Laplacian Coordinates: Theory and Methods for Seeded Image Segmentation

Wallace Casaca, João Paulo Gois, Harlen Costa Batagelo,
Gabriel Taubin, *Fellow, IEEE*, and Luis Gustavo Nonato, *Member, IEEE*

Abstract—Seeded segmentation methods have gained a lot of attention due to their good performance in fragmenting complex images, easy usability and synergism with graph-based representations. These methods usually rely on sophisticated computational tools whose performance strongly depends on how good the training data reflect a sought image pattern. Moreover, poor adherence to the image contours, lack of unique solution, and high computational cost are other common issues present in most seeded segmentation methods. In this work we introduce Laplacian Coordinates, a quadratic energy minimization framework that tackles the issues above in an effective and mathematically sound manner. The proposed formulation builds upon graph Laplacian operators, quadratic energy functions, and fast minimization schemes to produce highly accurate segmentations. Moreover, the presented energy functions are not prone to local minima, i.e., the solution is guaranteed to be globally optimal, a trait not present in most image segmentation methods. Another key property is that the minimization procedure leads to a constrained sparse linear system of equations, enabling the segmentation of high-resolution images at interactive rates. The effectiveness of Laplacian Coordinates is attested by a comprehensive set of comparisons involving nine state-of-the-art methods and several benchmarks extensively used in the image segmentation literature.

Index Terms—Seeded image segmentation, graph laplacian, laplacian coordinates, energy minimization models

1 Introduction

Image segmentation plays a crucial role in computer vision and pattern recognition. Prominent applications such as medical imaging [1], [2], [3], [4], machine vision [5], [6], [7] and tracking [8], [9], [10] have leveraged the development of an extensive number of methods for segmenting images. Among this diversity of approaches, the procedure of interpreting a digital image as a graph became a consolidated field. Converting images into graphs enables the use of solid mathematical tools such as energy minimization models [11], [12], [13], [14], spectral graph theory [15], [16], [17], [18] and numerical optimization schemes [19], [20] to effectively solve the segmentation problem. The versatility provided by graph representations as to data characterization, topological arrangement, and the freedom to properly set edge weights greatly increases the capability of segmentation methods in distinguishing patterns and shapes. However, it is unanimous to postulate that outperforming human skills in terms of recognition is a very challenging task, particularly due to the human subjectivity in identifying boundaries and clusters. To better deal with this criticism, the so-called *seeded/user-assisted image segmentation algorithms* [21], [22], [23], [24], [25], [26] have turned a trend, motivated by the user's autonomy, fast interactive feedback, and the versatility of processing graphs as energy minimization problems. This class of methods usually seek to accommodate graph Laplacians [27], [28], [29] and other well-posed graph operators [30], [31] into a similarity/affinity graph which encodes image attributes like colors, textures and gradients. The segmentation is then accomplished by minimizing an energy function defined on this affinity guidance graph.

Despite their powerfulness and effectiveness, interactive segmentation methods bear a number of drawbacks not properly tackled by several state-of-the-art partitioners [32], [33], [34], in particular:

1) The segmentation generally exhibits low adherence on the object contours, failing to capture the whole target or producing a low-quality segmentation.
2) The need for sophisticated optimization tools or many pre/post-processing or learning steps to ensure satisfactory results, leading to unacceptable computational cost especially for high-resolution images.
3) The algorithms are highly sensitive to the choice of parameters, especially the edge weights, as local changes can lead to very contrasting outputs.
4) Ensuring accuracy and uniqueness of solution is a hurdle for several seed-based segmentation methods.

In this paper, we present *Laplacian Coordinates* (LC), a new graph-based energy minimization framework for seeded image segmentation that tackles most of the

- *W. Casaca is with the Department of Energy Engineering, São Paulo State University (UNESP), Rosana 01049-010, Brazil. E-mail: wallace.casaca@unesp.br.*
- *J.P. Gois and H.C. Batagelo are with the Center for Mathematics, Computing and Cognition, Federal University of ABC (UFABC), Santo André 09210-580, Brazil. E-mail: {joao.gois, harlen.batagelo}@ufabc.edu.br.*
- *G. Taubin is with the School of Engineering, Brown University, Providence, RI 02912 USA. E-mail: taubin@brown.edu.*
- *L.G. Nonato is with the ICMC, University of São Paulo (USP), São Carlos 13566-590, Brazil. E-mail: gnonato@icmc.usp.br.*

Manuscript received 4 July 2019; revised 26 Dec. 2019; accepted 11 Feb. 2020. Date of publication 17 Feb. 2020; date of current version 1 July 2021. (Corresponding author: Wallace Casaca.)
Digital Object Identifier no. 10.1109/TPAMI.2020.2974475

issues listed above. Our framework relies on quadratic energy models that combine graph Laplacian and user-provided seeds into a mathematically well-posed clustering approach. In contrast to most existing methods which search for solutions that minimize the "distance" between pairwise pixel/superpixels (i.e., graph nodes), our formulation aims to minimize the average distance between the nodes, promoting a better anisotropic behavior to the label propagation while ensuring contour adherence and smoothness. Moreover, the solution given by the LC energy function is not prone to be trapped in local minima. In fact, it is guaranteed that the solution is the globally optimal one, a desirable property not always present in many image segmentation methods. Another important trait of Laplacian Coordinates is that its minimizer is given by the solution of a constrained sparse linear system, making it simple to be coded while still being computationally efficient.

A preliminary study of Laplacian Coordinates appears in our CVPR's paper [24] published a few years ago, which has served as baseline for other computer vision applications such as Moving Object Detection [35], 3D Reconstruction of Cone Beam Computed Tomography [36], Image Enhancement [6], Photo Colorization [37], and Mesh Cutting [38]. Going beyond our previous investigation, we design a new quadratic energy model that sets the user-specified seeds as *hard constraints* in the LC optimization. As a result, the Laplacian matrix involved in the minimization problem is of reduced size, therefore lessening the burden for getting the solution. We also extend the hard version of the Laplacian Coordinates energy to perform superpixel segmentation on high-resolution images at interactive rates, a task not occasionally confronted by several seeded segmentation methods in practice due to their high computational cost. We carry out an extensive experimental evaluation covering nine state-of-the-art methods and representative public benchmarks with several natural/real-world images. Finally, proofs and theoretical analysis involving the LC energy function in both hard and soft designs are also given (see the Supplementary Material, which can be found on the Computer Society Digital Library at http://doi.ieeecomputersociety.org/10.1109/TPAMI.2020.2974475). Fig. 1 illustrates some of the capabilities of the LC framework.[1]

Summary of Contributions. The main contributions introduced by the Laplacian Coordinates framework are:

- A functional and simple-to-solve energy minimization formulation for segmenting seeded images.
- The segmentation task is formulated as a quadratic minimization problem where the seeds can be conventionally imposed as soft or hard constraints, being also able to handle superpixel segmentation in high-resolution images at interactive rates.
- Laplacian Coordinates gathers attractive characteristics such as boundary fitting, anisotropic behavior, smoothness and uniqueness of solution.
- The segmentation trivially consists of solving a constrained sparse system of linear equations.

2 Related Work

To better contextualize our contributions while following the taxonomy given by Cousty *et al.* [39] and Couprie *et al.* [40], we focus our discussion on four very influential groups of graph-based segmentation methods:

1) *Graph Cuts* [22], [30], [41], [42], [43], [44].
2) *Random Walks* [21], [45], [46], [47], [48].
3) *Watersheds* [23], [39], [40], [49], [50], [51].
4) *Shortest Paths* [52], [53], [54], [55], [56], [57].

Additionally, we provide a discussion on seeded image segmentation under the deep learning paradigm, given its omnipresence and relevance in many fields of applications.

Graph Cuts. The *Graph-Cut* approach (GC) was first introduced by Boykov and Jolly [58] and improved later on [22] to address the problem of interactive N-dimensional image clustering. The rationale behind the GC is to consider the image as a graph, searching for the minimum cut between the seeded regions. For instance, assuming a 4- or 8-connected graph that represents the image, one can obtain foreground (object) and background segments by minimizing the following energy function:

$$E_{GC}(\mathbf{x}) = \sum_{i \in B \cup F} D_i(x_i) + \lambda \sum_{(i,j) \in E} V_{ij}(x_i, x_j), \tag{1}$$

where x_i is the unknown label w.r.t. pixel i, D_i is the data term, V_{ij} represents the regularization/boundary term, and $B \cup F$ gives the union of the background and foreground input labels. D_i measures the cost of assigning the label x_i to the pixel i, while V_{ij} imposes spatial smoothness to the segmentation, ensuring that pixels nearby the border of the object will assume opposite labels. Different choices of V_{ij} lead to different smoothness characterizations. To get the minimizer for Eq. (1), the GC generally utilizes the *min-cut/max-flow algorithm* [59], [60]. But if the energy involves multiple labels instead of only a binary partition, minimizing Energy (1) turns an NP-Hard problem [12], and an approximation of the global optimal is then required.

Many extensions of GC have been proposed in the literature, especially w.r.t. alternative ways of promoting user interaction and new variants to the graph-cut energy. For example, the Lazy snapping method [41] introduces a friendly user interface as a coarse-to-fine tool operated at interactive rates by the user. In a similar work, the so-called Grabcut [30] sets bounding boxes as input data to drive the graph cutting, where the fore/background regions are obtained by computing a Gaussian Mixture Model (GMM). Significant improvements of the primary approaches [22] and [30] have been also reported [61], [62], [63], [64], in addition to support other applications such as human face segmentation [65], medical imaging [4], [66], and mobile apps [67]. In this line, Tang *et al.* [63] redesigned the Grabcut [30] to deal with appearance models to improve the color separability of the image, while Bai *et al.* [68] presents a ratio-based function that takes the user scribbles as soft constraints while minimizing the graph-cut energy.

Although versatile and mathematically sound, techniques inspired on graph cuts tend to suffer from the generation of small segmented regions, which arise due to the underlying mathematical formulation that looks for

[1] Source code of our framework available at `http://github.com/hbatagelo/laplacianseg`.

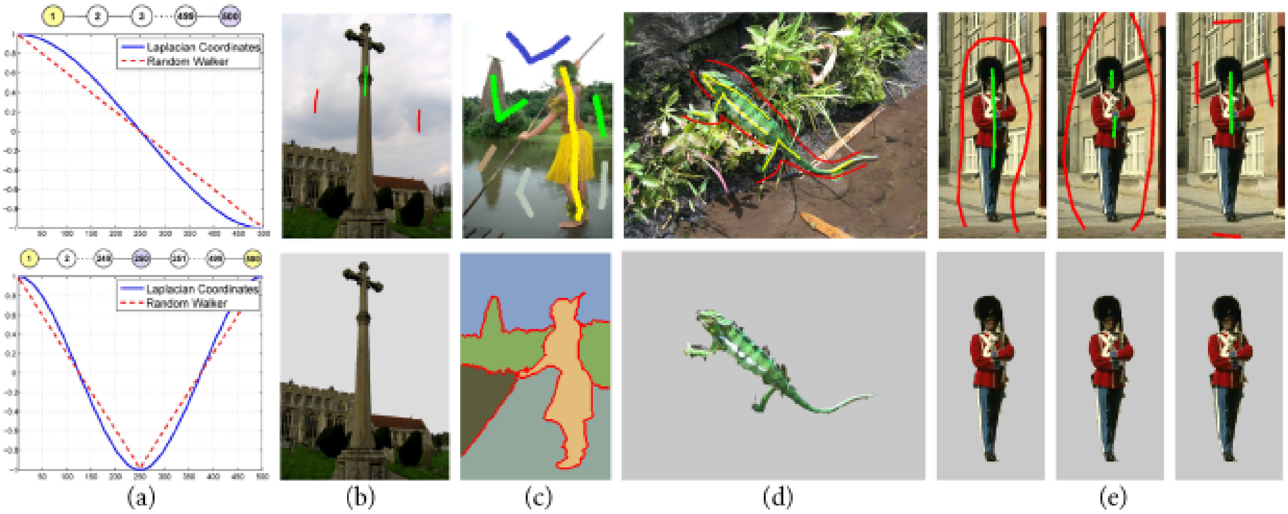


Fig. 1. A summary of different capabilities of the Laplacian Coordinates framework. From left to right, (a) Smoothness and seed propagability: Laplacian Coordinates solution × the well-established Random Walker [21] for line graphs as shown in the top row, with unitary weights and seeds in yellow and purple, (b) Contour adherence/boundary fitting, (c) Multiple-region segmentation, (d) Fast superpixel segmentation for large size images, (e) Invariance to seed placement.

solutions with minimal boundary length [45]. Another issue is the NP-hardness of the multilabel segmentation case [12].

Random Walks. The *Random Walker* segmentation (RW) [21] is a useful and easy-to-implement approach that relies on the standard graph Laplacian formulation $\mathbf{L}\mathbf{x} = 0$, where $\mathbf{L}$ is the graph Laplacian operator [27]. Given a graph G representing the image, the RW method minimizes the following quadratic energy function on G:

$$E_{RW}(\mathbf{x}) = \sum_{(i,j)\in E} w_{ij}(x_i - x_j)^2, \qquad (2)$$

constrained to the input seeded vertices: $x_i = x_B, i \in B$, and $x_i = x_F, i \in F$, with $\mathbf{W} = (w_{ij})$ denoting the weights assigned to the edges of G. Grady [21] presents an interesting interpretation of the standard Laplacian formulation $\mathbf{L}\mathbf{x} = 0$, by associating to each unseeded pixel, the probability of a random walker starting on it to reach a background seed. The segmentation is then performed by assigning a background label to a pixel if the probability is greater than 0.5 and a foreground label otherwise. In practice, instead of solving $\mathbf{L}\mathbf{x} = 0$, the RW solves $\mathbf{D}^{-1}\mathbf{W}\mathbf{x} = \mathbf{x}$, imposing the input seeds as constraints to ensure uniqueness of solution, where $\mathbf{D}$ is the diagonal weighted valency matrix.

