Fast Graph Algorithms for Superpixel Segmentation

Dimitris Floros* Tiancheng Liu† Nikos Pitsianis*† Xiaobai Sun†

*Department of Electrical and Computer Engineering
Aristotle University of Thessaloniki
Thessaloniki 54124, Greece

†Department of Computer Science
Duke University
Durham, NC 27708, USA

Fig. 1: A superpixel segmentation of an aircraft image by algorithm SLAM. LEFT: the aircraft image with ID 10081 in the Berkeley image segmentation benchmark suite BSDS-500 [19]. The color image has 154,401 ($= 481 \times 321$) pixels. MIDDLE: A superpixel segmentation overlayed on the original color image. The segmentation, rendered by SLAM, partitions the image into 256 superpixels, with the compression ratio 603, i.e., there are 603 pixels per superpixel on average. RIGHT: An image reconstructed from the compressed superpixel representation of the image. With respect to a prescribed compression ratio, the closer the reconstructed image is to the original, the better the superpixel segmentation is. REMARKS: In comparison to other state-of-the-art algorithms for superpixel segmentation, within the same time frame, SLAM is better at preserving diverse, distinctive local information and global composition structure. Specific to the aircraft image, the serial number 036 and the maple logo of the Canadian aircraft are well preserved and the compositional structure of the superpixels in the blue sky and tarmac regions are well simplified.

***Abstract*—We introduce the novel graph-based algorithm SLAM (simultaneous local assortative mixing) for fast and high-quality superpixel segmentation of any large color image. Superpixels are compact semantic image elements; superpixel segmentation is fundamental to a broad range of vision tasks in existing and emerging applications, especially, to safety-critical and time-critical applications. SLAM leverages a graph representation of the image, which encodes the pixel features and similarities, for its rich potential in implicit feature transformation and extra means for feature differentiation and association at multiple resolution scales. We demonstrate, with our experimental results on 500 benchmark images, that SLAM outperforms the state-of-art algorithms in superpixel quality, by multiple measures, within the same time frame. The contributions are at least two-fold: SLAM breaks down the long-standing speed barriers in graph-based algorithms for superpixel segmentation; it lifts the fundamental limitations in the feature-point-based algorithms.**

***Index Terms*—superpixel segmentation, segmentation latency, graph clustering, community detection, inter-scale consistency, intra-cluster homogeneity**

I. Introduction

Superpixel segmentation, since its introduction by Ren and Malik [26], has become fundamental to a wide range of applications involving vision tasks or real-time vision-guided decision-making, such as autonomous navigation in various circumstances. By superpixel segmentation, an image (often in multiple dynamic sequences or video streams) is quickly partitioned into nearly compact and homogeneous local regions, namely, superpixels, to arrive at a semantic image representation [22], [38], above the pixel array representation. See Figure 1. Superpixel segmentation facilitates and accelerates the following vision tasks, to name a few among many: (A-1) scene parsing and image understanding [12], [33], [41]; (A-2) image matching and registration between stereo views or across multiple scenes [4], [12]; (A-3) motion tracking or estimation [39], [42]; (A-4) saliency recognition or recommendation or background subtraction [18], [27], [34]; (A-5) feature and filter learning and training across a class of images [9], [33], [41]. An improvement of superpixel segmentation in quality or speed, or both, enhances the vision tasks at higher levels.

Superpixels are compact and semantic image elements, whereas the pixels are sensor elements (with or without being down/up-sampled). We assume a large color image $\mathcal{I}$ represented by an $m \times n$ pixel array. The pixels are of the same size and the same square or rectangle shape. The spatial neighborhood of a pixel is of a uniform pattern. A superpixel is a set of pixels that are (1) spatially close, connected and (2) chromatically similar, nearly homogeneous, and thereby compressible. Together, superpixels are a non-overlapping cover of the pixels. Superpixels vary in size, shape and neighbor connection patterns. The semantic meaning of a superpixel is encoded by its size, shape and shades of colors.

It is important to clarify the connection and distinction between superpixel segmentation in particular and image segmentation in general, in order to describe additional properties

The first two authors contributed equally to this work.

2022 IEEE High Performance Extreme Computing Conference (HPEC) | 978-1-6654-9786-2/22/$31.00 ©2022 IEEE | DOI: 10.1109/HPEC55821.2022.9926359

imposed on superpixels and to inspect and evaluate superpixel segmentation algorithms. (S-1) A superpixel segmentation is an image segmentation subject to certain regulatory conditions. It is expected to be consistent with a ground-truth image segmentation Ω_*. The consistency is assessed conventionally by two measurement scores among others: the boundary recall (BR) [20] and the achievable segmentation accuracy (ASA) [15], assuming Ω_* is available. See the ground-truth image segmentations and several superpixel segmentations in Figure 2 and Figure 3. (S-2) We also expect the consistency between two superpixel segmentations at different compression ratios. Suppose $k < k'$. Two superpixel segmentations Ω_k and $\Omega_{k'}$, with k and k' superpixels respectively, are consistent if $\Omega_{k'}$ is a finer partition of Ω_k. A discrepancy from such hierarchical consistency can also be measured by the scores BR and ASA, with Ω_* replaced by Ω_k. (S-3) By its fundamental functionality, superpixel segmentation is subject to operational and managerial regulations, which make the distinction of superpixel segmentation from image segmentation in general.

Among the operational regulations, the dominant ones are on size, shape and shade. The size regulation controls the number of image representation elements, from mn pixels to k superpixels, k is substantially smaller than mn, $k \ll mn$. Equivalently, it is a control over the average superpixel size, mn/k, which is the compression ratio. We expect chromatic homogeneity within each superpixel. We use the measurement EV [21]. The shape regulation discourages highly irregular shapes, such as long and narrow strips. The measurement SC [30] indicates the shape uniformity. The SC score is perceived very differently. A strong regulation on shape uniformity often comes at a significant loss of chromatic homogeneity in superpixels over most salient regions.

