TECHNICAL REPORT 2023

IM SCIENCE

A comprehensive review and new taxonomy on superpixel segmentation

(Submitted – journal – CSUR)

07

TECHNICAL REPORT 2023

A comprehensive review and new taxonomy on superpixel segmentation

CITATION DETAILS:
Isabela Borlido Barcelos, Felipe de Castro Belém, Leonardo de Melo João, Zenilton Kleber Gonçalves do Patrocínio Jr., Alexandre Xavier Falcão, Silvio Jamil F. Guimarães (2023). A comprehensive review and new taxonomy on superpixel segmentation. Technical Report ImScience/PUC Minas #07/2023.

07

Technical report – ImScience 07/2023

A comprehensive review and new taxonomy on superpixel segmentation

Isabela Borlido Barcelos · Felipe de Castro Belém · Leonardo de Melo João · Zenilton Kleber Gonçalves do Patrocínio Jr. · Alexandre Xavier Falcão · Silvio Jamil F. Guimarães

June 8, 2023

Abstract Superpixel segmentation consists of partitioning images into regions composed of similar and connected pixels. Its methods have been widely used in many computer vision applications since it allows for reducing the workload, removing redundant information, and preserving regions with meaningful features. Due to the rapid progress in this area, the literature fails to catch up on more recent works among the compared ones and to categorize the methods according to all existing strategies. This work fills this gap by presenting a comprehensive review with new taxonomy for superpixel segmentation, in which methods are classified according to their processing steps and processing levels of image features. We revisit the recent and popular literature according to our taxonomy and evaluate 20 strategies based on nine criteria: connectivity, compactness, delineation, control over the number of superpixels, color homogeneity, robustness, running time, stability, and visual quality. Our experiments show the trends of each approach in pixel clustering and discuss individual trade-offs. Finally, we provide a new benchmark for superpixel assessment, available at https://github.com/IMScience-PPGINF-PucMinas/superpixel-benchmark.

Keywords superpixel · image segmentation · survey · image processing

1 Introduction

Superpixel segmentation aims to divide images into homogeneous regions of connected pixels, such that unions of superpixels compose image objects. It has several benefits, such as reducing the workload (e.g., reducing millions of pixels to thousands/hundreds of superpixels) and providing higher-level content information than pixels. Consequently, methods for superpixel segmentation are used in several applications, such as object segmentation [18,56, 87], anomaly detection [78] semantic segmentation [119], saliency detection [117,120], and image classification [28, 83]. Superpixel segmentation has a vast literature counting with benchmarks to evaluate and compare methods [72,99, 90,68]. Some works also categorize the methods according to a given taxonomy [1,90]. However, more recent approaches with different strategies still need to be covered, leading to the need for a new taxonomy. This work provides a comprehensive review of superpixel segmentation with a new taxonomy in which the methods are categorized based on processing steps and processing levels of image features. For that, we analyze current and classical approaches.

In the literature, several authors identify the superpixel desired properties. Despite the absence of consensus, most authors agree that superpixels must be composed of connected pixels, adhere to the objects' borders, have smooth contours, and have regularly distributed and compact shapes [90,99]. Moreover, the methods must be computationally efficient and generate a controllable number of superpixels. However, superpixel methods usually meet part of those criteria, which often occurs when the improvement in a property leads to worse for another property. In this sense, the choice of an evaluation measure depends on the optimized property.

In contrast to the rapid progress in new superpixel strategies, the papers usually compare their proposals against classical approaches. Therefore, there are few comparisons among state-of-the-art methods, which impairs the judgment of their actual contribution. Furthermore, despite the extensive comparisons in previous works [72,99,90,89,68], several recent approaches have not been included. This work fills this gap by evaluating 20 superpixel segmentation meth-

Silvio Jamil F. Guimarães
ImScience/PUC-Minas – Belo Horizonte 31980-110, Brazil
E-mail: sjamil@pucminas.br

ods among the most recently proposed and commonly used ones. Our assessment covers connectivity, compactness, delineation, color homogeneity, robustness, running time, stability, control over the number of superpixels, and visual quality. The results provide valuable insights into the pros and cons of the methods, supporting the choice of the most suitable one for a given application.

This paper is organized as follows. Section 2 presents the essential works that evaluate superpixel methods and introduce benchmarks. Subsequently, Section 3 describes the proposed taxonomy and categorizes the most recently proposed and commonly used superpixel methods. A brief description of the evaluated methods is presented in Section 4 and an extensive discussion covering 54 methods is presented in Appendix **??**. Section 5 defines the experimental setup, and Section 6 presents our results. Finally, we draw conclusions and possible future work in Section 7.

2 Related works

The first benchmark for superpixel evaluation [72] compared eight algorithms and evaluated object delineation and robustness to affine transformations. To overcome the biased penalty in *Under-segmentation Error* (UE) measure [52] caused by the superpixel size, the authors proposed a modified UE to consider the smallest part of the superpixel leakage. Also, the evaluated superpixel methods presented similar results, demonstrating that the most appropriate methods for each task depend on the crucial characteristics of that task. In addition, algorithms less focused on compactness showed greater robustness to image transformations. Unlike Neubert and Protzel [72], Achanta et al. [1] demonstrate the effectiveness of *Simple Linear and Iterative Clustering* (SLIC) by comparing five superpixel methods to determine their benefits and limitations regarding their boundary adherence and efficiency. Achanta et al. [1] characterized the superpixel methods as *graph-based* and *gradient-ascent-based*. The former contains methods that model the segmentation problem based on graph theory generating superpixels by minimizing a cost function defined on the graph. The second iteratively refines its initial clusters until reaching a convergence criterion.

Schick et al. [81,82] investigated the importance of compactness in superpixel segmentation. They proposed a compactness measure based on the isoperimetric coefficient [75] and demonstrated a trade-off between Compactness and Boundary Recall [67]. The authors argue that a more accurate segmentation would not imply better overall performance. Thus, they claim that compact superpixels better capture spatially coherent information facilitating information extraction from their boundaries.

A new benchmark was proposed in [89] with two image datasets and fifteen superpixel methods, including algorithms and datasets that use depth information. According to their evaluation, depth inclusion may not represent improved results. Regarding visual quality, the authors settled that the high quantitative results in the delineation assessment do not necessarily reflect the segmentations' visual quality. Mathieu et al. [68] argue that more than two datasets, as used in [89], are needed for an exhaustive evaluation. They overcome this with a new dataset, called the *Heterogeneous Size Image Dataset* (HSID). The HSID mainly contains large images (with millions of pixels) and allows evaluating the superpixel methods according to the image size. Using the HSID, the authors analyzed the five best superpixel methods in [89] and Waterpixels [63] method. The evaluated methods do not achieve a satisfactory trade-off between adherence to contours, conciseness (smallest possible number of superpixels), and efficiency. Therefore, the authors argue that the superpixel method must be chosen according to the necessary superpixels' characteristics for the desired task.

Wang et al. [99] proposed a regularity measure for superpixels, allowing the quantitative regularity analysis. The authors also provided an overview of the superpixel methods and a benchmark with fifteen state-of-the-art methods and thirteen evaluation measures, including the proposed one. In [99], the superpixel methods are categorized as clustering-based (or gradient-based) and graph-based, following the characterization in [1]. According to Wang et al. [99], methods based on clustering showed greater efficiency, while those based on graphs presented an improved delineation. However, the running time could have been better, and the authors settled that the evaluated algorithms are hardly applicable in scenarios requiring real-time responses.

The authors in [90] present a more comprehensive evaluation in a benchmark with 28 state-of-the-art superpixel algorithms with five datasets that include indoor, outdoor, and people images. In addition to the benchmark, the authors also propose evaluation measures independent of the number of superpixels and based on existing delineation metrics: *Average Miss Rate* (AMR), *Average Under-segmentation Error* (AUE), and *Average Unexplained Variation* (AUV). Stutz et al. [90] evaluated the stability of the methods, considering the minimum, maximum, and standard deviation of each metric; and its robustness to noise, blur, and affine transformations. Based on the categorization in [1], they also categorize superpixel methods by their high-level approach, allowing them to relate their categories to experimental results. Despite the broad categorization in [90], the authors settled that some methods in the literature are not included in their categorization. Based on the proposed evaluation, they create a ranking of the evaluated methods, and they recommend six of them: ETPS [109], SEEDS [14], ERS [59], CRS [24], ERGC [19], and SLIC [1].

3 Taxonomy of superpixel methods

The existing categorizations for superpixel methods need to be revisited to cover the wide variety of approaches. Also, the rapid advance in this area hampers the establishment of a strict categorization composed of disjoint categories. In this work, we establish the following statements for an appropriate categorization: (i) the set of categories must be sufficient to cover all methods; and (ii) each category must be specific enough to allow comparing and merging strategies.

To categorize superpixel methods, a taxonomy based on different and non-strict aspects may be more appropriate to establish well-defined categories that respect both previous statements. Therefore, instead of providing a strict set of categories, we establish a taxonomy that categorizes methods according to their processing steps and the level of abstraction of the characteristics used. In addition, we also report the desired superpixel properties that each method satisfies.

3.1 Processing steps

To provide a comprehensive taxonomy with a more realistic representation, we identified that superpixel algorithms generally have up to three steps: (i) initial processing; (ii) main processing; and (iii) final processing. In initial processing, the superpixel methods usually perform seed sampling, an initial image partition, or feature extraction. The main processing contains the strategy for superpixel computation. We avoid dividing a loop into different processing steps. Therefore, the main processing includes the whole loop for superpixel generation, if any. Finally, post-processing operations are usually performed at the final step to ensure superpixel connectivity or to fine-tune the segmentation.

The processing steps divide superpixel approaches into specialized procedures, allowing their comparison. We identify categories that broadly define the process performed at each processing step in 52 superpixel segmentation methods. The categories shown in Table 1 identify the main processing of the analyzed methods that do not use neural networks. These categories were defined based on the main processing to obtain superpixels.

In the superpixel literature, some proposals encompass neural network architectures. In our analysis, the only neural architecture identified was the convolutional one. We classified the network in those methods according to its architecture and output. Moreover, these networks may not produce superpixels directly, relying on a differential clustering module for this [121, 105, 98] or only performing feature extraction [32, 74]. Therefore, one could classify their purpose as (i) feature extraction, (ii) segmentation with a differential module, or (ii) superpixel segmentation. Superpixel segmentation is also used as input of convolutional networks for segmentation refinement. However, methods with this approach perform object segmentation [44, 43, 16, 57].

3.2 Processing level of image features

Superpixel methods can compute features on-the-fly or obtain them from other algorithms. Several approaches extract information from the same features differently. For example, some methods combine local features (*e.g.*, color and pixel position) with higher-level ones (*e.g.*, edge or semantic information) in their optimization function [116, 13, 102]. On the other hand, others can extract abstract knowledge (*e.g.*, using strategies based on graph theory or linear algebra) by exploring only local information [21, 12, 31]. Despite this, for superpixel segmentation, there was no study in this regard. Such a study is beyond the scope of this work. However, the taxonomy proposed here also categorizes the methods according to the processing level of the characteristics used. To categorize a superpixel method based on the processing level of the image features, we assign the highest level used. The categories were defined as follows:

- **Pixel-level features:** Raw data resources in images — *e.g.*, pixel color, position, and depth;
- **Mid-level features:** features that can be computed based on a set of pixels, smaller than the entire image — *e.g.*, patch-based feature, path-based feature, gradient, or boundary;
- **High-level features:** features that combine pixel properties and high-level information. The high-level information cannot be extracted from a small set of pixels. They are given directly by the user or predicted by other models — *e.g.*, saliency map, semantic features, texture, or a desired object geometry;

3.3 The proposed taxonomy in superpixel literature

Table 2 presents superpixel methods categorized by the proposed taxonomy, color space, and inspiration method. In the processing steps, we specified categories according to the high-level purpose. In superpixel approaches whose paper's proposal contains Convolutional Neural Networks (CNN), we inform the network architecture (*arch*) and its output (*out*).

According to our analysis (in Table 2), most superpixel methods do not have CNNs in any processing step and perform *Seed sampling* in their initial processing. Furthermore, most networks produce superpixels directly [91, 110, 108, 97, 112]. Conversely, others rely on a differential clustering module [98, 105, 121, 115] or only perform feature extraction, requiring further superpixel clustering [74, 32]. The *Clustering Method* category in Table 2 indicates processing

Table 1 Main processing categories excluding methods based on neural networks

Categories	Explanation
Neighborhood-based clustering	Performs clustering based on the similarity between pixels restricted to a maximum spatial distance from some reference point in the image.
Boundary evolution clustering	These algorithms iteratively update the superpixels' boundaries to improve their superpixels, usually using a coarse-to-fine image block strategy.
Dynamic-center-update clustering	The dynamic-center-update algorithms perform clustering with a distance function based on the features of the clusters, dynamically updating their centers.
Path-based clustering	Path-based approaches generate superpixels by creating paths in the image graph based on some criteria. Usually, its clustering criterion is a path-based function to optimize during clustering.
Hierarchical clustering	These algorithms create regions in the image that form a hierarchical structure, obeying the criteria of locality and causality [41].
Density-based clustering	The superpixel methods rely on an optimization function to find the cluster centers, modeling the problem of finding superpixels in a problem of finding density peaks.
Sparse linear system clustering	Model the segmentation problem with a sparse matrix and use its properties to find superpixels.
Data distribution-based clustering	The approach assumes that the image pixels follow a specific distribution and perform the clustering based on this conjecture.
Regional feature extraction	Iteratively extracts regional features to perform clustering based on these features.
Polygonal decomposition clustering	Segmentation in these methods consists of decomposing the image into non-overlapping polygons.
Graph-based clustering	Perform superpixel segmentation based on graph topology

steps that use another superpixel method. Regarding the CNN architectures, most networks have a Fully Convolutional Network (FCN) [98,110,91] or an Encoder-Decoder [32, 108,97]. However, there are other architectures, such as an Interpolation Network [112], a Multiscale CNN [74,115], and a Weight-shared CNN [105].

