 2097–2104.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.