We briefly describe the state of the art in superpixel segmentation methods in the following two paragraphs. Many of the methods fall into two major types, type-P and type-G. The type-P methods cast image $\mathcal{I}$ as a cloud of feature points in a metric feature space [1], [13], [14], [36]. The feature vector of a pixel is a point in the metric space. A superpixel of $\mathcal{I}$ is a cluster of the feature points. The size and shape regulation conditions on superpixels are translated into the feature vector representation, via supervised or unsupervised feature learning and engineering [1], [14], [36]. There are two sub-categories by how the feature points are clustered, one is by geometric proximity, the other by probabilistic attachment to higher density peaks. The geometry-based methods are typically based on the k-means clustering method [1], [14]. Two particularly notable methods in this sub-category are (iterative) SLIC [1] and (non-iterative) SNIC [2]. The density-based methods include DBSCAN [8], MEAN-SHIFT [7], DP [28] and SD-DP [11]. The former two are iterative, the latter two are non- or semi-iterative. The type-G methods use a graph representation of image $\mathcal{I}$, where every pixel is associated with a graph vertex, as originally introduced by Shi and Malik [31], [32]. We may refer to such a graph as a pixel graph. Superpixels are vertex clusters. Graph clustering methods [14], [15], [42] are then used to obtain the superpixels. The regulative conditions on superpixels are translated into the weights on the vertices or edges or both. The feature-point representation and the graph representation of an image have mutual influence or impact on each other. In particular, a feature representation can be used to form a pixel graph [14], [15]. Reversely, the spectral space of a pixel graph, or a subspace of which, can be used as a geometric space for pixel feature embedding [4], [29]. Some other superpixel segmentation methods belong to neither of the two main categories. They are more specific to certain vision tasks, using certain specific image features such as contours and edges [13], [37] or certain specific algorithmic architectures such as the neural network architecture [40]. We are more interested in meeting the fundamental properties aimed at broader applications.

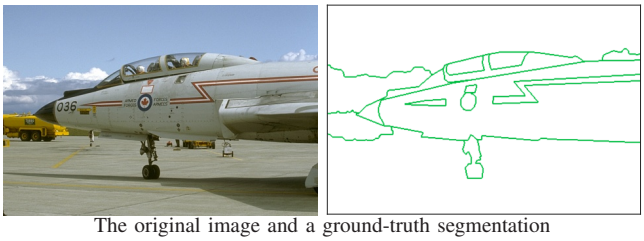
Existing type-G methods seemed outpaced, by up to two orders of magnitude in runtime, by the type-P methods, as they rely on combinatorial graph partition [15], [43] or sequential search for optimal splits and merges [3], [25], [35]. This computational latency eclipsed the extra and rich potential with graph-based image representation in (G-1) exploring and exploiting (and learning) various feature spaces implicitly and easily with kernel transforms [5], and (G-2) utilizing various, effective models for graph clustering or community detection, such as that by Newman [25]. The type-P methods on the other hand have fundamental drawbacks: (P-1) the lack of consistency, with the ground truth and with each other at different compression ratios; and (P-2) the loss of chromatic homogeneity within superpixels.

With this work, we introduce SLAM (simultaneous local assortative mixing), an efficient type-G algorithm for superpixel segmentation. It is the first type-G method competitive in execution time to the type-P methods and superior in superpixel qualities by several measures. In algorithm design, SLAM is novel in its construction of an internal architecture for primitive clustering operations—splits, merges and shifts of subgraphs. It makes combined use of the density principle [11], [28] and the assortative mixing principle [23]. By our experimental results on all 500 images in the data suite BSDS-500 [19], SLAM outperforms SLIC and SNIC, in the assessment scores BR, ASA and EV. In addition, SLAM shows superior consistency between superpixel segmentations at different compression ratios.

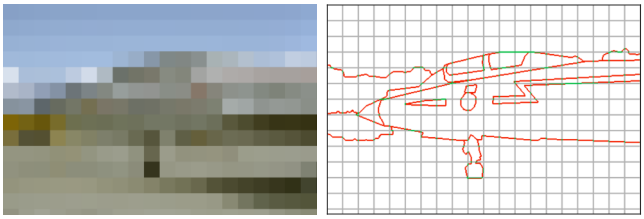
II. Superpixel segmentation with SLAM

We introduce SLAM for superpixel segmentation. We first describe how to apply a graph-based clustering algorithm to superpixel segmentation. We start with constructing a pixel graph $G = G(V, E)$ from the pixel feature vectors. Every pixel p is described by a 5-tuple attribute/feature vector, $p = (i, j, \ell, a, b)$, where (i, j) are the spatial coordinates and (ℓ, a, b) are the chromatic pixel values by the CIELAB color representation system [6]. This kind of pixel features is common to many existing algorithms for superpixel segmentation. We construct a pixel graph, following what is initially introduced by Ren and Malik [26]. There is a one-to-one correspondence between the image pixels and the graph

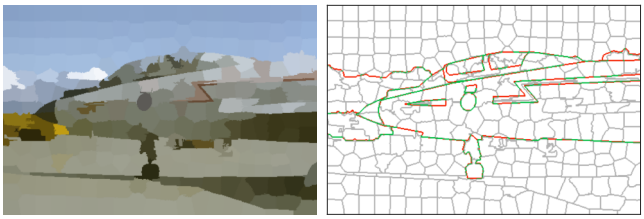
978-1-6654-9786-2/22/$31.00 ©2022 IEEE



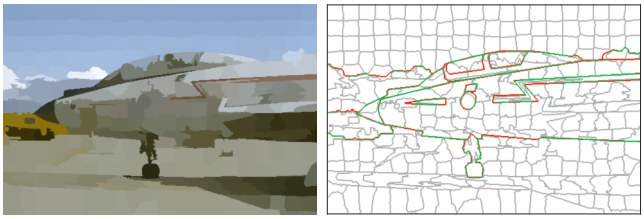
The original image and a ground-truth segmentation