Regarding methods without CNNs, SLIC or SLIC's variants inspired most of them. Furthermore, the most common main and final processings are the *Neighborhood-based clustering* and *Merging step*, respectively. Nevertheless, *Boundary evolution*, *Dynamic-center-update*, and *Path-based clustering* are also frequent. The Boundary Evolution strategy is usually the most efficient since it updates only the superpixels' boundaries, which may enforce their connectivity [17,14,24,109,106,113,76]. Similarly, most approaches with a Dynamic-center-update clustering guarantee connectivity during the clustering process, usually using a priority queue to find the best candidates for each superpixel [2, 35,48,54,61,118,100]. Path-based clustering methods iterate over pixels similarly, but their superpixels are spanning trees that optimize a path-based function [19,94,20,12,13, 11]. Most of these methods are based on the Image Foresting Transform (IFT) framework or its variants. In this work, we also consider *Hierarchical* approaches since they produce segmentations that conform to the abovementioned properties for superpixels [31,25,74,103]. Methods that perform Hierarchical clustering have the advantage of computing all hierarchy levels in a single execution, generating multiple segmentations for the same image.

4 Superpixel segmentation methods

Superpixel segmentation has a vast literature covering several techniques. In [90] a benchmark for superpixels is provided with an extensive evaluation of methods. Nevertheless, due to the rapid progress in developing new strategies for superpixel segmentation, an analysis of the most recent proposals becomes essential. This section reviews recent and commonly used literature on superpixel segmentation. For an extensive review, see Appendix **??**.

4.1 Neighborhood-based clustering

Neighborhood-based methods for superpixel segmentation perform clustering of image pixels based on the similarity between pixels restricted to a maximum spatial distance from some reference point in the image. For example, several methods constrain the clustering region of a superpixel to a fixed-size image patch around this superpixel [1,93,104, 58].

4.1.1 SLIC

SLIC [1] starts with a grid sampling of superpixel centers and iteratively assigns to each superpixel the most similar

Table 2 Recent methods for superpixel segmentation

Method	Iterative	#Iter.	#Superp.	Connec.	Compact.	Superv.	Color	Time complexity	Initial processing	Main processing	Final processing	Features: Pix.	Features: Mid.	Features: High.	Inspired
SLIC [1]	✓	✓	✓	✓ [2]	✓		CIELAB		Seed sampling	Neighborhood-based clustering	Merging step	✓			
LSC [55]	✓	✓	✓	✓ [2]	✓		CIELAB	$O(kn + nz)$ [8]	Seed sampling	Neighborhood-based clustering	Merging step	✓			
SCALP [34]	✓	✓	✓	✓	✓		CIELAB		Seed sampling	Neighborhood-based clustering			✓		SLIC [1]
TASP [104]	✓	✓	✓				CIELAB		Seed sampling	Neighborhood-based clustering			✓		SLIC [1]
MFGS [58]			✓ [1]	✓	✓		CIELAB		Seed sampling	Neighborhood-based clustering	Merging step		✓		SLICO [1]
DSR [116]	✓		✓		✓		CIELAB		Seed sampling	Neighborhood-based clustering	Merging step			✓	dSLIC [64]
Semasuperpixel [102]	✓	✓	✓	✓ [2]		✓	CIELAB		*arch:* Encoder-decoder *out:* Semantic map	Neighborhood-based clustering	Merging step			✓	SLIC [1]
AWkS [42]	✓	✓	✓				CIELAB		Seed sampling	Neighborhood-based clustering	Merging step	✓			W-k-means [45]
IBIS, IBIScuda [17]	✓		✓	✓ [2]	✓		CIELAB	$O(n)$	Grid segmentation	Boundary evolution clustering	Merging step	✓			SLIC [1]
SEEDS [14,15]	✓	✓	✓		✓		CIELAB		Grid segmentation	Boundary evolution clustering		✓			
CRS [24]	✓	✓	✓	✓	✓		YCrCb		Grid segmentation	Boundary evolution clustering			✓		CR [39,69]
ETPS [109]	✓	✓	✓	✓	✓		RGB		Grid segmentation	Boundary evolution clustering		✓			SEEDS [14]
CFBS [106]	✓		✓	✓	✓		CIELAB		Grid segmentation	Boundary evolution clustering		✓			SLIC [1]
SCAC [113]			✓ [1]	✓	✓		CIELAB		Grid segmentation	Boundary evolution clustering	Boundary evolution clustering		✓		WSBM [114]
LSC-Manhattan [76]	✓		✓	✓	✓				Classification	Boundary evolution clustering				✓	LSC [21]
SNIC [2]			✓	✓	✓		CIELAB	$O(n)$	Seed sampling	Dynamic-center-update clustering		✓			SLIC [1]
CONIC [35]			✓	✓	✓		CIELAB	$O(n)$	Seed sampling	Dynamic-center-update clustering			✓		SNIC [2], SCALP [34]
DRW [48]			✓	✓				$O(n)$	Seed sampling	Dynamic-center-update clustering	Label propagation		✓		RW [37]
FCSS [54]	✓	✓ [1]	✓	✓ [2]	✓		CIELAB	$O(n + nt)$ [4]		Dynamic-center-update clustering		✓			SNIC [2]
F-DBSCAN [61]			✓	✓			CIELAB	$O(n)$		Dynamic-center-update clustering		✓			RT-DBSCAN [36]
SCBP [118]			✓	✓	✓		RGB	$O(n)$		Dynamic-center-update clustering	Merging step		✓		DBSCAN [86]
A-DBSCAN [100]			✓	✓	✓		RGB	$O(n)$	Compute features	Dynamic-center-update clustering	Merging step			✓	DBSCAN [86]
ERGC [19]				✓	✓		CIELAB		Seed sampling	Path-based clustering			✓		
ISF [94]	✓	✓	✓	✓	✓		CIELAB	$O(n \log n)$	Seed sampling	Path-based clustering			✓		IFT [27]
RSS [20]			✓	✓	✓			$O(n)$	Seed sampling	Path-based clustering			✓		IFT [27]
DISF [12]	✓		✓	✓			CIELAB	$O(n \log n)$	Seed oversampling	Path-based clustering			✓		ISF [94]
ODISF [13]	✓		✓	✓			CIELAB	$O(n \log n)$ [6]	Seed oversampling	Path-based clustering				✓	DISF [12], OISF [10]
SICLE [11,9]	✓	✓ [1]	✓	✓			CIELAB	$O(n \log n)$ [6]	Seed oversampling	Path-based clustering				✓	ODISF [13]
SH [103]			✓	✓			RGB	$O(n)$		Hierarchical clustering			✓		
UOIFT [8]			✓	✓			CIELAB		Clustering method	Hierarchical clustering			✓		IFT [27], OIFT [65]
HMLI-SLIC [25]	✓	✓	✓ [1]	✓	✓		CIELAB	$O(nd)$ [5]	Clustering method	Hierarchical clustering	Merging step	✓			SLIC [1]
RISF [30,31]	✓	✓	✓	✓	✓		CIELAB			Hierarchical clustering	Hierarchical region merging		✓		ISF [94]
DAL-HERS [74]			✓	✓		✓	RGB	$O(n)$ [3]	*arch:* Multi-scale Residual CNN *out:* Affinity map	Hierarchical clustering				✓	SEAL [92], ERS [59]
PGDPC, SLIC-PGDPC [38]			✓	✓			CIELAB	$O(n \log n)$	Seed sampling	Density-based clustering			✓		DPC [95]
DPS [85]			✓ [1]				CIELAB		Compute features	Density-based clustering	Clustering method		✓		DP [80]
ANRW [96]			✓	✓			CIELAB	$O(n^2)$	Seed sampling	Sparse linear system clustering				✓	NRW [111]
GL$l_{1/2}$RSC [29]	✓		✓						Clustering method	Sparse linear system clustering	Encoding procedure		✓		CAWR [101]
SCSC [53]	✓	✓	✓				RGB		Clustering method	Sparse linear system clustering	Clustering method			✓	
EAM [4]			✓ [1]	✓			RGB	$O(\log^2 n)$	Noise remotion	Regional attributes extraction	Merging step		✓		
ECCPD [62]	✓	✓	✓	✓			RGB		Seed sampling	Polygonal decomposition clustering	Boundary evolution clustering		✓		
GMMSP [5] gGMMSP [6]	✓	✓	✓ [1]	✓ [2]	✓		CIELAB	$O(n)$ [7]		Data distribution-based clustering	Merging step		✓		SCGAGMM [47]
ERS [59]				✓	✓		RGB			Graph-based clustering		✓			
E2E-SIS [98]			✓	✓ [2]		✓	CIELAB			*arch:* FCN *out:* Superpixels	Superpixel pooling layer and merging step			✓	DEL [60], SSN [46]
ss-RIM [91]			✓ [1]				RGB			*arch:* FCN *out:* Image reconstruction and Superpixels				✓	DIP [51], RIM [50]
EW-RIM [110]			✓	✓	✓		RBG			*arch:* FCN *out:* Image reconstruction and Superpixels				✓	ss-RIM [91], DIP [51]
SEN [32]			✓				RGB		*arch:* Encoder-Decoder *out:* Deep embeddings	Clustering method				✓	RPEIG [49]
SSFCN [108]			✓ [1]			✓	CIELAB			*arch:* Encoder-Decoder *out:* Superpixels	Merging step			✓	SSN [46]
SENSS [97]			✓ [1]	✓	✓	✓	CIELAB			*arch:* Encoder-Decoder *out:* Superpixels				✓	SSFCN [108]
DAFnet [105]			✓	✓		✓	CIELAB			*arch:* Weight-shared CNN *out:* Superpixels				✓	SSFCN [108]
LNS-net [121]			✓		✓		LAB/RGB			*arch:* FCN *out:* Image reconstruction and Superpixels	Merging step			✓	
SIN [112]			✓ [1]	✓		✓				*arch:* Interpolation Network *out:* Superpixels				✓	
BP-net [115]			✓			✓	RGB-D		Seed sampling	*arch:* Multi-scale CNN and FCN *out:* Boundary map and superpixels	Merging step			✓	

[1] Partially, [2] With post-processing, [3] Time complexity in HERS module, [4] t is the number of relocations, [5] d is the number of hierarchy levels, [6] Without the saliency map computation, [7] Without parallelization, [8] k is the number of iterations and z represents the number of small isolated superpixels to be merged.

pixels in a limited region around the superpixel center. As post-processing, SLIC ensures connectivity by assigning unconnected superpixels to their nearest neighbors. SLIC reduces the segmentation complexity to linear concerning the number of pixels. Also, its distance function gives better control over the superpixel size and compactness. Although SLIC presents fair delineation and efficiency, it does not consider the relationship between adjacent pixels, resulting in worse delineation in regions with complex textures.

4.1.2 LSC

The authors [55, 21] investigated the relationship between the normalized cuts [79] and the weighted K-means to propose the LSC, which uses an NCut function that can obtain the same optimum result as the weighted kernel K-means. The LSC applies a kernel function to map pixels into a 10-dimensional feature space in a fixed limited region. LSC provides an efficient segmentation method and it obtains regular shapes. It also has linear time complexity with high memory efficiency. By considering a shape constraint, LSC achieves high boundary adherence without sacrificing spatial compactness. However, its fixed search range prevents LSC from ensuring connectivity, requiring post-processing.

4.1.3 SCALP

SCALP [33] considers image features and contour intensity on a linear path to the superpixel barycenter to improve SLIC's distance function with neighborhood information. It integrates the contour prior information as a soft constraint in the color distance to improve the adherence to the object boundaries and performs clustering in high-dimensional feature space [55]. SCALP is efficient, robust to noise, and produces compact superpixels. The authors further improve SCALP [34] with a hard constraint based on the contour prior to providing an initial segmentation. The hard constraint increases SCALP's robustness and its boundary adherence, but it slightly reduces regularity and smoothness.

4.2 Boundary evolution clustering

In boundary evolution clustering, the algorithm iteratively updates the superpixels' boundaries to improve delineation, usually using a coarse-to-fine image block strategy. SEEDS [14] and ETPS [109] are examples of superpixel methods using the boundary evolution strategy for clustering.

4.2.1 SEEDS

SEEDS [15] start from a regular grid partitioning and iteratively refine the superpixels' boundaries. The iterative process follows a coarse-to-fine approach with a hill-climbing algorithm for optimization. SEEDS is an efficient method that performs optimization based on a hill-climbing algorithm. SEEDS introduces an energy function that encourages color homogeneity, shape regularity, and smooth boundary shapes. However, the compactness constraint degrades the results, and the number of superpixels is challenging to control.

4.2.2 CRS

CRS [24] formulates the segmentation problem as an estimation task and transforms the model in [70, 40] into a superpixel approach. From an initial image partition, CRS generates superpixels under the constraint of maximum texture homogeneity inside of each image patch and maximum accordance of the contours with both the image content and a Gibbs-Markov random field model. CRS explicitly models the superpixel's shape and content as a statistical model, allowing it to handle an arbitrary number of feature channels. In addition, CRS allows direct control of the number of superpixels and their compacity.

4.2.3 ETPS

Inspired by SEEDS [14], ETPS [109] performs a coarse-to-fine approach to superpixel segmentation, starting from a grid partitioning. ETPS uses a priority list to optimize its energy function. Also, despite its energy function being at the pixel level, it measures shape regularization, color homogeneity, and smoothness of the contours. In addition, ETPS enforces connectivity and minimum size during the optimization process. The authors also presented a stereo version of the proposal and demonstrate that ETPS' efficiency surpasses SLIC [1]. Compared to [107], ETPS achieves a better convergence value in a single iteration.

4.2.4 IBIS

IBIS [17] starts with a grid segmentation and, using a SLIC's distance measure [1], compares the pixels located on the edge of the blocks, subdividing in 4 those blocks assigned to another superpixel. At each iteration, pixels in non-homogeneous blocks are assigned to the nearest superpixel according to the SLIC's distance measure. After the clustering step, IBIS performs the same merging stage as SLIC. The paper also presents a GPU variant aimed at real-time use cases, the IBIScuda. IBIS is faster than other methods and achieves similar results as SLIC. Also, its Cuda version can even improve its efficiency, reducing computational time. However, similar to SLIC, the IBIS's boundary adherence and accuracy are not competitive with other methods in the literature.