As the GC approach, the RW has also strongly influenced several interactive segmentation methods. For instance, aiming at quickly computing the solution of the RW energy, [46] and [69] optimize a spectral decomposition problem. Similarly, Casaca *et al.* [70] interpreted the RW model as a spectral clustering problem, where the image is subdivided into two particular bands: cartoon (smooth) and texture. While the cartoon band is taken to build the affinity graph, the texture is used to interactively tuning the clustering process. Segmenting textured images in a more efficient way was also the focus of Kim *et al.* [71], where the notion of "restart" was introduced into the RW model as a probabilistic measure reflecting the similarity between the graph nodes. Extensions of the above-mentioned method have recently appeared [72] to process dynamic graphs as well as to segment sequence of similar images [73]. Finally, there are still RW variants built to take advantage of natural observations and biological metaphors. The methodology described by Bampis and colleagues [48], [74] is a good representative of this kind of approach, where the RW is related to specific mathematical models of diseases [75]. The Grow-Cut [76] is another seeded segmentation method inspired on biological processes, as it simulates the grow and struggle for domination of the unseeded pixels by bacteria colonies, previously labeled by the user as a certain figurative type of "bacterial" seed.

Despite the good properties, high usability and ease of coding, methods derived from the classic RW formulation are prone to present an isotropic behavior when capturing object boundaries, leading to "flat" solutions, as previously shown in [24], [77]. Notice that our energy model shares many of the good properties of RW such as uniqueness of solution and a formulation given by a quadratic minimization problem while still propagating the seeds more smoothly (see Fig. 1a).

Watersheds. The core idea of *Watersheds/Maximum Spanning Forest* algorithms (MSF) is to represent the image objects as "catchment basins", performing the segmentation from these *basins* and their watershed lines (points equally likely to assume more than one minimum). Given a seeded image, trees are computed from the connected seed components by spanning the nodes of the weighted graph so that the set of trees (a forest) will be maximum w.r.t. weight intensities. The resulting segmentation is called a *Watershed* if the seeds correspond to the maxima. In practice, optimal spanning forests can be efficiently computed by the classic *Kruskal's* [78] or *Prim's* [79] algorithms.

Various watershed-based approaches have gained a lot of popularity in recent years [40], [50], [80], [81], being the unified framework *Power Watershed* (PWS) [40] one of the most influential methods for seeded segmentation due to mathematical links established between several popular clustering algorithms, represented as a common energy function. Other theoretical studies involving the PWS appeared recently in [82], [83]. Although watersheds are very traditional in the computer vision literature, they are not quite efficient in fitting objects where the gradient is locally irregular [24]. Also, reaching a trade-off between over-segmentation and high accuracy is another hurdle [84], as watersheds may suffer from drops and the degradation of the solution on the plateaus of the weight function [40].

Shortest Paths. The goal of *Shortest Paths/Geodesic* algorithms (SP) is to minimize the sum of the edge weights for geodesic paths connecting two vertices. More specifically, the segmentation problem is formulated in the sense of searching the minimum cost paths between two vertices so that the optimal solution can be effectively constructed from the given seeds. A typical representative of SP is the method proposed by Falcão *et al.* [52], [85], called Image Foresting Transform (IFT). This technique accomplishes the image partitioning by finding paths of minimum cost between seeded nodes while still allowing for user intervention to tune the edge weights. The use of IFT for high-precision interactive segmentation has been reported in previous studies [86], [87]. Another typical Geodesic-based method is the one proposed by Bai & Sapiro [54], [55], where the pixel labeling is accomplished by computing the shorter weighted path from the target pixel to the fore/background seeds. The energy function solved by this method is very attractive in terms of computational cost, but it highly depends on the fore/background color model designated to assign the probabilities to the pixels [88]. Considering the diversity of ways in traversing a graph and the good metrification properties of SP-inspired algorithms, new geodesic distances have been reported [89], [90], [91], which also perform well in terms of time complexity. However, these methods are highly influenced by the location where the seeds are placed [45].

Other Approaches (Deep Learning). Deep learning-based approaches for seeded segmentation combine either seed maps or dense ground-truth data, or both with solid label propagators such as RW, GC or Watersheds to train Deep Convolutional Neural Networks (CNNs). This is the case of [92], where seeds and label masks are taken to predict edge weights of image-derived graphs, used subsequently by the classic RW model for label propagation. Similarly, [93] trains a CNN to diffuse, via RW, the seeds to the unlabeled pixels to get dense image labels. Such a training process was also deployed in [94], but applying the Watershed framework instead. Since Watershed is a very popular segmentation method, it has been the basis of modern applications that rely on CNNs [95], [96]. Going deeper on CNNs, new interactive schemes have been proposed as part of deep learning models, as in [97], where the GC partitioner is tuned with distance maps and user-provided clicks, and in [98], where the targets are marked by loose rectangles. Finally, CNNs have even been trained to automatically mimic user-given trimaps, as in [99], where a spectral matting problem [17] is solved to produce soft segments, a set of multiple layers commonly utilized in image editing softwares [100]. Despite the recent advances of deep learning for interactive segmentation, CNN-based architectures require large amounts of data to successfully train the network parameters [101]. Another issue faced by CNNs is that they highly depend on dense ground-truth masks [92], and are computationally expensive, making interactive feedback unfeasible to the users [102]. Finally, as CNNs usually apply gradient descendent to optimize the parameters, differentiability is crucial in most cases [92].

As stated by Couprie *et al.* [40] and Cerrone *et al.* [92], a common characteristic of most visited groups of segmentation methods is that they minimize the same energy function, whose formulation takes into account the first-order pairwise pixels, differing only in terms of a set of exponent values. In contrast, the LC energy model introduced in this paper relies on the minimization of a weighted average of neighbor pixels, which leads to a better contour adherence of the segments. Moreover, the LC formulation admits a unique solution, which can be quickly computed by solving a sparse linear system. Further, the method can be extended to segment large images at iterative rates, in addition to addressing multiple segmentation targets. Finally, we observed that the LC energy is robust w.r.t. the seed placement, and it works well even for very elementary edge weight functions. This set of traits is hard to be concomitantly found in any other seed-based method.

In the following, we detail the technical aspects of the Laplacian Coordinates framework.

3 Laplacian Coordinates Framework

3.1 Affinity Graph Setup

Let I be a color or grayscale image. To compute the LC energy, we define a weighted graph $G = (V, E, W_E)$, where V is the set of nodes whose element i corresponds to the image pixel $P_i \in I$, E is the edges set linking pixels that are neighbors in an 8-connected lattice, and W_E is the set of weights associated to the edges. The local neighborhood set $N(i) = \{j : (i, j) \in E\}$ returns the list of nodes j that share an edge with node i, while $d_i = \sum_{j \in N(i)} w_{ij}$ accounts for the weighted valency of i.

There are many different ways to determine W_E, which includes pixel intensity, gradient, scalability and contour-based strategies [70], [103], [104], [105]. Aiming at keeping our framework as simple as possible, specially free of specific feature settings and tweaks, we only consider the local range of the pixel intensities I_i and I_j in both RGB and Lab components, as concatenated color vectors to determine the weights $w_{ij} \in W_E$. More precisely, w_{ij} is calculated as follows:

$$w_{ij} = \exp\left(-\frac{\beta\|I_i - I_j\|_\infty^2}{\sigma}\right), \sigma = \max_{(i,j)\in E} \|I_i - I_j\|_\infty, \tag{3}$$

where β is a tuning constant. Notice that the weights are positive and symmetric in the sense that $w_{ij} = w_{ji}$. In practice, a small constant $\epsilon = 10^{-6}$ is added into Eq. (3) to avoid null weights, as suggested by Grady [106].

3.2 Soft-Constrained Laplacian Coordinates

Given the background (B) and foreground (F) seeded pixels and their corresponding scalar labels x_B and x_F, the following energy function is minimized:

$$E_s(\mathbf{x}) = \sum_{i\in B}\|x_i - x_B\|_2^2 + \sum_{i\in F}\|x_i - x_F\|_2^2 + \sum_{i\in V}\left\|d_i x_i - \sum_{j\in N(i)} w_{ij}x_j\right\|_2^2, \tag{4}$$

where $\mathbf{x} = (x_1, x_2, \ldots, x_n)$ is the sought solution, that is, the scalar values assigned to the pixels $(P_1, P_2, \ldots, P_n)$ so as to minimize $E_s(\mathbf{x})$, and w_{ij} is computed as in Eq. (3).

Notice from Eq. (4) that the values x_i, $i \in S := B \cup F$ are enforced to be as close as possible to the reference values x_F and x_B in a soft manner, avoiding to strictly impose hard constraints to the optimization problem. Laplacian Coordinates Energy (4) is made up of two core components, one accounting for the soft constraints imposed by the seeds in B and F, called *fidelity term*, and a second component controlling the label spread in the neighborhood of each pixel, called *LC energy term*. In matricial form, the LC energy term can be rewritten as follows:

$$\sum_{i \in V} \left\| d_i x_i - \sum_{j \in N(i)} w_{ij} x_j \right\|_2^2 = \mathbf{x}^{\mathrm{T}} \mathbf{L}^2 \mathbf{x} = \|\mathbf{L}\mathbf{x}\|_2^2, \tag{5}$$

where $\mathbf{L} = \mathbf{D} - \mathbf{W}$ is the *graph Laplacian matrix*, $\mathbf{D}$ is the diagonal matrix $D_{ii} = d_i$, and $\mathbf{W}$ denotes the weighted adjacency matrix, i.e.,

$$W_{ij} = \begin{cases} w_{ij}, & \text{if } (i,j) \in E \\ 0, & \text{otherwise} \end{cases}. \tag{6}$$

Notice that each row in $\mathbf{Lx}$ corresponds to the *differential* (or average) *operator*

$$\delta_i = x_i - \frac{1}{d_i} \sum_{j \in N(i)} w_{ij} x_j, \text{ that is, } (\mathbf{Lx})_i = d_i \delta_i\,. \tag{7}$$

In less mathematical terms, δ_i measures how much each node deviates from the weighted average of its neighbors.

3.2.1 Soft-Constrained Energy Minimization

Let us denote by N_B and N_F the number of pixels marked as background and foreground in the image. Eq. (4) can be rewritten in a more general matrix form as follows [29]:

$$E_s(\mathbf{x}) = \mathbf{x}^{\mathrm{T}}(\mathbf{I}_S + \mathbf{L}^2)\mathbf{x} - 2\mathbf{x}^{t}\mathbf{b} + c, \tag{8}$$

where $\mathbf{I}_S$ is a diagonal matrix such that $I_{S_{ii}} = 1$, $i \in S$, and zero, otherwise, $\mathbf{b}$ is the vector whose $b_i = x_B, i \in B$, $b_i = x_F, i \in F$, and zero, otherwise, and $c = N_B x_B^2 + N_F x_F^2$ is a real constant. It is clear that Eq. (8) is a quadratic form, therefore it has a unique minimizer since $(\mathbf{I}_S + \mathbf{L}^2)$ is symmetric and positive definite (see the Appendix B, which can be found on the Computer Society Digital Library at http://doi.ieeecomputersociety.org/10.1109/TPAMI.2020.2974475, for the proof). Moreover, its minimizer vector $\widetilde{\mathbf{x}}$ is the solution of the following linear system [107]:

$$(\mathbf{I}_S + \mathbf{L}^2)\widetilde{\mathbf{x}} = \mathbf{b}. \tag{9}$$

Therefore, minimizing $E_s(\mathbf{x})$ is equivalent to solving the linear system (9), which, in turn, holds quite attractive properties such as symmetry, positive definiteness and sparsity.

3.2.2 Performing the Segmentation

Once the Energy (4) is minimized, the segmentation can be reached by applying a conventional clustering algorithm such as *K-Means*, *Otsu* or other thresholding schemes to the values x_i. For instance, assuming that $x_F > x_B$, a straightforward way to obtain partitions is to assign fore/background labels $y_i \in \{x_B, x_F\}, i \in V$, according to the following segmentation criterium:

$$y_i = \begin{cases} x_F, & \text{if } x_i \geq \dfrac{x_B + x_F}{2} \\ x_B, & \text{otherwise} \end{cases}. \tag{10}$$

3.3 Hard-Constrained Laplacian Coordinates

In Eq. (4), x_i, $i \in S$, are *soft constraints* in the sense that the solution $\widetilde{\mathbf{x}}$ does not strictly impose $x_i = x_B$ and $x_i = x_F$. To hold such equalities, i.e., *hard constraints*, we rewrite the LC Energy (4) as follows:

$$\begin{aligned} \underset{\mathbf{x}}{\text{minimize}} \quad & E_h(\mathbf{x}) = \mathbf{x}^{\mathrm{T}}\mathbf{L}^2\mathbf{x} \\ \text{subject to} \quad & x_i = x_B,\ i \in B \\ & x_i = x_F,\ i \in F. \end{aligned} \tag{11}$$

3.3.1 Hard-Constrained Energy Minimization

To minimize Eq. (11), we need to adequately group the labeled nodes apart from the un-labelled ones (unknown variables). This can be efficiently done by reordering the rows and columns of $\mathbf{L}$ by using the row-column permutation $\mathbf{PLP}^{\mathrm{T}}$, which preserves the topological relationships of the input Laplacian matrix $\mathbf{L}$

$$\begin{aligned} \mathbf{PLP}^{\mathrm{T}} &= \begin{bmatrix} \mathbf{D}_S & 0 \\ 0 & \mathbf{D}_U \end{bmatrix} - \begin{bmatrix} \mathbf{W}_S & \mathbf{R} \\ \mathbf{R}^{\mathrm{T}} & \mathbf{W}_U \end{bmatrix} \\ &= \begin{bmatrix} \mathbf{L}_S & -\mathbf{R} \\ -\mathbf{R}^{\mathrm{T}} & \mathbf{L}_U \end{bmatrix}, \end{aligned} \tag{12}$$

where $\mathbf{P}$ denotes the resulting permutation matrix so that Eq. (12) holds, and $U = S^c = (B \cup F)^c$, where ^{c} stands for the complementary set, gathering the un-labeled nodes. Notice that both submatrices $\mathbf{L}_S$ and $\mathbf{L}_U$ are symmetric, since the Laplacian matrix $\mathbf{L}$ is symmetric. Moreover, matrix $\mathbf{R}$ stands for the weighted adjacency matrix whose rows represent the nodes marked as seeds, and columns the un-labeled ones.