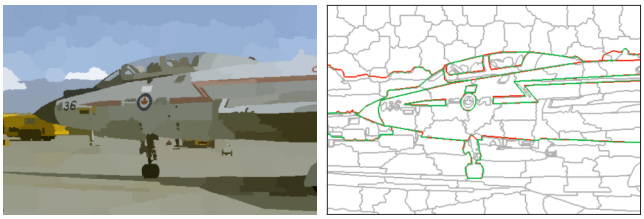
GRID: k = 294, (BR, ASA) = (0.31, 0.90), (EV, SC) = (0.88, 0.85)



SLIC: k = 262, (BR, ASA) = (0.80, 0.93), (EV, SC) = (0.95, 0.56)



SNIC: k = 260, (BR, ASA) = (0.84, 0.96), (EV, SC) = (0.96, 0.46)



SLAM: k = 256, (BR, ASA) = (0.89, 0.98), (EV, SC) = (0.96, 0.38)

Fig. 2: Three superpixel segmentations of the aircraft image by three different algorithms **SLIC, SNIC** and **SLAM**. A grid segmentation (down-sampling) is used as a base reference for quality assessment. TOP: The image of an aircraft with ID 10081 in the data suite BSDS-500, and a ground-truth image segmentation provided by the data suite. COMMON LAYOUT for the next four subfigures, one for each image segmentation: To the right is the superpixel segmentation in gray, with the ground-truth image segmentation overlayed on top. The agreement is in green, the disagreement is in red. To the left is the image reconstructed from the superpixel image representation. At the subcaption are the number k of superpixels and two pairs of quality assessment sores (BR, ASA) and (EV, SC). The pair (BR, ASA) is relative to the ground-truth image segmentation. The higher the scores with BR, ASA and EV are, the better. A higher SC score indicates higher uniformity in superpixel shape. The four segmentations are comparable in compression ratio, by their k values. REMARKS: The grid segmentation is the worst in BR, ASA and EV and the worst in the constructed image. This indicates that the shape uniformity shall be relaxed. The superpixel segmentation by SLAM is the best in BR, ASA and EV, and in the reconstructed image, at a relaxed constraint on shape uniformity. The reconstructed image by SLAM preserves the most salient local information, see in particular the serial number 036 and the Canadian maple logo on the aircraft body. The sky and tarmac zones are highly compressed.

vertices. The pixel vertices are pairwise connected by weighted edges. The weight on edge (u, v) between pixel u and pixel v is a function of the squared distance (or dissimilarity) in the feature space, $d^2(u, v) = (u - v)^{\mathrm{T}} M(u - v)$, where M is a 5×5 symmetric, positive definite matrix. In a simple setting, M is diagonal, $M = I_2 \oplus (r_c\, I_3)$, r_c is the ratio between the coefficients for the spatial elements and the coefficients for the chromatic elements, $r_c > 0$. Any non-identity kernel function effectively transforms the metric feature space nonlinearly and implicitly [5]. We use in particular the weight function on the edge (u, v) with the exponential kernel,

$$\alpha(u, v) = \exp(-\lambda\, d^2(u, v)), \tag{1}$$

where λ is the decay rate, $\lambda > 0$. The numerical support of the function at u is smaller with a larger decay rate. In practice, a hard window of size $w \times w$ centered at pixel u is used to specify the neighbor vertices of u. As a result, the pixel graph G is connected, undirected, edge-weighted and nearly regular in degrees. It has $|V| = mn$ vertices and approximately $(w^2 - 1)mn$ edges.

A graph-based clustering algorithm typically relies on a minimization model. In this paper, we use the following particular model,

$$\begin{aligned} \Omega_\gamma = \arg\min_{\Omega} h(\Omega; \gamma) = -\sum_{S\in\Omega} \frac{\alpha(S)}{\alpha(V)} + \gamma \sum_{S\in\Omega} \frac{\alpha^2(S, V)}{\alpha(V)}, \\ \alpha(S, T) = \sum_{\substack{u\in S, v\in T \\ (u,v)\in E}} \alpha(u, v), \quad \alpha(S) = \alpha(S, S). \end{aligned} \tag{2}$$

Here, Ω denotes a cluster configuration, namely, a partition of the vertex set of graph G, each partitioned subset induces a connected subgraph of G, i.e., a cluster. The number of clusters is not pre-specified. In the context of superpixel segmentation, the first term of clustering/objective function h encodes chromatic homogeneity within each superpixel, the second term encodes the relative volumes of superpixels and the image compression. At $\gamma = 1$, h recovers the negative of the modularity function originally proposed by Newman and Girvan [24], [25]. The resolution parameter γ, $\gamma \in (0, \infty)$, was introduced by Reichardt and Bornholdt as a critical step toward the solution to the resolution limit with any particular value of γ. It can be used to define the cluster resolution and to regulate the balance or bias between homogeneity within superpixels and heterogeneity across superpixels. A larger value of γ leads to a finer segmentation Ω_γ, i.e., with a larger number k of superpixels. The number of cluster configurations is exponential in $|V|$, the number of pixel vertices. The problem of (2) is NP-hard [3], [24]. Existing search algorithms for the problem (2), at any particular value of γ, are iterative with the basic operations on subgraphs—merges, splits and shifts. Among the state-of-the-art algorithms is LEIDEN [35], which is challenged, however, by superpixel segmentation in two aspects. First, it is necessary to link the resolution parameter γ to the number k of superpixels. Secondly, once an appropriate γ value is determined, the algorithm must be sped up by at

