Novel Arc-Cost Functions and Seed Relevance Estimations for Compact and Accurate Superpixels

Felipe C Belém, Isabela B Barcelos, Leonardo M João, Benjamin Perret, Jean Cousty, Silvio J F Guimarães, Alexandre X Falcão

Novel Arc-Cost Functions and Seed Relevance Estimations for Compact and Accurate Superpixels

Felipe C. Belém[1,3*], Isabela B. Barcelos[2], Leonardo M. João[1], Benjamin Perret[3], Jean Cousty[3], Silvio J. F. Guimarães[2] and Alexandre X. Falcão[1]

[1]LIDS, University of Campinas, Campinas, 13083-851, São Paulo, Brazil .
[2]ImScience, Pontifical Catholic University of Minas Gerais, Belo Horizonte, 30535-901, Minas Gerais, Brazil .
[3]LIGM, Univ. Gustave-Eiffel, CNRS, F-77454, Marne-la-Valée, France .

*Corresponding author(s). E-mail(s): felipe.belem@{ic.unicamp.br,esiee.fr};
Contributing authors: isabela_borlido@hotmail.com; leonardo.joao@ic.unicamp.br; benjamin.perret@esiee.fr; jean.cousty@esiee.fr; sjamil@pucminas.br; afalcao@ic.unicamp.br;

Abstract

On the verge of superpixel methods exploiting saliency information, the *Superpixels through Iterative CLEarcutting* (SICLE) framework has reported fast and accurate superpixel delineation. It is composed of three steps: (i) seed oversampling; (ii) superpixel generation; and (iii) seed removal. It starts from (i) and applies several iterations of (ii) and (iii) until reaching the desired superpixel quantity. In this work, we improve SICLE such that it can now generate compact superpixels with accurate delineation. We exploit differential computation and propose several novel functions for steps (ii) and (iii) for proper saliency incorporation, compact superpixel generation, and improvement in speed and delineation. Results show that, with our proposals, SICLE achieves state-of-the-art performance in delineation and speed whenever saliency is absent with on-par compacity. When an accurate saliency map is provided, its performance improves significantly and requires only two iterations for segmentation.

Keywords: Superpixel,Saliency,Multiscale,Segmentation

1 Introduction

In order to minimize workload and handle more meaningful data than simple pixels, medical applications [1–3], semantic segmentation methods [4], and video segmentation algorithms [5] often recur to superpixel segmentation. In brief, superpixels are disjoint groups of connected pixels that present similar characteristics, like color. Consequently, superpixel segmentation methods aim to represent any object by its comprising parts.

Despite the diversity of the superpixel literature, it is possible to list a set of crucial properties for any algorithm [6]: (i) boundary adherence; (ii) compactness; and (iii) efficiency. However, achieving simultaneously all three is a challenging task [6]. For instance, the popular *Simple Linear Iterative Clustering* (SLIC) [7] and the *Iterative Boundary Implicit Identification* (IBIS) [8] are two fast approaches for generating highly compact superpixels, but with moderate delineation performance. On the other hand, *Superpixel Hierarchy*

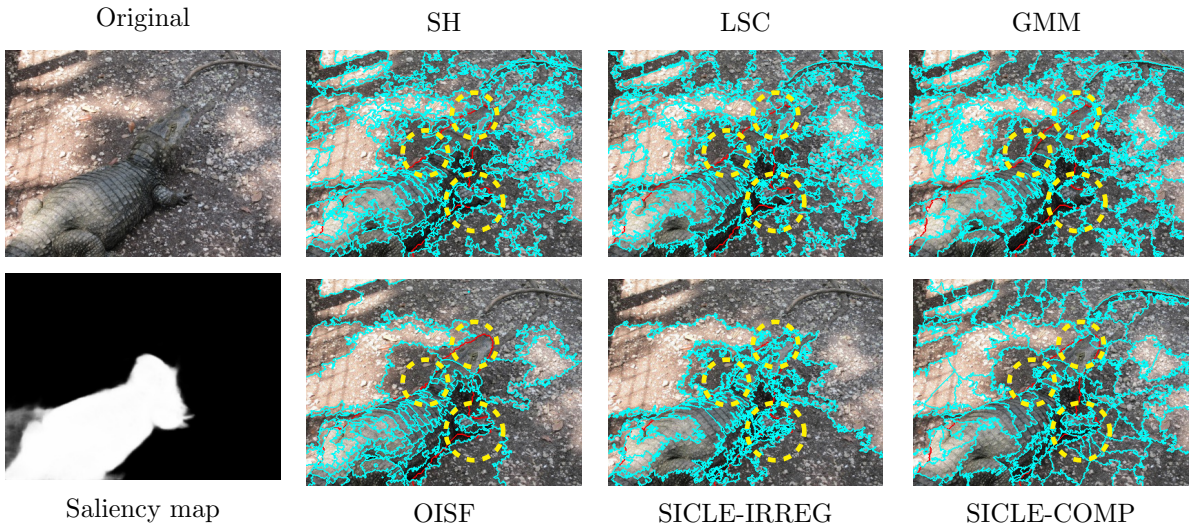

Fig. 1 Segmentation requiring 100 superpixels. The object boundaries and the superpixel borders are depicted in red and cyan, respectively. Yellow circles indicate regions where SICLE best approximates the object borders in contrast with the baselines.

(SH) [9] and *Dynamic and Iterative Spanning Forest* (DISF) [10] efficiently produce superpixels with high boundary adherence, but highly irregular as well. One may perceive such irregularity as an overfitting behavior [11]. Either way, these methods, by aiming the delineation of every possible border, are unaware of the user's desire to segment a specific object, nor might they be adaptable to do such a task.

By acknowledging the complex problem of detecting all borders, several recent methods exploit deep-learning strategies for learning a pattern (*e.g.*, relevant borders) for superpixel generation. Still, such algorithms often present moderate delineation performance and moderate compactness, demanding high computational power. Moreover, both supervised [12, 13] and unsupervised [14, 15] approaches are dependable on data, specially annotated for the former.

Using a different strategy, new algorithms exploit the probable object location, usually computed through deep learning and termed saliency, for adapting the segmentation to the user's needs. Thus, by maximizing the delineation of a specific object, these saliency-based approaches dismiss the segmentation of the remaining ones. To the best of our knowledge, the *Saliency-based Superpixels* (SS) [16] is the first method to adopt this paradigm, generating superpixels through a saliency-driven bottom-up merging process. The *Object-based Iterative Spanning Forest* (OISF) [17, 18], another example, can alter the superpixel morphology with respect to the given information, but is highly dependable on its quality. Conversely, the *Object-based DISF* (ODISF) [19] offers a less-dependable strategy with higher efficiency and more accurate delineation performance. However, ODISF is unable to produce compact superpixels and cannot further improve its segmentation for more accurate object information.

Another saliency-based framework, named *Superpixels through Iterative CLEarcutting* (SICLE) [20], generalizes both DISF and ODISF for computing superpixels more efficiently. SICLE is composed of three independent steps: (i) seed oversampling; (ii) path-based superpixel generation; and (iii) seed removal. Briefly, it starts from (i) and, for several iterations, applies (ii) and (iii) until obtaining a desired number of superpixels. Even though it inherits the drawbacks of both DISF and ODISF, like their irregularity, its architecture with separated and self-contained steps favors modifications and improvements with reduced effort while maintaining its state-of-the-art efficiency and efficacy.

In this work, we propose several functions for the SICLE framework for effective object delineation through compact or irregular superpixels, exploiting the quality of the saliency estimation. That is, we introduce two novel arc-cost functions

for step (ii), which provides a twofold property: for poor-quality saliency information (*e.g.*, inaccurate object boundary definition), our proposals sustain the effective delineation of ODISF; for state-of-the-art quality, the novel functions improve the border detection significantly without reinforcing erroneous estimations in the segmentation, differently from OISF. For step (iii), we present a novel seed criterion for spreading superpixels and three new penalizations for a more accurate seed removal by focusing on the object's salient borders. Furthermore, we performed an extensive study for building both irregular and compact SICLE variants which, as shown by the experimental results, have surpassed several state-of-the-art methods in delineation (see Figure 1) and speed. For the latter, SICLE variants require only two iterations for segmentation due to our proposed functions. Finally, since saliency usually depicts a single object, we debate the challenges of saliency-based superpixel segmentation using SICLE and, thus, using our proposals with multiple objects of interest.

In brief, we can list the following contributions:

- novel arc-cost functions which incorporate prior object information and produce compact and irregular superpixels robust to saliency errors;
- novel seed relevance criterion for superpixel spreading and three novel seed relevance saliency-based penalization for improving delineation and assuring overall compactness;
- thorough optimization of SICLE and an extensive performance evaluation in five datasets, considering well-known metrics and seven state-of-the-art methods as baselines, resulting in two SICLE variants for effective delineation and compact superpixel generation;
- debate over multiobject handling in SICLE.

This article extends and completes a previous work published at a conference [21]. Differently from it, the saliency importance in our arc-cost functions is not a binary parameter (thus, may be optimized), and the seed penalization is no longer associated with such importance, being independently defined. The remaining contributions are original and do not appear in [21].

This work is organized as follows. In Section 2, we present a brief overview of state-of-the-art superpixel segmentation algorithms. Subsequently, in Section 3, we introduce the theoretical framework necessary for describing our proposed functions in Section 4. Then, the experimental results and a debate over multiobject handling are presented in Sections 5 and 6. Finally, we conclude this work in Section 7 drawing possible future work.

2 Related Works

In this section, we present and debate the most notable state-of-the-art superpixel segmentation algorithms. We can broadly classify them based on their behavior towards prior object information: (i) classical; (ii) deep-learning-based; and (iii) saliency-based. Unfortunately, most surveys do not encompass (ii) and (iii), being recently developed strategies. Still, for a more detailed discussion on this topic, we recommend several distinguished works [6, 22, 23].

The majority of superpixel methods are classical and can be further divided into three subgroups based on their strategy: (i) clustering-based; (ii) graph-based; and (iii) path-based. As the label suggests, the algorithms in (i) model the segmentation task as a pixel-clustering problem, and the most well-known work is the *Simple Linear Iterative Clustering* (SLIC) [7]. In SLIC, superpixels are generated through an adapted K-means in a 5-dimensional feature space comprising the pixels' color and spatial coordinates. Given its simplicity and efficiency, several works extended or were inspired by SLIC, producing highly effective and efficient algorithms. One of such is the *Linear Spectral Clustering* (LSC) [24], which produces superpixels using a weighted K-means in a 10-dimensional feature space. Another example is the *Iterative Boundaries Implicit Identification* (IBIS) [8], an efficient approach for generating highly compact superpixels by minimizing the number of pixel operations without any performance degradation. Although certain methods classify themselves as content-sensitive [25], content-adaptive [26], or even texture aware [27], they often rely on clustering techniques, specially K-means. However, such approaches, including those that use *Gaussian Mixture Models* (GMM) [28] and *Density-based Spatial Clustering of Applications with Noise* (DBSCAN) [29] for clustering, tend not to ensure the desired number of superpixels [30]. Moreover, although they

generate compact superpixels, their delineation performance is usually moderate.