4.3 Dynamic-center-update clustering

The dynamic-center-update algorithms perform clustering with a distance function based on the features of the clusters, dynamically updating its centers. Unlike neighborhood-based clustering, this approach does not perform a limited regional search to calculate distances.

4.3.1 SNIC

SNIC [2] intends to overcome SLIC's limitations. The proposal starts with a sampling grid, but it dynamically updates the centroids during the clustering process. Furthermore, instead of searching limited to an image patch, SNIC uses a priority queue to group neighboring pixels — similar to a path-based approach, but with a distance function based on the superpixel centroid. Due to its clustering process based on neighboring pixels, SNIC enforces connectivity without requiring post-processing. Furthermore, SNIC requires less memory and is computationally more efficient than SLIC. The authors also proposed an algorithm for polygonal segmentation called SNICPOLY, which starts with superpixels generated with SNIC.

4.3.2 DRW

The DRW [48] model uses dynamic nodes, which reduces the redundant calculation by limiting the walking range. The proposed algorithm performs a new seed initialization strategy that creates a seed set with regular distribution in both 2D and 3D and can combine boundary prior information, such as gradient information or boundary probability [67]. DRW computes superpixels in linear time and allows control of the distribution of superpixels in complex and homogenous image regions. The proposed segmentation method has competitive performance and it is faster than existing RW models. However, DRW segmentation does not produce compact superpixels.

4.4 Path-based clustering

Path-based approaches generate superpixels by creating paths in the image graph based on some criteria. Usually, their clustering criteria are a path-based function to optimize during clustering. The ISF [94] is an example of a path-based method that calculates a forest of optimal paths based on a path cost function.

4.4.1 ERGC

First, the proposed ERGC [19] simplifies Computed Tomography (CT) images by computing superpixels based on the Eikonal algorithm. The superpixels start from seeds sampled in a regular grid and evolve according to the Fast Marching algorithm [84]. ERGC creates homogeneous superpixels with a spatial constraint to enforce compactness. The proposal demonstrated more efficiency and effectiveness than other methods of its period, in addition to being extensible to supervoxels and allowing control over the number of superpixels and compactness.

4.4.2 ISF

Based on IFT [27], the ISF [94] framework combines a seed sampling strategy, a connectivity function, an adjacency relation, and a seed recomputation procedure. The proposal's algorithm starts with (i) a seed sampling, followed by (ii) a spanning forest computed by the IFT algorithm, and (iii) a seed recomputation procedure. The ISF refines the segmentation by iteratively executing steps (ii) and (iii). The computational complexity of the ISF framework using a binary heap is linearithmic, independent of the number of superpixels. The computational cost can be reduced by using the *Differential Image Foresting Transform* (DIFT) [26,23,22] to compute the IFTs. However, the DIFT's effectiveness depends on the cost function used. In [94], the authors combine different components to present five ISF-based methods. They also demonstrated that ISF produces effective and efficient methods independent of the dataset.

4.4.3 RSS

The RSS [20] method follows the IFT [27] algorithm and can form a forest with optimal costs. To measure color similarity and spatial closeness, the authors proposed two path-based cost functions, that have proven to be more robust than the geodesic distance. Inspired by counting sort and bucket sort, the RSS computes optimal forest with buckets of queues and groups of seeds in an IFT [27]-based algorithm. Due to the sorting strategy, the proposal has $O(N)$ complexity. The proposal is fast and has competitive performance. The main strengths of RSS are the low computational complexity, great boundary adherence with stable performance, and adjustable compactness. However, besides the proposal extends to supervoxel segmentation, it performs poorly compared with the evaluated methods. Also, due to the initial seed sampling in a regular grid [1], RSS generates more superpixels in homogenous regions, which leads to a degrading in boundary adherence in complex regions.

4.4.4 DISF

Based on ISF [94], DISF [12] is a three-step superpixel framework that improves its delineation even for fewer superpixels. The proposal initializes with a seed over-segmentation that performs grid sampling [1] for a high number of seeds. Then, iteratively compute a forest rooted at the seeds with an IFT [27] execution followed by a seed set reduction by choosing the most relevant seeds. The IFT computation and seed set reduction are repeated until achieves the desired number of superpixels. DISF has an optimal delineation, especially for a few numbers of superpixels. Therefore, the proposal's segmentation is able to correctly selects relevant seeds, reducing its boundary adherence degradation when decreasing the number of final superpixels. Despite its iterative process increasing the running time, DISF performs a reduced and limited number of iterations. However, the proposal does not produce compact superpixels.

4.4.5 ODISF

Motivated by OISF [10] performance, ODISF [13] extends DISF [12] for an object-based proposal to improve the superpixel performance using object saliency maps. The proposal performs the same three-step pipeline in DISF. First, the ODISF performs a seed oversampling. Then, it iteratively computes a spanning forest rooted at the seed set with an IFT [27] execution followed by an object-based seed removal. In the remotion step, the algorithm maintains seeds closer to the object saliency boundaries or with higher saliency. The saliency maps were created using a U2-net [77]. The proposed method demonstrates a generalization ability by performing an effective superpixel segmentation in datasets with different object properties. Also, it demonstrates robustness to saliency map errors in comparison with OISF. Despite the ODISF delineation step being saliency-independent, its object-based removal strategy can circumvent the saliency errors. On the other hand, the ODISF does not allow controlling the number of iterations. Also, despite its computational complexity, it has a high running time, being faster only than the OISF.

4.4.6 SICLE

SICLE [11] generalizes ODISF [13] to control the number of iterations and to improve efficiency and delineation for poorly estimated saliency maps. SICLE starts with an (i) seed oversampling; and iteratively generates superpixels by (ii) computing the minimum forest rooted at the seed set [27], followed by the (iii) removal of the less relevant seeds. Similar to ODISF, SICLE incorporates saliency information during the seed removal step being robust to incorrect saliency estimations. However, SICLE's seed removal strategy allows controlling the number of iterations and avoids unnecessary iterations, improving efficiency. Since SICLE uses object information only on the removal step, its delineation is robust to saliency errors. However, SICLE cannot improve its delineation performance for more accurate saliency estimators. The authors overcome this drawback in [9], by encompassing a path cost function and a seed removal strategy to control the impact of object saliency information using a binary parameter. The proposal maintains its robustness for low-quality estimators and exploits the accurate information of high-quality estimators, improving performance with only two iterations. Despite the robustness and efficiency of SICLE, errors in the saliency map can still affect its results.

4.5 Hierarchical clustering

Hierarchical segmentation methods are generally not mentioned in the literature as superpixel methods. However, they fit most definitions for superpixels. Although hierarchical methods do not obtain a compact or regular segmentation, the regions produced are generally homogeneous. Furthermore, from the generated hierarchy, it is possible to control the desired number of regions without increasing the execution time.

4.5.1 SH

SH [103] uses the Borůvka algorithm to efficiently compute a minimum spanning tree (MST) in a bottom-up manner representing a hierarchy. To improve efficiency, SH uses edge contraction, contracting each tree to a vertex and recording the edge selection order. Also, to improve accuracy with local searching, SH incorporates edge information from an edge detector and combines it with color information. In experiments, SH achieved high accuracy and low computational time. The authors also demonstrate the SH's effectiveness in saliency detection, semantic segmentation, and stereo-matching. However, SH does not produce regular superpixels.

4.5.2 DAL-HERS

DAL-HERS [8] is a two-stage superpixel framework that consists of a *Deep Affinity Learning* (DAL) neural network architecture and a *Hierarchical Entropy Rate Segmentation* (HERS) method. The DAL network aggregates multiscale information to learn pairwise pixel affinities, and the HERS method builds a hierarchical tree structure by maximizing the graph's entropy rate. Using the DAL's affinity map, the proposed HERS algorithm constructs a hierarchy with Borůvka's algorithm [103]. The proposal preserves fine details on the objects by focusing on rich-structure parts rather than uniform regions, producing large superpixels in homogeneous regions and an over-segmentation in texture-rich regions. Also, compared with deep-based learning methods, the DAL-HERS running time is competitive, and it requires the same $O(N)$ time to produce any number of superpixels. Due to the highly adaptive nature of the produced superpixels, delineating fine details, the proposal has a low ASA score.

4.6 Graph-based, Data distribution-based, and CNN-based clustering

In superpixel segmentation, we name data distribution-based methods the approaches that assume that the image pixels follow a specific distribution. From this initial conjecture, the clustering step is performed. As far as we know, the distribution-based methods that perform superpixel segmentation are based on the gaussian mixture model and assume that the image pixels follow a Gaussian distribution. Conversely, Graph-based clustering methods perform superpixel segmentation based on graph topology. Finally, we present LNSNet which uses a Lifelong learning strategy for superpixels generation.

4.6.1 ERS

ERS [4] is a greedy algorithm that efficiently computes the entropy rate of a random walk on the image graph. The ERS's objective function is composed of an entropy rate term and a balancing term of the cluster distribution. While the entropy rate favors compact and homogeneous clusters, the balancing term encourages clusters with similar sizes. The authors demonstrated that ERS outperforms other methods in accuracy with less pixel leakage. The balancing term in ERS produces superpixels more similar in size and enforces control over the number of superpixels. However, they are irregular in shape.

4.6.2 GMMSP

GMMSP [5] models superpixel segmentation as a weighted sum of Gaussian functions, each one corresponding to a superpixel. The proposal produces superpixels of similar size by using a constant weight for the weighted sum of Gaussians. It also imposes two parameters during the expectation-maximization iterations to prevent singular covariance matrices and control superpixel regularity. GMMSP has a reduced computational complexity by using only a subset of pixels to estimate the parameters of a Gaussian function. The proposal has well-balanced accuracy and regularity but does not allow direct control over the number of superpixels. Also, GMMSP may produce irregular superpixels on strong gradient regions.

4.6.3 LNSNet

LNSNet [121] is an unsupervised CNN-based method that learns superpixels in a lifelong manner. It is composed of three major modules: a feature embedder module (FEM), a gradient rescaling module (GRM), and a non-iterative clustering module (NCM). FEM embeds the original feature into a cluster-friendly space. The NCM uses the embedded features to estimate the optimal cluster centers and assigns pixel labels based on similarity. Finally, the GRM solves the forgetting caused by lifelong learning during the backpropagation step using a Gradient Adaptive Layer (GAL) and a Gradient Bi-direction Layer (GBL). LNSNet demonstrates a high generalization capacity and generates competitive superpixels using less complex and computationally faster architecture. However, the proposal has some drawbacks. First, the proposed model cannot reach a complete convergence, due to the sequential training strategy, requiring post-processing to remove trivial regions. Secondly, GBL's boundary map may contain noises and lead to irregular superpixels when facing a background with a complex texture. Finally, the clustering step requires a distance matrix, which is inefficient when calculated by a CPU with a large number of superpixels.

5 Experimental setup

5.1 Methods and datasets

In this work, we identified 14 open source codes from the recent superpixel literature: DISF [12], RSS [20], ODISF [13], IBIS [17], DRW [48], DAL-HERS [74], ISF [94], GMMSP [5], SCALP [34], SNIC [2], SH [103], LNSNet [121], SICLE [9], and LSC [21,55]. In addition, we include the 6 methods recommended as state-of-the-art in [90]: SLIC [1], SEEDS [14], ERS [59], ETPS [109], CRS [24], and ERGC [19]. Finally, a grid segmentation (GRID) was used as a baseline. Regarding implementation, we used the code of SEEDS, CRS, and ERGC available in the benchmark of Stutz et. al [90]. Also, we implemented grid segmentation. For the other methods, we use the original authors' code. Furthermore, we set the parameters according to the recommended ones in their respective articles and we did not perform fine-tuning. All evaluated methods allow some control over the number of superpixels generated. In our experiments, we assess segmentations with $K \approx \{25, 50, 75, 100, 200, 300, 400, 500, 600, 700, 800, 900, 1000\}$ desired superpixels, except for the robustness evaluation, with only $K \approx 400$ superpixels.

We selected four different datasets which impose different challenges for superpixel segmentation: *Birds* [66]; *Insects* [66]; *Sky* [3]; and ECSSD [88]. *Birds* [66] consists of 150 natural images of birds with thin and elongated objects' parts. Similarly, *Insects* [66] has 130 images of invertebrates with less texture on background regions. *Sky* [3] has 60 images for sky segmentation with large homogeneous regions with subtle luminosity variations. Finally, the *Extended Complex Scene Saliency Dataset* (ECSSD) [88] is composed of 1000 images with objects and backgrounds whose textures are complex.

5.2 Evaluation measures

In general, the measures for superpixel evaluation can be divided into measures that evaluate: (i) superpixel delineation; (ii) its shape; or (iii) its color homogeneity. The delineation measures evaluate the overlap of the superpixel boundaries with the image object. The delineation-based evaluation is widespread in superpixel segmentation since the oversegmentation of the object and background regions is not penalized. On the other hand, the quality of the superpixels inside these regions is also not evaluated [90]. In this work, we evaluated boundary delineation using *Boundary Recall* (BR) [67] and *Undersegmentation Error* (UE) [72]. For color homogeneity assessment, we used the *Similarity between Image and Reconstruction from Superpixels* (SIRS) [7] and *Explained Variation* (EV) [71]. Finally, we assess superpixels' compactness using the *Compactness index* (CO) [81].