978-1-6654-9786-2/22/$31.00 ©2022 IEEE

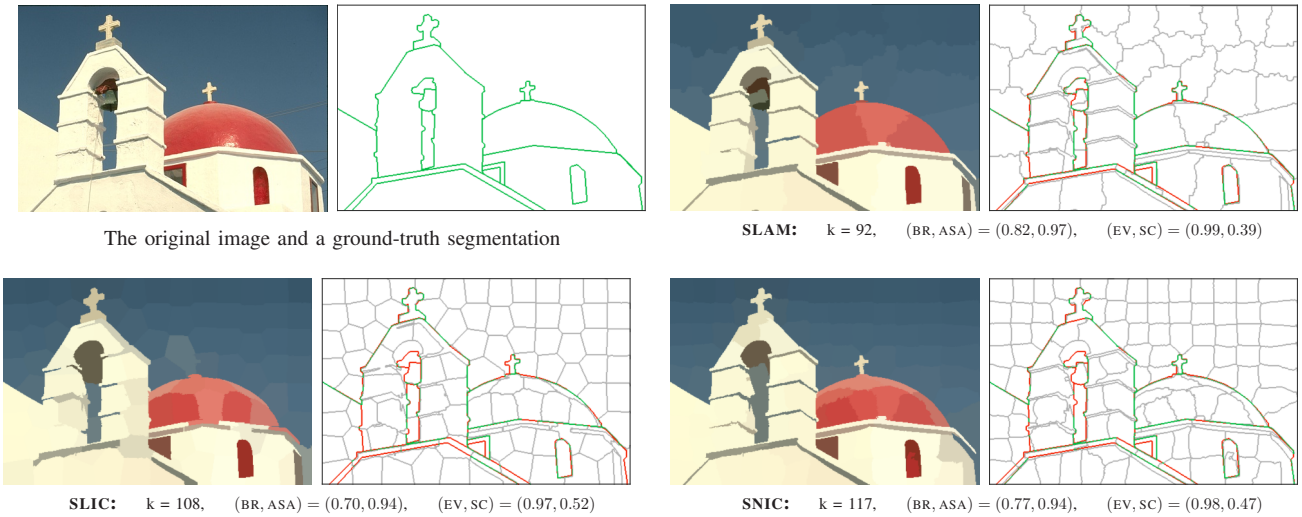


Fig. 3: Three superpixel segmentations of a church image by three different algorithms **SLIC, SNIC** and **SLAM**. TOP-LEFT: The image of the Church of Saint Anna in Mykonos, Greece, with ID 118035 in the data suite BSDS-500, and a ground-truth image segmentation provided by the data suite. COMMON LAYOUT for the other three subfigures, one for each algorithm: To the right is the superpixel segmentation in gray, with the ground-truth image segmentation overlayed on top. The agreement is in green, the disagreement is in red. To the left is the image reconstructed from the superpixel image representation. At the subcaption are the number k of superpixels and two pairs of quality assessment sores (BR, ASA) and (EV, SC). The higher the scores with BR, ASA and EV are, the better. A higher SC score indicates higher uniformity in superpixel shape. The three segmentations are comparable in compression ratio, by their k values. REMARKS: The superpixel segmentation by SLAM is the best by the quality scores and by visual inspection of the reconstructed images. The image reconstructed from SLAM superpixels preserves the cross atop the dome, all three windows in the original image, and the bell in the tower. The dome cross is missed by SLIC; the bell is missed by SLIC and SNIC; a window is missed by SNIC and the ground-truth image segmentation.

least an order of magnitude to be competitive in time with the type-P algorithms, SLIC and SNIC in particular.

In SLAM we have resolved both issues with type-G algorithms for superpixel segmentation. First, we make a connection between γ and k by the BLUERED theory and methodology [16], [17]. The BLUERED theory proves the existence of the intrinsic multi-resolution cluster structure in any graph G, viewed from an h-specific perspective. The BLUERED structure comprises a finite number of cluster configurations. Each BLUERED configuration is optimal over an γ-interval. The intervals together cover the entire γ-domain $(0, \infty)$. The theory also leads to practical algorithms to computationally obtain the intrinsic BLUERED structure autonomously, without resorting to an external selection, search or tuning of the resolution parameter [10], [16]. By the BLUERED structure, we have a mapping between γ-intervals and k values.

Next, we break down the long-standing speed barriers among type-G algorithms for superpixel segmentation. In SLAM, we make an approximation to the BLUERED structure. We describe the algorithmic operations of SLAM. There are two major stages: the construction of a SLAM architecture followed by simultaneous merges, splits, or shifts of vertices and subgraphs. The SLAM architecture is hierarchical. At the base layer, it is a special spanning forest of the pixel graph, by the following child-to-parent attachment rule,

$$\begin{aligned} p(v) &= \arg\min_{(v,u)\in E} \{\Delta h(u,v) \,|\, \Delta h(u,v) < 0 \,\&\, d(u) > d(v)\}, \\ \Delta h(u,v) &= \gamma\, d(u)\, d(v) - \alpha(u,v), \quad \forall v \in V, \end{aligned} \tag{3}$$

where $d(v) = \alpha(v, V)$ is the nodal density of vertex v. The first condition is a local version of attachment by the assortative mixing principle by Newman [23], the second is by the density principle that governs DP [28] and SDDP [11]. The child-to-parent attachments are local and simultaneous, the spanning forest is constructed instantaneously in an ideal computation condition. The objective function is decreased conditionally to each local neighborhood. The likelihood of a global descent due to simultaneous local descents is higher on a sparser graph. The pixel graph is sparse. At the next layer, the spanning rule is applied to the contracted graph each node of which is a tree at the base level. The spanning-contracting process can extend to higher layers.