Those termed as graph-based usually generate superpixels through edge operations in the image graph. For instance, the *Entropy Rate Superpixels* (ERS) [31] performs such task by optimizing the entropy of a random walk in the graph's topology. However, ERS tends to present high computational time. On the other hand, the *Superpixel Hierarchy* (SH) [9] builds a hierarchical superpixel segmentation through the Borůvka algorithms, achieving linear time complexity. In contrast to clustering-based approaches, both ERS and SH provides accurate object delineation at the expense of generating highly irregular superpixels. Furthermore, for SH, such delineation may be compromised by errors propagated from lower levels to upper ones in the hierarchy.

Similarly to SLIC and K-means, most path-based approaches base themselves on the *Image Foresting Transform* (IFT) framework for generating superpixels. In such category, superpixels are built by path searching from a subset of vertices (*i.e.*, seeds) to the remaining ones based on a similarity criterion. Aside from guaranteeing the desired number of superpixels, IFT-based methods often achieve top delineation performance. We may cite state-of-the-art methods, such as the *Iterative Spanning Forest* (ISF) [32] framework, which inspired later and more effective approaches. The *Recursive ISF* (RISF) [33] method, for instance, produces a sparse hierarchical superpixel segmentation by executing several ISFs at each level. As a different example, the *Dynamic ISF* (DISF) [10] uses a strategy of seed oversampling and iterative seed removal for computing a superpixel segmentation using a dynamic arc-cost estimation. Lastly, although the most common drawback of such category is the difficulty in producing compact superpixels [6], the *Waterpixels* [30] algorithm manages to produce compact superpixels using the watershed transformation.

Classical methods usually produce accurate object delineation but are unaware of the user's expectations towards the objects. On the other hand, in recent years, several novel approaches have tried to overcome such difficulty by exploiting deep-learning techniques to generate superpixels. One possible approach is to extend existing methods for incorporating features learned through deep-learning, such as the one proposed in [34]. However, the authors in [35] argue that directly incorporating deep features is insufficient for proper delineation. Thus, they propose the construction of pixel-affinities maps through deep learning using a new loss function, named *SEgmentation Aware Loss* (SEAL), and modify preexistent superpixel algorithms (*e.g.*, ERS) for considering such information, resulting in the SEAL-ERS approach. Similarly, the *Deep Affinity Learning for Hierarchical ERS* (DALHERS) [36] also computes pixel-affinity maps and generates superpixels hierarchically through the Borůvka method considering the entropy of the graph. Another strategy consists of building end-to-end supervised [12, 13, 37] or unsupervised [14, 15] networks for superpixel generation. However, although promising, both approaches demand more research [20, 35] since they present: (a) high data dependency; (b) moderate delineation; and (c) high computational cost.

In order to achieve the best of both strategies (*i.e.*, effectiveness and incorporating object information), current saliency-based algorithms, apart from [16], generalize classical algorithms for incorporating saliency information, often estimated through deep-learning. As an example, the *Object-based ISF* (OISF) [17, 18] is a generalization of ISF that allows user control over the superpixel's morphology and displacement with respect to an object saliency map. However, given that OISF is slow and highly dependable on the map's quality, the authors in [19] propose *Object-based DISF* (ODISF), a faster and more robust method for achieving accurate object delineation. Still, for both DISF and ODISF, it is difficult to establish the number of iterations for segmentation, and they do not produce compact superpixels. Moreover, although robust, they cannot improve their delineation for better saliency maps. Finally, in order to solve the iteration problem, the authors in [20] proposed *Superpixels through Iterative CLEarcutting* (SICLE), a generalization of both DISF and ODISF, in which the user can define a maximum number of iterations, producing a saliency-based multiscale segmentation.

3 Theoretical Background

In this section, we present the theoretical background of our proposed framework.

While Section 3.1 introduces basic concepts on images, saliency maps, and image digraphs, Section 3.2 presents the *Image Foresting Transform* (IFT) [38], the core algorithm for object delineation for the *Superpixels through Iterative CLEarcutting* (SICLE) [20] framework (described in Section 3.3).

3.1 Image Digraph

We may define an *image* I as a pair $\langle \mathcal{I}, \mathbf{F} \rangle$ such that every $p \in \mathcal{I} \subset \mathbb{Z}^2$ denotes a *picture element* (*i.e.*, pixel) whose *feature vector* (*e.g.*, CIELAB color in this work) is represented by $\mathbf{F}(p) \in \mathbb{R}^m$, for $m \in \mathbb{N}_{>0}$. Thus, I is a *grayscale* or a *color* image whenever $m = 1$ or $m > 1$, respectively. For example, an *object saliency map* $O = \langle \mathcal{I}, \mathbf{O} \rangle$ is a grayscale image in which $\mathbf{O}(p) \in [0, 1]$ defines a proportional object likelihood, often referred as *(object) saliency*, $\forall p \in \mathcal{I}$. Intuitively, brighter is the intensity $\mathbf{O}(p)$ of p, higher is its probability of belonging to the object of interest.

From I, we can build an *image digraph* $G = \langle \mathcal{V}, \mathcal{A} \rangle$ in which $\mathcal{V} \subseteq \mathcal{I}$ is the *vertex set* and $\mathcal{A} \subset \mathcal{V}^2$ is the *arc set*. A popular approach for establishing the arcs of G is through $\mathcal{A}_r = \{\langle x, y \rangle : \|x - y\|_2 \leq r\}$, given $x, y \in \mathcal{V}$ and $r \in \mathbb{R}_{\geq 0}$. One may note that $\mathcal{A}_{\sqrt{2}}$ represents the 8-adjacency relation. If $\exists \langle x, y \rangle \in \mathcal{A}$ then we term x as *adjacent of* y. Finally, in this work, every $\langle x, y \rangle \in \mathcal{A}$ is unique and $x \neq y$ (*i.e.*, G is *simple*).

A *(directed) path* $\rho = \langle v_1, \ldots, v_k \rangle$ is a sequence of distinct vertices such that $\langle v_i, v_{i+1} \rangle \in \mathcal{A}$, for $i, k \in \mathbb{N}_{>0}$ and $i < k$, and it is *trivial* when $k = 1$. Whenever ρ is *non-trivial*, we name v_i as *predecessor* of v_{i+1} and v_{i+1} as *sucessor* of v_i in ρ. Moreover, v_1 is the *root* of ρ while v_k is its *terminus*, and we may explicitly present both by $\rho_{v_1 \rightsquigarrow v_k}$. However, for simplicity, we may omit the root from such notation. For instance, $\rho_{s \rightsquigarrow x} \odot \langle x, y \rangle = \rho_x \odot \langle x, y \rangle = \langle v_1 = s, \ldots, v_k = x, y \rangle$ denotes the *concatenation* of the path ρ_x, irrespective of its origin $s \in \mathcal{V}$, with an arc $\langle x, y \rangle$, merging the two instances of x into one.

3.2 Image Foresting Transform

In [38], the authors proposed the *Image Foresting Transform* (IFT), a framework highly effective for object delineation [39, 40]. Briefly, the IFT, in this work, finds a path with optimum cost from a *seed* $s \in \mathcal{S}$ to every non-seed vertex $v \in \mathcal{V} \setminus \mathcal{S}$, being $\mathcal{S} \subset \mathcal{V}$ the *seed set*.

We can compute the *path-cost* $\mathbf{f}_*(\rho) \in \mathbb{R}_{\geq 0}$ of any path $\rho \in \mathscr{P}$, where $\mathscr{P}$ is the set of all possible ones in G, through a *path-cost function* $\mathbf{f}_*$. For instance, Equation 1 illustrates two different path-functions $\mathbf{f}_{\max}, \mathbf{f}_{\text{sum}}$ effective for superpixel segmentation [10, 17, 19, 32]:

$$\begin{aligned} \mathbf{f}_*(\langle x \rangle) &= \begin{cases} 0, & \text{if } x \in \mathcal{S} \\ \infty, & \text{otherwise} \end{cases} \\ \mathbf{f}_{\max}(\rho_x \odot \langle x, y \rangle) &= \max\{\mathbf{f}_{\max}(\rho_x), \mathbf{w}_*(x, y)\} \\ \mathbf{f}_{\text{sum}}(\rho_x \odot \langle x, y \rangle) &= \mathbf{f}_{\text{sum}}(\rho_x) + \mathbf{w}_*(x, y) \end{aligned} \tag{1}$$

in which $\mathbf{w}_*(x, y) \in \mathbb{R}_{\geq 0}$ denotes the *arc-cost function* of $\langle x, y \rangle \in \mathcal{A}$. A path ρ_x is said to be *optimum* if, for any other $\varrho_x \in \mathscr{P}$, $\mathbf{f}_*(\rho_x) \leq \mathbf{f}_*(\varrho_x)$.

Thus, by exploiting a generalization of the Dijkstra's algorithm, the IFT assigns an optimum path ρ_x, $\forall x \in \mathcal{V}$, while minimizing a *cost map* $\mathbf{C}(x) = \min_{\rho_x \in \mathscr{P}}\{\mathbf{f}_*(\rho_x)\}$. Concomitantly, it builds a *predecessor map* $\mathbf{P}$ that maps x to its predecessor in ρ_x or to a unique marker when x is its root (*i.e.*, $x \in \mathcal{S}$). Furthermore, one can recursively map, through $\mathbf{P}$, x to its *root* $s = \mathbf{R}(x) \in \mathcal{S}$ and, consequently, to its *tree* $\mathcal{T}_s = \{y : \mathbf{R}(y) = s\}$. Note that, in this work, every tree is a superpixel. Lastly, if $\mathbf{f}_*$ satisfy certain conditions [41], $\forall s \in \mathcal{S}$, $\mathcal{T}_s$ is an *optimum-path tree*, resulting in an *optimum-path forest*. Otherwise, $\mathcal{T}_s$ still exhibits important properties for segmentation [42].

3.3 Superpixels through Iterative CLEarcutting

Figure 2 presents the SICLE framework for superpixel generation. Therefore, the construction of a SICLE variant involves selecting: (i) a seed oversampling strategy; (ii) a path-cost and an arc-cost function (for the IFT); and for estimating the importance of a seed, (iii) a seed preservation curve, a seed relevance criterion, and a object penalization. Differently from most methods [7, 24, 25, 32], SICLE starts by selecting a number of N_0 of seeds significantly higher than the final number N_f of superpixels (*i.e.*, $N_0 \gg N_f$), given $N_0, N_f \in \mathbb{N}_{>0}$. The motivation is based on the *seed redundancy* premise: if two seeds are similar (*i.e.*, spatial position and features), then they are similarly *relevant* (*i.e.*, promote accurate

