Graph Laplacian for Spectral Clustering and Seeded Image Segmentation

Wallace Casaca
Institute of Mathematics and Computer Science (ICMC)
University of São Paulo (USP)
13566-590, São Carlos, SP, Brazil
https://sites.google.com/site/wallacecoc
wallace@icmc.usp.br

***Abstract*—Image segmentation is an indispensable tool to enhance the ability of computer systems to perform elementary cognitive tasks such as *detection*, *recognition* and *tracking*. In particular, interactive algorithms have gained much attention lately, specially due to their good performance in segmenting complex images and easy utilization. However, most interactive segmentation methods rely on sophisticated mathematical tools whose effectiveness strongly depends on the kind of image to be processed. In fact, sharp adherence to the contours of image segments, uniqueness of solution, high computational burden, and extensive user intervention are some of the weaknesses of most existing techniques. In this work we proposed two novel interactive image segmentation techniques that sort out the issues discussed above. The proposed methods rely on Laplace operators, spectral graph theory, and optimization approaches towards enabling highly accurate segmentation tools which demand a reduced amount of user interaction while still being mathematically simple and computationally efficient. The good performance of our segmentation algorithms is attested by a comprehensive set of comparisons against representative state-of-the-art methods. Indeed, qualitative and quantitative results obtained from well-known image benchmarks show that our methodologies outperform others. As additional contribution, we have also proposed two new algorithms for image inpainting and photo colorization, both of which rely on the accuracy of our segmentation apparatus.**

***Keywords*—Image segmentation, graph laplacian, spectral graph theory, inpainting, colorization, computer vision applications**

I. Introduction

Image segmentation is the key task for an enormous quantity of computer vision problems. A typical procedure in image segmentation is to interpret an image as a graph, which enables the use of powerful mathematical tools such as Laplace operators and spectral graph theory. Moreover, the flexibility introduced by a graph representation as to pixel connectivity and edge weighting greatly increases the capability of segmentation algorithms to distinguish patterns, structures, and shapes. However, outperforming human skills in terms of recognition is a difficult task. Therefore, *semi-supervised image segmentation methods* have become a trend by combining the human ability for object/background detection with the solid mathematical foundation of graph theory [1].

In this scenario, the use of interact mechanisms to properly settle Laplace operators on image graph representations have proven to be an effective alternative [2], [3]. Those user-assisted mechanisms typically define the Laplace operators in a similarity/affinity graph which encodes image information such as colors, textures and gradients. Moreover, it involves a cost function defined on the graph [4], [5], [6], [3], or solving a spectral-cut problem [7], [8], [9], [10]. However, as pointed out in [11], [12], [13], existing partitioning techniques are circumstantially prone to fail in many pragmatic situations. For instance, common drawbacks not tackled by state-of-the-art algorithms are:

1) The resulting segmentation generally exhibits low adherence on the contours of the image regions, failing to capture fine details or, in many cases, producing a low quality segmentation output.
2) Make use of sophisticated optimization tools to be effective, impacting negatively on the computational cost, implementation and portability of the code.
3) Demand extra computational effort to the user, specially for processing high resolution images, such those obtained nowadays by mobile devices.
4) They are highly sensitive to the adjustment of the edge weights in the graph.

A. Contributions

In this thesis we proposed two novel user-assisted image segmentation techniques that address the issues discussed above. The proposed algorithms rely on Laplace operators, spectral graph theory, and optimization tools towards reaching highly fitting on object boundaries which demand a reduced amount of user involvement while still being mathematically easy to solve and computationally efficient.

While most of our research has been focused on the particular problem of image segmentation, we develop as side results new methodologies for the problem of *image inpainting* [14], [15] and *photo colorization* [16], [17], both of which derived from the proposed segmentation methodologies. Figure 1 presents some illustrative examples while the list below provides the main publications originated during the development of this thesis:

Contributions in Computer Vision: [10], [18], [17], [19], [2], [15], [3], [20], [21].

Contributions in Graphics and Visualization: [22], [23], [24], [25].

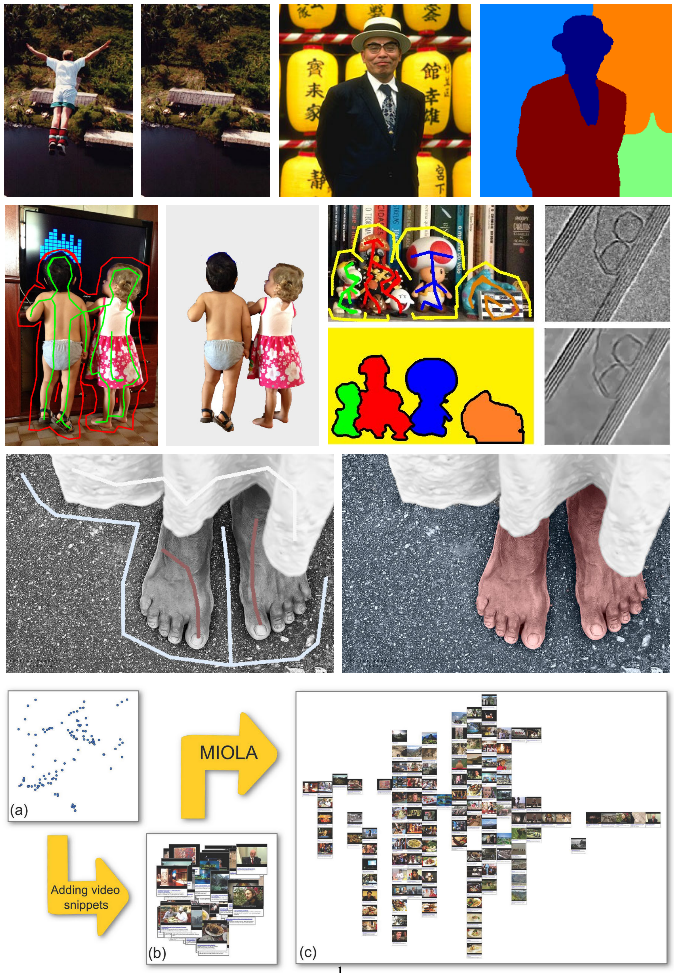


Fig. 1. A few results obtained during the PhD research.