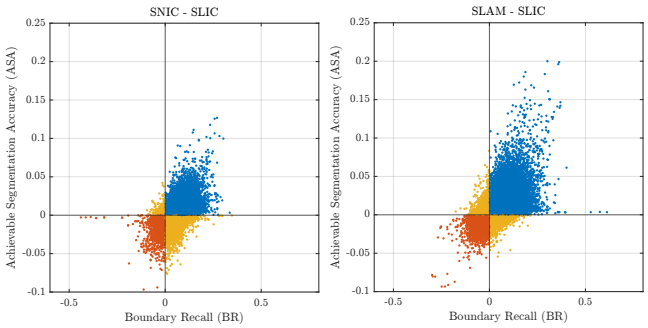
The SLAM structure is not only used for making the γ-k connection, but also used as a reference structure of subgraphs for subsequent search operations—splits, merges and shifts.

We show with experimental results that SLAM, a type-G algorithm, surpasses the two type-P algorithms, SLIC and SNIC, by superpixel segmentation quality measures within a comparable time frame. The SLAM approximation can also be extended to accelerate graph clustering in general [10].

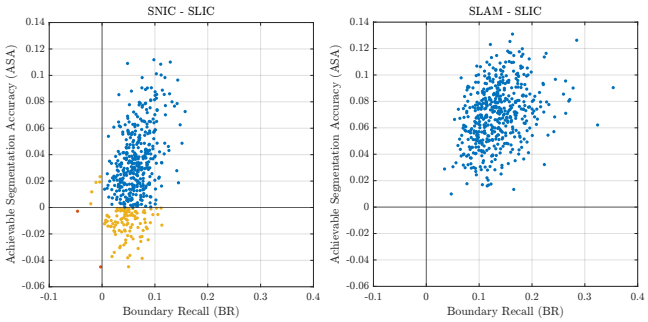
III. EXPERIMENTS

We present experimental comparisons in efficiency and accuracy between SLAM and three other algorithms. The first two other algorithms are SLIC and SNIC, the latter represents the state-of-the-art type-P algorithms for superpixel segmentation with the number of superpixels specified by k. The third algorithm is LEIDEN, representing the state-of-the-art graph clustering algorithms. We apply LEIDEN to superpixel segmentation, with the γ-k connection provided by SLAM. The

978-1-6654-9786-2/22/$31.00 ©2022 IEEE



(a) Comparisons in consistency with ground-truth segmentations



(b) Comparisons in inter-resolution-scale consistency

Fig. 4: Summaries of comparisons in consistency over the entire BSDS-500 data suite, with scatter plots in the BR-ASA plane. The comparisons are made over $500\times3\times10$ experiments with three algorithms, SLIC, SNIC and SLAM, at 10 different compression ratios. More than $60,000$ comparisons are made and summarized in four scatter plots. TOP: Consistency with the ground-truth segmentations. Every point in each of the two scatter plots is the difference in [ASA, BR] of SNIC (on the left), or SLAM (on the right), from SLIC on an image $\mathcal{I}$ against one of the ground-truth image segmentations provided for $\mathcal{I}$. There are q points for each image $\mathcal{I}$, q is the number of ground-truth segmentations provided for $\mathcal{I}$, $4 \le q \le 9$, over the entire data suite. There are $\mathbf{29{,}656}$ points (comparisons) in each scatter plot. A blue point in the northeast quadrant shows a winner case. By the top-left plot, SNIC is better than SLIC in $\mathbf{18{,}878}$ cases (blue points); by the top-right plot, SLAM is better than SLIC in $\mathbf{20{,}214}$ cases (blue points). BOTTOM: Inter-scale consistency at two compression ratios $1{,}500\times$ and $550\times$, or equivalently, the numbers of superpixels are $k \approx 100$ and $k' \approx 280$. There are $\mathbf{500}$ points (comparisons) in each scatter plot. By the bottom-left plot, SNIC is better than SLIC in $\mathbf{388}$ cases (blue points); by the bottom-right plot, SLAM is better, and much better, than SLIC in all $\mathbf{500}$ cases. See also Figure 5.

four algorithms are compared in their performance over all images at the Berkeley segmentation benchmark suite BSDS-500 [19], [44]. The data suite has 500 color images, several ground-truth image segmentations (at different detail levels) are provided for each image [44]. We use two particular images for demonstrating detailed performance measurements and visual inspection – the image of an aircraft with ID 10081 and the image of the Church of Saint Anna in Mykonos, Greece, with ID 118035. See Figures 1 to 3. We also provide summaries over the whole data suite in Figure 4 and Figure 6.

We assess the properties of a superpixel segmentation with the following performance measures. (M-1) consistency with ground-truth image segmentations, measured by a pair of complementary scores boundary recall BR [20] and achievable segmentation accuracy ASA [15], (M-2) inter-resolution-scale consistency between segmentations at different compression ratios, also measured by BR and ASA, as explained in Section I, (M-3) homogeneity within superpixels and compact compression and decompression quality, indicated by EV [21], (M-4) superpixel shape uniformity, indicated by SC [30], and (M-5) execution time. The architecture specification of the computers we use for benchmarking is in Table 1. The first two measures are complementary to each other, the next two may be conflicting. A strong shape uniformity across superpixels may cause a significant loss of chromatic homogeneity within superpixels or decrease the compression ratio.

The formulas for the score functions are as follows:

BR: Boundary recall [20]:

$$\mathrm{BR}(\Omega_k, \Omega_*) = \frac{\sum_{p\in\mathcal{B}(\Omega_*)} \chi[\min_{p'\in\mathcal{B}(\Omega_k)} \|p-p'\| < \epsilon]}{|\mathcal{B}(\Omega_*)|}, \quad (4)$$

where $\mathcal{B}$ is the set of pixels on the spatial boundary, $\chi[\cdot]$ is the set indicator function, and ϵ is the tolerance distance (in pixels) between the reference and target boundary. We set ϵ to 2. The best BR score is achieved when $\Omega = \Omega_*$.