From Eqs. (11) and (12), $E_h(\mathbf{x})$ becomes

$$E_h(\mathbf{x}) = \left[\mathbf{x}_S^{\mathrm{T}}\ \mathbf{x}_U^{\mathrm{T}}\right] \begin{bmatrix} \mathbf{L}_S & -\mathbf{R} \\ -\mathbf{R}^{\mathrm{T}} & \mathbf{L}_U \end{bmatrix}^{\mathrm{T}} \begin{bmatrix} \mathbf{L}_S & -\mathbf{R} \\ -\mathbf{R}^{\mathrm{T}} & \mathbf{L}_U \end{bmatrix} \begin{bmatrix} \mathbf{x}_S \\ \mathbf{x}_U \end{bmatrix}, \tag{13}$$

where $\mathbf{x} = \begin{bmatrix} \mathbf{x}_S \\ \mathbf{x}_U \end{bmatrix}$, $\mathbf{x}_S = \left[(x_B)_{1 \times N_B}\ (x_F)_{1 \times N_F}\right]^{\mathrm{T}}$, while $\mathbf{x}_U$ accounts for the unknown nodes.

Expanding Eq. (13), the energy function is reduced to

$$\begin{aligned} E_h(\mathbf{x}) = {} & \mathbf{x}_S^{\mathrm{T}}\mathbf{L}_S^{\mathrm{T}}\mathbf{L}_S\mathbf{x}_S - \mathbf{x}_S^{\mathrm{T}}\mathbf{L}_S^{\mathrm{T}}\mathbf{R}\mathbf{x}_U - \mathbf{x}_U^{\mathrm{T}}\mathbf{R}^{\mathrm{T}}\mathbf{L}_S\mathbf{x}_S \\ & + \mathbf{x}_U^{\mathrm{T}}\mathbf{R}^{\mathrm{T}}\mathbf{R}\mathbf{x}_U + \mathbf{x}_S^{\mathrm{T}}\mathbf{R}\mathbf{R}^{\mathrm{T}}\mathbf{x}_S - \mathbf{x}_S^{\mathrm{T}}\mathbf{R}\mathbf{L}_U\mathbf{x}_U \\ & - \mathbf{x}_U^{\mathrm{T}}\mathbf{L}_U^{\mathrm{T}}\mathbf{R}^{\mathrm{T}}\mathbf{x}_S + \mathbf{x}_U^{\mathrm{T}}\mathbf{L}_U^2\mathbf{x}_U \\ = {} & \mathbf{x}_S^{\mathrm{T}}(\mathbf{L}_S^2 + \mathbf{R}\mathbf{R}^{\mathrm{T}})\mathbf{x}_S - \mathbf{x}_S^{\mathrm{T}}(\mathbf{L}_S^{\mathrm{T}}\mathbf{R} + \mathbf{R}\mathbf{L}_U)\mathbf{x}_U \\ & - \mathbf{x}_U^{\mathrm{T}}(\mathbf{R}^{\mathrm{T}}\mathbf{L}_S + \mathbf{L}_U^{\mathrm{T}}\mathbf{R}^{\mathrm{T}})\mathbf{x}_S + \mathbf{x}_U^{\mathrm{T}}(\mathbf{R}^{\mathrm{T}}\mathbf{R} + \mathbf{L}_U^2)\mathbf{x}_U \\ = {} & \mathbf{x}_S^{\mathrm{T}}(\mathbf{L}_S^2 + \mathbf{R}\mathbf{R}^{\mathrm{T}})\mathbf{x}_S - 2\mathbf{x}_U^{\mathrm{T}}(\mathbf{R}^{\mathrm{T}}\mathbf{L}_S + \mathbf{L}_U^{\mathrm{T}}\mathbf{R}^{\mathrm{T}})\mathbf{x}_S \\ & + \mathbf{x}_U^{\mathrm{T}}(\mathbf{R}^{\mathrm{T}}\mathbf{R} + \mathbf{L}_U^2)\mathbf{x}_U. \end{aligned} \tag{14}$$

Again, one can observe that the resulting LC Energy (14) is quadratic. Our goal is to find the global minimum of Eq. (14), which corresponds to the solution of

$$\frac{\partial E_h}{\partial \mathbf{x}_U} = 0 \Rightarrow (\mathbf{R}^{\mathrm{T}}\mathbf{R} + \mathbf{L}_U^2)\mathbf{x}_U = (\mathbf{R}^{\mathrm{T}}\mathbf{L}_S + \mathbf{L}_U^{\mathrm{T}}\mathbf{R}^{\mathrm{T}})\mathbf{x}_S\,. \tag{15}$$

Eq. (15) remains sparse and admits unique solution in $\mathbf{x}_U$ if the associated affinity graph is connected and $B \cup F \neq \emptyset$ (see the Supplementary Material - Appendix A, available online).

After solving Eq. (15), the segmentation is obtained by labeling the un-labeled pixels as in the soft case.

3.4 Basic Features and Behavioral Properties

Besides being mathematically simple and ensuring a unique solution, Laplacian Coordinates has additional properties that render it suitable to segment images as discussed below.

3.4.1 Smoothness × Graph Laplacian Variants

Laplacian Coordinates mainly differs from other typical segmentation methods w.r.t. capability of propagating the seeds more smoothly. In Fig. 1a, we plot the LC solution against the one given by the RW for line graphs with unitary weights, where the seeds are marked in yellow and purple. One may notice that the LC energy tends to smoothly decrease while the RW diffuses the labels linearly. Such a behavior comes from the use of $\mathbf{L}^2$ operator in the LC energy instead of the standard graph Laplacian $\mathbf{L}$, as usually taken by other energy-based models. Moreover, as the combinatorial (graph) formulation of the well-known Laplace equation encodes the canonical graph Laplacian [21], it is not surprising that the $\mathbf{L}^2$ operator is strictly related to the *biharmonic equation*, a higher-order differential formulation which leads to smoother solutions as shown in other related fields [108], [109].

There are still other graph Laplacian variants besides $\mathbf{L}^2$, such as the one proposed by Levin *et al.* [17], named *matting Laplacian matrix*: a generalization that preserves several useful characteristics of the standard graph Laplacian [17], [99]. Just like the *matting Laplacian*, in our LC energy, the $\mathbf{L}^2$ operator extends the graph Laplacian to keep its good properties such as symmetry and positive-semidefiniteness. Despite the common facets of both *matting* [17] and our *squared* Laplacian-derived formulation, they differ substantially in the way of handling the energy functions as well as the nature of the problem: while the former solves a spectral segmentation problem for natural image matting, the latter minimizes an energy function in one shot to interactively segment the image.

3.4.2 Explicit Solution × Locally Extended Neighbors

An interesting interpretation of Laplacian Coordinates is that the solution x_i for each node i is written not in terms of the first-order neighbors, but rather involves more distant neighbors. Mathematically, for a non-labeled pixel P_i, the following equation holds:

$$(\mathbf{Lx})_i = \frac{1}{d_i}\sum_{j\in N(i)} w_{ij}(\mathbf{Lx})_j\,, \tag{16}$$

where $(\mathbf{Lx})_i$ is computed as in Eq. (7). The solution x_i is then mathematically expressed by

$$\begin{aligned} x_i &= \frac{1}{d_i}\sum_{j\in N(i)} w_{ij}\left(x_j + \frac{\delta_j}{d_i}\right) \\ &= \frac{1}{d_i}\sum_{j\in N(i)} w_{ij}x_j + \frac{1}{d_i^2}\sum_{j\in N(i)} w_{ij}\left(\sum_{p\in N(j)} w_{jp}(x_j - x_p)\right). \end{aligned} \tag{17}$$

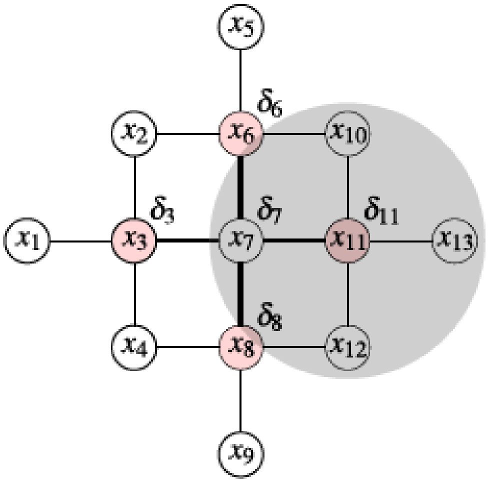


Fig. 2. Solution x_i (for $i = 7$) in terms of its local neighborhood nodes. The gray circle illustrates the points used to compute the differential coordinate δ_j at node $j = 11$.

The solution x_i involves both the weighted average of the first-order neighbors $j \in N(i)$ and the derivatives δ_j in the second-order neighbors (see Fig. 2). It differs from other well-known solutions given uniquely in terms of the first-order neighbors as, for example, the one computed from the standard graph Laplacian by RW. The resulting solution is largely region-expandable and can be said to be anisotropic *in behavior* [110], as the derivatives δ_j in Eq. (17) capture the image pixel discrepancies that are away from the central node as well. Therefore, the information coming from the constraints is better diffused by the Laplacian Coordinates to distant pixels while still ensuring smoothness to the segmentation solution, as previously discussed.

3.4.3 Multiple-Region Segmentation

Laplacian Coordinates can be easily extended to segment images into several parts, in both soft and hard modalities. Considering the soft solution given by Eq. (9), multiple segmentations can be reached by simply solving $(N-1)$ systems of linear equations, similar to the binary case:

$$(\mathbf{I_S} + \mathbf{L}^2)\mathbf{x}^{(\mathbf{j})} = \mathbf{b}^{(\mathbf{j})}, \tag{18}$$

but setting $I_{S_{ii}} = 1$ for all seeded pixels in the image, and specifying different $\mathbf{b}^{(\mathbf{j})}$ for each one of the given labels j, $1 \le j \le (N-1)$, whose indices are ranging in K_j, the set of the seeded nodes $K_j \in K = \bigcup_{i=1}^{N} K_i$. More specifically, let C be a positive constant. We set $b_i^{(j)} = C$, $i \in K_j$, $b_i^{(j)} = -C$, $i \in K \backslash K_j$, zero, otherwise.

Assuming that all $\mathbf{x}^{(\mathbf{j})}$ are bounded by $[-C, C]$, the last scalar map $\mathbf{x}^{(\mathbf{N})}$ is then obtained as follows:

$$x_i^{(N)} = C - \max_{1\le j\le (N-1)} \{x_i^{(j)}\}\,. \tag{19}$$

Finally, for each $j : 1 \le j \le N$, the segmentation $y^{(j)}$ (i.e., a binary image) is performed by

$$\mathbf{y}^{(\mathbf{j})} = \bigcap_{\substack{p=1,\ldots,N \\ p\neq j}} (\mathbf{x}^{(\mathbf{j})} > \mathbf{x}^{(\mathbf{p})})\,, \tag{20}$$

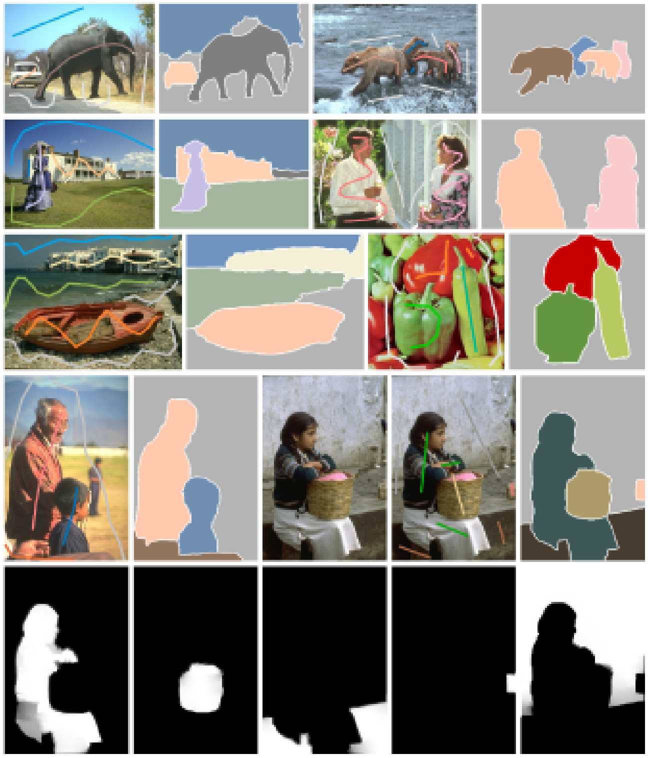


Fig. 3. The use of Eq. (21) for multiple segmentation. First to fourth rows: multiple seeds are sketched as colored strokes, from which Laplacian Coordinates produces the segmented regions. Bottom row: the five computed solutions $\mathbf{x}^{(j)}$ that give rise to the multiple partitions $\mathbf{y}^{(j)}$ merged in the woman's segmentation.

where the symbol $>$ is computed for all pixels of the image.

Similarly, for the hard constrained multiple segmentation, the following set of sparse linear systems must be solved:

$$(\mathbf{R}^{\mathrm{T}}\mathbf{R} + \mathbf{L}_{\mathrm{U}}^{2})\mathbf{x}_{\mathrm{U}}^{(\mathrm{j})} = (\mathbf{R}^{\mathrm{T}}\mathbf{L}_{\mathrm{S}} + \mathbf{L}_{\mathrm{U}}^{\mathrm{T}}\mathbf{R}^{\mathrm{T}})\mathbf{x}_{\mathrm{S}}^{(\mathrm{j})}\,. \tag{21}$$

Eq. (21) derives from the binary case as in Eq. (15). Moreover, matrices involved in Eq. (21) only require handling the full list of the seeded pixels, i.e., none of the particular sets of the user-prescribed seeds K_j are necessary to determine the left-hand matrix in Eq. (21). The individual sets of seeds K_j are later imposed into $\mathbf{x}_{\mathrm{S}}^{(\mathrm{j})}$, by setting $x_{\mathrm{S_i}}^{(j)} = C$, $i \in K_j$, $x_{\mathrm{S_i}}^{(j)} = -C$, $i \in K \backslash K_j$, zero, otherwise (see Fig. 3). Finally, to compute the last scalar mapping $\mathbf{x}^{(\mathrm{N})}$ as well as the outputs $\mathbf{y}^{(\mathrm{j})}$, we proceed similarly to the soft case.