B. Awards

The following papers have been awarded as “best papers” or received “honorable mention” in the renowned SIBGRAPI and ICCP conferences. According to Google Scholar, SIBGRAPI (Conference on Graphics, Patterns and Images) is the most relevant Latin American conference in the fields of *Graphics*, *Visualization* and *Computer Vision*. ICCP (IEEE International Conference on Computational Photography) is a prominent conference in the field of *Digital Photography*.

- **Best paper award** for the paper “Spectral Segmentation using Cartoon-Texture Decomposition and Inner Product-based Metric”, 24th SIBGRAPI, Maceio, Brazil, 2011.
- **Best paper award** for the paper “Mixed Integer Optimization for Layout Arrangement”, 26th SIBGRAPI, IEEE Computer Society, Arequipa, Peru, 2013[1].
- **Honorable Mention** for the poster presentation “Image Colorization based on Multidimensional Projection”, 5th IEEE ICCP, Harvard, Cambridge, United States, 2014.

C. Scientific Dissemination to the General Public

The research conducted during the thesis had a good impact in terms of diffusion to the general public (see Figure 2 for an illustration). In fact, part of our research has been advertised on TV news channels, news papers, and on the internet, as listed below (in portuguese):

Nunca foi tão fácil alterar uma imagem

Assessoria de Comunicação

26-Jun-2014

Pesquisa desenvolvida no ICMC cria ferramenta inovadora para facilitar a vida de quem precisa segmentar uma imagem, o que pode contribuir para aprimorar a personalização artística de fotografias, a identificação de indivíduos e a análise de imagens médicas

Pesquisadores explicam como funciona a nova ferramenta

02/08/2014 12h23 - Atualizado em 02/08/2014 12h31

Programa criado pela USP São Carlos remove pessoas de foto de forma fácil

Ferramenta contribui para aprimorar personalização artística de imagem. Segundo os desenvolvedores, até uma criança consegue manipulá-lo.

Do G1 São Carlos e Araraquara

Fig. 2. Dissemination of the technologies originated to the wider public.

1) **News published on *www.usp.br*** (2013): “Pesquisadores do ICMC recebem prêmio internacional em computação gráfica” (www.icmc.usp.br/e/53ddb).
2) **Report published on *www.usp.br*** (2014): “ICMC desenvolve ferramenta inovadora para segmentação de imagens” (www.icmc.usp.br/e/f37b3).
3) **Report published on *DCI (Newspaper)*** (2014): “Pesquisadores desenvolvem nova ferramenta para imagem” (www.icmc.usp.br/e/0ebf1).
4) **TV report broadcasted on *Rede Globo*** (2014): “Programa criado pela USP São Carlos remove pessoas de foto de forma fácil” (www.icmc.usp.br/e/49c86).
5) **TV report broadcasted on *TV Educativa de São Carlos*** (2014): “USP cria ferramenta que facilita a vida de quem trabalha com imagens” (www.icmc.usp.br/e/9daca).

A summary of the main results obtained during the development of this thesis is presented in the following sections.

II. Spectral Image Segmentation

Spectral graph theory [1] has been the basic tool for the so-called spectral cut methodology [7], [8], [26], [27], which exploits the eigenstructure of an image affinity graph so as to perform clustering. In fact, spectral graph theory enables great flexibility in the segmentation process, as different choices can be made towards defining the similarity graph connectivity as well as the assignment of weights to the edges of the graph. Such a flexibility has leveraged a multitude of techniques, making spectral cuts an attractive image segmentation tool.

Among the vast amount of techniques inspired in spectral cuts, three approaches have gain a lot of attention in recent years, being widely used as source of segmentations in many practical applications:

1) *Spectral and Normalized Cuts-based methods* [7], [28], [29], [30], [31], [8], [9], [32];
2) *Multiscale Segmentation-based methods* [33], [34], [35], [36], [37];
3) *Random Walker-based methods* [38], [5], [39], [40].

Despite their effectiveness and powerfulness, methods inspired on spectral cuts present some weaknesses that must be observed when performing segmentation. For example, the accuracy in detecting the boundaries between image regions

[1] See the video of our tool at https://www.youtube.com/watch?v=zGgIYX7oSqI

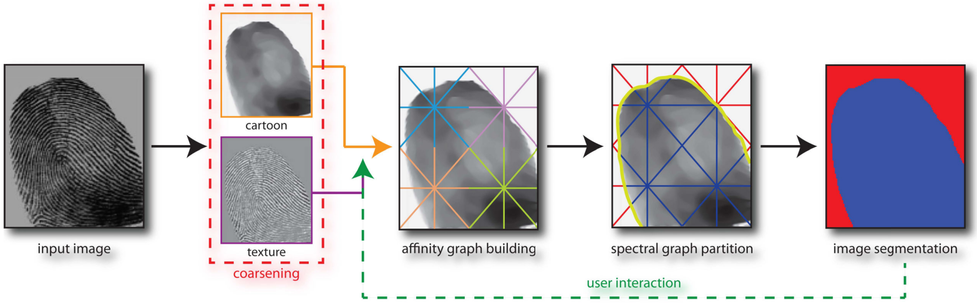


Fig. 3. Pipeline of the proposed image segmentation framework.

is highly dependent on the weights assigned to the edges of the graph. Although automatic schemes have been proposed to accurately compute those weights [7], [34], [41], [35], [42], it is well-known that user intervention is essential in many cases to correctly define object boundaries [39]. Therefore, incorporating user knowledge into the segmentation process is of paramount importance since the identification of boundary information is subject to human judgment in many practical situations. Another important issue in the context of spectral cuts is the computational cost, as computing the eigenstructure of a graph is a very time consuming task, hampering the direct use of spectral segmentation in high resolution images [43].

A. *Spectral Segmentation via Cartoon-Texture Decomposition and Inner Product-based Metric*

In this section we introduce a new methodology for image segmentation that relies on spectral cuts but addresses the issues raised above. We show that the proposed approach outperforms classical spectral segmentation techniques in aspects such as accuracy and robustness on the well-known image dataset from UC-Berkley [44]. Figure 4 shows an example of the proposed framework. We can summarize the novelties introduced by our methodology as:

1) An image segmentation technique that combines cartoon-texture decomposition and spectral cuts [2];
2) A novel method to compute and assign weights to the edges of the similarity graph using the cartoon component extracted from the image;
3) A new strategy to modify the weights of the graph according to user interaction, taking into account the texture component of the image.