ASA: Achievable segmentation accuracy [15]:

$$\mathrm{ASA}(\Omega_k, \Omega_*) = \frac{\sum_{S\in\Omega_k} \max_{T\in\Omega_*} |S\cap T|}{|V|}. \quad (5)$$

The best ASA score is achieved when $\Omega = \Omega_*$.

EV: Explained variation [21]:

$$\mathrm{EV}(\Omega) = \frac{\sum_{S\in\Omega}\sum_{p\in S} \|\mathrm{C}(p) - \mathrm{C}_\mu(\Omega)\|_2^2}{\sum_{S\in\Omega}\sum_{p\in S} \|\mathrm{C}(p) - \mathrm{C}_\mu(\Omega)\|_2^2}, \quad (6)$$

where $\mathrm{C}_\mu(S)$ denotes the average chromatic vector of the superpixel S in the CIELAB space, and $\mathrm{C}_\mu(\Omega)$ denotes the average chromatic vector of the whole image. A higher EV value indicates higher chromatic homogeneity of the superpixels. The measure reaches its maximum value 1 when each superpixel is chromatically constant.

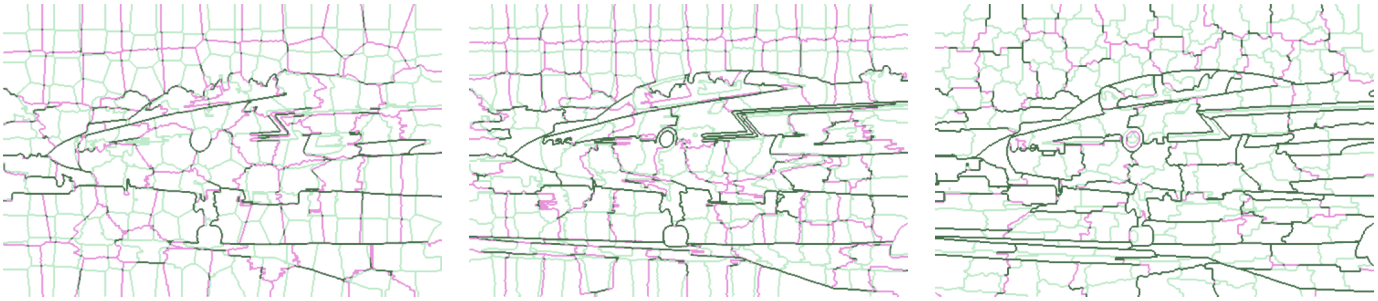
SC: Shape Compactness [30]:

$$\mathrm{SC}(\Omega) = \sum_{S\in\Omega} \frac{|S|}{|V|} \cdot \frac{4\pi|S|}{p^2(S)}, \quad (7)$$

where $p(S)$ is the perimeter of superpixel S. A higher SC value indicates higher uniformity in superpixel shape.

In addition, we provide a decompressed/reconstructed image for visual inspection of the overall quality of a superpixel segmentation. In the compressed form, the chromatic feature vector of each superpixel S is the average of the chromatic feature vectors of the pixels within S. In the decompressed image, the chromatic values of a superpixel S are assigned to all the pixels in S. The more chromatically homogeneous within superpixels, the closer the decompressed image is to the original image. At the same compression ratio, subject to similar shape-size regulations, the closer the decompressed image is to the original, the better quality the segmentation is.

978-1-6654-9786-2/22/$31.00 ©2022 IEEE



SLIC: $k \in \{76, 188\}$ **SNIC:** $k \in \{96, 216\}$ **SLAM:** $k \in \{82, 213\}$

Fig. 5: Comparisons in inter-scale segmentation consistency among **SLIC**, **SNIC** and **SLAM**. The pixel graph is for the aircraft image in Figure 1. COMMON LAYOUT for the three subfigures, one for each algorithm: A coarse segmentation in magenta (with a smaller k value) overlays with a finer segmentation in cyan. The number of superpixels rendered by the three algorithms is around 85 at the coarse scale and 205 at the fine scale. The inter-scale alignment is shown in green, the misalignment is shown by the exposed coarse boundaries in magenta. REMARKS: On the same pixel graph, the coarse segmentation by SLIC is hardly covered by the finer one. SNIC is far better in inter-scale consistency than SLIC, but outperformed by SLAM. See also Figure 4.

TABLE 1: Specification of the multi-core computers used for the experiments.

CPU	Clock (GHz)	Cores	L1 (KiB)	L2 (KiB)	L3 (MiB)	RAM (GiB)
Intel Xeon E5-2640 v4	2.4	10	10×32	10×256	25	256
Intel Core i5-4288U	2.6	2	2×64	2×256	30	16

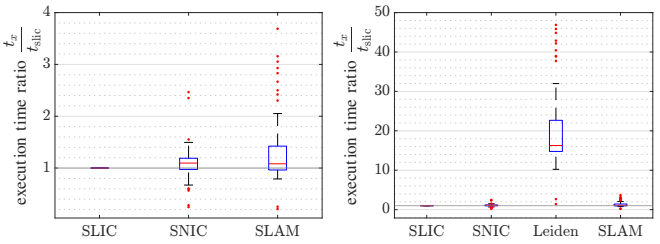


Fig. 6: Comparisons in execution time among different algorithms. LEFT: Summary statistics of the comparisons over 500 images among the three algorithms SLIC, SNIC, and SLAM is shown. On each image, the execution time is normalized by the SLIC time. All three algorithms are comparable in execution time, SLAM in its current and primitive version has a larger variation. RIGHT: Summary statistics of the comparisons among SLIC, SNIC, SLAM, and LEIDEN is shown. LEIDEN is a state-of-the-art graph-clustering algorithm. SLAM outpaces LEIDEN by up to 45 times on the BSDS-500 data set, with about the same segmentation quality. The benchmarks are done on the computers with their architecture specification in Table 1.