Boundary Recall (BR) [67] is a widely used measure for superpixel evaluation. It measures the fraction of ground-truth boundary pixels correctly detected, as presented in Equation 1, where TP is the number of boundary pixels that match in a segmentation S and a ground-truth G, and FN is the number of boundary pixels in G that does not match with S. The boundary pixels are matched within a local neighborhood of size $(2r+1)^2$, in which r is 0.0025 times the image diagonal.

$$\mathrm{BR}(S,G) = \frac{\mathrm{TP}(G,S)}{\mathrm{TP}(G,S) + \mathrm{FN}(G,S)} \tag{1}$$

Another widely used measure to assess the quality of superpixel segmentation delineation is the *Undersegmentation Error* (UE). Introduced by [52], the UE measures the adherence of the boundary pixels in S to the G contours based on the area between S and G regions. UE has different versions [90]. The most recommended was proposed by [72] that evaluated the adherence to contours based on the minimum area of overlap between S and G, as presented in Equation 2, where N is the number of pixels G and k is the number of regions in G.

$$\mathrm{UE}(S,G) = \frac{1}{N}\sum_{i}^{k}\sum_{S_j \cap G_i \neq \varnothing} \min\{|S_j \cap G_i|, |S_j - G_i|\} \tag{2}$$

Shape-based evaluation metrics assess whether the superpixels have compact shapes with smooth contours and are arranged regularly — *i.e.*, in a grid. Although these properties have an inverse relationship to the delineation, an improved boundary recall does not necessarily imply better segmentation [81,82]. Due to this, the quality of the superpixel methods has been evaluated in previous benchmarks according to the trade-off between its shape quality and delineation [89,99].

The *Compactness index* (CO) [81] measure uses the isoperimetric quotient to measure the similarity between the shape of a superpixel and a circle, which constitutes the most compact geometric shape. The CO measure is presented in Equation 3, in which $A(S_j)$ and $P(S_j)$ are the superpixel area and perimeter, respectively.

$$\mathrm{CO}(S) = \frac{1}{N}\sum_{S_j}|S_j|\frac{4\pi A(S_j)}{P(S_j)} \tag{3}$$

Although the desired properties of superpixels are not a consensus in the literature, the inner color similarity usually underlies their methods. The *Explained Variation* [71] defines homogeneity by comparing the variance of the superpixels' mean color $\mu(S_i)$ and the variance of the pixels' color $I(p)$ towards the image's mean color $\mu(\mathbf{I})$, resulting in a normalized measure (Equation 4). This measure is maximum when $|S| = |\mathbf{I}|$ or when $I(p) = \mu(S_i)$ for all $p \in S_i$ and for every $S_i \in S$. However, EV considers the superpixels' mean color, which is insufficient for describing perceptually homogeneous textures [71].

$$EV(S) = \frac{\sum_{S_i \in S}|S_i|\,\|\mu(S_i) - \mu(\mathbf{I})\|_1^2}{\sum_{p\in\mathbf{I}}\|I(p) - \mu(\mathbf{I})\|_1^2} \tag{4}$$

To overcome the mean color drawback, the *Similarity between Image and Reconstruction from Superpixels* (SIRS) [7] models the color homogeneity problem as an image reconstruction problem. The color descriptor *RGB Bucket Descriptor* (RBD) represents each superpixel as a small set of its most relevant colors. Let $G^{S_i} \in \mathbf{S}(S_i, 7)$ represent the set of 7 disjoint groups related to each RGB cube vertices, whose colors are $c_l \in [0,1]^3$, in which $1 \leq l \leq 7$. Then, we populate each $G_l^{S_i} \in G^{S_i}$ by assigning every $p \in S_i$ to its most similar group using a mapping function $M(p)$ (Equation 5)

$$M(p) = \operatorname*{argmin}_{c_i \in V}\{\|x - c_i\|_1\} \tag{5}$$

The colors in RBD are used to reconstruct the original image. The reconstruction error is measured by the *Mean Exponential Error* (MEE) between the original and reconstructed image. The MEE increases the error weight of heterogeneous colors based on the maximum distance between the colors of the RBD. The MEE's exponent interval varies between one and two (the absolute or the mean error). Finally, SIRS defines segmentation quality as the Gaussian weighted error of reconstruction using MEE.

$$\mathrm{MEE}(S) = \frac{1}{|\mathbf{I}|}\sum_{S_i\in S}\sum_{p\in S_i}\|R(p) - I(p)\|_1^{2-\psi} \tag{6}$$

$$\mathrm{SIRS}(S) = \exp^{-\frac{\mathrm{MEE}(S)}{\sigma^2}} \tag{7}$$

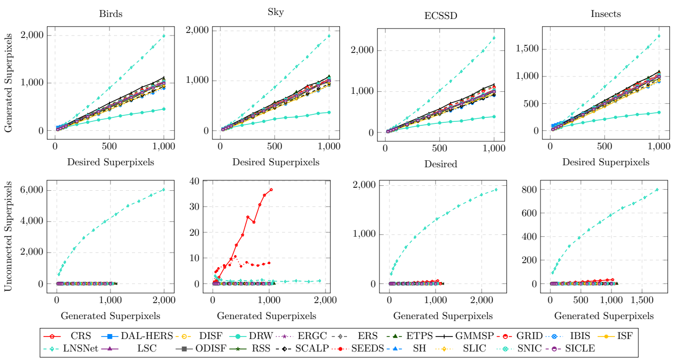

Fig. 1 Number of generated superpixels in relation to the desired number of superpixels (top row) and the number of not connected superpixels in relation to the number of generated superpixels (bottom row) on Birds, Sky, ECSSD, and Insects datasets.

6 Results

In this Section, we evaluate 21 superpixel methods according to different aspects. First, we evaluate connectivity, examining the number of superpixels with distinct labels regardless of their connectivity and the number of connected components (Section 6.1). Some superpixel methods include a merging step after the segmentation to ensure connectivity. In this work, we include a merging step as post-processing on methods that do not guarantee connectivity. In Section 6.2, we quantitatively evaluate object delineation, color homogeneity, and compactness. Also, we summarize these results with a boxplot analysis. As [73,90], we evaluated the methods' stability using the evaluation measures' minimum (min), maximum (max), and standard deviation (std) in Section 6.3. Afterward, we assessed robustness and runtime in Sections 6.4 and 6.5, respectively. Finally, in Section 6.6, we perform a qualitative evaluation concerning the smoothness of the contours of superpixels, their compactness, and their adherence to the images' boundaries. The superpixel methods and codes used in this work are available in our superpixel evaluation benchmark at https://github.com/IMScience-PPGINF-PucMinas/superpixel-benchmark.

6.1 Number of superpixels and connectivity

All superpixel methods evaluated in this work have a parameter for the desired number of superpixels. However, most of these methods generate a different number of superpixels than the desired one. Although control over the number of superpixels is a desirable property, some works reduce this control to produce a segmentation that better suits the image content. As one may note in the first row of Figure 1, only **DISF**, **ODISF**, **SICLE**, **SH**, and **ERS** generate exactly the desired number of superpixels. Despite this, most methods generate a number of superpixels close

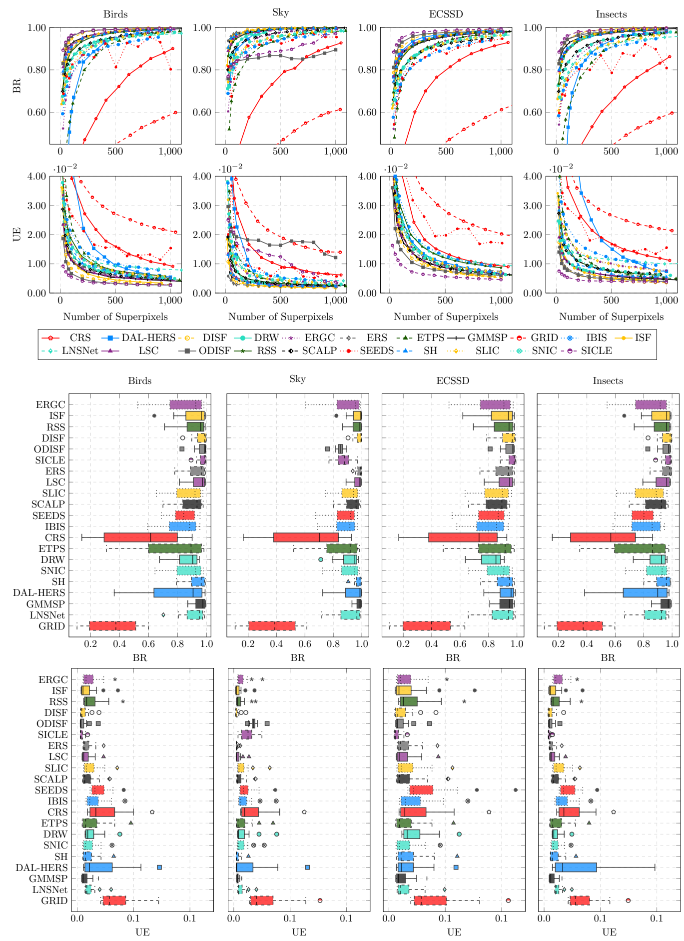

Fig. 2 Results for BR and UE on Birds, Sky, ECSSD, and Insects datasets.

to the desired one. In contrast, **LNSNet** and **DRW** generated quantities farther from the desired ones. While **DRW** usually produces fewer superpixels, **LNSNet** creates almost twice the desired superpixels.

Superpixel connectivity is also an important property to consider. However, many methods in the literature do not guarantee it. As one may see in the second row of Figure 1, **LNSNet**, **CRS**, and **SEEDS** do not guarantee the connectivity of their superpixels. In particular, **LNSNet** generated a high number of unconnected superpixels in most datasets. In contrast, **CRS** and **SEEDS** produces fewer unconnected superpixels. For the quantitative and stability experiments (Sections 6.2 and 6.3, respectively), we perform post-processing to enforce connectivity in **LNSNet**, **SEEDS**, and **CRS**. Let the similarity between two superpixels as the euclidean distance between their average colors. The merging step merges the smaller-area superpixels with their most similar neighbor (considering 8-neighborhood) until the number of superpixels reaches the number of segmentation labels.

6.2 Quantitative evaluation

6.2.1 Object delineation

As shown in Figure 2, the methods **GRID**, **CRS**, and **SEEDS** reach the worst results in all datasets. According to the evaluation with UE, most methods have low leakage. Similarly, the delineation measured by BR is generally high. The best scores in both UE and BR were achieved by **SICLE**, **ODISF**, **DISF**, **LSC**, **ISF**, **GMMSP**, **SH**, and **ERS**. However, followed by **GRID**, **ODISF** and **SICLE** obtained the lowest delineations on the Sky dataset. These results can also be observed in the boxplot visualization (bottom row in Figure 2). These results contrast with the other datasets, in which **SICLE** and **ODISF** had the best results. Furthermore, **RSS** has a competitive BR, but with worse UE. This observation is more evident in the boxplot results. In contrast, **DAL-HERS**, **ETPS**, **IBIS**, and **SLIC**, obtained a low delineation, only superior to **GRID**, **SEEDS**, and **CRS**. Their results are followed by **SNIC**, **SCALP**, **DRW**, and **LNSNet**. One may see in the first row of Figure 2 that **DAL-HERS** obtains low delineation for a number of superpixels smaller than 400, approximately, on Birds, ECSSD, and Insects datasets. However, **DAL-HERS** presents a competitive delineation after 400 superpixels. These low results occur because this method may generate tiny regions, resulting in segmentations with low delineation and low color homogeneity.

6.2.2 Compactness

Figure 3 show the compactness evaluation. As expected, **GRID** obtains the most compact segmentations. Aside from **GRID**, the methods **CRS** and **ETPS** had the highest compactness, followed by **SCALP** and **SNIC**. **SLIC** and **IBIS** achieve similar compactness, usually lower than **SCALP** and **SNIC**. All these methods have a parameter to determine the compactness. While **CRS** and **ETPS** produce superpixels by optimizing the contours of a grid segmentation, the others use different approaches based on **SLIC**. In contrast, **LSC** and **GMMSP** present similar and moderate compactness. Among the evaluated methods, only **SEEDS** had high variability in its compactness. More delineation-focused methods, such as **SICLE**, **ODISF**, **DISF**, **SH**, and **DAL-HERS** produced less compact segmentations.

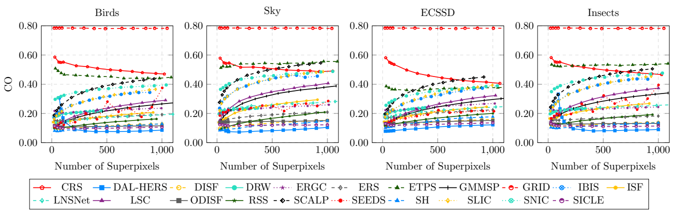

Fig. 3 Results for CO on Birds, Sky, ECSSD, and Insects datasets.

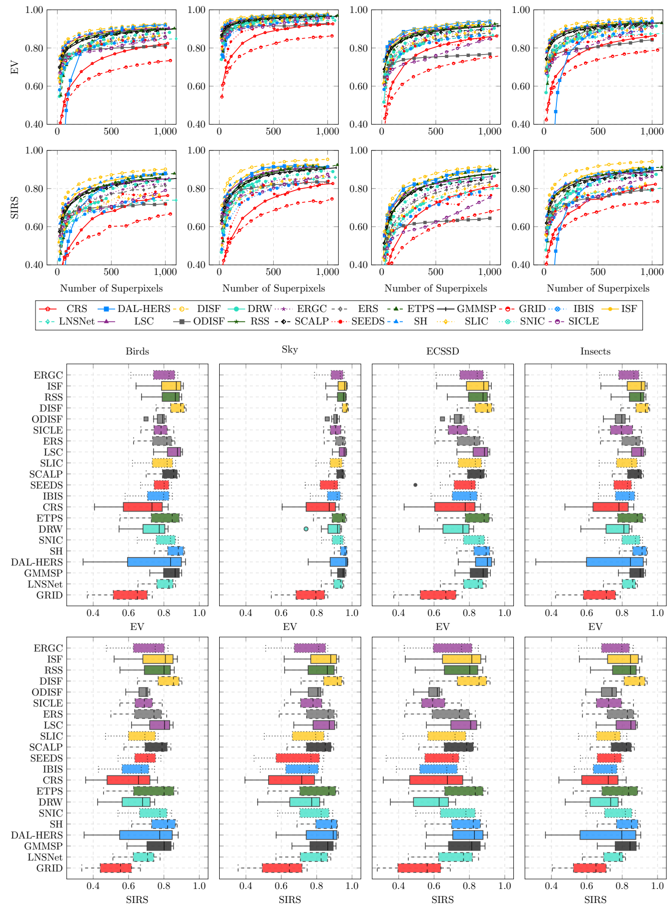

Fig. 4 Results for EV and SIRS on Birds, Sky, ECSSD, and Insects datasets.