3.4.4 Seeding Flexibility and Adaptability

In Fig. 4, we illustrate the robustness of the Laplacian Coordinates in producing different segmentations just by marking distinct objects in the image. From the three input configurations in Fig. 4 (left, middle and right columns), multiple targets can be accurately segmented by simply seeding them from both soft and hard LC models.

4 Laplacian Coordinates Superpixel Segmentation

We also introduce a superpixel version of Laplacian Coordinates designed to operate on large digital images. As the matrices in Eqs. (9) and (15) are fairly sparse, the LC-based models perform the segmentation at interactive rates when handling images of moderate size (~ 2 seconds for 600 × 600 images). However, the computational time can grow for large images. To reduce the computational burden when dealing with large images, we have adapted the Laplacian Coordinates framework to deal with superpixel domains. The *Superpixel Laplacian Coordinates* (SPLC) performs segmentations at interactive frame rates while providing competitive results as to segmentation quality and accuracy when compared against the regular pixel-based version.

In the following, we detail the basic steps of the proposed SPLC segmentation approach.

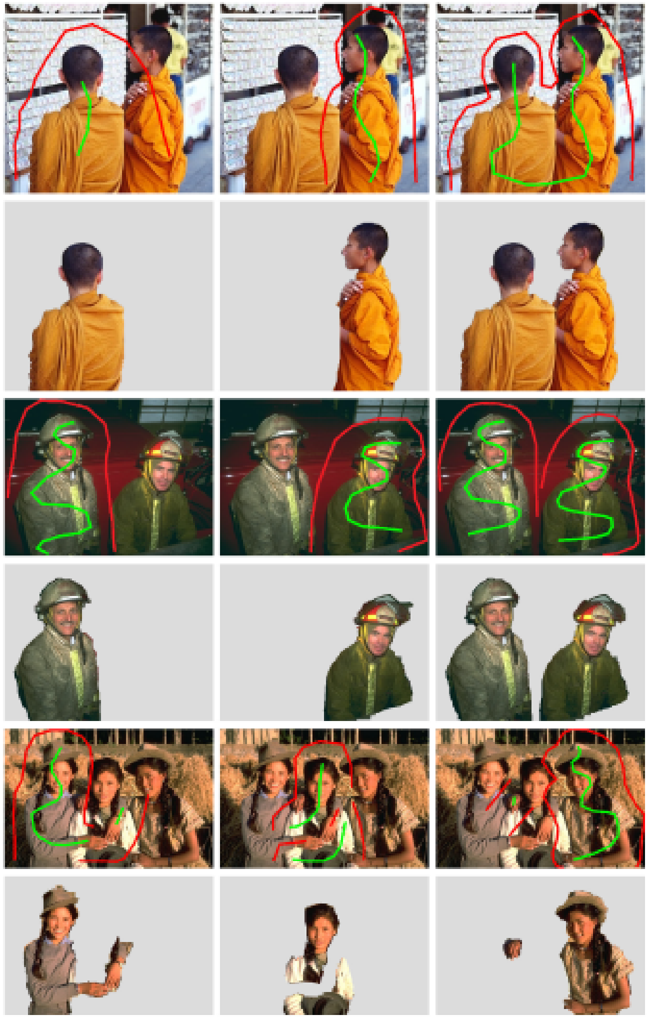


Fig. 4. Selecting different targets by exploiting the seeding flexibility of Laplacian Coordinates. Odd rows: multiple markings are given as input to the soft (first example) and hard (middle and bottom examples) LC modalities.

4.1 Superpixel Initialization

There are several approaches to create an initial superpixel representation of an image. Those approaches vary considerably as to the clustering rules, image descriptors, shape analysis, among other factors. In our approach, we use SLIC superpixels [111], because it is broadly used and it has been proved to be very effective in practice.

Superpixels are marked as *background* or *foreground* depending on the underlying labeling of their pixels. A simple

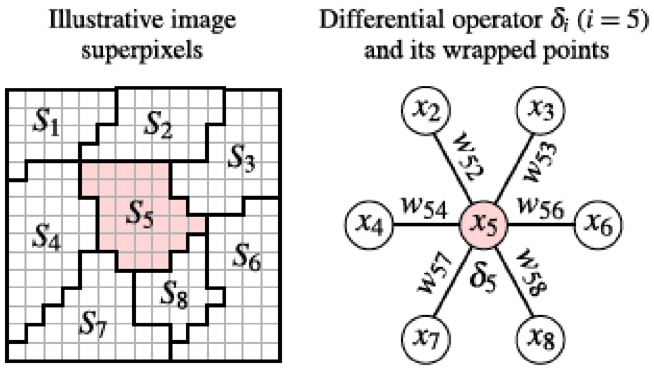


Fig. 5. Superpixel graph representation. (Left) A superpixel partition of the image. (Right) Differential operator δ_i evaluated at node $i = 5$.

voting scheme is used to label a superpixel, that is, if the number of pixels labeled as background is greater than the number of pixels marked as foreground, the superpixel is labeled as background, and vice versa. If a superpixel does not contain any labeled pixel, then it is set as *unknown*.

4.2 SPLC Affinity Graph Setup

Once the seeds are given, we apply the differential operator (7) to the superpixel domain, as illustrated in Fig. 5. In contrast to the pixel-based modality, the number of neighbors $N(i)$ in the SPLC varies beyond 4 or 8 neighbors. In our approach, we consider that two superpixels i and j are adjacent if at least one pixel in i has a pixel of j in its 4-connected neighborhood (e.g., see superpixels in Fig. 5).

The weight function w_{ij} used in SPLC also differs from the pixel-based approach. While in the pixel-based LC the weight depends only on the pixel color attributes, in the superpixel formulation, we take into account both the image colors (as a *feature vector*) and the geometric data, given by the distance between the centroids of the superpixels. As a result, assuming that we have computed the feature vectors f_i and f_j, as well as the centroids c_i and c_j of two adjacent superpixels i and j, the weight w_{ij} assigned to the edge (i, j) is computed as follows:

$$w_{ij} = \exp\left(-\beta_f \| f_i - f_j \|_\infty\right) \exp\left(-\beta_d \| c_i - c_j \|_2\right), \tag{22}$$

where we find β_d over a range $[p, p + \Delta . . q]$ using a simple weight-difference heuristic (see the Supplementary Material - Appendix C, available online). In our experiments, we set $\beta_f = 88$, $p = 0.1$, $q = 0.5$, and $\Delta = 0.01$. For color images, we set as feature vector $f_i = (\overline{R_i}, \overline{G_i}, \overline{B_i})$ in Eq. (22), where $\overline{R_i}$, $\overline{G_i}$ and $\overline{B_i}$ are the arithmetic mean for each color channel *Red* $(\overline{R_i})$, *Green* $(\overline{G_i})$ and *Blue* $(\overline{B_i})$ of the pixels within superpixel i.

4.3 SPLC Segmentation

In the last step of the SPLC, we minimize one of the LC Energies: soft, $E_s(\mathbf{x})$ (8), or hard, $E_h(\mathbf{x})$ (14), but assuming as input the affinity graph resulting from the superpixel model. In either case, the linear system to be solved is sparse with the number of rows and columns given by the number of nodes in the input graph (see Eqs. (9)-(15)).

After solving the linear system corresponding to the chosen energy model (i.e., soft or hard), the foreground superpixels are assigned by using Otsu's thresholding, thereby producing as output the sought segmentation of the image (see Fig. 1d for an illustration).

5 Results and Evaluations

In this section, we provide a comprehensive experimental evaluation of the proposed LC and SPLC methods in the hard mode, referred here as LCH and SPLCH, respectively. We then compare the LCH and SPLCH against nine well-established graph-based approaches for seeded image segmentation, namely Grow Cuts (GRO) [76], One Cut (ONE) [63], Power Watershed (PWS) [40], Maximum Spanning Forest with Kruskal's algorithm (MSFK) [23], [40], Image Foresting Transform with Sub-pixel Precision (IFTS) [86], Random Walker (RW) [21], Normalized Random Walker (NRW) [74], Normalized Lazy Random Walker (NLRW) [48], and the prior Laplacian Coordinates (LC) method [24]. We use publicly available implementations of those methods with default parameters or tuned according to what is reported in the corresponding papers. For the SLIC parameters in SPLCH, we use *compactness* 10, superpixel size 100 and centers initialized in a hexagon distribution.

5.1 Metrics and Datasets

In order to quantitatively assess the segmentation results, we compare the quality of the segmented objects in terms of object/region detection as well as the accuracy in preserving the ground-truth boundaries. For that, we get a number of well-known metrics, namely: Rand Index (*RI*) [112], Variation of Information (*VoI*) [113], Boundary Displacement Error (*BDE*) [114], Dice Coefficient (*Dice*) [40], F-Score [115], and Precision × Recall curves [115]. We performed a battery of experiments employing these metrics to assess the segmentation results in two widely used benchmarks in the context of seed-based image segmentation: the Grabcut Microsoft Research (MSRC) dataset [30], and the BSD dataset [116]. Both benchmark datasets are publicly available for non-commercial purposes. MSRC dataset contains 50 real-world images with corresponding ground-truth segmentations (obtained by manual human labeling) and seeded maps marking the foreground and background regions of the images. BSD database contains a larger number of segmented images (96 real-world scenes) collected from the Berkeley Segmentation dataset [117], most of them bearing patterns that make the segmentation a harder task, as for example the presence of textures, cluttering and different types of lighting conditions.

To go further in our analysis, we also make use of two variants of the MSRC with very sparsely seeded images, as described by Andrade & Carrera [118]. These sets of scribbles, namely here as S1 and S2, allow us to better check the level of assertiveness of the methods in certain pragmatic scenarios where only a limited amount of seeds are available for use. Finally, we also compute the time performance of SPLCH on the INRIA Holidays [119] data set, which contains 500 image groups of large size photos.

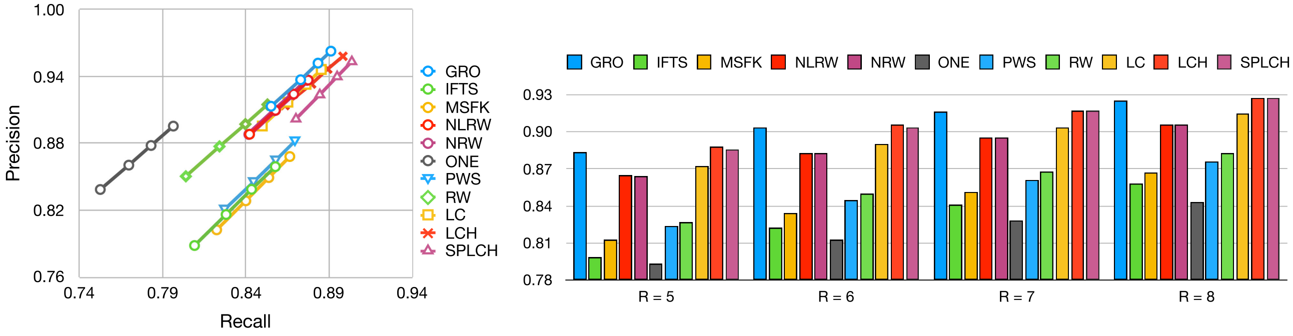
5.2 The Classic MSRC Benchmark

We start our experimental analysis by building Table 1 with some statistics such as *median*, *mean* and *standard deviation* for the metrics RI, VoI, BDE and Dice coefficient when applied to assess the segmentations of the images in the

TABLE 1
RI, VoI, BDE and Dice Scores Obtained from the Original MSRC Dataset for Each Segmentation Method

Method	Rand Index (RI)			Variation of Information (VoI)			Boundary Displ. Error (BDE)			Dice Coefficient (Dice)		
	Median (↑)	Mean (↑)	Std (↓)	Median (↓)	Mean (↓)	Std (↓)	Median (↓)	Mean (↓)	Std (↓)	Median (↑)	Mean (↑)	Std (↓)
GRO	0.9767	0.9742	0.0165	0.1632	0.1725	0.0884	1.9669	3.2309	5.9175	0.9807	0.9760	**0.0184**
IFTS	0.9746	0.9683	0.0189	0.1804	0.2060	0.1014	2.4261	3.3637	2.9908	0.9762	0.9704	0.0215
MSFK	0.9746	0.9689	0.0205	0.1833	0.2012	0.1074	2.1856	3.1917	2.9824	0.9763	0.9714	0.0219
NLRW	0.9770	0.9733	**0.0146**	0.1640	0.1792	**0.0811**	1.9631	2.9225	2.9511	0.9791	0.9742	0.0192
NRW	0.9769	0.9733	0.0147	0.1642	0.1793	0.0812	2.1117	2.8833	**2.9108**	0.9792	0.9741	0.0193
ONE	0.9707	0.9651	0.0237	0.1920	0.2120	0.1181	3.1892	4.1075	3.6868	0.9737	0.9645	0.0303
PWS	0.9761	0.9703	0.0203	0.1745	0.1931	0.1079	2.1909	3.1284	3.0549	0.9776	0.9725	0.0225
RW	0.9745	0.9699	0.0194	0.1738	0.1933	0.1022	2.1888	3.4522	3.5499	0.9784	0.9715	0.0235
LC	0.9774	0.9745	0.0155	0.1561	0.1691	0.0829	1.8768	2.7421	3.0259	0.9813	0.9759	0.0224
LCH	**0.9802**	**0.9752**	0.0159	**0.1437**	**0.1687**	0.0859	**1.8551**	**2.5852**	2.9124	**0.9822**	**0.9763**	0.0220
SPLCH	**0.9778**	**0.9764**	**0.0133**	**0.1559**	**0.1611**	**0.0726**	**1.7730**	**2.3079**	**1.9647**	**0.9825**	**0.9770**	**0.0182**

Values in bold indicate the best scores while the red ones indicate the second best.