We make $500 \times 3 \times 10$ superpixel segmentations with three algorithms, SLIC [45], SNIC [46], and SLAM, at 10 different compression ratios. We present summaries over more than 60,000 comparisons by performance measures. In Figure 2, we present three segmentations of the aircraft image rendered by the three algorithms under study. The detailed description is in the figure caption. We use a regular grid as a reference. A regular grid partition is simply the pixels down-sampled. It is no surprise that the grid is the best in the SC score on shape uniformity and the worst in all other quality measures. The reconstructed image from the grid partition is very blurry. Furthermore, any two grid partitions have the worst inter-scale consistency unless one is a super-grid of the other. In contrast, SLAM is the best in all other performance measures except a relaxation on shape uniformity. The superpixel segmentation by SLAM is evidently superior. It preserves distinctive and most salient information and it makes adaptive compression. By achieving higher homogeneity and relaxing on shape uniformity, SLAM admits a higher compression ratio with a lower error rate in reconstruction. Similar observations are made from Figure 3 with the church image.

Figure 5 and Figure 4 present comparisons in inter-scale consistency. Figure 5 shows the overlay of two super-segmentations on a particular image. Figure 4 is a summary made of 60,312 pairwise comparisons. Details are described in the captions. SLAM shows absolute superiority in inter-scale consistency. By sampling the suite images, we found certain obvious problems with some ground-truth image segmentations, such as the missing window in the ground-truth segmentation in Figure 3.

We show in Figure 6 that in the same time frame, SLAM achieves higher segmentation quality compared to SLIC and SNIC over the data suite BSDS-500. SLAM outpaces the state-of-art algorithm LEIDEN [35] on CPUs, by up to 45 times, without compromising superpixel quality. The speed up is much higher in parallel computation [10], because LEIDEN has inherently sequential bottlenecks, SLAM is highly parallelizable.

IV. DISCUSSION

Our contributions with SLAM for superpixel segmentation are multi-fold. (D-1) SLAM breaks down the long-standing speed barrier in graph-based superpixel segmentation algorithms. (D-2) SLAM lifts the fundamental limitations in the quality of superpixel segmentation. Specifically, SLAM renders superpixel segmentations with significantly higher quality than SLIC and SNIC, within the same time frame. (D-3) SLAM shows greater potential with the use of a graph representation in exploring implicit, non-linear feature transformation, in probabilistic feature learning. (D-4) It is possible to extend SLAM to higher-dimensional image data or dynamic image streams, and to speed up algorithms for graph clustering in general [10].

978-1-6654-9786-2/22/$31.00 ©2022 IEEE

Acknowledgments. The authors thank the anonymous reviewers for their comments that helped to clarify the following: (i) How the BLUERED theory underlies SLAM, (ii) SLAM for superpixel segmentation can be extended to graph clustering in general, and (iii) SLAM is parallelizable.

REFERENCES

[1] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Süsstrunk, "SLIC superpixels compared to state-of-the-art superpixel methods," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 11, pp. 2274–2282, 2012.

[2] R. Achanta and S. Susstrunk, "Superpixels and polygons using simple non-iterative clustering," in *2017 IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 4895–4904.

[3] V. D. Blondel, J.-L. Guillaume, R. Lambiotte, and E. Lefebvre, "Fast unfolding of communities in large networks," *Journal of Statistical Mechanics: Theory and Experiment*, vol. 2008, no. 10, p. P10008, Oct. 2008.

[4] A. Bodis-Szomoru, H. Riemenschneider, and L. Van Gool, "Fast, approximate piecewise-planar modeling based on sparse structure-from-motion and superpixels," in *2014 IEEE Conference on Computer Vision and Pattern Recognition*, Jun. 2014, pp. 469–476.

[5] B. E. Boser, I. M. Guyon, and V. N. Vapnik, "A training algorithm for optimal margin classifiers," in *Proceedings of the Fifth Annual Workshop on Computational Learning Theory - COLT '92*, 1992, pp. 144–152.

[6] CIE, "CIE 015:2018 Colorimetry, 4th Edition," International Commission on Illumination (CIE), Tech. Rep., Oct. 2018.

[7] D. Comaniciu and P. Meer, "Mean shift: A robust approach toward feature space analysis," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 5, pp. 603–619, May 2002.

[8] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," in *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, 1996, pp. 226–231.

[9] L. Fang, S. Li, X. Kang, and J. A. Benediktsson, "Spectral–spatial classification of hyperspectral images with a superpixel-based discriminative sparse model," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 53, no. 8, pp. 4186–4201, 2015.

[10] D. Floros, "Efficient analysis of local and global structures in large networks," Ph.D. dissertation, Aristotle University of Thessaloniki, Greece, Sep. 2022.

[11] D. Floros*, T. Liu*, N. Pitsianis, and X. Sun, "Sparse dual of the density peaks algorithm for cluster analysis of high-dimensional data," in *IEEE High Performance Extreme Computing Conference*, 2018.

[12] A. Gupta, A. A. Efros, and M. Hebert, "Blocks world revisited: Image understanding using qualitative geometry and mechanics," in *Proceedings of the 11th European Conference on Computer Vision: Part IV*, vol. 6314, 2010, pp. 482–496.

[13] A. Levinshtein, A. Stere, K. Kutulakos, D. Fleet, S. Dickinson, and K. Siddiqi, "Turbopixels: Fast superpixels using geometric flows," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 12, pp. 2290–2297, 2009.

[14] Z. Li and J. Chen, "Superpixel segmentation using linear spectral clustering," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 1356–1363.

[15] M.-Y. Liu, O. Tuzel, S. Ramalingam, and R. Chellappa, "Entropy rate superpixel segmentation," in *2011 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2011, pp. 2097–2104.