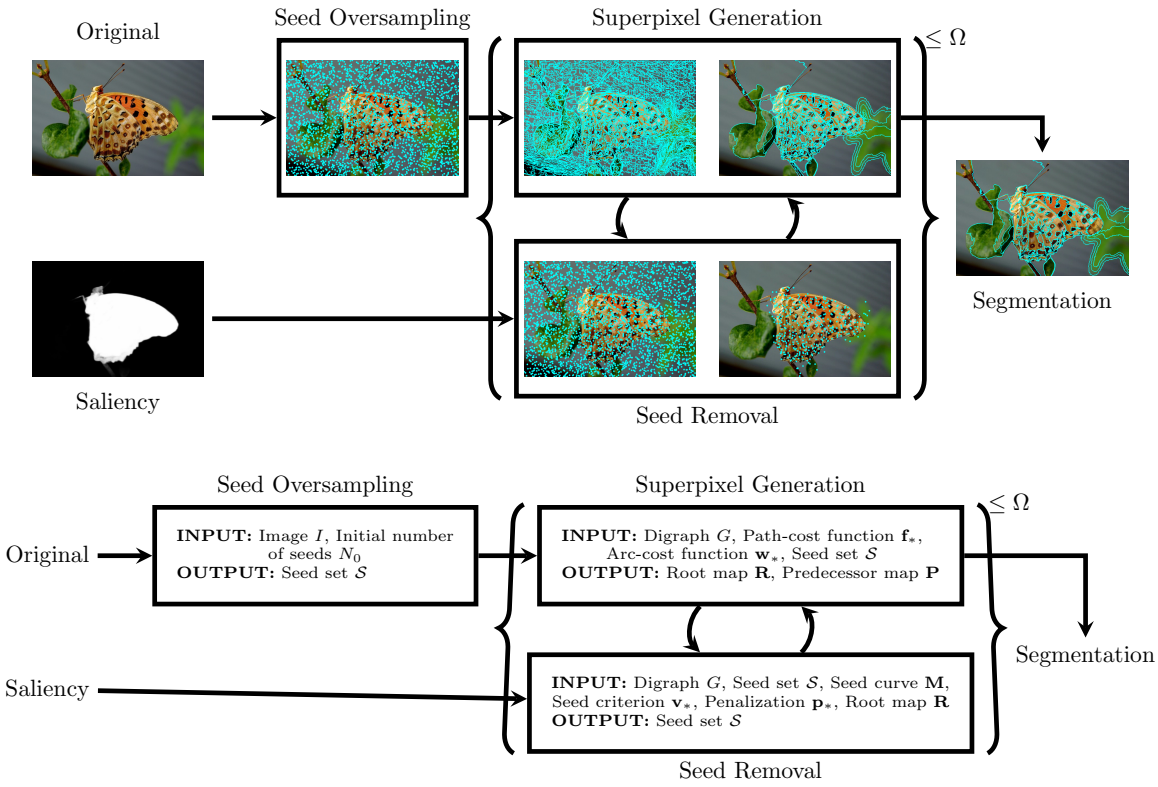

Fig. 2 The top image shows the SICLE framework for generating 100 superpixels using a deep-learning saliency estimator [43]. Superpixel borders and seeds are depicted by cyan lines and dots, respectively. Whereas the bottom image explicit the inputs and outputs of each process in the pipeline.

object delineation). This premise can be empirically perceived since different approaches achieve equivalent delineation [20] and, consequently, one may opt for a simple random sampling, especially given the reported loss of efficiency of saliency-based samplings [44].

In the subsequent step, superpixels are generated using the seed-restricted version of the IFT and, so far, it has been implemented considering the $\mathbf{f}_{\max}$ path-cost function. For estimating the arc-cost $\langle x, y\rangle \in \mathcal{A}$, SICLE considered a dynamic arc-cost estimation [39] $\mathbf{w}_{\text{dyn}}(x, y) = \left\|\overline{\mathbf{F}}(\mathcal{T}_s) - \mathbf{F}(y)\right\|_2$, given that $s = \mathbf{R}(x)$, which considers the mean features of the tree $\mathcal{T}_s$ *growing* during the IFT execution. Due its instability [20], the $\mathbf{w}_{\text{root}}(x, y) = \|\mathbf{F}(\mathbf{R}(x)) - \mathbf{F}(y)\|_2$ was also evaluated in which uses the immutable features of the seed s, being preferred over $\mathbf{w}_{\text{dyn}}$.

Due to oversampling, it is necessary to remove a total of $N_0 - N_f$ seeds from $\mathcal{S}$ in at most $\Omega \in \mathbb{N}_{>0} > 1$ iterations to ensure N_f superpixels. For that, a portion of the most irrelevant are removed *per* iteration $i \in \mathbb{N}_{>0} < \Omega$. Or, conversely, the $\mathbf{M}(i) = \max\left\{(N_0)^{1-i/(\Omega-1)}, N_f\right\}$ most relevant seeds are maintained for the subsequent iteration $i + 1$ while the remaining ones are discarded. As one may note, exactly $\lceil(\Omega - 1)(1 - \log_{N_0} N_f)\rceil + 1$ iterations are required for reaching the desired quantity of superpixels, being ideally two.

We can estimate the relevance of a seed s based on certain information (or attributes) extracted from its respective tree $\mathcal{T}_s$, like color-based (*e.g.*, mean color, color histograms, pattern spectra) or shape-based (*e.g.*, size, circularity, compactness) ones. First, in terms of particular features, we can compute its *mean features* $\overline{\mathbf{F}}(\mathcal{T}_s) = \sum_{v\in\mathcal{T}_s} \mathbf{F}(v)/\left|\mathcal{T}_s\right|$, its *mean saliency* $\overline{\mathbf{O}}(\mathcal{T}_s) = \sum_{v\in\mathcal{T}_s} \mathbf{O}(v)/\left|\mathcal{T}_s\right|$, and its *centroid* $\overline{\mathbf{c}}(\mathcal{T}_s) = \sum_{v\in\mathcal{T}_s} v/\left|\mathcal{T}_s\right|$. From that, we can compute the *color gradient* between two trees $\mathcal{T}_s$ and $\mathcal{T}_t$ by $\nabla_{\overline{\mathbf{F}}}(\mathcal{T}_s, \mathcal{T}_t) = \left\|\overline{\mathbf{F}}(\mathcal{T}_s) - \overline{\mathbf{F}}(\mathcal{T}_t)\right\|_2$. Similarly, we may compute the *saliency gradient* and the *distance between centroids* by

$\nabla_{\overline{\mathbf{O}}}(\mathcal{T}_s, \mathcal{T}_t) = \left\|\overline{\mathbf{O}}(\mathcal{T}_s) - \overline{\mathbf{O}}(\mathcal{T}_t)\right\|_1$ and $\nabla_{\overline{\mathbf{c}}}(\mathcal{T}_s, \mathcal{T}_t) = \left\|\overline{\mathbf{c}}(\mathcal{T}_s) - \overline{\mathbf{c}}(\mathcal{T}_t)\right\|_2$, respectively.

Also, given a seed $s \in \mathcal{S}$, we can establish the *(immediate) adjacents* of its tree $\mathcal{T}_s$ from $\overline{\mathbf{A}}(\mathcal{T}_s) = \{\mathcal{T}_t : \exists\, \langle x, y\rangle \in \mathcal{A}\}$ given $x \in \mathcal{T}_s$, $y \in \mathcal{T}_t$, and $s \neq t$. Consequently, we can compute the *maximum* and *minimum color contrast* of $\mathcal{T}_s$ amongst its neighbors through $\nabla^{+}_{\overline{\mathbf{F}}}(\mathcal{T}_s) = \max_{\mathcal{T}_t \in \overline{\mathbf{A}}(\mathcal{T}_s)} \left\{\nabla_{\overline{\mathbf{F}}}(\mathcal{T}_s, \mathcal{T}_t)\right\}$ and $\nabla^{-}_{\overline{\mathbf{F}}}(\mathcal{T}_s) = \min_{\mathcal{T}_t \in \overline{\mathbf{A}}(\mathcal{T}_s)} \left\{\nabla_{\overline{\mathbf{F}}}(\mathcal{T}_s, \mathcal{T}_t)\right\}$, respectively. Furthermore, we can indirectly estimate the proximity of the tree $\mathcal{T}_s$ to an object through its *maximum saliency contrast* $\nabla^{+}_{\overline{\mathbf{O}}}(\mathcal{T}_s) = \max_{\mathcal{T}_t \in \overline{\mathbf{A}}(\mathcal{T}_s)} \left\{\nabla_{\overline{\mathbf{O}}}(\mathcal{T}_s, \mathcal{T}_t)\right\}$. Lastly, the *shortest adjacent distance* of $\mathcal{T}_s$, with respect to $\overline{\mathbf{c}}$, can be calculated by $\nabla^{-}_{\overline{\mathbf{c}}}(\mathcal{T}_s) = \min_{\mathcal{T}_t \in \overline{\mathbf{A}}(\mathcal{T}_s)} \{\nabla_{\overline{\mathbf{c}}}(\mathcal{T}_s, \mathcal{T}_t)\}$.

Therefore, given the previous definitions, the *relevance* $\mathbf{V}_*(s) \in \mathbb{R}_{\geq 0}$ of any seed $s \in \mathcal{S}$ can be computed through a *criterion* $\mathbf{v}_*(s) \in \mathbb{R}_{\geq 0}$ based on the characteristics of its tree $\mathcal{T}_s$. For instance, in IFT, trees grow by minimizing the path-costs to vertices, making the *size-based* criterion $\mathbf{v}_{\text{size}}(s) = |\mathcal{T}_s| \,/\, |\mathcal{V}|$ a straightforward option. However, considering that it may favor background seeds [20], contrast is vital in accurate relevance estimation. For improving delineation, one could exploit either the minimum contrast or maximum contrast as indicatives of object borders through a *size- and minimum-contrast-based* $\mathbf{v}_{\text{minsc}}(s) = \mathbf{v}_{\text{size}}(s)\nabla^{-}_{\overline{\mathbf{F}}}(\mathcal{T}_s)$ and a *size- and maximum-contrast-based* $\mathbf{v}_{\text{maxsc}}(s) = \mathbf{v}_{\text{size}}(s)\nabla^{+}_{\overline{\mathbf{F}}}(\mathcal{T}_s)$ criteria, respectively.

Still, any criterion $\mathbf{v}_*$ is subjected to the existence of an object of interest [19, 20]. For example, a seed that lies far from the object is intrinsically irrelevant for delineation. Consequently, the relevance computation of a seed $s \in \mathcal{S}$ can be then defined as $\mathbf{V}_*(s) = \mathbf{v}_*(s)\mathbf{p}_*(s)$, where $\mathbf{p}_*(s) \in [0, 1]$ denotes an object-based *penalization*. Whenever an object saliency map is absent, $\mathbf{p}_{\text{none}}(s) = 1$ is applied $\forall s \in \mathcal{S}$. For concentrating seeds within and nearby the objects, one may exploit the mean saliency $\overline{\mathbf{O}}(\mathcal{T}_s)$ and the maximum saliency gradient $\nabla_{\overline{\mathbf{O}}}(\mathcal{T}_s)$ of the tree $\mathcal{T}_s$, as considered in the *object* penalization $\mathbf{p}_{\text{obj}}(s) = \max\left\{\overline{\mathbf{O}}(\mathcal{T}_s), \nabla^{+}_{\overline{\mathbf{O}}}(\mathcal{T}_s)\right\}$.

4 Compact and Accurate Superpixel Generation

In this section, we present our proposals for producing compact and accurate superpixels in the SICLE framework. Thus, our contributions are related to the Superpixel Generation and Seed Removal steps (Figure 2) for creating more efficient and effective SICLE variants. In Section 4.1, we define two novel saliency-based arc-cost functions and compare them with previous approaches, saliency-based or not, for superpixel segmentation. Similarly, Section 4.2 presents one novel seed relevance criterion and three new seed penalizations, contrasting with those proposed in SICLE.

4.1 Arc-cost Estimation

The arc-cost function $\mathbf{w}_{\text{root}} = \|\mathbf{F}(\mathbf{R}(x)) - \mathbf{F}(y)\|_2$, considered in [19, 20], led to effective superpixel delineation whenever a saliency map is absent. Moreover, its performance is equivalent to a dynamic arc-cost function [39], whose results surpass state-of-the-art methods whenever such map is absent. Also, since it is based on immutable features (*i.e.*, the seed's feature vector), $\mathbf{w}_{\text{root}}$ exhibits more stability than the aforementioned function [20]. Finally, the use of $\mathbf{w}_{\text{root}}$ allows differential computation of the IFT [45] throughout the iterations. However, both functions are unable to improve their delineation for superior saliency maps.