(a) Precision-Recall curves

(b) F-score

Fig. 6. Quantitative evaluation concerning the boundary-based metrics Recall, Precision and F-measure score over different neighborhood radius R [115] for the MSRC dataset with usual seeds.

TABLE 2
RI, VoI, BDE and Dice Scores Obtained from the MSRC With the S1 Seed Set for Each Segmentation Method

Method	Rand Index (RI)			Variation of Information (VoI)			Boundary Displ. Error (BDE)			Dice Coefficient (Dice)		
	Median (↑)	Mean (↑)	Std (↓)	Median (↓)	Mean (↓)	Std (↓)	Median (↓)	Mean (↓)	Std (↓)	Median (↑)	Mean (↑)	Std (↓)
GRO	0.8128	0.8025	0.1061	0.7566	0.7893	**0.3131**	22.4489	26.8187	17.8120	0.8260	0.8203	**0.0862**
IFTS	0.8040	0.7992	0.1166	0.7573	0.7734	0.3544	22.9537	26.2728	16.6349	0.7986	0.8022	0.1067
MSFK	0.8468	0.8064	0.1258	0.6339	0.7388	0.3848	21.4500	23.4021	13.5844	0.8181	0.8147	0.0980
NLRW	0.8882	0.8534	**0.1045**	**0.5022**	0.6039	0.3304	17.1275	19.1723	13.4278	0.8580	0.8522	0.0916
NRW	0.8868	0.8513	0.1087	0.5090	0.6084	0.3341	16.6015	19.0219	**13.2143**	0.8595	0.8498	0.0974
ONE	0.7377	0.7552	0.1345	0.6688	0.6409	**0.2603**	51.5384	56.7088	34.1423	0.5114	0.5993	0.1812
PWS	0.8380	0.8127	0.1239	0.6386	0.7167	0.3829	20.8103	22.6402	14.2947	0.8054	0.8190	0.1070
RW	0.8458	0.8174	0.1266	0.6035	0.6907	0.3802	23.1282	24.7029	16.4130	0.8212	0.8168	0.1075
LC	0.8809	0.8394	0.1189	0.5467	0.6601	0.3774	18.3263	19.9549	14.2795	0.8612	0.8525	0.0987
LCH	**0.8900**	**0.8584**	**0.1015**	0.5555	**0.6022**	0.3462	**15.2866**	**17.4865**	13.2537	**0.8772**	**0.8729**	**0.0911**
SPLCH	**0.8904**	**0.8678**	0.1130	**0.5052**	**0.5566**	0.3622	**13.2476**	**15.3452**	**12.1757**	**0.8803**	**0.8700**	0.1096

Values in bold indicate the best scores while the red ones indicate the second best.

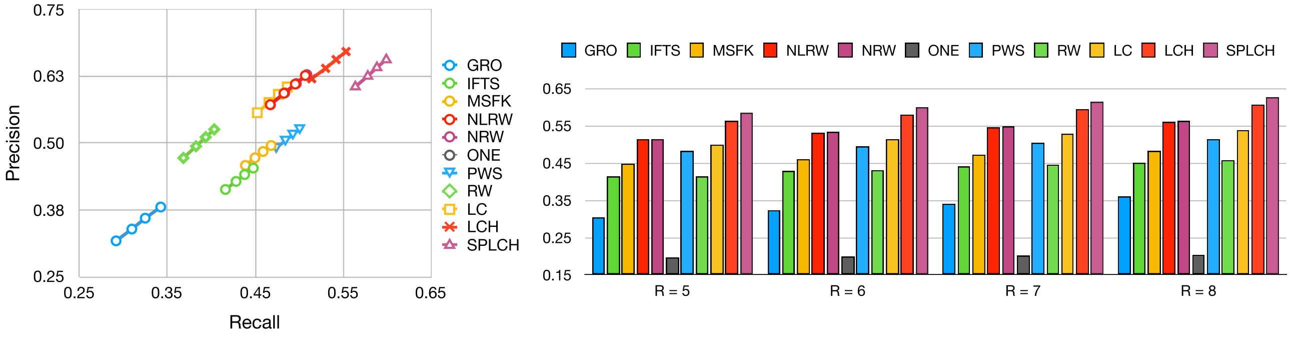
MSRC dataset. From the collected statistics, one can see that both Laplacian Coordinates approaches in the hard versions, LCH and SPLCH, outperform existing competitors in almost all the quality measurements, mainly regarding the median and mean scores. In fact, for these particular summaries, LCH and SPLCH obtain either the best or the second best scores in all numerical verifications.

The Precision-Recall curves plotted in Fig. 6a endorses the previous claim, but now assessing how well a certain segmentation method behaves in terms of boundary fitting capability. Here, the higher the Precision and Recall, the better the method's performance. A composite index that balances the Precision and Recall is the F-score, whose values are exposed in Fig. 6b. LCH, SPLCH, and GRO are the methods that produce higher F-scores in all the four evaluated scenarios.

5.3 MSRC Benchmark With Sparse Seed Maps

A. S1 Seed Set

We also performed a deeper and more comprehensive evaluation of the segmentation techniques when they take as input sets of sparsely seeded annotations. Table 2 summarizes the comparative analysis when using the sparsely seeded of MSRC called S1 set [118]. By analyzing the tabulated scores, one may conclude that LCH and SPLCH lead to better accuracies for the mean and median calculations for most of the validation metrics, attesting their efficiency when handling images with a small portion of annotated pixels. They also yield the best scores, being followed by NLRW and NRW, when one computes boundary adherence criteria, as shown in the Precision-Recall curves in Fig. 7a and F-score computations in Fig. 7b.



(a) Precision-Recall curves

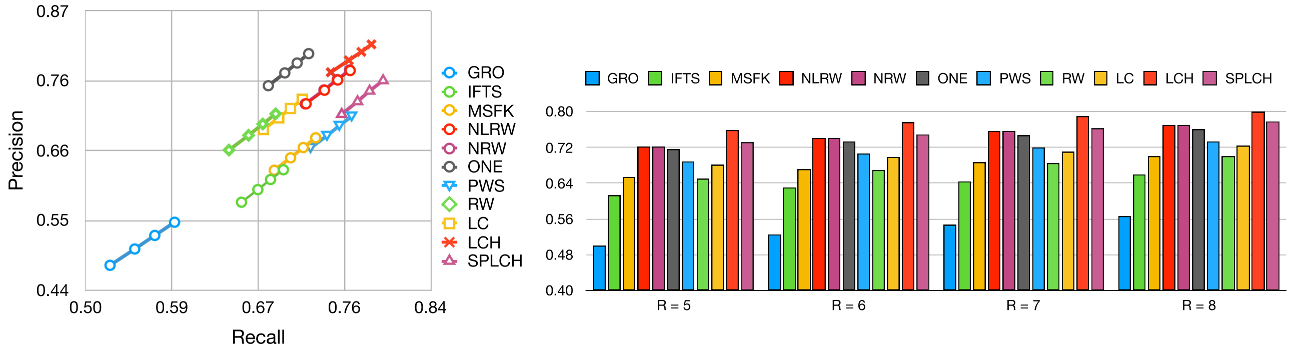
(b) F-score

Fig. 7. Quantitative evaluation concerning the boundary-based metrics Recall, Precision and F-measure score over different neighborhood radius R [115] for the MSRC with the S1 seed set.

TABLE 3
RI, VoI, BDE and Dice Scores Obtained from the MSRC With the S2 Seed Set for Each Segmentation Method

Method	Rand Index (RI)			Variation of Information (VoI)			Boundary Displ. Error (BDE)			Dice Coefficient (Dice)		
	Median (↑)	Mean (↑)	Std (↓)	Median (↓)	Mean (↓)	Std (↓)	Median (↓)	Mean (↓)	Std (↓)	Median (↑)	Mean (↑)	Std (↓)
GRO	0.8585	0.8485	0.0969	0.7057	0.6609	0.3347	14.3006	16.8822	11.1239	0.8892	0.8714	0.0794
IFTS	0.9115	0.8817	0.0962	0.4636	0.5269	0.3228	8.2894	10.3872	7.1862	0.9274	0.8980	0.0825
MSFK	0.9370	0.9021	0.0814	0.3461	0.4602	0.2960	6.1550	8.9817	6.9261	0.9376	0.9156	0.0637
NLRW	0.9471	0.9208	0.0799	0.3114	0.3936	0.2920	4.6857	6.8990	6.3904	0.9527	0.9329	0.0607
NRW	0.9480	0.9205	0.0806	0.3157	0.3941	0.2938	4.7458	6.9122	6.4770	0.9534	0.9327	0.0615
ONE	**0.9642**	0.9286	0.0964	**0.2268**	0.3126	0.2528	4.1616	8.3735	14.1352	**0.9614**	0.9189	0.1199
PWS	0.9461	0.9181	0.0726	0.3085	0.4031	0.2728	5.5394	7.3470	5.5165	0.9505	0.9308	0.0546
RW	0.9309	0.8980	0.0888	0.3665	0.4682	0.3080	7.0661	9.1247	7.7347	0.9375	0.9138	0.0651
LC	0.9319	0.8937	0.0959	0.3623	0.4964	0.3594	6.5975	9.1522	8.1019	0.9389	0.9120	0.0728
LCH	0.9495	0.9300	0.0723	0.3027	0.3102	0.2672	4.0952	6.6815	6.4085	0.9512	0.9392	0.0482
SPLCH	0.9510	**0.9336**	**0.0609**	0.3032	**0.3053**	**0.2509**	**4.0650**	**5.7406**	**4.2043**	0.9562	**0.9442**	**0.0426**

Values in bold indicate the best scores while the red ones indicate the second best.



(a) Precision-Recall curves

(b) F-score

Fig. 8. Quantitative evaluation concerning the boundary-based metrics Recall, Precision and F-measure score over different neighborhood radius R [115] for the MSRC with the S2 seed set.

B. S2 Seed Set

Table 3 and Fig. 8 show another comparison involving the 50 MSRC images, but on a different set of sparse seeds, the S2 set, as previously described in [118]. The proposed LCH and SPLCH approaches surpass other image segmentation methods acquiring higher average and lower std values. For the median, the One Cut algorithm performs better than others in three assessments, however, one may note that LCH and SPLCH give rise to the second-best scores for these particular cases. Considering the boundary preservation criteria, the LCH and SPLCH clearly overcome all the nine competitors, specially the LCH, as one can see by the F-Score values exposed in Fig. 8b.

5.4 The BSD Benchmark

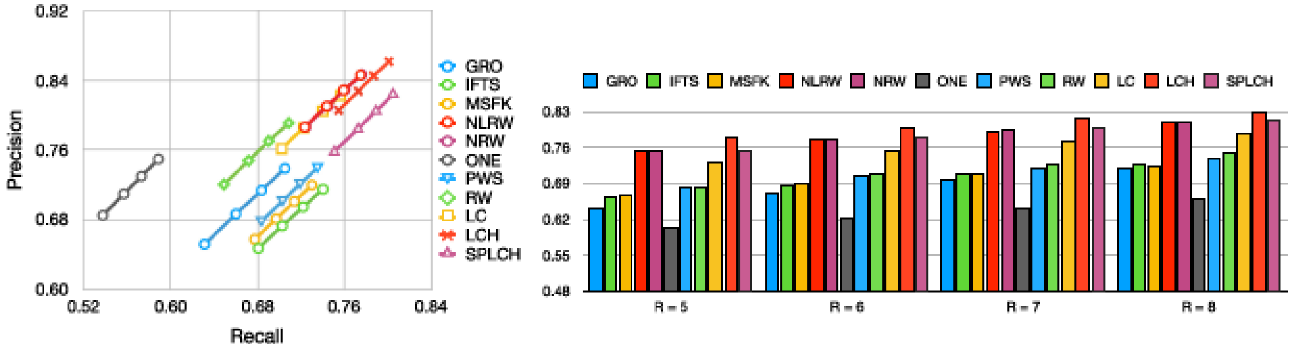
5.4.1 *Quantitative Assessments*

Table 4 shows RI, VoI, BDE, Dice of the eleven analyzed techniques when applied to the classic BSD data set [116]. Notice that the new proposed methods, LCH and SPLCH, resulted in the best mean and std scores for most of the quality metrics. Regarding the median, LCH yields the higher scores, while the SPLCH turns out to be very competitive with the recently released image segmentation methods NRW and NLRW. Similarly, the Recall-Precision curves and the F-score chart displayed in Fig. 9 also emphasize the suitable performance of our LC Hard-based frameworks, but in the sense of quality boundary metrics as well.

TABLE 4
RI, VoI, BDE and Dice Scores Obtained from the BSD Dataset for Each Segmentation Method

Method	Rand Index (RI)			Variation of Information (VoI)			Boundary Displ. Error (BDE)			Dice Coefficient (Dice)		
	Median (↑)	Mean (↑)	Std (↓)	Median (↓)	Mean (↓)	Std (↓)	Median (↓)	Mean (↓)	Std (↓)	Median (↑)	Mean (↑)	Std (↓)
GRO	0.9464	0.9291	0.0689	0.3057	0.3635	0.2476	5.4492	7.6941	10.2368	0.9343	0.9240	0.0707
IFTS	0.9522	0.9334	0.0716	0.3042	0.3400	0.2429	4.7352	6.2361	5.5460	0.9408	0.9257	0.0733
MSFK	0.9538	0.9329	0.0718	0.2880	0.3383	0.2462	4.5209	6.6009	6.8137	0.9435	0.9256	0.0731
NLRW	**0.9691**	0.9560	0.0424	0.2039	0.2597	0.1810	3.3820	4.2788	3.8083	0.9579	0.9496	**0.0314**
NRW	0.9691	0.9561	0.0425	**0.2033**	0.2591	0.1901	**3.3761**	4.2653	3.7948	0.9575	0.9498	0.0315
ONE	0.9530	0.9263	0.0831	0.2816	0.3236	0.2340	6.1044	9.1217	10.0022	0.9280	0.8898	0.1154
PWS	0.9561	0.9391	0.0604	0.2766	0.3199	0.2354	4.4855	5.9641	4.9819	0.9492	0.9341	0.0505
RW	0.9559	0.9446	0.0483	0.2919	0.3018	0.2041	4.5958	5.6490	4.8573	0.9471	0.9381	0.0419
LC	0.9654	0.9498	0.0598	0.2220	0.2735	0.1986	4.0386	5.1452	5.8171	0.9538	0.9429	0.0659
LCH	**0.9720**	**0.9629**	**0.0351**	**0.1889**	**0.2265**	**0.1653**	**3.0905**	**3.8497**	**3.0547**	**0.9671**	**0.9585**	**0.0276**
SPLCH	0.9664	**0.9581**	**0.0297**	0.2201	**0.2548**	**0.1533**	3.7065	**4.0702**	**2.5653**	**0.9589**	**0.9521**	0.0334

Values in bold indicate the best scores while the red ones indicate the second best.