B. *Pipeline Overview*

The proposed approach, first reported in [10], [2], comprises five main steps, as illustrated in Figure 3. The first step, *Cartoon-Texture Decomposition*, separates the target image $\mathcal{I}$ into two images, $\mathcal{C}$ and $\mathcal{T}$, where $\mathcal{C}$ and $\mathcal{T}$ hold the cartoon and texture information contained in $\mathcal{I}$. In the second step, an *Image Coarsening* is applied to $\mathcal{C}$ and $\mathcal{T}$ so as to build smaller affinity graphs in the third step of the pipeline, namely *Affinity Graph Construction*. Besides speeding up the spectral decomposition, the reduced number of edges also lessen the computational burden during the weight assignment stage, which allow us to handle large images. Weights are derived from an inner product-based metric defined on the coarse cartoon image. The spectral decomposition is carried out in the *Spectral Partition* step, being the result mapped back to the original image through a coarse-to-fine interpolation procedure. The user can change the partition by stroking the resulting segmentation. This step is performed by combining the coarse texture component with a recent technique of harmonic analysis [45], [46] in order to incorporate the high-level oscillatory information into the spectral cut process. Details about each step of the pipeline are provided below.

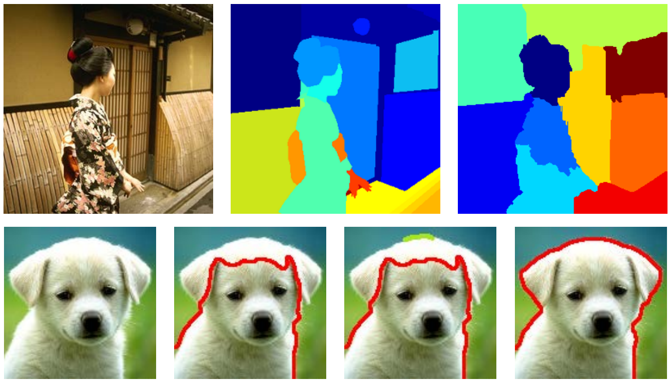
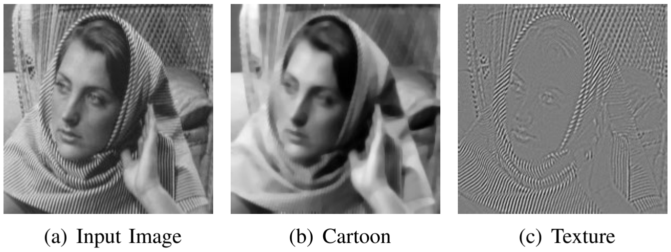


Fig. 4. First row: from the left to right, the ground truth image and the result obtained with our method without user intervention. Second row: illustrates the user interaction (green scribes) to improve the segmentation.

1) Cartoon-Texture Image Decomposition: CTD splits the input image $\mathcal{I}$ into two disjoint images, $\mathcal{C}$ and $\mathcal{T}$. The cartoon component $\mathcal{C}$ holds the geometric structures, isotopes and smooth-pieces of $\mathcal{I}$ while the texture component contains textures, oscillating patterns, fine details and noise. This decomposition schemes satisfies the important relationship $\mathcal{I} = \mathcal{C} + \mathcal{T}$ (see [47], [18] and the underlying mathematical theory proposed in [48]). Similar to [47], where a functional minimization problem has been formulated and solved through



(a) Input Image (b) Cartoon (c) Texture

Fig. 5. Image decomposition into a cartoon and texture component.

a system of partial differential equations, both cartoon $\mathcal{C}$ and texture $\mathcal{T}$ components are computed by solving the following system of equations:

$$\left\{\begin{array}{l} \mathcal{C} = \mathcal{I} - \partial_x g_1 - \partial_y g_2 + \dfrac{1}{2\lambda}\,\mathrm{div}\left(\dfrac{\nabla\mathcal{C}}{|\nabla\mathcal{C}|}\right) \\ \mu\dfrac{g_1}{\sqrt{g_1^2+g_2^2}} = 2\lambda\left[\dfrac{\partial}{\partial x}(\mathcal{C}-\mathcal{I}) + \partial^2_{xx}g_1 + \partial^2_{xy}g_2\right] \\ \mu\dfrac{g_2}{\sqrt{g_1^2+g_2^2}} = 2\lambda\left[\dfrac{\partial}{\partial y}(\mathcal{C}-\mathcal{I}) + \partial^2_{xy}g_1 + \partial^2_{yy}g_2\right] \end{array}\right. , \tag{1}$$

with initial conditions for $\mathcal{C}$, g_1, and g_2 given by

$$\left\{\begin{array}{l} \dfrac{\nabla\mathcal{C}}{|\nabla\mathcal{C}|}\cdot(n_x, n_y) = 0 \\ (\mathcal{I} - \mathcal{C} - \partial_x g_1 - \partial_y g_2)\cdot nx = 0 \\ (\mathcal{I} - \mathcal{C} - \partial_x g_1 - \partial_y g_2)\cdot ny = 0 \end{array}\right. . \tag{2}$$

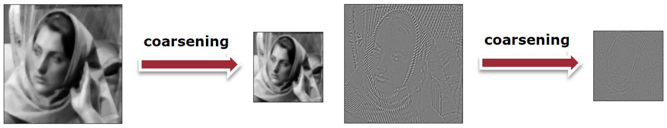
Mathematically, the cartoon component $\mathcal{C}$ is a bounded variation function, $\overrightarrow{g} = (g_1, g_2) \in L^2(\mathbb{R}^2)$ where the texture component $\mathcal{T} = \mathrm{div}(\overrightarrow{g})$, and the constants $\lambda, \mu > 0$ are tuning parameters. Equations (1) are usually discretized by a semi-implicit finite difference schemes and solved using an iterative algorithm based on fixed point iteration (for more details about numerical aspects, see [47], [49]). Figure 5 shows the result of the CTD scheme applied to a digital image.