A key idea of our proposals is to exploit a function $\mathbf{o}(x, y) \in \mathbb{R}_{\geq 0}$ that indicates, for a given vertex $y \in \mathcal{V}$, if it is likely to be concatenated to an optimal path $\rho_x \in \mathscr{P}$ ending at a vertex $v \in \mathcal{V}$, with respect to the object saliency $\mathbf{O}$. This likelihood value $\mathbf{o}(x, y)$ is weighted by a constant parameter $\alpha \in \mathbb{R}_{\geq 0}$ that intuitively indicates the degree of trust we have towards the *saliency variation* between the root of x and y. More precisely, we define such function as $\mathbf{o}(x, y) = \alpha \left\|\mathbf{O}(\mathbf{R}(x)) - \mathbf{O}(y)\right\|_1$. Given that, we propose a generalization of $\mathbf{w}_{\text{root}}$, as shown in Equation 2:

$$\mathbf{w}^{\alpha}_{\text{root}}(x, y) = \left(\mathbf{w}_{\text{root}}(x, y)\right)^{1+\mathbf{o}(x,y)} \tag{2}$$

Note that, by disregarding any saliency variation, thus $\mathbf{w}_{\text{root}}^{\alpha=0} = \mathbf{w}_{\text{root}}$, our proposal does not incorporate incorrect estimations and, therefore, maintains the effective delineation provided by $\mathbf{w}_{\text{root}}$ (*i.e.*, robustness). Conversely, by setting $\alpha > 0$, the arc costs that cross a saliency border increase significantly, promoting a higher adherence to it and preventing superpixel leaking.

However, even with $\mathbf{w}_{\text{root}}^{\alpha}$, such SICLE configuration is still unable to produce compact superpixels. For that, one may suggest using the $\mathbf{f}_{\text{sum}}$ and the $\mathbf{w}_{\text{sum}}$ (Equation 3) functions as proposed in [32]:

$$\mathbf{w}_{\text{sum}}(x,y) = \left(\iota \left\|\mathbf{F}(\mathbf{R}(x)) - \mathbf{F}(y)\right\|_2\right)^{\beta} + \left\|x - y\right\|_2 \tag{3}$$

in which $\iota \in \mathbb{R}_{\geq 0}$ and $\beta \in \mathbb{R}_{\geq 0}$ control the superpixels' irregularity and boundary adherence, respectively. Clearly, this function is unaware of any object information, and consequently, one could recur to the generalization proposed in [17, 18]. Nevertheless, although it presents high effectiveness in delineation, especially with few superpixels, incorrect saliency estimations may deteriorate its performance significantly [19].

Therefore, we aim to develop an arc-cost function that maximizes object delineation, but with compact superpixels, while minimizing errors derived from inaccurate saliency estimation. For that, we propose a novel generalization (Equation 4) in which the saliency information only impacts the irregularity factor:

$$\mathbf{w}_{\text{sum}}^{\alpha}(x,y) = \left(\left(\iota + \mathbf{o}(x,y)\right) \left\|\mathbf{F}(\mathbf{R}(x)) - \mathbf{F}(y)\right\|_2\right)^{\beta} + \left\|x - y\right\|_2 \tag{4}$$

As one can see, similarly with $\mathbf{w}_{\text{root}}^{\alpha}$, $\mathbf{w}_{\text{sum}}^{\alpha=0} = \mathbf{w}_{\text{sum}}$ neglects the saliency information while, for $\alpha > 0$, the irregularity factor changes substancially whenever the arc crosses the saliency border. That is, for the former, $\mathbf{w}_{\text{sum}}^{\alpha}$ promotes incorrect saliency robustness whereas, for the latter, promotes adherence to saliency borders and prevents leakings.

4.2 Seed Criteria and Penalization

When regularity is not desired, the criteria $\mathbf{v}_{\text{size}}$, $\mathbf{v}_{\text{maxsc}}$, and $\mathbf{v}_{\text{minsc}}$ promote accurate seed relevance estimation. On the other hand, they often lead to a higher superpixel concentration in different regions of the image (*e.g.*, background and near object borders), which may favor irregularity. Therefore, in this work, we introduce a novel criterion $\mathbf{v}_{\text{spread}}(s) = \mathbf{v}_{\text{size}}(s)\nabla_{\mathbf{c}}^{-}(\mathcal{T}_s)$ for *spreading* superpixels across the image (*i.e.*, favoring regularity) by exploiting the shortest adjacent distance $\nabla_{\mathbf{c}}^{-}(\mathcal{T}_s)$ of each tree $\mathcal{T}_s$.

In terms of seed penalization, $\mathbf{p}_{\text{obj}}$, as expected, promotes a high superpixel concentration within and near the objects. Still, as one may note, it favors internal superpixels which exert limited influence on delineation. Thus, when irregularity and delineation are of utmost importance, we propose favoring only trees $\mathcal{T}_s$ which are near a saliency border, increasing competition near the objects. More formally, the object *border* criterion is based on the tree's maximum saliency contrast such that $\mathbf{p}_{\text{bord}}(s) = \nabla_{\mathbf{O}}^{+}(\mathcal{T}_s)$.

Clearly, by focusing on delineation, the previous penalizations $\mathbf{p}_{\text{obj}}$ and $\mathbf{p}_{\text{bord}}$ favor irregularity (*e.g.*, near object borders), which compromises compactness when it is desired. Conversely, if saliency is provided, it is expected to favor the object delineation to the detriment of the others. Thus, it is essential to provide a penalization that minimizes the impacts on compactness whilst improving the object delineation.

One possible approach is to exploit the quantity of seeds in each region. That is, for those majorly in the object, a higher concentration and, thus, a higher competition can suppress eventual leakings, which produce irregularity. Conversely, for those mainly in the background, we argue in favor of spreading instead of selecting near the object. That is, we propose favoring those trees located within the *object*, indicated by their mean saliency, and *spreading* those in the *background*, with respect to their shortest adjacent distance. Simply put, such penalization is defined by $\mathbf{p}_{\text{osb}}(s) = \max\left\{\overline{\mathbf{O}}(\mathcal{T}_s), (1 - \overline{\mathbf{O}}(\mathcal{T}_s))\nabla_{\mathbf{c}}^{-}(\mathcal{T}_s)/d\right\}$, considering $d \in \mathbb{R}_{>0}$ as the maximum distance between any two vertices.

By the same assumption presented for $\mathbf{p}_{\text{bord}}$, we may infer that superpixels within the object, due to their size, also present limited influence on the overall compactness, such that prioritizing them may not favor delineation nor compactness. Therefore, we may focus on those near the object *borders* to improve delineation while, for those at

the *background*, *spreading* for promoting regularity and compactness. We argue that such selection tends to impact the compactness slightly since the major influence lies on the superpixels in the background. Finally, by using the tree's $\mathcal{T}_s$ maximum saliency contrast $\nabla^{+}_{\overline{\mathbf{O}}}(\mathcal{T}_s)$ instead of its mean saliency $\overline{\mathbf{O}}(\mathcal{T}_s)$, the new penalization differs from $\mathbf{p}_{\text{osb}}$ and can be formally defined as $\mathbf{p}_{\text{bobs}}(s) = \max\left\{\nabla^{+}_{\overline{\mathbf{O}}}(\mathcal{T}_s), (1-\overline{\mathbf{O}}(\mathcal{T}_s))\nabla^{-}_{\overline{\mathbf{c}}}(\mathcal{T}_s)/d\right\}$.

5 Experimental Results

In this section, we describe the experimental framework for evaluating SICLE with our proposals against state-of-the-art approaches. In Section 5.1, we present the experimental setup, including datasets, metrics, and baselines. Then, in Sections 5.2 and 5.3, we provide a quantitative and qualitative analysis whenever an object saliency map is absent or not, respectively. Finally, we discuss, analyze and exemplify some limitations of our proposals in Section 5.4.

5.1 Experimental Setup

In order to provide a wide-ranging analysis of the methods' performance in generating superpixels, we have selected five distinct medical and natural image datasets. First, we considered the *Berkeley Segmentation Dataset* (BSDS500) [46], a popular segmentation dataset containing 500 natural images. However, since it is contour-driven, such dataset is not applicable whenever a single object is desired. To cope with this drawback, we chose the well-known *Extended Complex Scene Saliency* (ECSSD) [47] dataset, which contains 1000 natural images with complex objects and backgrounds. Third, we selected the *Insects* [42] dataset (130 images), given the challenge of delineating the thin legs of various bugs. In terms of medical images, the *Liver* [32] dataset (40 images) imposes a major difficulty in segmenting CT slices of the human liver, given its monochromaticity and smooth borders. Finally, the fifth dataset, named *Parasites* [17] is composed of 72 images of human intestinal parasite eggs which, although colored, are often attached to impurities and also exhibit smooth borders. We randomly selected 30% for optimization and 70% for testing for each dataset, except for BSDS500, which already establishes such division. For producing the saliency maps, we opted for the U^2-Net estimator [43], fine-tuned using the same training set and its default parameters.

For measuring the performance of all methods, we selected popular evaluation metrics. First, let the two partitions $\mathscr{R} = \{\mathcal{R}_1, \ldots, \mathcal{R}_n\} \in \mathbb{P}(\mathcal{V})$ and $\mathscr{O} = \{\mathcal{O}_1, \ldots, \mathcal{O}_k\} \in \mathbb{P}(\mathcal{V})$ denote, respectively, the set of all superpixels and the set of all objects, considering $n, k \in \mathbb{N}_{\geq 1}$ and $\mathbb{P}$ to be the *power set*. The *Boundary Recall* (BR) [6] evaluates the ratio between object boundaries and superpixel borders and it is defined by Equation 5:

$$\mathsf{BR}(\mathscr{R}, \mathscr{O}) = \frac{|\mathbf{tp}^r(\mathscr{R}, \mathscr{O})|}{|\mathbf{tp}^r(\mathscr{R}, \mathscr{O})| + |\mathbf{fn}^r(\mathscr{R}, \mathscr{O})|} \tag{5}$$

in which $\mathbf{tp}^r(\mathscr{R}, \mathscr{O}) \in \mathbb{N}$ (resp. $\mathbf{fn}^r(\mathscr{R}, \mathscr{O}) \in \mathbb{N}$) computes the number of true positive (resp. false negatives) boundary pixels of the segmentation $\mathscr{R}$ with respect to $\mathscr{O}$, considering a tolerance radius of $r \in \mathbb{N}_{\geq 1}$ pixels. Similarly to [6], we set $r = 0.0025d\sqrt{2}$, being d the image's diagonal size. *Under-segmentation Error* (UE) [48] measures the error (Equation 6) from object overlap with either internal or external superpixel (*i.e.*"leaking"):

$$\mathsf{UE}(\mathscr{R}, \mathscr{O}) = \frac{1}{|\mathcal{V}|} \sum_{\mathcal{O}_j \in \mathscr{O}} \sum_{\mathcal{R}_i \in \mathscr{R}} \min\{|\mathcal{R}_i \cap \mathcal{O}_j|, |\mathcal{R}_i \setminus \mathcal{O}_j|\} \tag{6}$$

considering that $\mathcal{R}_i \cap \mathcal{O}_j \neq \emptyset$. *Compactness* (CO) [6] is also a famous metric, but it differs from both previous ones by evaluating whether the superpixels present a compact shape (being maximum when it is a circle). Thus, it is defined by Equation 7:

$$\mathsf{CO}(\mathscr{R}) = \frac{1}{|\mathcal{V}|} \sum_{\mathcal{R}_i \in \mathscr{R}} \frac{4\pi|\mathcal{R}_i|}{\wp(\mathcal{R})^2} \tag{7}$$

in which $\wp(\mathcal{R}) \in \mathbb{N}_{>0}$ denotes the perimeter of $\mathcal{R}_i$. Finally, we estimated the elapsed time of each method in a 64-bit Intel(R) Core(TM) i7-4790S PC with a CPU Speed of 3.20GHz.