(a) Precision-Recall curves

(b) F-score

Fig. 9. Quantitative evaluation concerning the boundary-based metrics Recall, Precision and F-score over different neighborhood radius R [115] for the BSD dataset.

5.4.2 Qualitative Results

To further examine the performance of the state-of-the-art methods in another light on the BDS dataset, we depict in Fig. 10 the segmentation obtained by the techniques for nine sample images. Overall, many of the methods produce similar results, but there are nuances that deserve to be highlighted. For instance, in the church image, ONE, RW, LC, LCH, and SPLCH produced the best results. When comparing the result of NLR and NLRW in the starfish picture against other methods, one can notice that the three techniques produce good results, but slightly worse than LCH and SPLCH. Comparing the results of proposed approaches in the remaining images, one can clearly notice that LC and SPLC deliver more accurate outputs, giving rise to best contour adherence and full object segmentation.

5.5 Seed Quantity and Placement Invariance

We also evaluate the sensitivity of LCH and SPLCH as to seed quantity and seed placement, following the protocol given in [25], [45], [120]. More precisely, we first take as benchmark the segmentations from the original trimaps as provided by the MSRC dataset. From the input trimaps, seeds are randomly chosen from 50 to 1 percent of total seeds. The segmentations are then computed from the "perturbed" seeds and compared against the ground-truth. The normalized overlap measure $a_0 = \left|\frac{F_1 \cap F_2}{F_1 \cup F_2}\right|$ is applied to check the amount of overlap between the computed and ground-truth segmentations, where F_1 and F_2 account for the set of pixels labeled as foreground in the two segmentations [45].

Table 5 presents the normalized overlap averages when the amount of seeds is reduced to 50, 30, 10 and 1 percent of total input seeds. Notice that for reductions of 50 and 30 percent, the methods produce similar results. However, when the number of seeds is dropped to 10 and 1 percent, both LCH and SPLCH approaches produce considerably better results, thus being less dependent on user-inputs than other methods.

5.6 Superpixel Laplacian Coordinates × High-Resolution Images

We also attested the performance of the superpixel version of the LC method for large size images. We use as benchmark the INRIA Holidays dataset [119],[2] which is formed by several high-resolution photos covering both natural and man-made scenes. Fig. 11 shows the accuracy of SPLCH for different choices of superpixel sizes in six of these images. In our tests, we noticed that satisfactory results are obtained for superpixel sizes ranging from 100 to 600.

The main advantage of the SPLCH is that the segmentation can be done at interactive rates. Fig. 12 presents the timings of the main steps of the SPLCH when applied to segment images from the INRIA dataset. Our approach took less than 1 second to segment an image of size 1600 × 1200, and around 1 second for images with size 2048 × 1536. For images with resolution as large as 2560 × 1920, our approach took approximately 1.5 seconds and, in the scenario with superpixel size of 100 pixels, the method took around 2.3 seconds to segment the image completely.

[2] Publicly available at http://lear.inrialpes.fr/~jegou/data.php

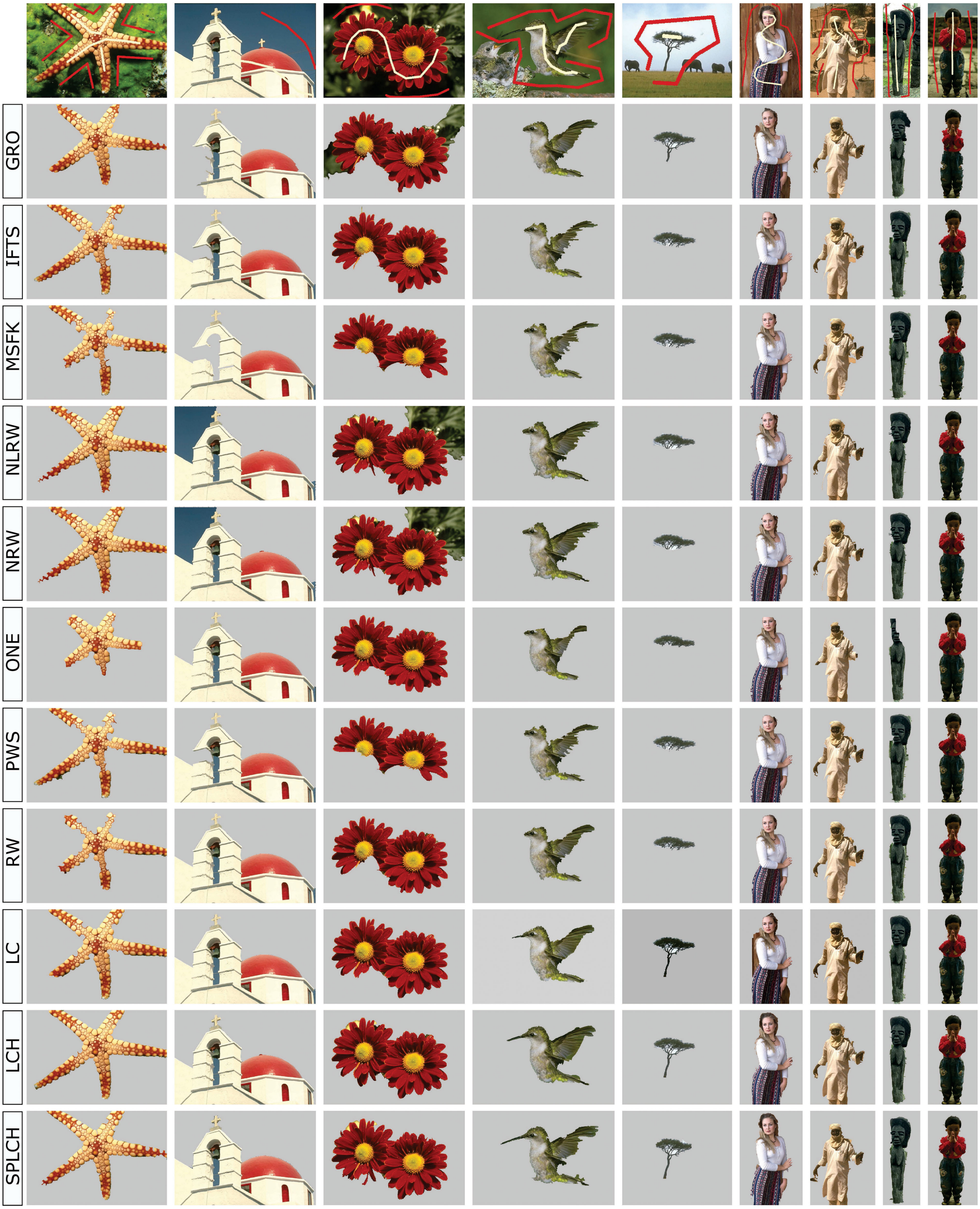


Fig. 10. From top to bottom: sample images + scribbles from BSD dataset, and the outputs produced by each method.

An important aspect that one may take into consideration from Fig. 12 is that the SLIC pre-partitioner [111] is the most time-consuming task of our pipeline and it is not tied to the size of superpixel, but rather to the image resolution. In future versions, other superpixel pre-clusterizers can be used, including hardware-accelerated implementations as well. Finally, it is worth mentioning that, in our prototype, the computation of *Superpixel Graph*, *Features* and the *System Setup* has not been optimized to support parallelism, a feasible architecture which could turn the implementation even faster in practice, and no effective computational gain was found between the pixel and superpixel versions for images

TABLE 5
Sensitivity of the Segmentation Methods w.r.t. Seed Quantity and Seed Placement When Measured by the Average of a_0 Overlap Measure

Method	Perturbed seed quantity			
	50%	30%	10%	1%
GRO	0.9968	0.9938	0.9880	0.9648
IFTS	0.9988	0.9964	0.9867	0.9540
MSFP	0.9969	0.9968	0.9880	0.9565
NLRW	0.9932	0.9932	0.9654	0.9632
NRW	0.9924	0.9932	0.9653	0.9629
ONE	0.9928	0.9914	0.9806	0.8770
PWS	0.9975	0.9963	0.9870	0.9630
RW	0.9984	0.9966	0.9892	0.9630
LC	0.9974	0.9967	0.9886	0.9694
LCH	**0.9992**	**0.9969**	**0.9928**	**0.9815**
SPLCH	0.9972	0.9941	0.9894	0.9700

The percentages regulate the quantity of seeds used to produce the segmentations by the methods. Values in bold indicate the best scores while the red ones indicate the second best.

with small and moderate sizes. We ran our tests on a MacBook Pro Intel Core i7 Quad Core with 2.2 GHz and 16 GB of RAM. Our superpixel prototype was implemented in C++ using the Qt framework with the Eigen Library to solve the linear system, and OpenCV for image routines.

6 Conclusion

In this paper we propose Laplacian Coordinates, a quadratic energy minimization framework which combines simplicity, uniqueness of solution and accuracy into a new seeded image segmentation approach. Our formulation enables freely managing the seeds as both soft or hard constraints into the optimization problem so that the minimizer lies in the solution of a sparse system of linear equations, making the technique mathematically concise and simpler to be coded and run. Moreover, its solution is not prone to be trapped in local minima, on the contrary, it is guaranteed to be globally optimal,

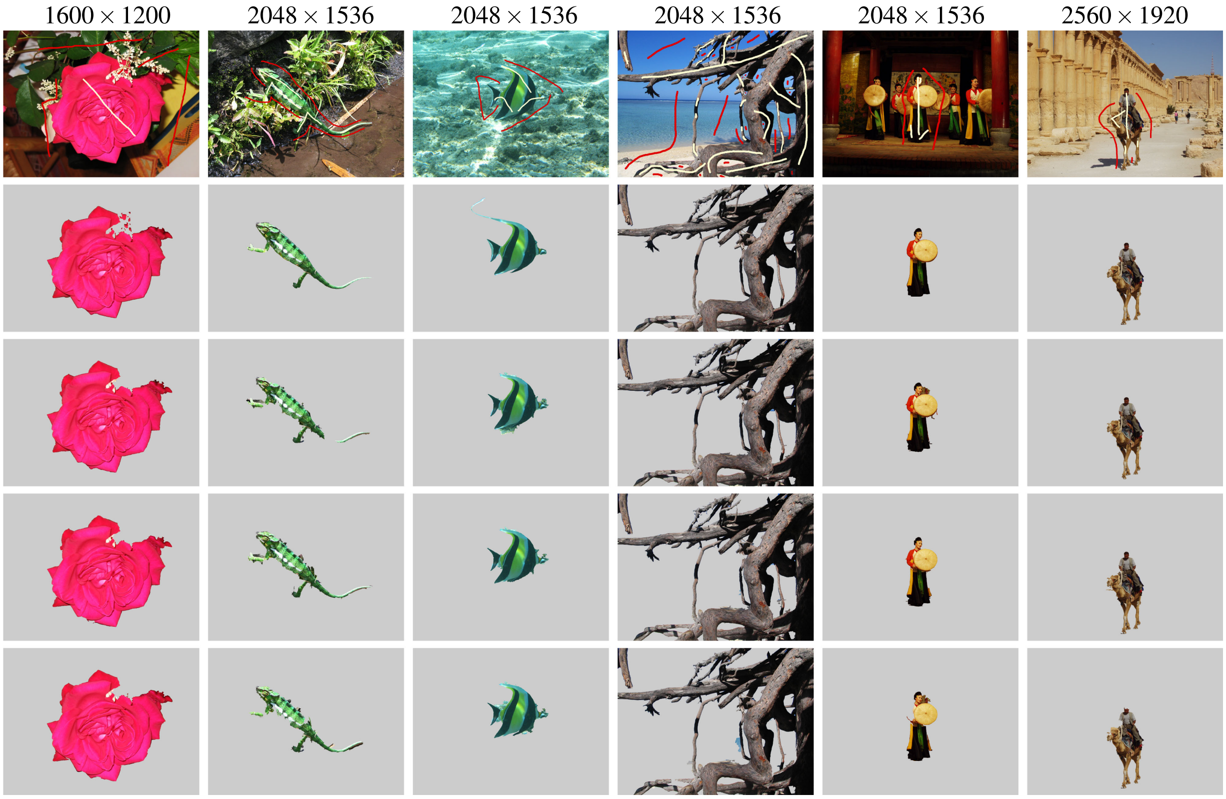


Fig. 11. From top to bottom: Six seeded high-resolution images from the INRIA dataset, ground-truth images, and SPLCH segmentations for superpixels of size 100, 300, and 600, respectively.

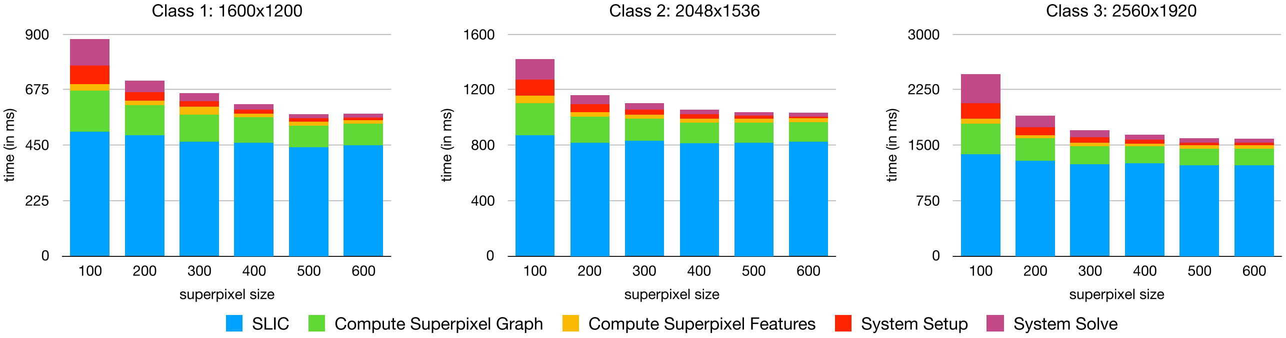


Fig. 12. Time measurements (on average) of the main steps of the Superpixel Laplacian Coordinates.

an important trait not always present in many other image segmentation methods. We also extend the hard version of Laplacian Coordinates to perform superpixel segmentation on high-resolution images at interactive rates, which is another issue not always confronted by several interactive segmentation methods in practice. Our energy model minimizes the average of distances between image nodes instead of the pairwise nodes as other existing methods, which was found experimentally to have high contour adherence and full object segmentation.