In our methodology, both $\mathcal{C}$ and $\mathcal{T}$ are used to compute the weights assigned to the edges of the affinity graph. Since $\mathcal{C}$ is a texture-free denoised image, edge and shape detectors work well when applied to $\mathcal{C}$ as pointed out in [47]. This fact is exploited to define the weights, as we detail later. Information contained in $\mathcal{T}$ is handled only at the end of pipeline, during user interaction stage.

2) Image Coarsening: In order to reduce the size of the affinity graph towards alleviating the computational burden during the spectral decomposition, we perform a fine-to-coarse transformation on $\mathcal{C}$ (resp. $\mathcal{T}$), resulting in a coarse scale $\tilde{\mathcal{C}}$ (resp. $\tilde{\mathcal{T}}$) of $\mathcal{C}$ (resp. $\mathcal{T}$). Such a transformation is accomplished using the bicubic interpolation method described in [50], which minimizes the blurring effect while still preserving gradients in the coarse image (see Figure 6 for an illustration). Other downsampling techniques such as [51] can be alternatively used to convey essential image information among scales. Our experiments showed that coarsening the image to one-fourth of its original resolution is a good trade-off between computational time and accuracy, speeding up the processing up to 6 times, as outlined in the comparison section.

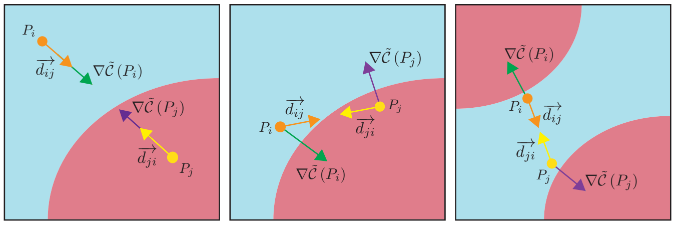
3) Building the Affinity Graph: The affinity graph G is built by associating each pixel from $\tilde{\mathcal{C}}$ to a node of the graph, connecting the nodes according to the distance r between corresponding pixels, in mathematical words,

$$\|P_i - P_j\|_\infty < r\,. \tag{3}$$



(a) Cartoon image coarsening (b) Texture image coarsening

Fig. 6. Fine-to-coarse step illustration obtained from [50].


P_i
$\overrightarrow{d_{ij}}$
$\nabla\tilde{\mathcal{C}}(P_i)$
$\nabla\tilde{\mathcal{C}}(P_j)$
$\overrightarrow{d_{ji}}$
P_j
$\nabla\tilde{\mathcal{C}}(P_j)$
$\overrightarrow{d_{ij}}$
P_j
P_i
$\overrightarrow{d_{ji}}$
$\nabla\tilde{\mathcal{C}}(P_i)$
$\nabla\tilde{\mathcal{C}}(P_i)$
P_i
$\overrightarrow{d_{ij}}$
$\overrightarrow{d_{ji}}$
P_j
$\nabla\tilde{\mathcal{C}}(P_j)$

(a) Same direction (b) Oblique direction (c) Opposite direction

Fig. 7. Geometric interpretation of the inner product-based metric. Maximum weights occur when the gradient and the direction defined from the graph edge point to the same direction (a). Moderate weight is highlighted in (b) and the third case, where opposite directions (c) produce minimum weights (zero).

The weight assigned to each edge of G is derived from the proposed inner product-based metric. Our metric considers the variation of the image in the directions defined by the edges of the graph. More specifically, the weight w_{ij} associated to the edge e_{ij} is defined as:

$$w_{ij} = \frac{1}{1+\eta h_{ij}^2},\quad h_{ij} = \max\left\{\frac{\partial\tilde{\mathcal{C}}(P_i)}{\partial\overrightarrow{d_{ij}}}, \frac{\partial\tilde{\mathcal{C}}(P_j)}{\partial\overrightarrow{d_{ji}}}, 0\right\}, \tag{4}$$

$$\frac{\partial\tilde{\mathcal{C}}(x)}{\partial\overrightarrow{d_{ij}}} = \langle\nabla\tilde{\mathcal{C}}(x), \overrightarrow{d_{ij}}\rangle,\quad \text{with}\quad \overrightarrow{d_{ij}} = \frac{\overrightarrow{P_iP_j}}{|\overrightarrow{P_iP_j}|}\,. \tag{5}$$

The left most term in Equation (5) is the directional derivative of $\tilde{\mathcal{C}}$ in the direction $\overrightarrow{d_{ij}}$, which is defined from the graph G and $\eta > 0$ is a tuning constant. Therefore, image properties as well as the adjacency structure of the affinity graph is taken into account when assigning weights to the edges of G. Furthermore, our formulation accounts for the intensity and geometric information to define the weights through the inner-product in the edge direction. Figure 7 provides a geometric interpretation of the proposed metric.

The effective weights w_{ij} are chosen from Eq. (4) rather then using the exponential measure usually employed by other authors [7], [30], [36]. The scheme proposed in Eq. (4) does not push values to zero as fast as the exponential function, which allows for considering the influence of a larger number of edges when carrying out the spectral decomposition. Eq. (4) is indeed derived from the Malik-Perona diffusivity term [52], [53], which was originally used for establishing the notion of anisotropy in partial differential equations. Moreover, the inner product-based metric (4) holds that $w_{ij} = w_{ji}$, which ensures symmetry for the graph Laplacian matrix $\mathbf{L}$. This fact is of paramount importance to guarantee that the eigenstructure of $\mathbf{L}$ is made up of only real numbers.

4) Spectral Cutting and Coarse-to-Fine: Given the affinity graph G built from $\tilde{\mathcal{C}}$ and the number of partitions initially defined by the user, we carry out the spectral decomposition using the same methodology proposed in [7]. More specifically, we first decompose the graph Laplacian matrix as $\mathbf{L} = \mathbf{D} - \mathbf{W}$, where $\mathbf{D}$ and $\mathbf{W}$ contain the diagonal and off-diagonal elements of $\mathbf{L}$. Then, the Fiedler vector $\mathbf{f}$ is obtained by solving the generalized eigenvalue problem

$$(\mathbf{D} - \mathbf{W})\mathbf{x} = \lambda\mathbf{D}\mathbf{x}, \tag{6}$$

getting $\mathbf{f}$ as the eigenvector associated to the smallest non-zero eigenvalue. The Fiedler vector splits $\tilde{\mathcal{C}}$ into two subsets,

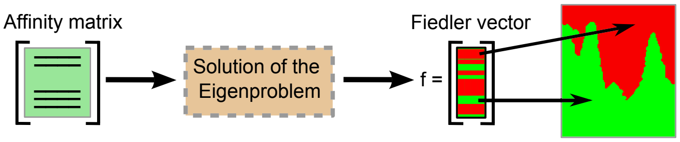


Fig. 8. Spectral cut pipeline to partition the image from the zero-set of the Fiedler vector.