In this work, our goal is to provide SICLE variants for generating effective delineation with irregular and compact superpixels (hereafter named SICLE-IRREG and SICLE-COMP, respectively). Moreover, differently from [20, 21], these novel versions of SICLE exploit *differential* computations of the IFT (DIFTs) [45] for improving speed. For

comparison, we selected the following state-of-the-art algorithms as baselines based on their reported performance: (i) ERS [31] [1]; (ii) SLIC [7] [2]; (iii) SH [9] [3]; (iv) IBIS [8] [4]; (v) LSC [24] [5]; (vi) GMM [28] [6]; (vii) OISF [17, 18] [7]; and (viii) DAL-HERS [36] [8]. For all of the aforementioned methods, we used their default parameter setting. Regarding CO, SLIC, IBIS, and GMM offer highly compact superpixels. Conversely, SH, LSC, OISF, and ERS present highly irregular ones but with accurate object delineation (*i.e.*, BR and UE). Finally, we included the recent DAL-HERS algorithm for comparison.

5.2 Non-object-based Analysis

We first optimized SICLE-IRREG and SICLE-COMP through grid-search under the situation of not having any object information (*i.e.*, $\alpha = 0$ and $\mathbf{p}_{\text{none}}$). For all variants, we optimized considering $N_0 \in \{2000, 3000, 5000, 8000\}$, $\Omega \in \{2, 3, 5, 7, 10\}$, $\mathbf{f}_* = \{\mathbf{f}_{\max}, \mathbf{f}_{\text{sum}}\}$, $\mathbf{w}_* = \{\mathbf{w}_{\text{root}}, \mathbf{w}_{\text{sum}}\}$, $\iota \in \{0.05, 0.10, 0.12, 0.15, 0.20, 0.25, 0.50, 0.75, 1.00\}$, and $\mathbf{v}_* \in \{\mathbf{v}_{\text{size}}, \mathbf{v}_{\text{minsc}}, \mathbf{v}_{\text{maxsc}}, \mathbf{v}_{\text{spread}}\}$. For SICLE-IRREG, we found the following values: (i) $N_0 = 3000$; (ii) $\Omega = 5$; (iii) $\mathbf{f}_* = \mathbf{f}_{\max}$ and $\mathbf{w}_* = \mathbf{w}^{\alpha}_{\text{root}}$; and (iv) $\mathbf{v}_* = \mathbf{v}_{\text{minsc}}$. These values converge to those found in [10, 19, 20]. For the novel compact variant SICLE-COMP, we fixed $\beta = 12$ as suggested in [32], and we obtained the following optimized configuration: (i) $N_0 = 3000$; (ii) $\Omega = 7$; (iii) $\mathbf{f}_* = \mathbf{f}_{\text{sum}}$ and $\mathbf{w}_* = \mathbf{w}^{\alpha}_{\text{sum}}$; (iv) $\iota = 0.12$; and (v) $\mathbf{v}_* = \mathbf{v}_{\text{maxsc}}$. The optimization of ι prioritized delineation, crucial in segmentation, at the expense of on-par compactness performance. From Ω, we can infer that compactness is sensible to the quantity of seeds removed *per* iteration, requiring slightly more than SICLE-IRREG. Also, from $\mathbf{v}_{\text{maxsc}}$, we may assume that seeds positioned near high contrast boundaries favor the generation of compact superpixels without compromising delineation. We believe that, due to the proximity, such border is assured at first by competition while possible leakings, often in low contrast regions, are prevented by the spatial limitation imposed by $\mathbf{w}_{\text{sum}}$.

The quantitative results of all methods are shown in Figure 3. It is possible to see that SICLE-IRREG achieves superior delineation in all datasets for BR, except for Insects. It is important to notice that, although ERS excels on such dataset, its performance is fair for both medical image datasets. For BSDS500 and Parasites, DAL-HERS achieves on-par performance with our proposal but presents a poor delineation for Liver and Insects, which shares similarities with BSDS500. In fact, apart from SICLE-IRREG and SH, the remaining irregular methods do not present a consistent performance throughout the baselines, indicating a tendency to be best suited for specific conditions rather than a general solution for superpixel segmentation. Regarding UE, we can see that SICLE is continuously on pair with the best methods on each dataset.

We can also notice such effective performance and stability in SICLE-COMP for all metrics considered. For BR and UE, our proposal achieves on-par performance with GMM and LSC in BSDS500 and Insects while surpassing them in Liver and Parasites by a significant margin. For CO, we can see that SICLE-COMP consistently achieves a similar absolute performance irrespective of the dataset. Under the same perspective, IBIS consistently produces the most compact superpixels in all datasets considered. On the other hand, similarly to the irregular baselines, the remaining methods, such as SLIC, GMM, and LSC, are unable to exhibit the same behavior seen on BSDS500 and Insects, considering all metrics on Liver and Parasites.

Regarding speed, Table 1 compares the performance of the fastest methods in the BSDS500 dataset. It is important to notice that, although SICLE is $O(|\mathcal{V}| \log |\mathcal{V}|)$, bounded by the IFT, and the other referred methods are $O(|\mathcal{V}|)$, our proposal executes fewer iterations as N_f increases. As a result, it becomes faster for usual superpixel quantities, whereas SLIC and LSC, for instance, perform a strict number of 10 iterations independently of N_f. Also, in a single execution, SICLE generates the exact number of scales in a multi-scale segmentation, while SH may unnecessarily

[1] https://github.com/mingyuliutw/EntropyRateSuperpixel
[2] https://www.epfl.ch/labs/ivrl/research/slic-superpixels/
[3] https://github.com/semiquark1/boruvka-superpixel
[4] https://github.com/xapha/IBIS
[5] https://jschenthu.weebly.com/projects.html
[6] https://github.com/ahban/GMMSP-superpixel
[7] https://github.com/LIDS-UNICAMP/OISF
[8] https://github.com/hankuipeng/DAL-HERS

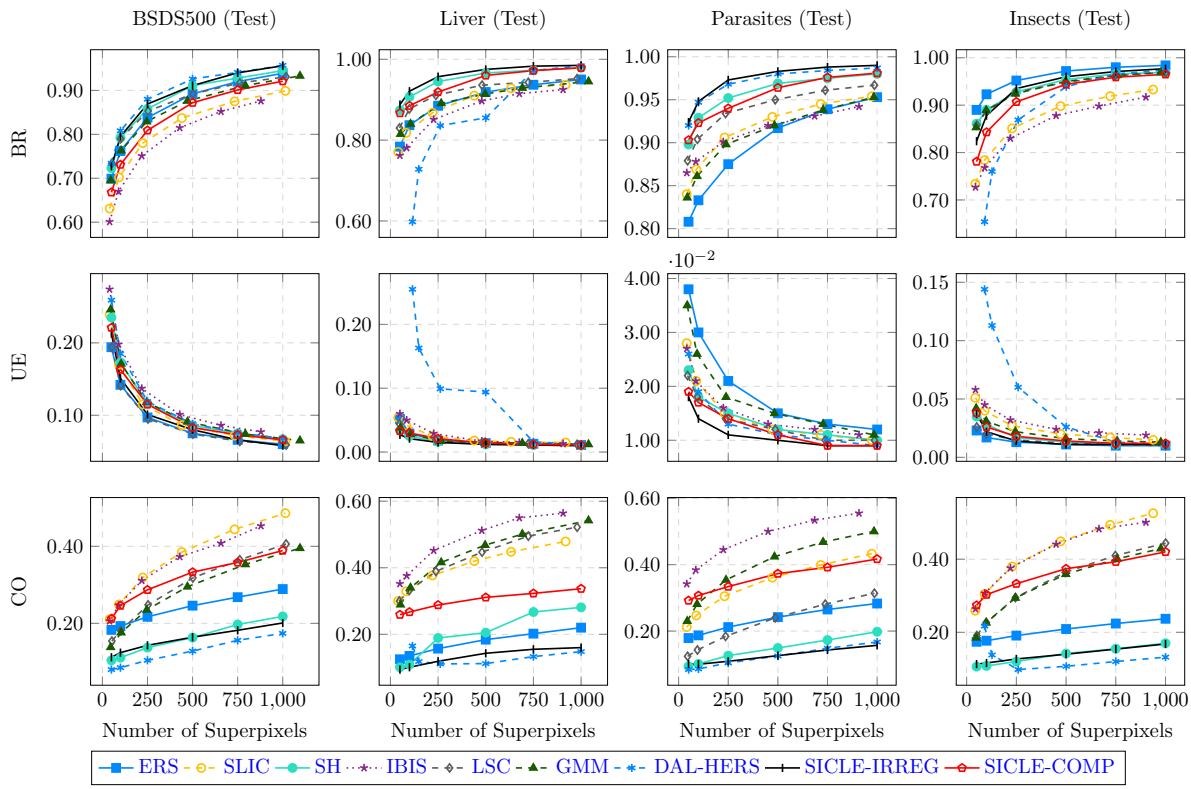

Fig. 3 Quantitative results on the test set of each dataset when no object information is provided.

N_f	SLIC	SH	IBIS	LSC	SICLE-IRREG	SICLE-COMP
50	**0.610±0.027**	0.856±0.026	**0.762±0.027**	**0.774±0.026**	0.808±0.065	1.031±0.047
250	**0.612±0.027**	0.857±0.028	0.753±0.025	0.789±0.027	**0.678±0.030**	**0.746±0.035**
500	**0.615±0.027**	0.853±0.026	0.752±0.025	0.787±0.029	**0.567±0.024**	**0.718±0.030**
1000	**0.620±0.028**	0.857±0.026	0.748±0.025	0.803±0.030	**0.538±0.021**	**0.566±0.023**

Table 1 Speed performance, in seconds, of the fastest methods on the BSDS500 dataset when no object information is provided. The values depicted in bold, blue, and red indicate, respectively, the first, second, and third faster performances for each N_f.

exceed such quantity by computing the whole hierarchy [20].

From Figure 4, we can perceive the effective performance of the novel SICLE variants by their delineation of the parasite egg. Compared with irregular methods, SICLE-IRREG best approximates the object borders and presents minimal leaking errors. Although SH also manages to present accurate object delineation, it still presents leakings around the object, like the segmentations from ERS and LSC. For those approaches which aim for compactness, it is possible to see that the result from SICLE-COMP achieves an accurate delineation of the object with compact superpixels, while the remaining ones, although generated highly compact and regular superpixels, exhibit major segmentation errors.