We attested the performance of our framework against several state-of-the-art methods, from well-established benchmarks and quantitative measures systematically used in the image segmentation field. The technique remains stable and quite assertive for a great variety of seed sets, including very popular data sets. All those properties render Laplacian Coordinates a useful and compelling graph clustering framework to segment images interactively.

Acknowledgments

The authors acknowledge the Center for Mathematical Sciences Applied to Industry (CeMEAI-FAPESP, grant #2013/07375-0), São Paulo Research Foundation (FAPESP, grant #2014/16857-0), National Council for Scientific and Technological Development (CNPq, grant #2016/04391-2), and CAPES-Brasil (Finance Code 001) for providing resources that contributed greatly to this research.

References

[1] C. T. N. Suzuki, J. F. Gomes, A. X. Falcão, J. P. Papa, and S. Hoshino-Shimizu, "Automatic segmentation and classification of human intestinal parasites from microscopy images," *IEEE Trans. Biomed. Eng.*, vol. 60, no. 3, pp. 803–812, Mar. 2013.

[2] R. Huang *et al.*, "A locally constrained statistical shape model for robust nasal cavity segmentation in computed tomography," in *Proc. IEEE Int. Symp. Biomed. Imag.*, 2016, pp. 1334–1337.

[3] C. Dong, X. Zeng, and L. Lin, "An improved random walker with bayes model for volumetric medical image segmentation," *J. Healthcare Eng.*, vol. 2017, no. 1, pp. 1–11, 2017.

[4] X. Chen and L. Pan, "A survey of graph cuts/graph search based medical image segmentation," *IEEE Rev. Biomed. Eng.*, vol. 11, pp. 112–124, 2018.

[5] J. Garcia-Rodriguez and M. Quevedo, *Robotic Vision: Technologies for Machine Learning and Vision Applications*. London, U.K.: IGC Global, 2012.

[6] W. Casaca, D. Motta, G. Taubin, and L. G. Nonato, "A user-friendly interactive image inpainting framework using Laplacian coordinates," in *Proc. IEEE Int. Conf. Image Process.*, 2015, pp. 862–866.

[7] X. Li, H. Zhao, H. Huang, L. Xiao, Z. Hu, and J. Shao, "Stereoscopic image recoloring," *J. Electron. Imag.*, vol. 25, no. 5, pp. 1–13, 2016.

[8] S. Karayev, M. Fritz, and T. Darrell, "Anytime recognition of objects and scenes," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2014, pp. 572–579.

[9] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 779–788.

[10] L. Dai, J. Yang, L. Chen, and J. Li, "Category-specific object segmentation via unsupervised discriminant shape," *Pattern Recognit.*, vol. 64, pp. 202–214, 2017.

[11] V. Kolmogorov and R. Zabih, "What energy functions can be minimized via graph cuts," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 2, pp. 147–159, Feb. 2004.

[12] A. T. Delong, "Advances in graph-cut optimizaiton: Multi-surface models, label costs, and hierarchical costs," Electronic Thesis and Dissertation Repository, University of Western Ontario, 2011. [Online]. Available: https://ir.lib.uwo.ca/etd/298

[13] K. I. Kim, J. Tompkin, H. Pfister, and C. Theobalt, "Context-guided diffusion for label propagation on graphs," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2015, pp. 2776–2784.

[14] M. Pelillo and E. Hancock, *Energy Minimization Methods in Computer Vision and Pattern Recognition*, vol. 1. Berlin, Germany: Springer, 2018.

[15] F. Chung, *Spectral Graph Theory*. Providence, RI, USA: American Mathematical Society, 1997.

[16] B. Bollobás, *Modern Graph Theory*. New York, NY, USA: Springer, 2002.

[17] A. Rav-Acha, D. Lischinski, and A. Levin, "Spectral matting," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 30, no. 10, pp. 1699–1712, Oct. 2008.

[18] M. C. V. Nascimento and A. C. P. L. F. de Carvalho, "Spectral methods for graph clustering: A survey," *Eur. J. Oper. Res.*, vol. 211, no. 2, pp. 221–231, 2011.

[19] I. Koutis, G. L. Miller, and D. Tolliver, "Combinatorial preconditioners and multilevel solvers for problems in computer vision and image processing," in *Proc. Int. Symp. Vis. Comput.*, 2009, pp. 1067–1078.

[20] R. Tutunov, H. Bou-Ammar, and A. Jadbabaie, "Distributed SDDM solvers: Theory and applications," 2015. [Online]. Available: https://arxiv.org/pdf/1508.04096.pdf

[21] L. Grady, "Random walks for image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 11, pp. 1768–1783, Nov. 2006.

[22] Y. Boykov and G. Funka-Lea, "Graph cuts and efficient N-D image segmentation," *Int. J. Comput. Vis.*, vol. 70, no. 7, pp. 109–131, 2006.

[23] J. Cousty, G. Bertrand, L. Najman, and M. Couprie, "Watershed cuts: Minimum spanning forests and the drop of water principle," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 8, pp. 1362–1374, Aug. 2009.

[24] W. Casaca, L. G. Nonato, and G. Taubin, "Laplacian coordinates for seeded image segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2014, pp. 384–391.

[25] T. Wang, Z. Ji, Q. Sun, Q. Chen, Q. Ge, and J. Yang, "Diffusive likelihood for interactive image segmentation," *Pattern Recognit.*, vol. 1, no. 1, pp. 1–15, 2018.

[26] V. R. P. Borges, M. C. F. D. Oliveira, T. G. Silva, A. A. H. Vieira, and B. Hamann, "Region growing for segmenting green microalgae images," *IEEE/ACM Trans. Comput. Biol. Bioinf.*, vol. 15, no. 1, pp. 257–270, Jan./Feb. 2018.

[27] R. Merris, "Laplacian matrices of graphs: A survey," *Linear Algebra Appl.*, vol. 197/198, pp. 143–176, 1994.

[28] M. Wardetzky, S. Mathur, F. Kalberer, and E. Grinspun, "Discrete laplace operators: No free lunch," in *Proc. 15th Eurographics SGP*, 2007, pp. 33–37.

[29] Wallace Correa de Oliveira Casaca, "Graph laplacian for spectral clustering and seeded image segmentation," Doctoral Thesis, Instituto de Ciências Matemáticas e de Computação, University of São Paulo, University of São Paulo, 2014, doi: 10.11606/T.55.2014.tde-24062015-112215.

[30] C. Rother, V. Kolmogorov, and A. Blake, "GrabCut: Interactive foreground extraction using iterated graph cuts," *ACM Trans. Graph.*, vol. 23, no. 3, pp. 309–314, 2004.

[31] P. Kohli, *Minimizing Dynamic and Higher Order Energy Functions Using Graph Cuts*. LAP LAMBERT Academic Publishing, R: Riga, Latvia, 2010.

[32] F. Yi and I. Moon, "Image segmentation: A survey of graph-cut methods," in *Proc. Int. Conf. Syst. Informat.*, 2012, pp. 1936–1941.

[33] B. Peng, L. Zhang, and D. Zhang, "A survey of graph theoretical approaches to image segmentation," *Pattern Recognit.*, vol. 46, no. 3, pp. 1020–1038, 2013.

[34] K. Suresh and P. S. Rao, "Various image segmentation algorithms: A survey," in *Smart Intelligent Computing and Applications*. Berlin, Germany: Springer, 2019, pp. 233–239.

[35] H. Ramadan and H. Tairi, "Moving object segmentation in video using spatiotemporal saliency and laplacian coordinates," in *Proc. IEEE/ACS Int. Conf. Comput. Syst. Appl.*, 2016, pp. 1–7.

[36] O. C. Linares, J. Bianchi, D. Raveli, J. B. Neto, and B. Hamann, "Mandible and skull segmentation in cone beam computed tomography using super-voxels and graph clustering," *Vis. Comput.*, vol. 1, pp. 1461–1474, 2019.

[37] W. Casaca, M. Colnago, and L. G. Nonato, "Interactive image colorization using laplacian coordinates," in *Proc. Int. Conf. Comput. Anal. Images Patterns*, 2015, pp. 675–686.

[38] B. Liu, W. Wang, J. Cao, and X. Liu, "Interactive mesh cutting with laplace coordinates and gradient," *Commun. Inf. Syst.*, vol. 17, no. 2, pp. 65–83, 2017.

[39] J. Cousty, G. Bertrand, L. Najman, and M. Couprie, "Power watersheds: A new image segmentation framework extending graph cuts, random walker and optimal spanning forest," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2009, pp. 731–738.

[40] C. Couprie, L. Grady, L. Najman, and H. Talbot, "Power watershed: A unifying graph-based optimization framework," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 7, pp. 1384–1399, Jul. 2011.

[41] Y. Li, J. Sun, C. K. Tang, and H. Y. Shum, "Lazy snapping," *ACM Trans. Graph.*, vol. 23, no. 3, pp. 303–308, 2004.

[42] S. Vicente, V. Kolmogorov, and C. Rother, "Graph cut based image segmentation with connectivity priors," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2008, pp. 1–8.

[43] F. Yi and I. Moon, "Image segmentation: A survey of graph-cut methods," in *Proc. Int. Conf. Syst. Informat.*, 2012, pp. 1936–1941.

[44] Y. Qi, G. Zhang, and Y. Li, "An auto-segmentation algorithm for multi-label image based on graph cut," *Sens. Imag.*, vol. 19, no. 1, pp. 1–13, 2018.

[45] A. K. Sinop and L. Grady, "A seeded image segmentation framework unifying graph cuts and random walker which yields a new algorithm," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2007, pp. 1–8.

[46] L. Grady and A. K. Sinop, "Fast approximate random walker segmentation using eigenvector precomputation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2008, pp. 1–8.

[47] W. Yang, J. Cai, J. Zheng, and J. Luo, "User-friendly interactive image segmentation through unified combinatorial user inputs," *IEEE Trans. Image Process.*, vol. 19, no. 9, pp. 2470–2470, Sep. 2010.

[48] C. G. Bampis, P. Maragos, and A. C. Bovik, "Graph-driven diffusion and random walk schemes for image segmentation," *IEEE Trans. Image Process.*, vol. 26, no. 1, pp. 35–50, Jan. 2017.

[49] F. Monteiro and A. Campilho, "Watershed framework to region-based image segmentation," in *Proc. Int. Conf. Pattern Recognit.*, 2008, pp. 1–4.

[50] J. Cousty, G. Bertrand, L. Najman, and M. Couprie, "Watershed cuts: Thinnings, shortest path forests, and topological watersheds," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 5, pp. 925–939, May 2010.

[51] F. Malmberg, C. L. L. Hendriks, and R. Strand, "Exact evaluation of targeted stochastic watershed cuts," *Discrete Appl. Math.*, vol. 216, no. 2, pp. 449–460, 2017.

[52] A. Falcão, J. Stolfi, and R. Lotufo, "The image foresting transform: Theory, algorithms, and applications," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 1, pp. 19–29, Jan. 2004.

[53] A. Criminisi, T. Sharp, and A. Blake, "GeoS: Geodesic image segmentation," in *Proc. Eur. Conf. Comput. Vis.: Part I*, 2008, pp. 99–112.

[54] X. Bai and G. Sapiro, "A geodesic framework for fast interactive image and video segmentation and matting," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2007, pp. 1–8.

[55] X. Bai and G. Sapiro, "Geodesic matting: A framework for fast interactive image and video segmentation and matting," *Int. J. Comput. Vis.*, vol. 82, no. 2, pp. 113–132, 2009.

[56] N. T. N. Anh, J. Cai, J. Zhang, and J. Zheng, "Robust interactive image segmentation using convex active contours," *IEEE Trans. Image Process.*, vol. 21, no. 8, pp. 3734–3743, Aug. 2012.

[57] B. Summa, J. Tierny, and V. Pascucci, "Visualizing the uncertainty of graph-based 2D segmentation with min-path stability," *Comput. Graph. Forum*, vol. 36, no. 3, pp. 133–143, 2017.

[58] Y. Boykov and M.-P. Jolly, "Interactive graph cuts for optimal boundary & region segmentation of objects in N-D images," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2001, pp. 105–112.

[59] Y. Boykov and V. Kolmogorov, "An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 9, pp. 1124–1137, Sep. 2004.

[60] O. Juan and Y. Boykov, "Active graph cuts," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2006, pp. 1023–1029.

[61] D. Chen, B. Chan, G. Mamic, C. Fookes, and S. Sridharan, "Improved grabcut segmentation via GMM optimisation," in *Proc. Digit. Image Comput.: Techn. Appl.*, 2008, pp. 39–45.

[62] V. S. Lempitsky, P. Kohli, C. Rother, and T. Sharp, "Image segmentation with a bounding box prior," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2009, pp. 277–284.

[63] M. Tang, L. Gorelick, O. Veksler, and Y. Boykov, "Grabcut in one cut," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2013, pp. 1769–1776.

[64] X. Chen and L. Pan, "A reduction method for graph cut optimization," *Pattern Anal. Appl.*, vol. 17, no. 3, pp. 361–378, 2014.

[65] D. Khattab, C. Theobalt, A. S. Hussein, and M. F. Tolba, "Modified grabcut for human face segmentation," *Ain Shams Eng. J.*, vol. 5, no. 4, pp. 1083–1091, 2014.