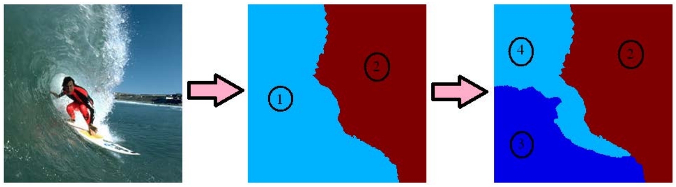


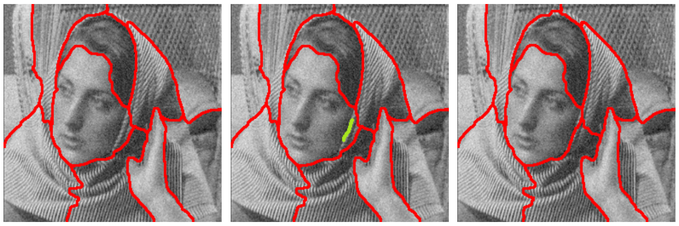
Fig. 9. Hierarchical segmentation by recursively computing the spectral decomposition for multiple parts of the image.

one containing the pixels corresponding to nodes of the graph where the entries of $\mathbf{f}$ are positive and other containing the pixels with negative values of $\mathbf{f}$. Therefore, the zero-set of $\mathbf{f}$ is a curve that separates the regions with different signs. Figure 8 portrays the spectral cut procedure. The partitioning created in $\tilde{\mathcal{C}}$ is brought back to $\mathcal{C}$ using bicubic interpolation from $\mathbf{f}$.

Multiple partitions can also be reach by recursively computing the spectral decomposition for each part of the image before the interpolation process, as depicted in Figure 9. In fact, the recursive process may be driven by the user, who can specify the highest level of recursion, moreover, the user can brush any pieces of the image during each one of recursion steps in order to better set weights and thus improve the segmentation quality (see the next section for details).

Figure 10 shows the result of applying our methodology to segment a fingerprint image. For the sake of comparison, we show in Fig. 10(a) the result of computing our framework directly from the original image $\mathcal{I}$, that is, skipping the CTD, while Fig.10(b) depicts the result using CTD and the classical weighting metric [7] instead of Eq.(4) to define the graph weights. Notice, from Fig.10(c), how better the segmentation is when Eq.(4) and CTD are combined.

5) Interactive Weight Manipulation: Weights can be interactively tuned so as to force the spectral cut to accurately fit boundaries between textured regions of the image. Our tuning scheme relies on the texture component $\mathcal{T}$ obtained from the cartoon-texture decomposition. The component $\mathcal{T}$ is processed by an harmonic analysis tool called *Wave Atoms* [45], [46]. Wave atoms-based techniques bear high directional anisotropy and sensitivity to noise, which makes them suited to identifying oscillatory patterns in high-frequency domains. In our approach, we use this tool to assign a scalar $S(\mathcal{T}_i) \in [0, 1)$ to each pixel $\mathcal{T}_i$ of $\mathcal{T}$, where values close to "one" means the pixel belongs to the "wave" of a texture pattern, similar to that used in [54] to produce a texture mapping. Therefore, pixels nearby the boundary between two textured regions tend to be identified as not belonging to a texture wave, thus assuming values close to "zero".



(a) Initial segmentation (b) Brush made by user (c) Final result

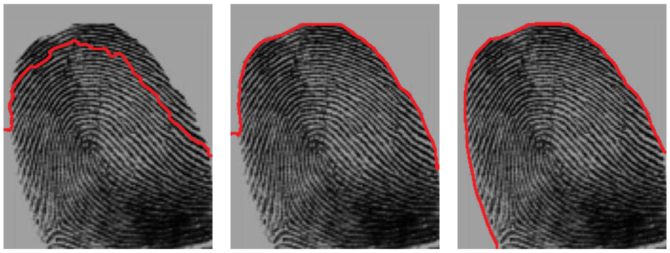
Fig. 11. Improving segmentation of a noise-textured image.

Starting from this premise, the weights of edges incident to pixels brushed by the user are modified as follows:

$$w_{ij} = \rho \left(\min_{e_{ij} \in E, w_{ij} \neq 0} w_{ij} \right) \left(1 - \max\{S(\tilde{\mathcal{T}}_i), S(\tilde{\mathcal{T}}_j)\} \right), \quad (7)$$

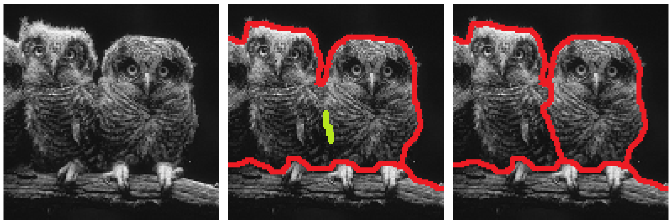
where the constant $\rho \in (0, 1)$ is the smallest non-zero weight of the edges in G obtained during the automatic spectral cutting performed on $\tilde{\mathcal{C}}$, and $\tilde{\mathcal{T}}$ is the coarse version of $\mathcal{T}$. The constant ρ enforces a more drastic change of weights in the region stroked by the user since it holds the new weights that will have the lowest possible non-zero value within the target graph.

Figure 11 shows the result of segmenting a noisy image using our method setting 10 partitions. Notice from Fig. 11(a) that most parts of the image is accurately segmented, attesting the accuracy of the proposed method for the case where the image contains texture and moderate gaussian noise. The spectral cut deviates from the correct boundary in just a few small regions which are easily fixed through user interaction, as depicted in Fig.11(b) and Fig. 11(c). This post-segmentation was only feasible because the texture mapping used to accomplish this task is sensitive to noise. Figure 12 shows that is not necessary to perform a large number of user interventions to achieve the desired segmentation. The simple greenish stroke depicted on the texture region between the two owls were enough to separate the birds, as depicted in Fig.12(c).