5.3 Object-based Analysis

In this section, we further optimized the configuration found in the previous section for considering prior object information. First, we analyzed the impact of the saliency quality only into delineation (*i.e.*, $\mathbf{p}_* = \mathbf{p}_{\text{none}}$) by considering three different cases based on their indirect object delineation: (i) poor-quality, represented by the object's minimum bounding box (BB); (ii)

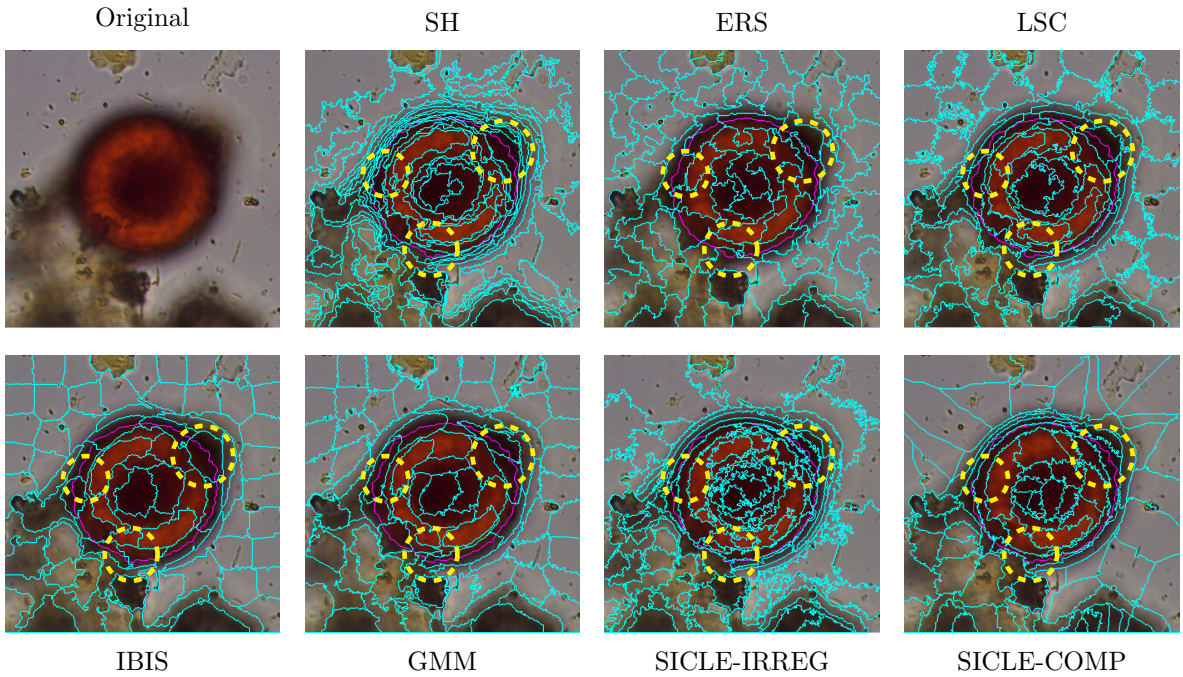

Fig. 4 Qualitative results with an image from Parasites requiring 100 superpixels. The object boundaries and the superpixel borders are depicted in magenta and cyan, respectively. The images were cropped for visualization purposes. Yellow circles indicate regions where SICLE best approximates the object borders in contrast with the baselines.

Fig. 5 Quantitative results on the train set of each dataset using the ground-truth (GT), U^2-Net map [43] (U^2) and the object's minimum bounding box (BB) as saliency maps.

state-of-the-art [43] (U^2); and (iii) ideal, represented by the object's ground-truth (GT). Concurrently, we optimized the saliency importance $\alpha \in \{0.0, 0.5, 1.0, 1.5, 2.0, 2.5, 3.0\}$ with respect to each case. We emphasize that the GT is only used in this experiment. Coincidently, the best values of α for each case, for both SICLE-IRREG and SICLE-COMP, were 0, 2, and 3, respectively, and they are represented in Figure 5. As expected, we note that the trustiness of the information correlates with its quality. For BB, $\alpha = 0$ avoids considering saliency inaccuracies and maintains an effective delineation performance. On the other hand, by increasing α, our method can significantly improve its performance for better quality estimations, as seen for U^2 and GT. Moreover, we found that such improvement does not impair the compactness achieved by SICLE-COMP. Thus, our approach can maximize compactness and delineation, whereas other approaches only offer a compromise. It is worth noticing the discrepancy between the performances when considering U^2 and GT, indicating that, although accurate, the state-of-the-art estimation is not ideal.

Subsequently, we evaluated the best penalization $\mathbf{p}_* \in \{\mathbf{p}_{\text{none}}, \mathbf{p}_{\text{obj}}, \mathbf{p}_{\text{osb}}, \mathbf{p}_{\text{bobs}}\}$ for each variant. As expected, $\mathbf{p}_{\text{bord}}$ is more suited for SICLE-IRREG by increasing the number of seeds around the object, thus increasing competition in crucial regions. But, for SICLE-COMP, $\mathbf{p}_{\text{bobs}}$ presented the best results without compromising compactness, indicating the importance of spreading background superpixels to suppress the irregularity within the object. Finally, in the last experiment, we verified whether such final relevant seed estimation is accurate by varying values for $\Omega \in \{2, 3, 5, 7, 10\}$. That is, the more accurate it is, the fewer iterations are required to achieve effective delineation (*i.e.*, ideally two). For both SICLE-IRREG and SICLE-COMP, $\Omega = 2$ suffice for achieving top performance with no significant variation on BR, UE, and CO.

Figure 6 shows the quantitative results of all methods. It is possible to see that, contrasting

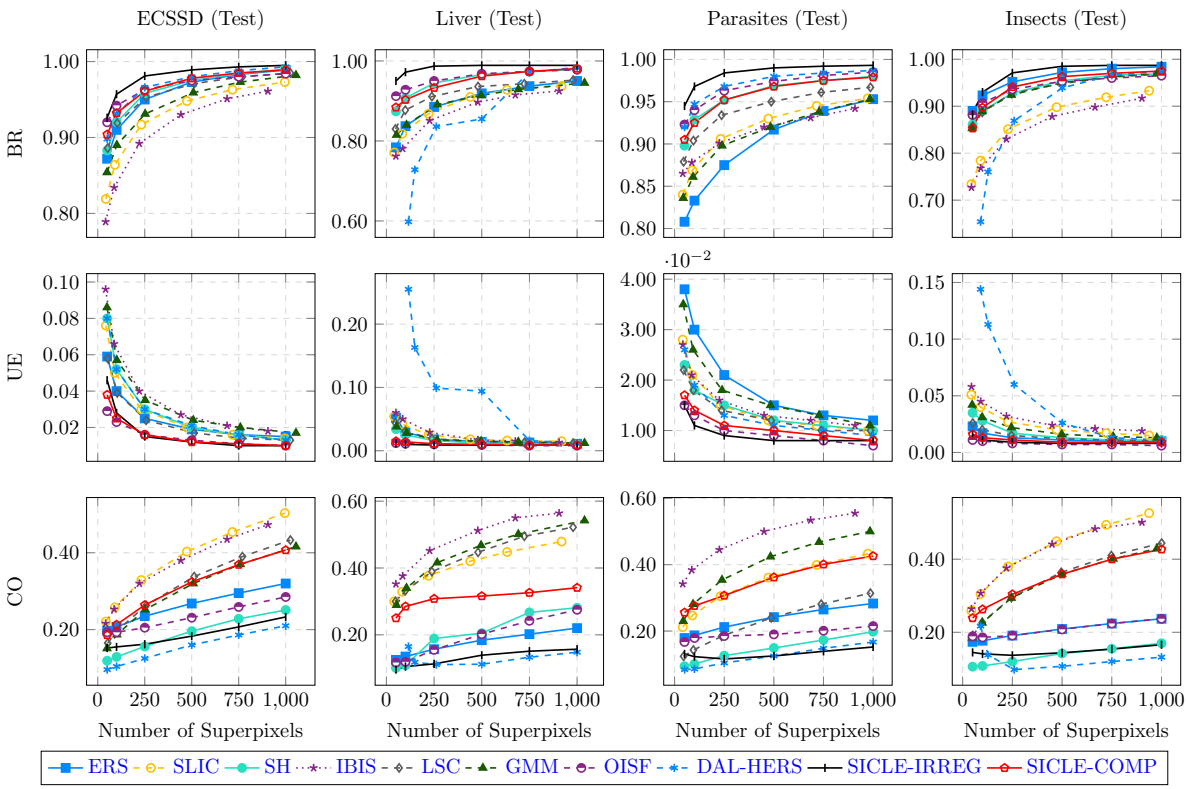

Fig. 6 Quantitative results on the test set of each dataset using the U^2-Net map [43].

with Figure 3, SICLE variants improved their performance significantly, up to surpassing ERS in Insects. Although the saliency quality plays an essential role in such improvement, we argue that our proposals intelligently incorporate such information. For instance, OISF, which uses the same information, is unable to achieve similar performance in all datasets, especially in Insects. Similarly to SICLE-IRREG being the best amongst all baselines in BR and UE, SICLE-COMP presents the best performance in both metrics for all compact methods, being on par with state-of-the-art methods in delineation, such as SH. Finally, by comparing with the non-object-based scenario, this upgrade did not deteriorate SICLE's performance in CO, as previously stated. We further argue that our object-based approaches offer a more suitable methodology for object incorporation, especially when compared to the performance instability of DAL-HERS in both BR and UE, for producing either irregular or compact superpixels.

The speed performance of the fastest methods in ECSSD is presented in Table 2. It is important to state that the performance of SICLE-IRREG differs from the results shown in [21]. We argue that the speedup achieved from exploiting DIFTs is limited when a significantly high quantity of seeds is removed between iterations, which is a direct result when $\Omega = 2$. Intuitively, removing superpixels can generate inconsistencies during the IFT execution, which are absent in the sequential version and must be handled in the differential one. Thus, the higher the number to be removed, the higher the quantity of inconsistencies to be solved, impacting the overall performance. We may note such behavior on the performance discrepancy for $N_f = 1000$ and $N_f = 500$. Still, differently from [21], our implementation benefits from multiscale segmentation. As the number of scales $k \in \mathbb{N}_{\geq 1} \leq \Omega$ increase, more iterations are required and, consequently, $\mathbf{M}$ estimates higher quantities of relevant seeds to be maintained *per* iteration. Therefore, by removing fewer superpixels at each iteration, fewer inconsistencies

N_f	SLIC	SH	IBIS	LSC	SICLE-IRREG	SICLE-COMP
50	**0.576±0.029**	0.813±0.028	0.710±0.027	0.702±0.036	**0.501±0.055**	**0.514±0.046**
250	**0.581±0.027**	0.816±0.028	0.700±0.028	0.713±0.036	**0.503±0.040**	**0.511±0.043**
500	**0.583±0.028**	0.813±0.028	0.694±0.026	0.714±0.039	**0.507±0.039**	**0.517±0.042**
1000	**0.583±0.028**	0.812±0.027	0.694±0.028	0.718±0.036	**0.420±0.090**	**0.482±0.060**

Table 2 Speed performance, in seconds, of the fastest methods on ECSSD dataset using the U^2-Net map [43]. The values depicted in bold, blue, and red indicate, respectively, the first, second, and third faster performances for each N_f.

are generated and need to be handled by DIFT. Lastly, when comparing with the baselines, both SICLE-IRREG and SICLE-COMP still exhibit the fastest segmentation performances irrespective of N_f.