[66] Y.-W. Lu, J.-G. Jiang, M.-B. Qi, S. Zhan, and J. Yang, "Segmentation method for medical image based on improved grabcut," *Int. J. Imag. Syst. Technol.*, vol. 27, no. 4, pp. 383–390, 2017.

[67] M. Pizenberg, A. Carlier, E. Faure, and V. Charvillat, "Outlining objects for interactive segmentation on touch devices," in *Proc. ACM Multimedia Conf.*, 2017, pp. 1734–1742.

[68] J. Bai and X. Wu, "Error-tolerant scribbles based interactive image segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2014, pp. 392–399.

[69] S. Andrews, G. Hamarneh, and A. Saad, "Fast random walker with priors using precomputation for interactive medical image segmentation," in *Proc. Int. Conf. Med. Image Comput. Comput.-Assisted Intervention*, 2009, pp. 9–16.

[70] W. Casaca, A. Paiva, E. Gomez-Nieto, P. Joia, and L. G. Nonato, "Spectral image segmentation using image decomposition and inner product-based metric," *J. Math. Imag. Vis.*, vol. 45, no. 3, pp. 227–238, 2013.

[71] T. H. Kim, K. M. Lee, and S. U. Lee, "Generative image segmentation using random walks with restart," in *Proc. Eur. Conf. Comput. Vis.*, 2008, pp. 264–275.

[72] W. Yu and J. McCann, "Random walk with restart over dynamic graphs," in *Proc. IEEE Int. Conf. Data Mining*, 2016, pp. 589–598.

[73] C. Lee, W. D. Jang, J. Y. Sim, and C. S. Kim, "Multiple random walkers and their application to image cosegmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 3837–3845.

[74] C. G. Bampis and P. Maragos, "Unifying the random walker algorithm and the SIR model for graph clustering and image segmentation," in *Proc. IEEE Int. Conf. Image Process.*, 2015, pp. 2265–2269.

[75] F. M. Ruziska, T. Tome, and M. J. de Oliveira, "Susceptible-infected-recobered model with recurrent infection," *Physica A: Statist. Mech. Appl.*, vol. 467, pp. 21–29, 2017.

[76] V. Vezhnevets and V. Konouchine, "Growcut - interactive multi-label N-D image segmentation by cellular automata," in *Proc. Graphicon*, 2005, pp. 150–156.

[77] J. Zhang, J. Zheng, and J. Cai, "A diffusion approach to seeded image segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2010, pp. 2125–2132.

[78] J. B. Kruskal, "On the shortest spanning subtree of a graph and the traveling salesman problem," *Proc. Amer. Math. Soc.*, vol. 7, pp. 48–50, 1956.

[79] R. C. Prim, "Shortest connection newtworks and some generalizations," *Bell Syst. Tech. J.*, vol. 36, pp. 1389–1401, 1957.

[80] C. Allène, J.-Y. Audibert, M. Couprie, and R. Keriven, "Some links between extremum spanning forests, watersheds and min-cuts," *Image Vis. Comput.*, vol. 28, pp. 1460–1471, 2010.

[81] J. Cousty, G. Bertrand, L. Najman, and M. Couprie, "Watershed cuts," in *Proc. Int. Symp. Math. Morphology*, 2007, pp. 301–312.

[82] L. Najman, "Extending the power watershed framework thanks to γ-convergence," *SIAM J. Imag. Sci.*, vol. 10, no. 4, pp. 2275–2292, 2017.

[83] J. Jordan and E. Angelopoulou, "Supervised multispectral image segmentation with power watersheds," in *Proc. IEEE Int. Conf. Image Process.*, 2012, pp. 1585–1588.

[84] T. Lei, X. Jia, T. Liu, S. Liu, H. Meng, and A. Nandi, "Adaptive morphological reconstruction for seeded image segmentation," *IEEE Trans. Image Process.*, vol. 28, no. 11, pp. 5510–5523, Nov. 2019.

[85] F. P. Bergo, A. X. Falcão, P. A. Miranda, and L. M. Rocha, "Automatic image segmentation by tree pruning," *J. Math. Imag. Vis.*, vol. 29, pp. 141–162, 2007.

[86] F. Malmberg, J. Lindblad, and I. Nyström, "Sub-pixel segmentation with the image foresting transform," in *Proc. Int. Workshop Combinatorial Image Anal.*, 2009, pp. 201–211.

[87] P. A. V. Miranda and L. A. C. Mansilla, "Oriented image foresting transform segmentation by seed competition," *IEEE Trans. Image Process.*, vol. 23, no. 1, pp. 389–398, Jan. 2014.

[88] B. L. Price, B. Morse, and S. Cohen, "Geodesic graph cut for interactive image segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2010, pp. 3161–3168.

[89] R. Cardenes, C. Alberola-Lopez, and J. Ruiz-Alzola, "Fast and accurate geodesic distance transform by ordered propagation," *Image Vis. Comput.*, vol. 28, no. 3, pp. 307–316, 2010.

[90] J. Gaura and E. Sojka, "Resistance-geodesic distance and its use in image segmentation," *Int. J. Artif. Intell. Tools*, vol. 25, no. 05, 2016, Art. no. 1640002.

[91] M. Holusa and E. Sojka, "The k-max distance in graphs and images," *Pattern Recognit. Lett.*, vol. 98, pp. 103–109, 2017.

[92] L. Cerrone, A. Zeilmann, and F. A. Hamprecht, "End-to-end learned random walker for seeded image segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 12551–12560.

[93] P. Vernaza and M. Chandraker, "Learning random-walk label propagation for weakly-supervised semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 2953–2961.

[94] S. Wolf, L. Schott, U. Köthe, and F. Hamprecht, "Learned watershed: End-to-end learning of seeded segmentation," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 2030–2038.

[95] D. Eschweiler, T. Spina, R. Choudhury, E. Meyerowitz, A. Cunha, and J. Stegmaier, "CNN-based preprocessing to optimize watershed-based cell segmentation in 3D confocal microscopy images," in *Proc. IEEE Int. Symp. Biomed. Imag.*, 2019, pp. 223–227.

[96] M. Kowal, M. Zejmo, M. Skobel, J. Korbicz, and R. Monczak, "Cell nuclei segmentation in cytological images using convolutional neural network and seeded watershed algorithm," *J. Digit. Imag.*, pp. 1–12, 2019. [Online]. Available: https://link.springer.com/article/10.1007/s10278-019-00200-8

[97] N. Xu, B. Price, S. Cohen, J. Yang, and T. Huang, "Deep interactive object selection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 373–381.

[98] N. Xu, B. L. Price, S. Cohen, J. Yang, and T. S. Huang, "Deep grabcut for object selection," in *Proc. Brit. Mach. Vis. Conf.*, 2017, pp. 1–12.

[99] Y. Aksoy, T.-H. Oh, S. Paris, M. Pollefeys, and W. Matusik, "Semantic soft segmentation," *ACM Trans. Graphs*, vol. 37, no. 4, pp. 72:1–72:13, 2018.

[100] Y. Aksoy, T. O. Aydin, A. Smolić, and M. Pollefeys, "Unmixing-based soft color segmentation for image manipulation," *ACM Trans. Graph.*, vol. 36, no. 2, pp. 19:1–19:19, 2017.

[101] T. Wang, J. Yang, Q. Sun, Z. Ji, P. Fu, and Q. Ge, "Global graph diffusion for interactive object extraction," *Inf. Sci.*, vol. 460/461, pp. 103–114, 2018.

[102] Y. Liang, Y. Gan, M. Chen, D. Gutierrez, and A. Muñoz, "Generic interactive pixel-level image editing," *Comput. Graph. Forum*, vol. 38, no. 7, pp. 23–34, 2019.

[103] T. Cour, F. Bénézit, and J. Shi, "Spectral segmentation with multiscale graph decomposition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2005, pp. 1124–1131.

[104] T. H. Kim, K. M. Lee, and S. U. Lee, "Learning full pairwise affinities for spectral segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 7, pp. 1690–1703, Jul. 2013.

[105] A. Saglam and N. A. Baykan, "Effects of color spaces and distance norms on graph-based image segmentation," in *Proc. 3rd Int. Conf. Front. Signal Process.*, 2017, pp. 130–135.

[106] L. Grady, "Targeted image segmentation using graph methods," in, *Image Processing and Analysis with Graphs*. Boca Raton, FL, USA: CRC Press, 2012.

[107] V. Jankovic, "Quadratic functions in several variables," *Teaching Math.*, vol. 8, pp. 53–60, 2005.

[108] O. Stein, E. Grinspun, M. Wardetzky, and A. Jacobson, "Natural boundary conditions for smoothing in geometry processing," *ACM Trans. Graph.*, vol. 37, no. 2, pp. 23:1–23:13, 2018.

[109] A. Elmahmudi and H. Ugail, "The biharmonic eigenface," *Signal Image Video Process.*, vol. 13, pp. 1639–1647, 2019.

[110] C. N. Ochotorena and Y. Yamashita, "Anisotropic guided filtering," *IEEE Trans. Image Process.*, vol. 29, pp. 1397–1412, 2020.

[111] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Susstrunk, "SLIC superpixels compared to state-of-the-art superpixel methods," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 11, pp. 2274–2282, Nov. 2012.

[112] R. Unnikrishnan, C. Pantofaru, and M. Hebert, "Toward objective evaluation of image segmentation algorithms," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 6, pp. 929–944, Jun. 2007.

[113] M. Meliva, "Comparing clusterings: An axiomatic view," in *Proc. Int. Conf. Mach. Learn.*, 2005, pp. 577–584.

[114] R. Unnikrishnan, C. Pantofaru, and M. Hebert, "A measure for objective evaluation of image segmentation algorithms," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. Workshops*, 2005, pp. 34–39.

[115] F. J. Estrada and A. D. Jepson, "Benchmarking image segmentation algorithms," *Int. J. Comput. Vis.*, vol. 85, no. 2, pp. 167–181, 2009.

[116] K. McGuinness and N. E. O'Connor, "A comparative evaluation of interactive segmentation algorithms," *Pattern Recognit.*, vol. 43, no. 2, pp. 434–444, 2010.

[117] P. Arbeláez, M. Maire, C. Fowlkes, and J. Malik, "Contour detection and hierarchical image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 5, pp. 898–916, May 2011.

[118] F. Andrade and E. V. Carrera, "Supervised evaluation of seed-based interactive image segmentation algorithms," in *Proc. IEEE Symp. Comput. Intell. Multimedia Signal Vis.*, 2015, pp. 1–7.

[119] H. Jegou, M. Douze, and C. Schmid, "Hamming embedding and weak geometry consistency for large scale image search," in *Proc. Eur. Conf. Comput. Vis.*, 2008, pp. 304–317.

[120] T. H. Kim, K. M. Lee, and S. U. Lee, "Nonparametric higher-order learning for interactive segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2010, pp. 3201–3208.

Wallace Casaca received the BSc and master's degrees in pure and applied mathematics from São Paulo State University (UNESP), São Paulo, Brazil, in 2008 and 2010, respectively, and the PhD degree in computer sciences and applied mathematics from the University of São Paulo (USP), São Paulo, Brazil, in 2014. As part of his doctoral studies, he also worked as a visiting researcher with Brown University, the School of Engineering. His research interests include image segmentation, image restoration, image retrieval, remote sensing, information visualization, optimization techniques, partial differential equations, and numerical analysis.

João Paulo Gois received the mathematics degree from São Paulo State University (UNESP), São Paulo, Brazil, in 2001, and the MSc and PhD degrees in computer science and computational mathematics from the University of São Paulo (USP), São Paulo, Brazil, in 2004 and 2008, respectively. Currently, he is associate professor with the Center of Mathematics, Computing, and Cognition, Federal University of ABC, UFABC, Brazil. His research interests include interactive computer graphics, geometric modeling, and computer animation.

Harlen Costa Batagelo received the BSc degree in computer science from the University of West Santa Catarina, Brazil, and the MSc and PhD degrees in electrical engineering from the State University of Campinas, UNICAMP, Campinas, Brazil. He is currently a professor with the Federal University of ABC, UFABC, Brazil. His research interests include interactive computer graphics, realtime rendering, user interface, games, and geometric modeling.

Gabriel Taubin (Fellow, IEEE) is professor of engineering and computer science with Brown University. In 1990, he joined the IBM Research Division, where he held various positions, including research staff member and research manager. In 2003, he joined the Brown University School of Engineering as an associate professor of engineering and computer science. During the 2000-2001 academic year, on sabbatical from IBM, he was visiting professor of electrical engineering with the California Institute of Technology. During the Spring semester of 2010, on sabbatical from Brown, he was visiting associate professor of media arts and sciences with MIT. He was the editor-in-chief of the *IEEE Computer Graphics and Applications Magazine* from 2010 to 2013, and has served as a member of the editorial board of the *Geometric Models Journal*, and as associate editor of the *IEEE Transactions on Visualization and Computer Graphics*. He is also a member of the Fulbright Specialist Roaster, and a Fulbright Specialist grantee and was named IBM master inventor. The 3D geometry compression technology that he has developed at IBM was incorporated into the MPEG-4 standard, and became an integral part of IBM products.

Luis Gustavo Nonato (Member, IEEE) received the PhD degree in applied mathematics from the Pontifícia Universidade Católica, Rio de Janeiro, Brazil, in 1998. He is currently a full professor with the Institute of Mathematical and Computer Sciences, University of São Paulo, São Carlos, Brazil. He was a visiting professor with the Center for Data Science, New York University, New York from 2016 to 2018. From 2008 to 2010, he was a visiting scholar with the Scientific Computing and Imaging Institute, University of Utah, Salt Lake City. Besides having served in several program committees, including IEEE SciVis, IEEE InfoVis, and EuroVis, he was associate editor of the *Computer Graphics Forum* and currently he is associate editor of the *IEEE Transactions on Visualization and Computer Graphics*. He is also the editor of the SBMAC SpringerBriefs in *Applied Mathematics and Computational Sciences*. His research interests include visualization, visual analytics, machine learning, and data science.