(a) Without CTD (b) Without our metric (c) Complete pipeline

Fig. 10. Automatic result with the proposed framework.



(a) Original image (b) Small stroke (c) Final result

Fig. 12. A simple stroke (greenish region between the two owls) is sufficient to improve the segmentation.

C. Results, Comparisons and Evaluation

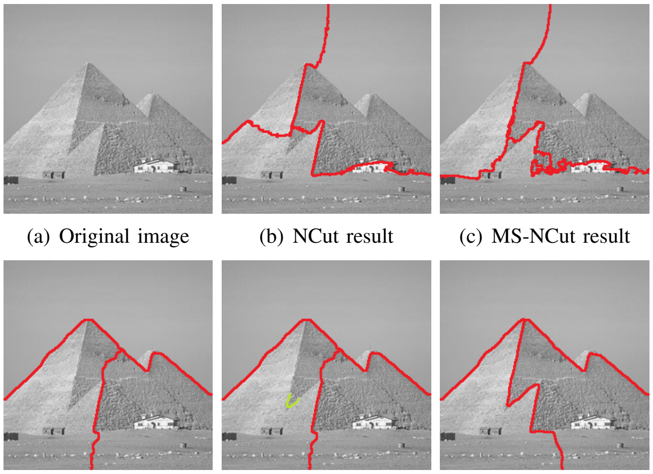
Now we present the results obtained with the proposed framework and a comparative study involving four other state-of-the-art methods. We split this section into three classes of experiments:

1) *Comparison with Automatic Methods.*
2) *Evaluation using Ground-Truth Images.*
3) *Comparison with Interactive Methods.*

The following parameters were used in all experiments presented in this section: $\lambda = 0.05$ and $\mu = 0.1$ in the cartoon-texture decomposition, the default parameters suggested in [50] for the bicubic interpolation and a hard threshold at 3σ (noise-to-signal ratio of the image) combined with cycle spinning [45] for the wave atom transform. We set $r = 1$ and $\eta = 5$ in Equations (3)-(4), respectively. Finally, in order to check the segmentation quality of the proposed approach, we provide comparisons against two automatic and two user-assisted eigenspectrum-based techniques:

- k-way Normalized Cuts method (NCut)[2] [7], [32];
- Multiscale Normalized Cuts method (MS-NCut)[3] [34] (with radius 2.3);
- Random Walker-based Segmentation with pure Eigenvector Precomputation (RWS-EP) [39];
- Random Walker based on Eigenvector Precomputation and Prior scheme (MSFP)[4] (with 80 precomputed eigenvectors) [40];

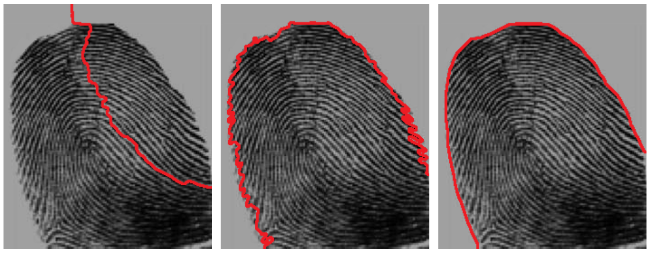
1) Comparison with Automatic Methods: The first experiment shown in Figure 13 presents a comparative analysis of our technique against the non-interactive NCut and MS-NCut approaches. We can see that both classical NCut (Fig. 13(b)) and MS-NCut (Fig. 13(c)) badly segment parts of the image. Our approach results in a better partitioning (Fig. 13(d)), although some regions are also segmented in an incorrect way. After user intervention, shown in Fig. 13(e), the result improves considerably (Fig. 13(f)).



(a) Original image (b) NCut result (c) MS-NCut result

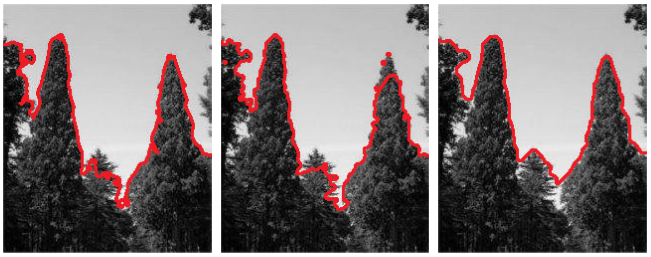
(d) Our initial result (e) With brush (green) (f) Final segmentation

Fig. 13. The influence of the user intervention in comparison with static approaches.



(a) NCut partition (b) MS-NCut partition (c) Our result

Fig. 14. The result of applying NCut, MS-NCut, and the proposed approach (in automatic mode) in a fingerprint image.



(a) NCut result (b) MS-NCut result (c) Our result

Fig. 15. Our approach (in automatic mode) produces smoother segmentation curves when compared to NCut and MS-NCut.

The result of applying the three above-mentioned methods in a fingerprint image for two partitions is shown in Figure 14. Notice that the NCut (Fig. 14(a)) does not segment the fingerprint correctly while the MS-NCut and our approach do a good job. It is easy to see from Fig. 14(b) that the MS-NCut tends to produce a jagged segmentation curve while our method results in a smoother curve, as shown in Fig. 14(c). It becomes clear from Figure 15 that the smoothness of the result produced by our approach also help increase robustness. While NCut and MS-NCut tend to generate a segmentation curve with many artifacts and some cognition errors, our approach produces a much pleasant result.

2) Evaluation using Ground-Truth Images: We evaluate the performance of the proposed methodology by means of quantitative and visual analysis against the NCut and MS-NCut algorithms. To accomplish the numerical evaluation, we make use of *Recall*, *Precision* and *F-Score* measures [55] on the well-known *Berkeley Segmentation Dataset (BSDS)* [56], which provides 300 natural images with their human-drawn ground-truth segmentations organized by a large number of different human subjects. The metrics are defined as follows:

Recall: is computed by the following expression

$$\text{Recall} = \frac{\text{Matched}(S_{target}, S_{source})}{|S_{target}|}, \tag{8}$$

where S_{source} represents the ground truth segmentation (obtained from the average of the provided BSDS segmentations), S_{target} the partitioning to be evaluated and $|\cdot|$ indicates the total number of boundary pixels in the current segmentation. Recall (8) can be understood as the proportion of boundary pixels in S_{target} for which it is possible to find a matching boundary pixel in S_{source}.