6.2.3 Color homogeneity

When evaluating the color homogeneity (Figure 4) with EV and SIRS, the results of the first measure were generally higher and closer to each other compared to the second one. However, their results show some similarities. **GRID** and **CRS** had the worst results in all datasets in both measures, followed by **ODISF** and **SICLE**. From these methods, only **ODISF** and **SICLE** have an accurate delineation, and their low color homogeneity is a result of fewer superpixels in the non-salient image region. It is not easy to define the best method on all datasets according to EV scores, but **DISF** obtains the best results according to SIRS scores. Finally, one may see that **DISF**, **SH**, **ISF**, **LSC**, **RSS**, **GMMSP**, and **SCALP** achieve competitive results in both color homogeneity measures.

6.2.4 Overall

As one may see in Figures 2, 4, and 3, most path-based clustering methods (**ERGC**, **ISF**, **DISF**, **RSS**, **ODISF**), and **SICLE** had similar performance in object delineation, compactness, and homogeneity. Among these methods, **DISF** had better delineation and color homogeneity according to all measures used. On the other hand, **SICLE** and **ODISF** obtained a similar delineation on most datasets but with low color homogeneity. The significant performance reduction of **SICLE** and **ODISF** on Sky is due to the saliency maps identifying wrong objects. Since the sky is not a salient region in this dataset, the saliency map is less reliable, being necessary to fine-tune or switch the saliency map trustiness in **SICLE**. Although path-based methods had optimal delineation, their superpixels have low compactness. With a similar clustering approach, **ERS** performs clustering based on graphs and obtains excellent delineation on Sky and Insects datasets.

Region-based clustering approaches (**SLIC**, **LSC**, and **SCALP**) had more variate results. While **LSC** achieved better delineation and more homogeneous superpixels, **SLIC** had superpixels with moderate compactness and worse delineation. On the other hand, **SCALP** obtained a competitive delineation with homogeneous and more compact superpixels than **SLIC**. Methods that perform clustering based on contour optimization (**SEEDS**, **IBIS**, **CRS**, and **ETPS**) also reached different results due to the distinction between their optimization functions. Among these, **IBIS** achieved better object delineation and color homogeneity, with results similar to **SLIC** in all evaluation measures. On the other hand, **CRS** and **SEEDS** had the worst delineation and homogeneity but greater compactness among all methods. Therefore, concerning the main processing approaches, clustering based on contour evolution produced the worst results in object delineation and color homogeneity but with higher compactness.

Regarding clustering with a dynamic center update (**D-RW** and **SNIC**), they use strategies to adapt the number of generated superpixels to the image content. Despite their similarities, **DRW** and **SNIC** use different features and optimization functions, which explains the contrast in their results. While **DRW** has better delineation and fewer superpixels, **SNIC** generates more compact and homogeneous superpixels. The lower color homogeneity of **DRW** compared to **SNIC** is due to the smaller number of superpixels produced by **DRW** than the other methods. Concerning hierarchical approaches (**SH** and **DAL-HERS**), they have low compacity and high color homogeneity. However, **SH** had competitive delineation in contrast with worse results with **DAL-HERS**. Finally, **GMMSP** and **LNSNet**, unique in their clustering category, presented excellent delineation with BR. Concerning UE and color homogeneity, **LNSNet** had heterogeneous superpixels with moderate compactness and more leakage. On the other hand, **GMMSP** achieved competitive results in all evaluated measures.

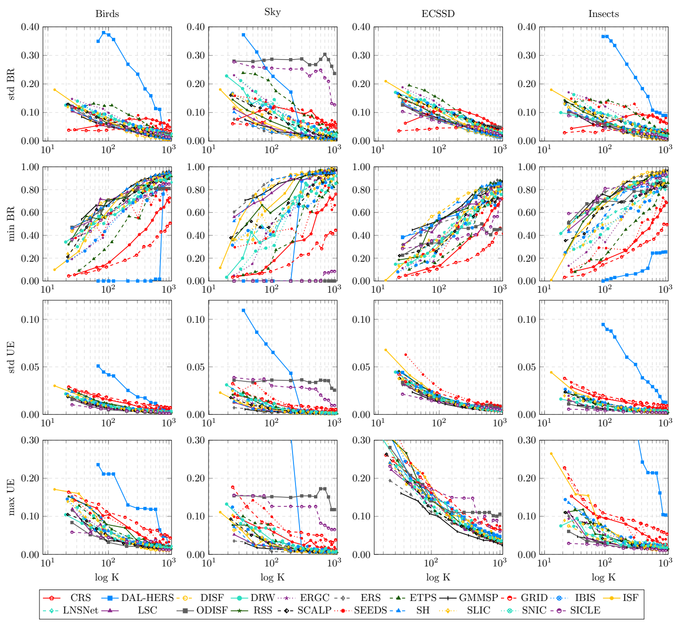

Fig. 5 Results for minimum BR, maximum UE, and standard deviation of BR and UE on Birds, Sky, ECSSD, and Insects datasets.

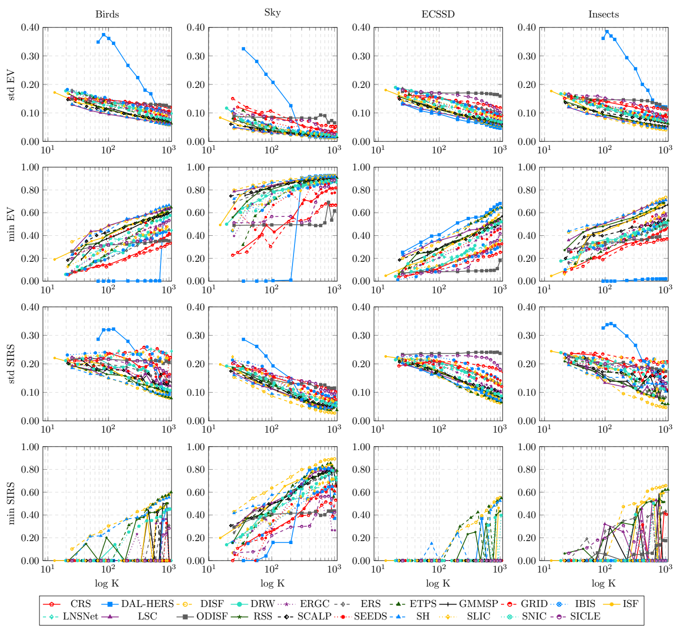

Fig. 6 Results for minimum and standard deviation of EV and SIRS on Birds, Sky, ECSSD, and Insects datasets.

6.3 Superpixels stability

6.3.1 Object delineation

As one may see in Figure 5, most methods present high stability regarding object delineation, since most of them present a performance that monotonically increases in BR and decreases in UE. Most of the methods also present low std BR and std UE. In contrast, **DAL-HERS**, **SEEDS**, **ETPS**, and **CRS** show lower BR stability on all datasets. Also, **ODISF** and **SICLE** only presents instability on the Sky dataset due to its high and almost constant standard deviations. Despite being a delineation-focused method, the **ODISF**'s and **SICLE**'s performance of UE standard deviation on the Sky dataset achieves worse results than **GRID** for more than 200 superpixels, contrasting with its high stability on the other datasets. The **ODISF**'s and **ODISF**'s instability explains their inferior mean BR and UE (Section 6.2) performance on Sky. On the other hand, **DAL-HERS** presented greater instability due to its creation of tiny regions, as mentioned in Section 6.2. As one may note in Figure 5, the low min BR of **DAL-HERS** indicates that the tiny regions are created independent of the number of superpixels. Based on the **DAL-HERS** results, we consider that its low performance in this work results from some bug. As shown in Figure 5, **DISF**, **GMMSP**, **LSC**, **SH**, and **ERS** showed high stability. On the other hand, **ISF** and **RSS** present stable and low std BR and std UE, but with some instability in max UE and min BR. Concerning min BR, **GRID**, **CRS**, and **SEEDS** had the worst results while **SH**, **ISF**, **RSS**, **GMMSP**, **DISF**, **LSC**, and **ERS** had the highest ones.

6.3.2 Color homogeneity

The color homogeneity stability evaluation with EV and SIRS is presented in Figure 6. Concerning min EV and min SIRS, most of the methods with the former had increasing values while the second showed more rigorous minimum scores with increasing values only in the Sky dataset. In both minimum measures, the methods with the highest minimum differ, except for **DISF**, which presents higher results in all datasets, followed by **SH**. Among the evaluations with min EV, **ODISF** and **SICLE** had almost constant values and worse results than GRID in the Sky dataset. These results are due to the saliency map and the concentration of superpixels in the salient region, as aforementioned. Furthermore, std SIRS and std EV also showed distinct variations. While the std EV results presented less stable results, the std SIRS evaluation presented more increasing results, indicating greater instability in some methods. For the std EV assessment, the methods **DISF**, **SH**, and **LSC** showed high stability on all datasets. In addition, the methods **ISF**, **RSS**, and **SCALP** also showed high stability on at least one dataset. Unlike std EV, in std SIRS the methods **LNSNet**, **GRID**, **IBIS**, **ODISF**, and **SLIC** showed less stability on Birds and Insects datasets. On the other hand, **DISF** showed high stability in SIRS, followed by **SH** and **ETPS**.

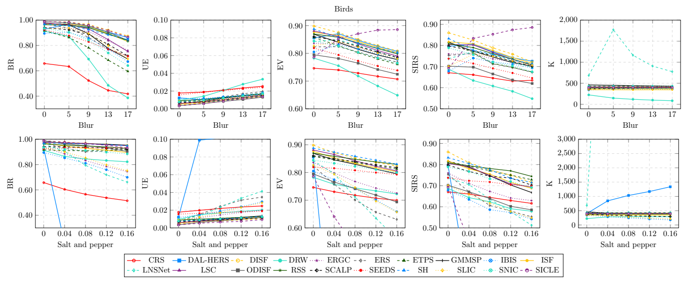

Fig. 7 Influence of average blur (top row) and salt and pepper noise (bottom row) for $K \approx 400$ on Birds dataset.

6.4 Robustness

Noise and blur robustness evaluate, respectively, the susceptibility of the algorithm to strong and irrelevant edges and potentially relevant but soft edges. Similar to [90], we evaluated robustness against salt and pepper noise and average blur. In this experiment, we varied the average blur filter size by $\{0, 5, 9, 13, 17\}$ and the noise probability by $\{0, 0.4, 0.08, 0.12, 0.16\}$ in the Birds dataset images with approximately 400 superpixels. The evaluation measures used were BR, UE, EV, SIRS, and the number of superpixels produced (K) in the segmentations.

As one may see in Figure 7, blur and noise generally tend to have a similar impact. **DISF**, **ERGC**, **RSS**, **ISF**, **ODISF**, **SEEDS**, and **SH** were robust in blur and noise. On the other hand, **DAL-HERS** showed the lowest noise robustness, followed by **LNSNet** and **ERS**. Despite being the least robust to noise, **DAL-HERS** achieved considerable robustness to blur. A similar sensitivity to noise can be observed in **SICLE** regarding homogeneity. However, **SICLES**'s homogeneity highly increased with blur. Also, it presents high robustness to noise and blur concerning delineation. On the other hand, **DRW** was the most influenced by blur. One can also see that some methods presented a slightly better evaluation when adding blur or noise. That is the case for **LNSNet**, **IBIS**, and **SLIC** with blur. The same occurred less perceptibly in **SEEDS**, **DAL-HERS**, **ERGC**, and **ERS**.

As shown in Figure 7, some methods try to compensate for noise and blur by producing more or fewer superpixels. Among the evaluated methods, **LNSNet** was the most impacted in the number of superpixels generated, especially when adding noise. As seen in Section 6.1, **LNSNet** produced superpixels that were more discrepant in quantity, many of those disconnected. The second with the most

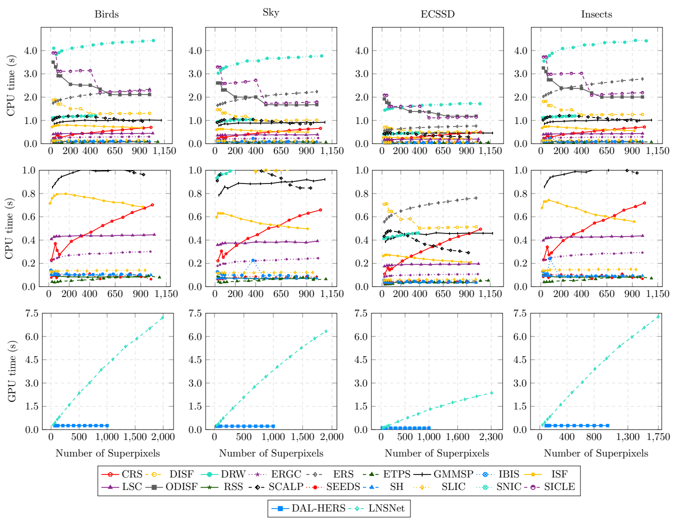

Fig. 8 Runtime in seconds on Birds, Sky, ECSSD, and Insects datasets.

influenced number of superpixels was **DAL-HERS** when adding noise. In addition to these, **IBIS**, **DRW**, **SLIC**, **GMMSP**, and **SCALP** showed a moderate susceptibility to the number of superpixels. Finally, the addition of noise or blur did not modify the number of superpixels generated in the **CRS**, **DISF**, **ERGC**, **ERS**, **ETPS**, **SICLE**, **ODISF**, **RSS**, **SH**, and **SNIC** methods.

6.5 Runtime

Execution time may be a critical aspect in superpixel methods, especially for real-time applications. Figure 8 shows the CPU[1] and GPU[2] time in seconds without the post-processing of Section 6.1. For **SCALP** and **ODISF**, we do not include the edge maps and saliency maps computation. As one may see in Figure 8, due to the images of the ECSSD and Sky datasets being generally smaller than the others, the runtime in these datasets was usually shorter.