As an example of how SICLE variants profit from our proposals and use object information appropriately, we may note, in Figure 7, that OISF replicates the errors of the saliency map in its segmentation, whereas our proposals overcame them and provided an accurate object delineation. Compared to non-object-based baselines, the proper incorporation of object information assisted SICLE-IRREG and SICLE-COMP in avoiding severe leaks. Finally, it is interesting to notice that SICLE-COMP favored the superpixels within the object less than SICLE-IRREG but more than GMM and LSC, as represented by the larger background superpixels.

5.4 Limitations

Although the performance of SICLE was improved after considering our proposed modifications, it still presents several limitations. From Figure 8, one can see that, even though the saliency map partially detects the insect's thin parts (*e.g.*, antenna and legs), SICLE cannot, at least, delineate the same regions with similar accuracy. We argue that this behavior is a drawback of seed-based methods since far away seeds have more difficulty to offer strongly connected paths to pixels in thin parts of the object, even with help of a saliency map. That is, for small quantities of superpixels, seeds tend to be farther, and consequently, it is more challenging to conquer the pixels associated with its classification (*i.e.*, object or background seed). In such case, one may propose a more meticulous path-cost or arc-cost function for improving the delineation of thin object parts or, conversely, conceive different seed strategies (*i.e.*, seed sampling or removal) for increasing the seed resolution in such regions and, consequently, increasing competition in such crucial region. Either way, knowing the peculiarities and particularities of the object is utterly vital for solving this difficulty.

6 Multiobject Handling

In this section, we will debate briefly over the challenges of segmenting multiple objects of interest in an image using SICLE. First, let $\mathscr{O} = \{\mathcal{O}_1, \ldots, \mathcal{O}_k\} \in \mathbb{P}(\mathcal{I})$ as the set of all $k \in \mathbb{N}_{>0}$ objects within the image I and, equivalently, in the digraph G. As one may see, $k = 1$ consists of the usual object saliency task.

Difficulties arise when $k > 1$, especially on manipulating different object informations in SICLE. A straightforward approach could be incorporating multiple saliencies, one dimension for each object, such that $\mathbf{O}(v) = \langle \mathbf{O}_1(v), \ldots, \mathbf{O}_k(v) \rangle \in [0,1]^k$ given $v \in \mathcal{V}$. However, we argue that storing k saliencies is memory inefficient since background vertices are redundantly marked as such $\forall \mathbf{O}_j$, given $1 \leq j \leq k$. Instead, we claim that, for SICLE, a single saliency $\mathbf{O}(v) = \max_{1 \leq j \leq k} \{\mathbf{O}_j(v)\}$, for $v \in \mathcal{V}$, is sufficient for handling multiple objects. Aside being more memory efficient, conceptually speaking, our proposed saliency still depicts the probable likelihood of a vertex belonging to an object.

For such case, and without loss of generality and abuse of notation, we analyze three situations, as exemplified in Figure 9, considering two distinct objects $\mathcal{O}_x, \mathcal{O}_y \in \mathscr{O}$ and ideal object estimation: (i) $\mathcal{O}_y \notin \overline{\mathbf{A}}(\mathcal{O}_x)$; (ii) $\mathcal{O}_y \in \overline{\mathbf{A}}(\mathcal{O}_x)$; and (iii) $\mathcal{O}_x \cap \mathcal{O}_y \neq \varnothing$. Thus, Figure 9(a) depicts (i), while Figure 9(c) relates to (iii). Both Figures 9(b,d) are associated with (ii) in which, for the former, the distinction of both triangles of equal shape is facilitated by the colors, and highly difficulted for the latter due to the color gradient. Intuitively, (i) presents the case of two non-adjacent objects and, consequently, every path $\rho_{u \rightsquigarrow v}$ in which $u \in$

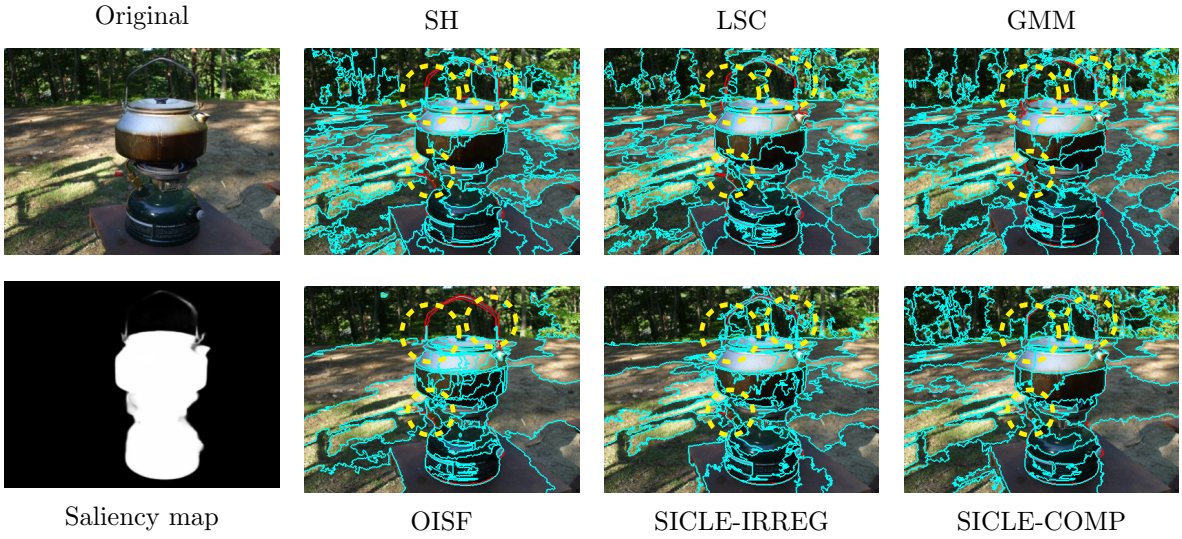

Fig. 7 Qualitative results with an image from ECSSD requiring 100 superpixels and using the U^2-Net estimator [43]. The object boundaries and the superpixel borders are depicted in red and cyan, respectively. Yellow circles indicate regions where SICLE best approximates the object borders in contrast with the baselines.

$\mathcal{O}_x$ and $v \in \mathcal{O}_y$ is ensured to cross a saliency border, being properly penalized by $\mathbf{o}$. However, for (ii) and (iii), $\rho_{u \rightsquigarrow v}$ can be composed of non-background vertices of distinct objects, resulting in no penalization. Nevertheless, we can verify that $\mathbf{w}^{\alpha}_{\text{root}}$ and $\mathbf{w}^{\alpha}_{\text{sum}}$ tend to estimate a distinction between v, u, being $v \in \mathcal{O}_x$ and $u \in \mathcal{O}_y$, even for $\mathbf{o}(v, u) = 0$, if they exhibit sufficient color divergence. Finally, for case (iii), we argue that it is an ill-posed problem due to the impossibility to define and, thus, to detect the overlapped object's boundaries, either in $\mathbf{F}$ or $\mathbf{O}$, in a bi-dimensional representation.

7 Conclusion and Future Work

This work proposes novel arc-cost functions, a new seed relevance criterion, and three new seed relevance penalizations for improving the effectiveness and efficiency of the state-of-the-art superpixel segmentation method *Superpixels through Iterative CLEarcutting* (SICLE). Due to our proposals, SICLE can now compute compact superpixels with high adherence to object boundaries and can exploit the accurate saliency information during delineation. Moreover, our proposals maintain the properties of previous SICLE variants, such as saliency inaccuracy robustness and saliency-based multiscale segmentation. Results show that, with our modifications, SICLE produces accurate object delineation with irregular and compact superpixels whenever a saliency map is absent, achieving a faster performance than several state-of-the-art methods analyzed in this work. When object information is provided, the novel proposals led SICLE variants to achieve superior performance in delineation and speed for distinct objects of interest, requiring only two iterations for segmentation. Finally, we propose a brief discussion on the challenges and a possible solution for multiobject handling in SICLE. For future work, we intend to extend SICLE for video supervoxel segmentation and evaluate its applicability as an interactive segmentation tool.

Acknowledgements

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 – CAPES – Finance code 001 (COFECUB

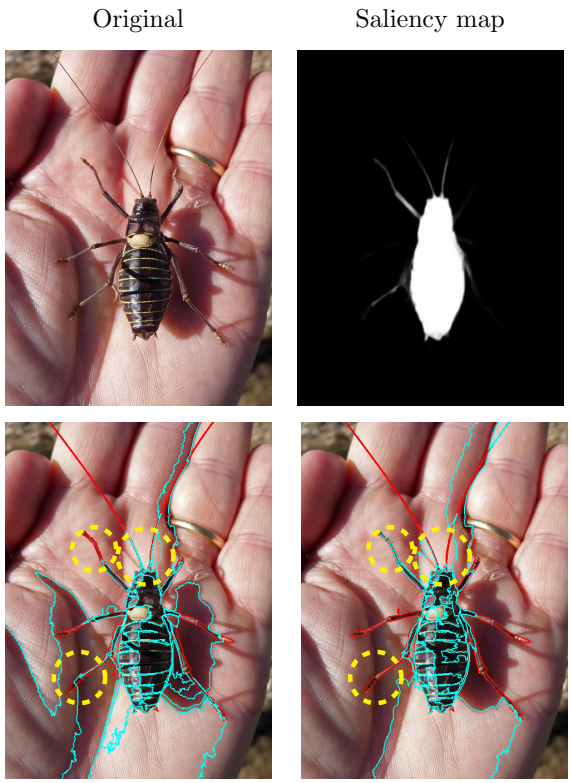

Fig. 8 Examples of the limitations of our approaches, even when assisted by a saliency map from the U^2-Net estimator [43], requiring 50 superpixels. The object boundaries and the superpixel borders are depicted in red and cyan, respectively. Yellow circles indicate regions where SICLE could not properly approximate to the object borders.

88887.191730/2018-00, 88887.712530/2022-00) for the financial support.

References

[1] Dhore, S., Abin, D.: Chest x-ray segmentation using watershed and super pixel segmentation technique. In: International Conference on Communication Information and Computing Technology (ICCICT), pp. 1–4 (2021). https://doi.org/10.1109/ICCICT50803.2021.9510078

[2] Liu, C., Zhao, R., Pang, M.: A fully automatic segmentation algorithm for ct lung images based on random forest. Medical Physics **47**(2), 518–529 (2019). https://doi.org/10.1002/mp.13939

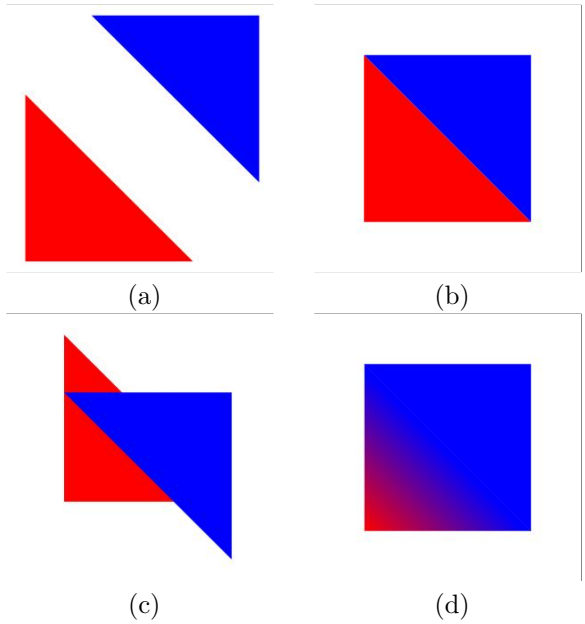

Fig. 9 Graphical examples of possible situations for multiobject handling, in which the objects are: (a) separated; (b,d) adjacents; or (c) overlapping.