[2] available at http://note.sonots.com/SciSoftware/NcutImageSegmentation.html

[3] available at http://www.cis.upenn.edu/∼jshi/software/

[4] available at http://fastrw.cs.sfu.ca

Precision: holds the opposite situation, that is,

$$\text{Precision} = \frac{\text{Matched}(S_{source}, S_{target})}{|S_{source}|}. \tag{9}$$

F-score: is a standard statistical measure that summarizes the recall and precision measures into a unified metric. It accounts for the trade-off between sensitivity and positive predictive values:

$$F\text{-Score} = 2\frac{\text{Precision Recall}}{\text{Precision} + \text{Recall}}. \tag{10}$$

Figure 16 presents the computation of Recall (8), Precision (9) and the F-Score (10) for the BSDS dataset. One notices that our framework is better than other methods for most of the metrics, being fairly stable in all cases. In fact, only the recall quantity leads to a similar behavior between the MS-NCut and the proposed method. Figure 17 depicts the partitioning produced by NCut, MS-NCut, and our approach when applied to illustrative images randomly extracted from the BSDS dataset [56]. Notice that the MS-NCut and our method produced much better results than the classical NCut (the ground truth is shown in the last column). In contrast to MS-NCut, our method has two advantages: it produces smooth boundaries between the segmented regions and it clusters the image into slightly wider regions, two characteristics also present in the ground truth images. Moreover, it can be seen that our approach is more robust to identify objects and structures contained in the images. For instance, the input images *monk*, *geisha*, and *horse* were better captured by our technique.

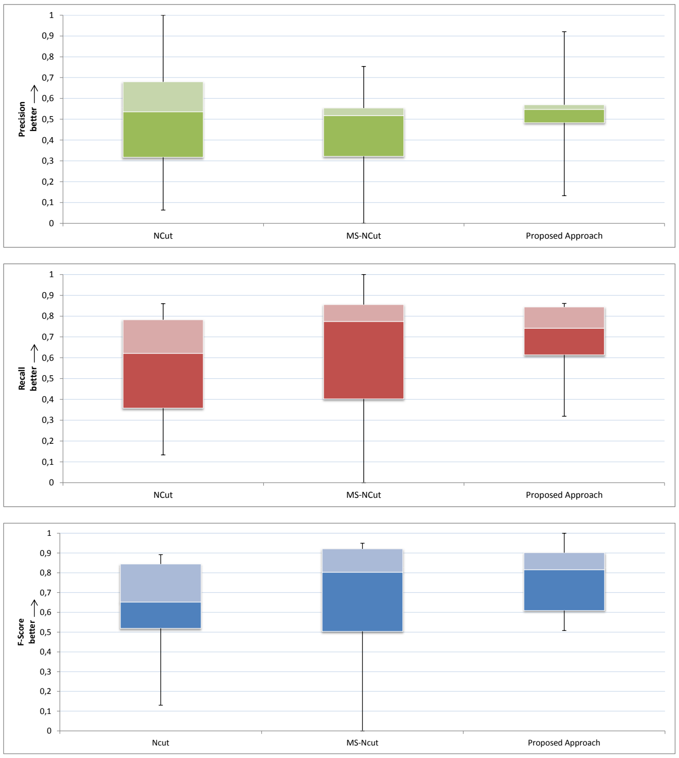


Fig. 16. Quantitative comparison for the Recall, Precision and the F-score segmentation measures. For all images, we compute the average of the recall and precision quantities for $R = 8, 9, 10, 11$ and 12 [55].

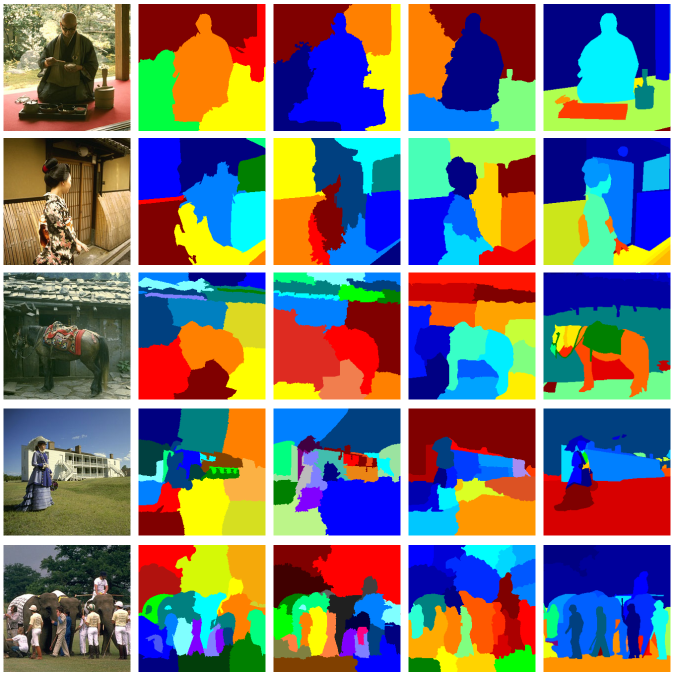
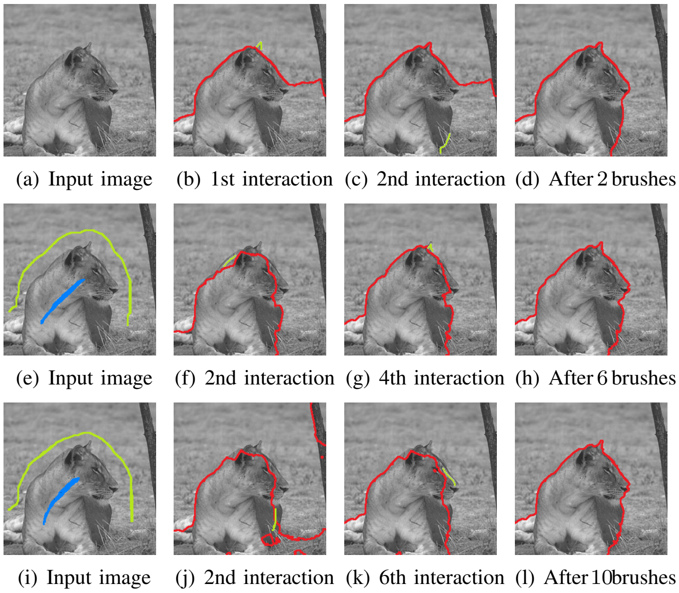


Fig. 17. From left to right: The input image, the segmentations resulting from NCut, MS-NCut, our approach and the human-drawn ground-truth.