According to the CPU runtimes on the first and second rows in Figure 8, methods whose main processing is boundary evolution clustering (except **CRS**) achieved the lowest execution times. **SLIC** performed similarly, while other region-based clustering methods (**LSC** and **SCALP**) varied in efficiency. Among these, **LSC** has a runtime of up to 0.4 seconds, while **SCALP** needs approximately twice as long. Path-based clustering methods (**ERGC**, **ISF**, **RSS**, **DISF**, **SICLE** and **ODISF**) also showed varied efficiency. In this category, the highest runtimes were achieved by **SICLE**, **ODISF**, and **DISF**, while **ISF** and **ERGC** required less than 1 second per image. In contrast, **RSS** had a competitive execution time (less than 0.1 seconds). Similarly, **SH**, the only method with hierarchical clustering in CPU execution, also achieved competitive runtime.

Methods with a dynamic center update clustering category (**DRW** and **SNIC**) had distinct runtimes. **SNIC** was the most time-consuming among all methods running on the CPU, while **DRW** was more efficient, with execution times of up to 1 second per image. Considering graph-based clustering, **ERS** required a high execution time, similar to **ODISF**. And the **GMMSP**, the only one with clustering based on data distribution, achieved similar efficiency to **SCALP**. As one may see on the bottom row of Figure 8, only **LNSNet** and **DAL-HERS** were executed on a GPU. The former had the worst execution time of all evaluated methods, while the second had an excellent execution time (less than 0.3 seconds per image). Finally, **SH** and **DAL-HERS** were the only ones whose execution time was constant since they produced a hierarchy of superpixels in a single execution. From cuts on the hierarchy, they produce different numbers of superpixels.

6.6 Qualitative evaluation

In this section, we evaluated the segmentations' visual quality regardless of their ground-truth since the image object may vary according to the application. We assess visual quality based on the superpixels' adherence to the image boundaries, smoothness, compactness, and regularity. The smoothness is inversely related to the superpixel's boundary length. On the other hand, the superpixels' compactness relates to their area. Moreover, regularity refers to their shape, size, and arrangement. Figures 9 and 10 present segmentations with approximately 100 and 700 superpixels on Birds, Sky, ECSSD, and Insects datasets.

6.6.1 Path-based clustering

Relative to path-based clustering methods, **RSS** (Figure 9) does not produce compact superpixels. Instead, their superpixels may have elongated and thin shapes at strong image boundaries but with an optimal delineation. However, by reducing the number of superpixels, the delineation quality dramatically decreases at smooth image boundaries. In contrast, **ISF** produces regular superpixels in homogeneous regions. However, it has a high sensitivity to color variations, leading to non-smooth superpixels, highly variable in size, on less homogeneous. For a higher number of superpixels, **ISF** has excellent delineation. However, reducing the number of superpixels implies a worse delineation.

As we may observe in Figure 9, **DISF** achieved an improved segmentation, in which its superpixels are neither compact nor smooth, but they present a high adherence to the image boundaries. **DISF** also maintains good adherence

[1] CPU Intel® Core™ i5-7200U @ 2.5GHz x 4, 64bit with 24GB RAM.

[2] CPU Intel® Core™ i7-8700 @ 3.20GHz x 12, 64bit with 31GB RAM and a GPU Nvidia GeForce GTX 1080 with 8GB RAM.

to the image boundaries and generates larger superpixels in more homogeneous regions, even with a smaller number of them. Based on **DISF**, both **ODISF** and **SICLE** present different results from the other methods. Both produce more superpixels on the salient area identified by the saliency map, which can improve the delineation of this region. However, their superpixels are neither compact nor smooth. Due to this, there is a low number of superpixels in regions not identified by the saliency map, leading to a worse delineation in these regions but a superior delineation in the salient ones. Also, by fine-tuning the saliency maps, their results can improve. Between **SICLE** and **ODISF**, we can observe more accurate delineation in **SICLE** segmentations. Similar to the previous ones, the segmentation with **ERGC** has good adherence to the image boundaries. In addition, its superpixels have some regularity, without significant variations in size, and their contours are smooth. However, for a smaller number of superpixels, the boundary adherence of **ERGC** segmentation reduces significantly.

6.6.2 Region-based clustering

Regarding the region-based methods, **SLIC** (Figure 9) produces very compact superpixels with good adherence to the image boundaries. Its superpixels are also regular in more homogeneous regions. However, **SLIC** generates superpixels with slightly non-smooth contours in less homogeneous regions. By reducing the number of superpixels, the compactness is slightly reduced, even in complex areas of the image. On the other hand, the delineation is more affected. In contrast, **SCALP** produces superpixels with excellent delineation that are more compact, smooth, and regular than **SLIC**. The compactness of **SCALP** segmentation reduces for a reduced number of superpixels, and its delineation also reduces slightly. However, the superpixels' contours remain smooth. Unlike **SLIC** and **SCALP**, **LSC** only produces smooth superpixels in more homogeneous regions. However, its high sensitivity to minor color variations results in superpixels with less smooth contours and compactness in regions with simpler textures. Furthermore, **LSC** may generate more elongated and thin superpixels in the strong image boundaries, obtaining a great delineation but without compactness. For a reduced number of superpixels, the visual quality of **LSC** delineation suffers a slight reduction, and its superpixels have significantly fewer smooth contours in regions with textures.

6.6.3 Dynamic center update clustering

With a segmentation visually very similar to **SLIC**, **SNIC** (Figure 9) also produces superpixels with high compactness and better delineation. In contrast, as one may see in Figure 10, **DRW** does not generate compact superpixels. Also, the number of superpixels produced is noticeably smaller than expected. Despite this, **DRW** generates superpixels with good adherence and fewer superpixels in homogeneous regions.

6.6.4 Boundary evolution clustering

Similarly to DRW, the superpixels created by **SEEDS** (Figure 10) are not compact and have non-smooth boundaries. The segmentation with a higher number of superpixels in **SEEDS** has moderate delineation with small leakage regions. For a reduced number of superpixels, the compactness and smoothness do not increase in **SEEDS**, and there is a noticeable reduction in delineation. In contrast to **SEEDS**, **CRS** generates superpixels very compact, regular, and with smooth contours independent of the number of superpixels, but with low adherence to the image boundaries. In a segmentation with 100 superpixels, the image boundaries seem to be almost completely ignored. Similarly, **ETPS** produces very regular, smooth, and compact superpixels. For a higher number of superpixels, the segmentation generated with **ETPS** has high adherence to the boundaries. The compactness, smoothness, and regularity reduce slightly by reducing the number of superpixels, but the delineation suffers drastically.

IBIS also generates significantly compact pixels, whose compactness and smoothness vary depending on the region's homogeneity. Also, it produces regular superpixels at the homogeneous image regions. For a higher number of superpixels, their compactness in homogeneous regions is very high, and **IBIS** has good adherence to the image contours, even in more complex regions. However, its sensitivity to color variations reduces compactness and smoothness in less homogeneous areas. Also, by reducing the number of superpixels, its adherence to contours is significantly reduced.

6.6.5 Hierarcical clustering

Regarding the hierarchical methods, **SH** has an excellent delineation, but its superpixels are not regular nor compact and have non-smooth contours in more textured regions. In addition, it generates elongated and thin superpixels at some of the strong image boundaries. **DAL-HERS** also has superb delineation but generates rough superpixels and some tiny ones, resulting in visibly poor segmentation.

6.6.6 Others

LNSNet produces a significantly higher number of superpixels than desired. Similarly to **ISF**, **LNSNet** produces compact superpixels in homogeneous regions, but its sensitivity to color variations implies very rough superpixels. It has good delineation when the number of superpixels is

Fig. 9 Segmentation comparison with images from Birds, Sky, ECSSD, and Insects with 100 and 700 superpixels.

Fig. 10 Segmentation comparison with images from Birds, Sky, ECSSD, and Insects with 100 and 700 superpixels.

higher. However, their non-smooth contours do not have a high delineation even in regions with a more prominent boundary, causing small leaks. In **ERS**, for a higher number of superpixels, they do not vary much in size, but their shape varies considerably, and they have low smoothness but good boundary adherence. By reducing the number of superpixels, its boundary adherence reduces, but not drastically. In comparison, **GMMSP** produces significantly more compact superpixels in homogeneous regions. In less homogeneous ones, **GMMSP** produces fewer compact superpixels but usually with smoother contours. By reducing the number of superpixels, the compactness of the less homogeneous image region barely changes. However, the compactness and smoothness drastically reduce in less homogeneous regions.

6.6.7 Overall

As one may see, **CRS**, **ERGC**, **ETPS**, **SCALP**, **SLIC**, **SNIC**, **IBIS**, and **GMMSP** produced visibly smooth, compact, and regular superpixels. These properties are noticeable on **CRS**, **SCALP**, and **ETPS**. Nevertheless, high compactness may lead to a worse delineation, as in **CRS** and **ETPS**. Conversely, **ERGC**, **IBIS**, **SNIC**, and **SLIC** had medium boundary adherence, with worse results on smooth image boundaries. Among these methods, only **SCALP** and **GMMSP** achieved excellent boundary delineation.

Concerning methods with less (or no) compactness and smoothness, **LNSNet** and **SEEDS** had the worse delineations. In contrast, **DRW**, **ERS**, **RSS**, **LSC**, and **ISF** had some compactness and smoothness along with good boundary adherence, especially **ISF** and **LSC**. One may observe the best delineation in **DISF**, **LSC**, **ODISF**, **SICLE**, **DAL-HERS**, **ISF**, and **SH**, although most do not present compactness or regularity. In particular, **DISF**, **ODISF**, **SICLE**, and **GMMSP** had visually better adherence to the image boundaries. Among these methods, **ODISF** and **SICLE** had exceptional adherence to image boundaries belonging to a specific image region, indicated as an object in the saliency map, due to the high number of superpixels in this region. Conversely, they generate fewer superpixels in non-salient image regions, reducing their color homogeneity. As observed in the quantitative evaluation, when the saliency map corresponds to the desired object in the image, the **ODISF**'s and **SICLE**'s delineations outperform the other methods.

In relation to the main processing categories, methods with contour evolution-based clustering usually produce the most compact and regular superpixels, although they have low boundary adherence. Conversely, those with neighborhood-based clustering usually had good delineation with high compactness and regularity. Similarly, dynamic-center-update clustering methods also achieved good boundary adherence. However, only **SNIC** showed compactness and regularity, whereas **DRW** only had smooth superpixels contours. Finally, **GMMSP** had great compactness and competitive delineation. Hierarchical methods, path-based methods, **LNSNet**, and **ERS**, produced superpixels with low compactness. Among these, only **LNSNet** had poor delineation. Also, the superpixels in both **LNSNet** and hierarchical methods are neither compact nor smooth. Conversely, **ERS** and most path-based methods generated superpixels with low compactness and smoothness but with excellent boundary adherence.

7 Conclusions

In this work, we present a taxonomy for superpixel methods, which categorizes them according to their processing steps and the level of abstraction of the features used. Our taxonomy separates each superpixel approach into up to three processing steps and categorizes the task performed at each one. We demonstrate our taxonomy and inform other significant properties of 52 of the most recently and commonly used superpixel methods. We also provide a comprehensive literature review encompassing these methods. We present an extensive comparison among 20 superpixel methods considering: superpixels' connectivity, control of the number of superpixels, compactness, adherence to object contours, color homogeneity, stability, robustness to noise and blur, execution time, and visual quality.

According to our experiments, methods with clustering based on contour evolution generally present greater efficiency, compactness, and regularity. Nevertheless, they have worse boundary adherence and color homogeneity. In addition, methods with dynamic-update-clustering are less efficient and generate slightly less compact and regular superpixels. In addition, they have better delineation and homogeneity than those based on contour evolution. Conversely, methods with region-based clustering present more varied performances. For instance, **LSC** achieved good boundary adherence, compactness, and smoothness. On the other hand, **SLIC** and **SCALP** had higher compactness but worse delineation than **LSC**.

Regarding methods with clustering based on data distribution, we evaluated only **GMMSP**, which obtained a competitive delineation, good compactness, and smooth superpixels' contours, although no regularity. In addition, **ERS**, the only evaluated method that performs graph-based clustering, had a similar delineation to **GMMSP** but with worse efficiency, compactness, and color homogeneity. Hierarchical methods also produced superpixels with excellent boundary adherence. They have low execution time, but their superpixels were neither visually compact nor smooth.

LNSNet, the only evaluated method which performs clustering with a CNN, presents a visually poor delineation and the worst efficiency. In addition, it has color homogeneity than methods with hierarchical clustering. In our evaluation, the path-based clustering methods generally had the best delineation and the most homogeneous superpixels. However, they had varied efficiency, low compactness, and low smoothness.

Most evaluated methods produce connected superpixels. Also, they usually generate superpixels in a similar number to the desired one. In particular, **DISF**, **ODISF**, **SICLE**, **SH**, and **ERS** generated the exact number of desired superpixels. In contrast, among the evaluated methods, only **LNSNet**, **SEEDS**, and **CRS** produced non-connected superpixels. **LNSNet** creates almost twice the superpixels, many of these disconnected. On the other hand, the number of superpixels produced by **DRW** is usually lower than desired.

When evaluating robustness, most methods achieved good robustness to noise and blur. The worst results were observed in **DAL-HERS**, followed by **SICLE**, **LNSNet**, and **ERS**. In contrast, the most robust methods were **DISF**, **ERGC**, **RSS**, **ISF**, **ODISF**, **SEEDS**, and **SH**. We could also see that some evaluated methods produce a different number of superpixels according to the addition of noise or blur. For $K \approx 400$, **LNSNet** had more sensitivity in this criterion, creating more than 30000 superpixels. **DAL-HERS** also produced significantly more superpixels, reaching almost 1500 when increasing noise. In addition, the number of superpixels produced by **IBIS**, **DRW**, **SLIC**, **GMMSP**, and **SCALP** is slightly different from the desired one when increasing noise or blur.

Due to the trade-off between delineation and compactness, it is hard to establish which method had the best performance. Considering object delineation and color homogeneity, **DISF**, **ISF**, **LSC**, **GMMSP**, and **SH** showed the best average performance and stability. **SH** has greater efficiency, followed by the **LSC** and **ISF**. On the other hand, **GMMSP** has more compact superpixels, followed by **ISF**. When delineation and homogeneity are more critical than compactness, **DISF** is the most recommended. We also recommend **SICLE** and **ODISF** when only object delineation is crucial. Despite not having good results when the saliency map does not find the desired object, their superior performance in other datasets may indicate that fine-tuning the saliency detector can improve the results. However, for greater compactness at the expense of delineation, both **SCALP** and **SLIC** are recommended. Between these, **SLIC** has more compactness and low execution time but worse delineation and less color homogeneity.