[3] Zhou, J., Ruan, J., Wu, C., Ye, G., Zhu, Z., Yue, J., Zhang, Y.: Superpixel segmentation of breast cancer pathology images based on features extracted from the autoencoder. In: IEEE 11th International Conference on Communication Software and Networks (ICCSN), pp. 366–370 (2019). https://doi.org/10.1109/ICCSN.2019.8905358

[4] Yi, S., Ma, H., Wang, X., Hu, T., Li, X., Wang, Y.: Weakly-supervised semantic segmentation with superpixel guided local and global consistency. Pattern Recognition **124**, 108504 (2022). https://doi.org/10.1016/j.patcog.2021.108504

[5] Liang, Y., Zhang, Y., Wu, Y., Tu, S., Liu, C.: Robust video object segmentation via propagating seams and matching superpixels. IEEE Access **8**, 53766–53776 (2020). https://doi.org/10.1109/ACCESS.2020.2981140

[6] Stutz, D., Hermans, A., Leibe, B.: Superpixels: An evaluation of the state-of-the-art. Computer Vision and Image Understanding **166**, 1–27 (2018). https://doi.org/10.1016/j.cviu.2017.03.007

[7] 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). https://doi.org/10.1109/TPAMI.2012.120

[8] 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**, 77250–77263 (2021). https://doi.org/10.1109/ACCESS.2021.3081919

[9] Wei, X., Yang, Q., Gong, Y., Ahuja, N., Yang, M.: Superpixel hierarchy. IEEE Transactions on Image Processing **27**(10), 4838–4849 (2018). https://doi.org/10.1109/TIP.2018.2836300

[10] Belém, F., Guimarães, S., Falcão, A.: Superpixel segmentation using dynamic and iterative spanning forest. Signal Processing Letters **27**, 1440–1444 (2020). https://doi.org/10.1109/LSP.2020.3015433

[11] Schick, A., Fischer, M., Stiefelhagen, R.: An evaluation of the compactness of superpixels. Pattern Recognition Letters **43**, 71–80 (2014). https://doi.org/10.1016/j.patrec.2013.09.013

[12] 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: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1225–1234 (2021). https://doi.org/10.1109/CVPR46437.2021.00128

[13] Yang, F., Sun, Q., Jin, H., Zhou, Z.: Superpixel segmentation with fully convolutional networks. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 13961–13970 (2020). https://doi.org/10.1109/CVPR42600.2020.01398

[14] Suzuki, T.: Superpixel segmentation via convolutional neural networks with regularized information maximization. In: IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 2573–2577 (2020). https://doi.org/10.1109/ICASSP40776.2020.9054140

[15] Yu, Y., Yang, Y., Liu, K.: Edge-aware superpixel segmentation with unsupervised convolutional neural networks. In: IEEE International Conference on Image Processing (ICIP), pp. 1504–1508 (2021). https://doi.org/10.1109/ICIP42928.2021.9506289

[16] Xu, L., Zeng, L., Wang, Z.: Saliency-based superpixels. Signal, Image and Video Processing **8**(1), 181–190 (2014). https://doi.org/10.1007/s11760-013-0520-8

[17] Belém, F., Guimarães, S., Falcão, A.: Superpixel segmentation by object-based iterative spanning forest. In: Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications, vol. 11401, pp. 334–341 (2019). https://doi.org/10.1007/978-3-030-13469-3_39

[18] Belém, F., Guimarães, S., Falcão, A.: Superpixel generation by the iterative spanning forest using object information. In: 33rd Conference on Graphics, Patterns and Images (SIBGRAPI), pp. 22–28 (2020). https://doi.org/10.5753/sibgrapi.est.2020.12979. Workshop of Thesis and Dissertations

[19] Belém, F., Cousty, J., Perret, B., Guimarães, S., Falcão, A.: Towards a simple and efficient object-based superpixel delineation framework. In: 34th Conference on Graphics, Patterns and Images (SIBGRAPI), pp. 346–353 (2021). https://doi.org/10.1109/SIBGRAPI54419.2021.00054

[20] Belém, F., Perret, B., Cousty, J., Guimarães, S., Falcão, A.: Efficient Multiscale Object-based Superpixel Framework (2022). https://doi.org/10.48550/ARXIV.2204.03533

[21] Belém, F., Borlido, I., João, L., Perret, B., Cousty, J., Guimarães, S., Falcão, A.: Fast and effective superpixel segmentation using accurate saliency estimation. In: Discrete Geometry and Mathematical Morphology, pp. 261–273 (2022). https://doi.org/10.1007/978-3-031-19897-7_21

[22] Wan, L., Xu, X., Zhao, Q., Feng, W.: Spherical superpixels: Benchmark and evaluation. In: Asian Conference on Computer Vision (ACCV), vol. 11366, pp. 703–717 (2019). https://doi.org/10.1007/978-3-030-20876-9_44

[23] Wang, M., Liu, X., Gao, Y., Ma, X., Soomro, N.: Superpixel segmentation: A benchmark. Signal Processing: Image Communication **56**, 28–39 (2017). https://doi.org/10.1016/j.image.2017.04.007

[24] Li, Z., Chen, J.: Superpixel segmentation using linear spectral clustering. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1356–1363 (2015). https://doi.org/10.1109/CVPR.2015.7298741

[25] Liu, Y., Yu, M., Li, B., He, Y.: Intrinsic manifold SLIC: A simple and efficient method for computing content-sensitive superpixels. IEEE Transactions on Pattern Analysis and Machine Intelligence **40**(3), 653–666 (2018). https://doi.org/10.1109/TPAMI.2017.2686857

[26] Xiao, X., Zhou, Y., Gong, Y.: Content-adaptive superpixel segmentation. IEEE Transactions on Image Processing **27**(6), 2883–2896 (2018). https://doi.org/10.1109/TIP.2018.2810541

[27] Wu, J., Liu, C., Li, B.: Texture-aware and structure-preserving superpixel segmentation. Computers and Graphics **94**, 152–163 (2021). https://doi.org/10.1016/j.cag.2020.12.002

[28] Ban, Z., Liu, J., Cao, L.: Superpixel segmentation using gaussian mixture model. IEEE Transactions on Image Processing **27**(8), 4105–4117 (2018). https://doi.org/10.1109/TIP.2018.2836306

[29] 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). https://doi.org/10.1109/TIP.2016.2616302

[30] Machairas, V., Faessel, M., Cárdenas-Peña, D., Chabardes, T., Walter, T., Decenciere, E.: Waterpixels. IEEE Transactions on Image Processing **24**(11), 3707–3716 (2015). https://doi.org/10.1109/TIP.2015.2451011

[31] Liu, M., Tuzel, O., Ramalingam, S., Chellappa, R.: Entropy rate superpixel segmentation. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 2097–2104 (2011). https://doi.org/10.1109/CVPR.2011.5995323

[32] Vargas-Muñoz, J., Chowdhury, A., Alexandre, E., Galvão, F., Miranda, P., Falcão, A.: An iterative spanning forest framework for superpixel segmentation. IEEE Transactions on Image Processing **28**(7), 3477–3489 (2019). https://doi.org/10.1109/TIP.2019.2897941

[33] Galvão, F., Falcão, A., Chowdhury, A.: RISF: recursive iterative spanning forest for superpixel segmentation. In: 31st Conference on Graphics, Patterns and Images (SIBGRAPI), pp. 408–415 (2018). https://doi.org/10.1109/SIBGRAPI.2018.00059

[34] Awaisu, M., Li, L., Peng, J., Zhang, J.: Fast superpixel segmentation with deep features. In: Advances in Computer Graphics, vol. 11542, pp. 410–416 (2019). https://doi.org/10.1007/978-3-030-22514-8_38

[35] Tu, W., Liu, M., Jampani, V., Sun, D., Chien, S., Yang, M., Kautz, J.: Learning superpixels with segmentation-aware affinity loss. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 568–576 (2018). https://doi.org/10.1109/CVPR.2018.00066

[36] Peng, H., Aviles-Rivero, A., Schonlieb, C.: Hers superpixels: Deep affinity learning for hierarchical entropy rate segmentation. IEEE/CVF Winter Conference on Applications of Computer Vision (WACV), 72–81 (2022). https://doi.org/10.1109/WACV51458.2022.00015

[37] Jampani, V., Sun, D., Liu, M., Yang, M., Kautz, J.: Superpixel sampling networks.

In: European Conference on Computer Vision (ECCV), vol. 11211, pp. 363–380 (2018). https://doi.org/10.1007/978-3-030-01234-2_22

[38] Falcão, A., Stolfi, J., Lotufo, R.: The image foresting transform: Theory, algorithms, and applications. IEEE Transactions on Pattern Analysis and Machine Intelligence **26**(1), 19–29 (2004). https://doi.org/10.1109/TPAMI.2004.1261076

[39] Bragantini, J., Martins, S., Castelo-Fernandez, C., Falcão, A.: Graph-based image segmentation using dynamic trees. In: Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications, pp. 470–478 (2018). https://doi.org/10.1007/978-3-030-13469-3_55

[40] Borlido, I., Belém, F., Miranda, P., Falcão, A., Patrocínio, Z., Guimarães, S.: Towards interactive image segmentation by dynamic and iterative spanning forest. In: Discrete Geometry and Mathematical Morphology, pp. 351–364 (2021). https://doi.org/10.1007/978-3-030-76657-3_25

[41] Ciesielski, K., Falcão, A., Miranda, P.: Path-value functions for which dijkstra's algorithm returns optimal mapping. Journal of Mathematical Imaging and Vision **60**(7), 1025–1036 (2018). https://doi.org/10.1007/s10851-018-0793-1

[42] Mansilla, L., Miranda, P.: Oriented image foresting transform segmentation: Connectivity constraints with adjustable width. In: 29th Conference on Graphics, Patterns and Images (SIBGRAPI), pp. 289–296 (2016). https://doi.org/10.1109/SIBGRAPI.2016.047

[43] Qin, X., Zhang, Z., Huang, C., Dehghan, M., Zaiane, O., Jagersand, M.: U2-net: Going deeper with nested u-structure for salient object detection. Pattern Recognition **106**, 107404 (2020). https://doi.org/10.1016/j.patcog.2020.107404

[44] Belém, F., Melo, L., Guimarães, S., Falcão, A.: The importance of object-based seed sampling for superpixel segmentation. In: 32nd Conference on Graphics, Patterns and Images (SIBGRAPI), pp. 108–115 (2019). https://doi.org/10.1109/SIBGRAPI.2019.00023

[45] Falcão, A., Bergo, F.: Interactive volume segmentation with differential image foresting transforms. IEEE Transactions on Medical Imaging **23**(9), 1100–1108 (2004). https://doi.org/10.1109/TMI.2004.829335

[46] Arbelaez, P., Maire, M., Fowlkes, C., Malik, J.: Contour detection and hierarchical image segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence **33**(5), 898–916 (2011). https://doi.org/10.1109/TPAMI.2010.161

[47] 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). https://doi.org/10.1109/TPAMI.2015.2465960

[48] Neubert, P., Protzel, P.: Superpixel benchmark and comparison. In: Forum Bildverarbeitung, vol. 6, pp. 1–12 (2012)