[16] T. Liu, "On Graph Clustering or Community Detection: A Characteristic Analysis and Its Implications," Ph.D. dissertation, Duke University, Durham, NC, USA, May 2021.

[17] T. Liu, D. Floros, N. Pitsianis, and X. Sun, "Digraph clustering by the BlueRed method," in *2021 IEEE High Performance Extreme Computing Conference (HPEC)*, 2021, pp. 1–7.

[18] Z. Liu, X. Zhang, S. Luo, and O. Le Meur, "Superpixel-based spatiotemporal saliency detection," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 24, no. 9, pp. 1522–1540, Sep. 2014.

[19] D. Martin, C. Fowlkes, D. Tal, and J. Malik, "A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics," in *Proceedings Eighth IEEE International Conference on Computer Vision*, vol. 2, 2001, pp. 416–423.

[20] D. Martin, C. Fowlkes, and J. Malik, "Learning to detect natural image boundaries using local brightness, color, and texture cues," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 5, pp. 530–549, May 2004.

[21] A. P. Moore, S. J. D. Prince, J. Warrell, U. Mohammed, and G. Jones, "Superpixel lattices," in *2008 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2008, pp. 1–8.

[22] M. Mostajabi, P. Yadollahpour, and G. Shakhnarovich, "Feedforward semantic segmentation with zoom-out features," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2015, pp. 3376–3385.

[23] M. E. J. Newman, "Assortative Mixing in Networks," *Physical Review Letters*, vol. 89, no. 20, p. 208701, 2002.

[24] M. E. J. Newman and M. Girvan, "Finding and evaluating community structure in networks," *Physical Review E*, vol. 69, no. 2, p. 026113, Feb. 2004.

[25] M. E. J. Newman, "Modularity and community structure in networks," *Proceedings of the National Academy of Sciences*, vol. 103, no. 23, pp. 8577–8582, 2006.

[26] X. Ren and J. Malik, "Learning a classification model for segmentation," in *Proceedings Ninth IEEE International Conference on Computer Vision*, vol. 1, 2003, pp. 10–17.

[27] Z. Ren, S. Gao, L.-T. Chia, and I. W.-H. Tsang, "Region-based saliency detection and its application in object recognition," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 24, no. 5, pp. 769–779, May 2014.

[28] A. Rodriguez and A. Laio, "Clustering by fast search and find of density peaks," *Science*, vol. 344, no. 6191, pp. 1492–1496, Jun. 2014.

[29] S. T. Roweis and L. K. Saul, "Nonlinear dimensionality reduction by locally linear embedding," *Science*, vol. 290, no. 5500, pp. 2323–2326, Dec. 2000.

[30] A. Schick, M. Fischer, and R. Stiefelhagen, "Measuring and evaluating the compactness of superpixels," in *Proceedings of the 21st International Conference on Pattern Recognition (ICPR2012)*, Nov. 2012, pp. 930–934.

[31] J. Shi and J. Malik, "Normalized cuts and image segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 1997, pp. 731–737.

[32] ——, "Normalized cuts and image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pp. 888–905, 2000.

[33] J. Tighe and S. Lazebnik, "Superparsing: Scalable nonparametric image parsing with superpixels," in *European Conference on Computer Vision*, vol. 6315, 2010, pp. 352–365.

[34] N. Tong, H. Lu, L. Zhang, and X. Ruan, "Saliency detection with multi-scale superpixels," *IEEE Signal Processing Letters*, vol. 21, no. 9, pp. 1035–1039, Sep. 2014.

[35] V. A. Traag, L. Waltman, and N. J. van Eck, "From Louvain to Leiden: Guaranteeing well-connected communities," *Scientific Reports*, vol. 9, no. 1, p. 5233, Dec. 2019.

[36] W.-C. Tu, M.-Y. Liu, V. Jampani, D. Sun, S.-Y. Chien, M.-H. Yang, and J. Kautz, "Learning superpixels with segmentation-aware affinity loss," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, pp. 568–576.

[37] M. Van den Bergh, X. Boix, G. Roig, and L. Van Gool, "SEEDS: Superpixels extracted via energy-driven sampling," *International Journal of Computer Vision*, vol. 111, no. 3, pp. 298–314, 2015.

[38] A. Vezhnevets, V. Ferrari, and J. M. Buhmann, "Weakly supervised semantic segmentation with a multi-image model," in *2011 International Conference on Computer Vision*, Nov. 2011, pp. 643–650.

[39] S. Wang, H. Lu, F. Yang, and M.-H. Yang, "Superpixel tracking," in *2011 International Conference on Computer Vision*, Nov. 2011, pp. 1323–1330.

[40] F. Yang, Q. Sun, H. Jin, and Z. Zhou, "Superpixel segmentation with fully convolutional networks," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2020, pp. 13 961–13 970.

[41] J. Yang, B. Price, S. Cohen, and M.-H. Yang, "Context driven scene parsing with attention to rare classes," in *2014 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2014, pp. 3294–3301.

[42] D. Yeo, J. Son, B. Han, and J. H. Han, "Superpixel-based tracking-by-segmentation using Markov chains," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jul. 2017, pp. 511–520.

978-1-6654-9786-2/22/$31.00 ©2022 IEEE

[43] S. Yin, Y. Qian, and M. Gong, "Unsupervised hierarchical image segmentation through fuzzy entropy maximization," *Pattern Recognition*, vol. 68, pp. 245–259, 2017.
[44] www2.eecs.berkeley.edu/Research/Projects/CS/vision/bsds, "Berkeley segmentation benchmark suite."
[45] www.epfl.ch/labs/ivrl/research/slic-superpixels, "SLIC superpixels."
[46] www.epfl.ch/labs/ivrl/research/snic-superpixels, "SNIC superpixels."

978-1-6654-9786-2/22/$31.00 ©2022 IEEE