Acknowledgments

The authors thank the Conselho Nacional de Desenvolvimento Científico e Tecnológico – CNPq – (Universal 407242/2021-0, PQ 303808/2018-7, 310075/2019-0), the Fundação de Amparo a Pesquisa do Estado de Minas Gerais – FAPEMIG – (PPM-00006-18), the Fundação de Amparo a Pesquisa do Estado de São Paulo – FAPESP – (2014/12236-1) and the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (COFECUB 88887.191730/2018-00 and Finance code 001) for the financial support.

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**(11), 2274–2282 (2012). DOI 10.1109/TPAMI.2012.120
2. 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) (2017)
3. Alexandre, E.B., Chowdhury, A.S., Falcao, A.X., Miranda, P.A.V.: Ift-slic: A general framework for superpixel generation based on simple linear iterative clustering and image foresting transform. In: 2015 28th SIBGRAPI Conference on Graphics, Patterns and Images, pp. 337–344 (2015). DOI 10.1109/SIBGRAPI.2015.20
4. An, J., Shi, Y., Han, Y., Sun, M., Tian, Q.: Extract and merge: Superpixel segmentation with regional attributes. In: European Conference on Computer Vision, pp. 155–170. Springer (2020)
5. Ban, Z., Liu, J., Cao, L.: Superpixel segmentation using gaussian mixture model. IEEE Transactions on Image Processing **27**(8), 4105–4117 (2018). DOI 10.1109/TIP.2018.2836306
6. Ban, Z., Liu, J., Fouriaux, J.: Gmmsp on gpu. Journal of Real-Time Image Processing **17**(2), 245–257 (2020)
7. Barcelos, I.B., Belém, F.D.C., João, L.D.M., Falcão, A.X., JF, G.S.: Improving color homogeneity measure in superpixel segmentation assessment. In: 2022 35th SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI), vol. 1, pp. 79–84 (2022). DOI 10.1109/SIBGRAPI55357.2022.9991772
8. Bejar, H.H., Guimarães, S.J.F., Miranda, P.A.: Efficient hierarchical graph partitioning for image segmentation by optimum oriented cuts. Pattern Recognition Letters **131**, 185–192 (2020). DOI https://doi.org/10.1016/j.patrec.2020.01.008. URL `https://www.sciencedirect.com/science/article/pii/S0167865520300088`
9. Belém, F., Borlido, I., João, L., Perret, B., Cousty, J., Guimarães, S.J., Falcão, A.: Fast and effective superpixel segmentation using accurate saliency estimation. In: International Conference on Discrete Geometry and Mathematical Morphology, pp. 261–273. Springer (2022)
10. Belém, F., Guimarães, S.J.F., Falcão, A.X.: Superpixel segmentation by object-based iterative spanning forest. In: Iberoamerican Congress on Pattern Recognition, pp. 334–341. Springer (2018)
11. Belém, F., Perret, B., Cousty, J., Guimarães, S.J., Falcão, A.: Efficient multiscale object-based superpixel framework. arXiv preprint arXiv:2204.03533 (2022)

12. Belém, F.C., Guimarães, S.J.F., Falcão, A.X.: Superpixel segmentation using dynamic and iterative spanning forest. IEEE Signal Processing Letters **27**, 1440–1444 (2020). DOI 10.1109/LSP.2020.3015433
13. Belém, F.C., Perret, B., Cousty, J., Guimarães, S.J., Falcão, A.X.: Towards a simple and efficient object-based superpixel delineation framework. In: 2021 34th SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI), pp. 346–353. IEEE (2021)
14. Van den Bergh, M., Boix, X., Roig, G., de Capitani, B., Van Gool, L.: Seeds: Superpixels extracted via energy-driven sampling. In: A. Fitzgibbon, S. Lazebnik, P. Perona, Y. Sato, C. Schmid (eds.) Computer Vision – ECCV 2012, pp. 13–26. Springer Berlin Heidelberg, Berlin, Heidelberg (2012)
15. Van den Bergh, M., Boix, X., Roig, G., Van Gool, L.: Seeds: Superpixels extracted via energy-driven sampling. International Journal of Computer Vision **111**(3), 298–314 (2015). DOI 10.1007/s11263-014-0744-2
16. Bhugra, S., Kaushik, V., Mateos, I.C., Chaudhury, S., Lall, B.: Unsupervised learning of affinity for image segmentation: An inpainting based approach. In: 2021 36th International Conference on Image and Vision Computing New Zealand (IVCNZ), pp. 1–6. IEEE (2021)
17. 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 **9**, 77,250–77,263 (2021)
18. Borlido Barcelos, I., Belém, F., Miranda, P., Falcão, A.X., do Patrocínio, Z.K., Guimarães, S.J.F.: Towards interactive image segmentation by dynamic and iterative spanning forest. In: International Conference on Discrete Geometry and Mathematical Morphology, pp. 351–364. Springer (2021)
19. Buyssens, P., Gardin, I., Ruan, S.: Eikonal based region growing for superpixels generation: Application to semi-supervised real time organ segmentation in ct images. IRBM **35**(1), 20–26 (2014). DOI https://doi.org/10.1016/j.irbm.2013.12.007. URL `https://www.sciencedirect.com/science/article/pii/S1959031813001437`. Biomedical image segmentation using variational and statistical approaches
20. Chai, D.: Rooted spanning superpixels. International Journal of Computer Vision **128**(12), 2962–2978 (2020)
21. Chen, J., Li, Z., Huang, B.: Linear spectral clustering superpixel. IEEE Transactions on Image Processing **26**(7), 3317–3330 (2017). DOI 10.1109/TIP.2017.2651389
22. Condori, M.A., Cappabianco, F.A., Falcão, A.X., Miranda, P.A.: An extension of the differential image foresting transform and its application to superpixel generation. Journal of Visual Communication and Image Representation **71**, 102,748 (2020). DOI https://doi.org/10.1016/j.jvcir.2019.102748. URL `https://www.sciencedirect.com/science/article/pii/S1047320319303694`
23. Condori, M.A.T., Cappabianco, F.A.M., Falcão, A.X., De Miranda, P.A.V.: Extending the differential image foresting transform to root-based path-cost functions with application to superpixel segmentation. In: 2017 30th SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI), pp. 7–14. Ieee (2017)
24. Conrad, C., Mertz, M., Mester, R.: Contour-relaxed superpixels. In: Energy Minimization Methods in Computer Vision and Pattern Recognition, pp. 280–293. Springer Berlin Heidelberg, Berlin, Heidelberg (2013)
25. Di, S., Liao, M., Zhao, Y., Li, Y., Zeng, Y.: Image superpixel segmentation based on hierarchical multi-level li-slic. Optics & Laser Technology **135**, 106,703 (2021)
26. Falcão, A.X., Bergo, F.P.: Interactive volume segmentation with differential image foresting transforms. IEEE Transactions on Medical Imaging **23**(9), 1100–1108 (2004)
27. Falcão, A.X., Stolfi, J., de Alencar Lotufo, R.: The image foresting transform: Theory, algorithms, and applications. IEEE transactions on pattern analysis and machine intelligence **26**(1), 19–29 (2004)
28. Fang, L., Li, S., Kang, X., Benediktsson, J.A.: Spectral–spatial classification of hyperspectral images with a superpixel-based discriminative sparse model. IEEE Transactions on Geoscience and Remote Sensing **53**(8), 4186–4201 (2015)
29. Francis, J., Baburaj, M., George, S.N.: An $l_{1/2}$ and graph regularized subspace clustering method for robust image segmentation. ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM) **18**(2), 1–24 (2022)
30. Galvão, F.L., Falcão, A.X., Chowdhury, A.S.: Risf: recursive iterative spanning forest for superpixel segmentation. In: 2018 31st SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI), pp. 408–415. IEEE (2018)
31. Galvão, F.L., Guimarães, S.J.F., Falcão, A.X.: Image segmentation using dense and sparse hierarchies of superpixels. Pattern Recognition **108**, 107,532 (2020). DOI https://doi.org/10.1016/j.patcog.2020.107532. URL `https://www.sciencedirect.com/science/article/pii/S0031320320303356`
32. Gaur, U., Manjunath, B.S.: Superpixel embedding network. IEEE Transactions on Image Processing **29**, 3199–3212 (2020). DOI 10.1109/TIP.2019.2957937
33. Giraud, R., Ta, V.T., Papadakis, N.: Scalp: Superpixels with contour adherence using linear path. In: 2016 23rd International Conference on Pattern Recognition (ICPR), pp. 2374–2379 (2016). DOI 10.1109/ICPR.2016.7899991
34. 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)
35. Gong, J., Liao, N., Li, C., Ma, X., He, W., Guo, B.: Superpixel segmentation via contour optimized non-iterative clustering. In: International Conference on Neural Computing for Advanced Applications, pp. 645–658. Springer (2021)
36. Gong, Y., Sinnott, R.O., Rimba, P.: Rt-dbscan: Real-time parallel clustering of spatio-temporal data using spark-streaming. In: Y. Shi, H. Fu, Y. Tian, V.V. Krzhizhanovskaya, M.H. Lees, J. Dongarra, P.M.A. Sloot (eds.) Computational Science – ICCS 2018, pp. 524–539. Springer International Publishing, Cham (2018)
37. Grady, L.: Random walks for image segmentation. IEEE transactions on pattern analysis and machine intelligence **28**(11), 1768–1783 (2006)
38. Guan, J., Li, S., He, X., Chen, J.: Peak-graph-based fast density peak clustering for image segmentation. IEEE Signal Processing Letters **28**, 897–901 (2021)
39. Guevara, A., Conrad, C., Mester, R.: Boosting segmentation results by contour relaxation. In: 2011 18th IEEE International Conference on Image Processing, pp. 1405–1408. IEEE (2011)
40. Guevara, A., Conrad, C., Mester, R.: Boosting segmentation results by contour relaxation. In: 2011 18th IEEE International Conference on Image Processing, pp. 1405–1408. IEEE (2011). DOI 10.1109/ICIP.2011.6115703
41. Guigues, L., Cocquerez, J.P., Le Men, H.: Scale-sets image analysis. International Journal of Computer Vision **68**(3), 289–317 (2006)
42. Gupta, A.K., Seal, A., Khanna, P., Krejcar, O., Yazidi, A.: Awks: adaptive, weighted k-means-based superpixels for improved saliency detection. Pattern Analysis and Applications **24**(2), 625–639 (2021)

43. Huang, J.Y., Ding, J.J.: Generic image segmentation in fully convolutional networks by superpixel merging map. In: Proceedings of the Asian Conference on Computer Vision, pp. 723–737 (2020)
44. Huang, J.Y., Ding, J.J., Huang, P.C.: Learning based superpixel merging model for image segmentation. In: 2021 29th European Signal Processing Conference (EUSIPCO), pp. 661–665. IEEE (2021)
45. Huang, J.Z., Ng, M.K., Rong, H., Li, Z.: Automated variable weighting in k-means type clustering. IEEE transactions on pattern analysis and machine intelligence **27**(5), 657–668 (2005)
46. Jampani, V., Sun, D., Liu, M.Y., Yang, M.H., Kautz, J.: Superpixel sampling networks. In: Proceedings of the European Conference on Computer Vision (ECCV), pp. 352–368 (2018)
47. Ji, Z., Huang, Y., Sun, Q., Cao, G.: A spatially constrained generative asymmetric gaussian mixture model for image segmentation. Journal of Visual Communication and Image Representation **40**, 611–626 (2016)
48. Kang, X., Zhu, L., Ming, A.: Dynamic random walk for superpixel segmentation. IEEE Transactions on Image Processing **29**, 3871–3884 (2020). DOI 10.1109/TIP.2020.2967583
49. Kong, S., Fowlkes, C.C.: Recurrent pixel embedding for instance grouping. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 9018–9028 (2018)
50. Krause, A., Perona, P., Gomes, R.: Discriminative clustering by regularized information maximization. Advances in neural information processing systems **23**, 775–783 (2010)
51. Lempitsky, V., Vedaldi, A., Ulyanov, D.: Deep image prior. In: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 9446–9454. IEEE (2018)
52. Levinshtein, A., Stere, A., Kutulakos, K.N., Fleet, D.J., Dickinson, S.J., Siddiqi, K.: Turbopixels: Fast superpixels using geometric flows. IEEE transactions on pattern analysis and machine intelligence **31**(12), 2290–2297 (2009)
53. Li, H., Jia, Y., Cong, R., Wu, W., Kwong, S.T.W., Chen, C.: Superpixel segmentation based on spatially constrained subspace clustering. IEEE Transactions on Industrial Informatics **17**(11), 7501–7512 (2020)
54. Li, X., Zhou, Y., Zhang, X., Xu, S., Yu, P.: Cluster-based fine-to-coarse superpixel segmentation. Engineering Applications of Artificial Intelligence **102**, 104,281 (2021)
55. Li, Z., Chen, J.: Superpixel segmentation using linear spectral clustering. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1356–1363 (2015)
56. Liang, Y., Zhang, Y., Wu, Y., Tu, S., Liu, C.: Robust video object segmentation via propagating seams and matching superpixels. IEEE Access **8**, 53,766–53,776 (2020)
57. Lin, Q., Zhong, W., Lu, J.: Deep superpixel cut for unsupervised image segmentation. In: 2020 25th International Conference on Pattern Recognition (ICPR), pp. 8870–8876. IEEE (2021)
58. Liu, G., Duan, J.: Rgb-d image segmentation using superpixel and multi-feature fusion graph theory. Signal, Image and Video Processing **14**(6), 1171–1179 (2020)
59. Liu, M.Y., Tuzel, O., Ramalingam, S., Chellappa, R.: Entropy rate superpixel segmentation. In: CVPR 2011, pp. 2097–2104 (2011). DOI 10.1109/CVPR.2011.5995323
60. Liu, Y., Jiang, P.T., Petrosyan, V., Li, S.J., Bian, J., Zhang, L., Cheng, M.M.: Del: Deep embedding learning for efficient image segmentation. In: IJCAI, pp. 864–870 (2018). URL `https://doi.org/10.24963/ijcai.2018/120`
61. Loke, S.C., MacDonald, B.A., Parsons, M., Wünsche, B.C.: Accelerated superpixel image segmentation with a parallelized dbscan algorithm. Journal of Real-Time Image Processing **18**(6), 2361–2376 (2021)
62. Ma, D., Zhou, Y., Xin, S., Wang, W.: Convex and compact superpixels by edge-constrained centroidal power diagram. IEEE Transactions on Image Processing **30**, 1825–1839 (2020)
63. Machairas, V., Faessel, M., Cárdenas-Peña, D., Chabardes, T., Walter, T., Decenciere, E.: Waterpixels. IEEE Transactions on Image Processing **24**(11), 3707–3716 (2015)
64. Maierhofer, G., Heydecker, D., Aviles-Rivero, A.I., Alsaleh, S.M., Schonlieb, C.B.: Peekaboo-where are the objects? structure adjusting superpixels. In: 2018 25th IEEE International Conference on Image Processing (ICIP), pp. 3693–3697 (2018). DOI 10.1109/ICIP.2018.8451822
65. Mansilla, L.A., Miranda, P.A.: Image segmentation by oriented image foresting transform: Handling ties and colored images. In: 2013 18th International Conference on Digital Signal Processing (DSP), pp. 1–6. IEEE (2013)
66. Mansilla, L.A.C., Miranda, P.A.V.: Oriented image foresting transform segmentation: Connectivity constraints with adjustable width. In: 2016 29th SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI), pp. 289–296 (2016). DOI 10.1109/SIBGRAPI.2016.047
67. Martin, D.R., Fowlkes, C.C., Malik, J.: Learning to detect natural image boundaries using local brightness, color, and texture cues. IEEE transactions on pattern analysis and machine intelligence **26**(5), 530–549 (2004). DOI 10.1109/TPAMI.2004.1273918
68. Mathieu, B., Crouzil, A., Puel, J.B.: Oversegmentation methods: a new evaluation. In: Iberian Conference on Pattern Recognition and Image Analysis, pp. 185–193. Springer (2017)
69. Mester, R., Conrad, C., Guevara, A.: Multichannel segmentation using contour relaxation: fast super-pixels and temporal propagation. In: Scandinavian conference on image analysis, pp. 250–261. Springer (2011)
70. Mester, R., Conrad, C., Guevara, A.: Multichannel segmentation using contour relaxation: fast super-pixels and temporal propagation. In: A. Heyden, F. Kahl (eds.) Scandinavian conference on image analysis, pp. 250–261. Springer, Berlin, Heidelberg (2011)
71. Moore, A.P., Prince, S.J., Warrell, J., Mohammed, U., Jones, G.: Superpixel lattices. In: 2008 IEEE conference on computer vision and pattern recognition, pp. 1–8. IEEE (2008)
72. Neubert, P., Protzel, P.: Superpixel benchmark and comparison. In: Proc. Forum Bildverarbeitung, vol. 6, pp. 1–12. KIT Scientific Publishing (2012)
73. Neubert, P., Protzel, P.: Evaluating superpixels in video: Metrics beyond figure-ground segmentation. In: British Machine Vision Conference (2013)
74. Peng, H., Aviles-Rivero, A.I., Schönlieb, C.B.: Hers superpixels: Deep affinity learning for hierarchical entropy rate segmentation. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, pp. 217–226 (2022)
75. Polya, G.: Mathematics and plausible reasoning: Induction and analogy in mathematics, vol. 1. Princeton University Press (2020)
76. Qiao, N., Di, L.: An improved method of linear spectral clustering. Multimedia Tools and Applications **81**(1), 1287–1311 (2022)
77. Qin, X., Zhang, Z., Huang, C., Dehghan, M., Zaiane, O.R., Jagersand, M.: U2-net: Going deeper with nested u-structure for salient object detection. Pattern Recognition **106**, 107,404 (2020)

78. Ren, L., Zhao, L., Wang, Y.: A superpixel-based dual window rx for hyperspectral anomaly detection. IEEE Geoscience and Remote Sensing Letters **17**(7), 1233–1237 (2019)
79. Ren, X., Malik, J.: Learning a classification model for segmentation. In: Proceedings Ninth IEEE International Conference on Computer Vision, vol. 2, pp. 10–17. IEEE Computer Society (2003). DOI 10.1109/ICCV.2003.1238308
80. Rodriguez, A., Laio, A.: Clustering by fast search and find of density peaks. Science **344**(6191), 1492–1496 (2014). DOI 10.1126/science.1242072. URL `https://www.science.org/doi/abs/10.1126/science.1242072`
81. Schick, A., Fischer, M., Stiefelhagen, R.: Measuring and evaluating the compactness of superpixels. In: Proceedings of the 21st international conference on pattern recognition (ICPR2012), pp. 930–934. IEEE (2012)
82. Schick, A., Fischer, M., Stiefelhagen, R.: An evaluation of the compactness of superpixels. Pattern Recognition Letters **43**, 71–80 (2014)
83. Sellars, P., Aviles-Rivero, A.I., Schönlieb, C.B.: Superpixel contracted graph-based learning for hyperspectral image classification. IEEE Transactions on Geoscience and Remote Sensing **58**(6), 4180–4193 (2020)
84. Sethian, J.A.: Level set methods and fast marching methods: evolving interfaces in computational geometry, fluid mechanics, computer vision, and materials science, vol. 3. Cambridge university press (1999)
85. Shah, S.A.H., Li, L., Li, Y., Zhang, J.: Superpixel segmentation via density peaks. In: 2021 4th International Conference on Information and Computer Technologies (ICICT), pp. 93–98. IEEE (2021)
86. Shen, J., Hao, X., Liang, Z., Liu, Y., Wang, W., Shao, L.: Real-time superpixel segmentation by dbscan clustering algorithm. IEEE Transactions on Image Processing **25**(12), 5933–5942 (2016). DOI 10.1109/TIP.2016.2616302
87. Sheng, B., Li, P., Mo, S., Li, H., Hou, X., Wu, Q., Qin, J., Fang, R., Feng, D.D.: Retinal vessel segmentation using minimum spanning superpixel tree detector. IEEE transactions on cybernetics **49**(7), 2707–2719 (2018)
88. Shi, J., Yan, Q., Xu, L., Jia, J.: Hierarchical image saliency detection on extended cssd. IEEE transactions on pattern analysis and machine intelligence **38**(4), 717–729 (2015)
89. Stutz, D.: Superpixel segmentation: An evaluation. In: German conference on pattern recognition, pp. 555–562. Springer (2015)
90. Stutz, D., Hermans, A., Leibe, B.: Superpixels: An evaluation of the state-of-the-art. Computer Vision and Image Understanding **166**, 1–27 (2018)
91. Suzuki, T.: Superpixel segmentation via convolutional neural networks with regularized information maximization. In: ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 2573–2577 (2020). DOI 10.1109/ICASSP40776.2020.9054140
92. Tu, W.C., Liu, M.Y., Jampani, V., Sun, D., Chien, S.Y., Yang, M.H., Kautz, J.: Learning superpixels with segmentation-aware affinity loss. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 568–576 (2018)
93. Ullah, S., Bhatti, N., Zia, M.: Adaptive tuning of slic parameter k. Multimedia Tools and Applications **80**(17), 25,649–25,672 (2021)
94. Vargas-Muñoz, J.E., Chowdhury, A.S., Alexandre, E.B., Galvão, F.L., Miranda, P.A.V., Falcão, A.X.: An iterative spanning forest framework for superpixel segmentation. IEEE Transactions on Image Processing **28**(7), 3477–3489 (2019)
95. Wang, G., Wei, Y., Tse, P.: Clustering by defining and merging candidates of cluster centers via independence and affinity. Neurocomputing **315**, 486–495 (2018). DOI https://doi.org/10.1016/j.neucom.2018.07.043. URL `https://www.sciencedirect.com/science/article/pii/S0925231218308786`
96. Wang, H., Shen, J., Yin, J., Dong, X., Sun, H., Shao, L.: Adaptive nonlocal random walks for image superpixel segmentation. IEEE Transactions on Circuits and Systems for Video Technology **30**(3), 822–834 (2020). DOI 10.1109/TCSVT.2019.2896438
97. Wang, J., Luan, Z., Yu, Z., Ren, J., Gao, J., Yuan, K., Xu, H.: Superpixel segmentation with squeeze-and-excitation networks. Signal, Image and Video Processing pp. 1–8 (2022)
98. Wang, K., Li, L., Zhang, J.: End-to-end trainable network for superpixel and image segmentation. Pattern Recognition Letters **140**, 135–142 (2020). DOI https://doi.org/10.1016/j.patrec.2020.09.016. URL `https://www.sciencedirect.com/science/article/pii/S0167865520303421`
99. Wang, M., Liu, X., Gao, Y., Ma, X., Soomro, N.Q.: Superpixel segmentation: A benchmark. Signal Processing: Image Communication **56**, 28–39 (2017)
100. Wang, N., Zhang, Y.: Adaptive and fast image superpixel segmentation approach. Image and Vision Computing **116**, 104,315 (2021)
101. Wang, W., Wu, C.: Image segmentation by correlation adaptive weighted regression. Neurocomputing **267**, 426–435 (2017). DOI https://doi.org/10.1016/j.neucom.2017.06.046. URL `https://www.sciencedirect.com/science/article/pii/S0925231217311748`
102. Wang, X., Zhao, Q., Fan, L., Zhao, Y., Wang, T., Yan, Q., Chen, L.: Semasuperpixel: A multi-channel probability-driven superpixel segmentation method. In: 2021 IEEE International Conference on Image Processing (ICIP), pp. 1859–1863. IEEE (2021)
103. Wei, X., Yang, Q., Gong, Y., Ahuja, N., Yang, M.H.: Superpixel hierarchy. IEEE Transactions on Image Processing **27**(10), 4838–4849 (2018)
104. Wu, J., Liu, C., Li, B.: Texture-aware and structure-preserving superpixel segmentation. Computers & Graphics **94**, 152–163 (2021)
105. Wu, R., Du, Y., Li, H., Dai, Y.: Stereo superpixel segmentation via dual-attention fusion networks. In: 2021 IEEE International Conference on Multimedia and Expo (ICME), pp. 1–6. IEEE (2021)
106. Wu, X., Chen, Y., Liu, X., Shen, J., Zhuo, K., Zhao, W.: Superpixel via coarse-to-fine boundary shift. Applied Intelligence **50**(7), 2079–2092 (2020)
107. Yamaguchi, K., McAllester, D., Urtasun, R.: Efficient joint segmentation, occlusion labeling, stereo and flow estimation. In: F. David, T. Pajdla, B. Schiele, T. Tuytelaars (eds.) European Conference on Computer Vision, vol. 8693, pp. 756–771. Springer, Cham (2014). DOI 10.1007/978-3-319-10602-1_49
108. Yang, F., Sun, Q., Jin, H., Zhou, Z.: Superpixel segmentation with fully convolutional networks. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2020)
109. Yao, J., Boben, M., Fidler, S., Urtasun, R.: Real-time coarse-to-fine topologically preserving segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 2947–2955 (2015)
110. Yu, Y., Yang, Y., Liu, K.: Edge-aware superpixel segmentation with unsupervised convolutional neural networks. In: 2021 IEEE International Conference on Image Processing (ICIP), pp. 1504–1508. IEEE (2021)

111. Yuan, H., Wu, S., Cheng, P., An, P., Bao, S.: Nonlocal random walks algorithm for semi-automatic 2d-to-3d image conversion. IEEE Signal Processing Letters **22**(3), 371–374 (2015). DOI 10.1109/LSP.2014.2359643
112. Yuan, Q., Lu, S., Huang, Y., Sha, W.: Sin: Superpixel interpolation network. In: Pacific Rim International Conference on Artificial Intelligence, pp. 293–307. Springer (2021)
113. Yuan, Y., Zhang, W., Yu, H., Zhu, Z.: Superpixels with content-adaptive criteria. IEEE Transactions on Image Processing **30**, 7702–7716 (2021)
114. Yuan, Y., Zhu, Z., Yu, H., Zhang, W.: Watershed-based superpixels with global and local boundary marching. IEEE Transactions on Image Processing **29**, 7375–7388 (2020). DOI 10.1109/TIP.2020.3002078
115. Zhang, B., Kang, X., Ming, A.: Bp-net: deep learning-based superpixel segmentation for rgb-d image. In: 2020 25th International Conference on Pattern Recognition (ICPR), pp. 7433–7438. IEEE (2021)
116. Zhang, J., Aviles-Rivero, A.I., Heydecker, D., Zhuang, X., Chan, R., Schönlieb, C.B.: Dynamic spectral residual superpixels. Pattern Recognition **112**, 107,705 (2021)
117. Zhang, J., Chen, J., Wang, Q., Chen, S.: Spatiotemporal saliency detection based on maximum consistency superpixels merging for video analysis. IEEE Transactions on Industrial Informatics **16**(1), 606–614 (2019)
118. Zhang, Y., Guo, Q., Zhang, C.: Simple and fast image superpixels generation with color and boundary probability. The Visual Computer **37**, 1061–1074 (2021). DOI 10.1007/s00371-020-01852-2
119. Zhao, W., Fu, Y., Wei, X., Wang, H.: An improved image semantic segmentation method based on superpixels and conditional random fields. Applied Sciences **8**(5), 837 (2018)
120. Zhou, X., Wang, Y., Zhu, Q., Xiao, C., Lu, X.: Ssg: superpixel segmentation and grabcut-based salient object segmentation. The Visual Computer **35**(3), 385–398 (2019)
121. Zhu, L., She, Q., Zhang, B., Lu, Y., Lu, Z., Li, D., Hu, J.: Learning the superpixel in a non-iterative and lifelong manner. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 1225–1234 (2021)