3) Comparison with Interactive Methods: The results provided by the proposed methodology considering user intervention has been compared against two user-assisted eigenspectrum-based methods, RWS-EP [39] and RWS-EPP [40], as depicted in Figure 18. In contrast to the Random Walker-based techniques RWS-EP and RWS-EPP, our method does not require an initial user setup to produce the first segmentation result.



(a) Input image (b) 1st interaction (c) 2nd interaction (d) After 2 brushes

(e) Input image (f) 2nd interaction (g) 4th interaction (h) After 6 brushes

(i) Input image (j) 2nd interaction (k) 6th interaction (l) After 10brushes

Fig. 18. Segmentation driven by user produced by our technique (top row), RWS-EP (middle row) and RWS-EPP (bottom row). Fig. 18(a) is required by our approach, while the target image and the initial setup with 2000 seeds (blue and green pieces) must be provided by the user in random walker-based methods (Figs. 18(e) and 18(i))

TABLE I. COMPARATIVE TIMING TABLE (IN SECONDS) WITH RESPECT TO EXPERIMENT INVOLVING USER AUTONOMY (FIG. 18, IMAGE DIMENSION: 256×256). THE TIMING $+4$ MEANS THE TIME TAKEN FOR THE INCLUSION OF SEEDS.

Processing stage	RWS-EP	RWS-EPP	Our method
To produce the 1st segmentation	23 (+4)	21 (+4)	5
Average time per interaction	0.5	0.6	0.4
To produce the desired result	29.4	32.3	5.83

Notice that differently from RWS-EP and RWS-EPP, which require much more user interference to reach a pleasant result (Figs. 18(h) and 18(l)), our approach needed only two brushes to yield the desired result (Fig. 18(d)). Besides, our technique is considerably faster when compared against RWS-EP and RWS-EPP (see Table I).

III. SEEDED IMAGE SEGMENTATION

A growing number of semi-supervised image segmentation methods have been proposed in the last few years, motivated mainly by the human capability of recognizing and detecting patterns. In fact, *seeded image segmentation* figures among the most relevant image segmentation methodologies, where traditionally algorithms make use of user's prior knowledge as input data in order to operate suitably. Seed-based image segmentation methods typically rely on a given set of labeled pixels ("seeds") and on affinity weighted graphs whose nodes correspond to image pixels and edges reflect the neighborhood structure of the pixels. Edge weights encode image attributes such as texture, color or gradients and they are used to guide the propagation of the seeded labels throughout the image.

Many mathematical formulations and algorithms have been recently proposed to perform segmentation by seed propagation [57], [5], [4], [58], [59], [6], [60], [61], [62], [63], most of them making use of energy functional minimization on graphs. In terms of energy minimization formulation, [64], [65] showed that most seed-based image segmentation methods can be understood as variations of a small group of basic techniques which differ from each other in terms of their mathematical formulation, pairwise pixel distance and weight computation. In fact, those algorithms were reinterpreted as special cases of the following generalized energy functional [65]:

$$E_{PWS}(\mathbf{x}) = \sum_{i\in B} w_{Bi}^{p}|x_i - x_B|^q + \sum_{i\in F} w_{Fi}^{p}|x_i - x_F|^q + \frac{1}{2}\sum_{i\in V}\left(\sum_{j\in N(i)} w_{ij}^{p}|x_i - x_j|^q\right), \quad (11)$$

where x_i is the sought solution, that is, the saliency map w.r.t. target image, w_{ij} is the weight assigned to the edge (i,j), x_B and x_F represent the values of seeded pixels in the background and foreground, w_{Bi} and w_{Fi} are their penalizations, respectively, and the constant p and q are positive parameters. Functional (11) assumes the form of specific image segmentation algorithms when p and q vary as shown in Table II.

TABLE II. VARIATION OF THE PARAMETERS p AND q IN EQUATION (11) VERSUS CLASSICAL SEGMENTATION ALGORITHMS [65].

q\p	0	finite	∞
1	Seed collapse	*Graph Cuts*	*P. Watershed* ($q = 1$)
2	l_2 norm	*R. Walker*	*P. Watershed* ($q = 2$)
∞	l_1 norm	l_1 norm	*S. Path Forest*

Most of the work from the literature particularly focus on the four major groups mentioned in Table II, more specifically:

1) *Graph Cuts-based methods* [66], [57], [4], [58], [67];
2) *Random Walker-based methods* [5], [68], [39], [40];
3) *Watersheds-based methods* [69], [6], [64], [70], [65];
4) *Shortest Path Forest/Geodesic-based methods* [71], [72], [73], [74], [75], [76],

which minimize the same energy functional whose formulation takes into account only first-order pairwise pixels, differing only in terms of exponent values. Furthermore, most existing methods from this set rely on non-quadratic energies, thus demanding the use of sophisticated and computationally cost optimization tools. Ensuring accuracy and smooth solution is also an issue for existing techniques.

A. Laplacian Coordinates for Image Segmentation

Aiming at dealing with the issues raised above, in this section we present a new technique for seeded image segmentation, first reported in [3], which relies on the minimization of a novel quadratic energy functional defined on an affinity graph. The new approach, called *Laplacian Coordinates*, allows for user intervention while leading to smoother and accurate solutions.

The notion of Laplacian Coordinates has initially appeared in [77], [78] to address the problem of surface editing in the field of *Geometry Processing*. In contrast to most existing algorithms, in particular the four ones mentioned earlier that minimize the "distance" between pairwise pixels, the proposed approach minimizes the average of distances while controlling anisotropic propagation of labels during the segmentation process. Therefore, pixels sharing similar attributes are kept closer to each other while jumps are naturally imposed on the boundary between image regions, thus ensuring better fitting on image boundaries as well as a pretty good neighborhood preservation (on average). Moreover, the proposed formulation is guaranteed to have a unique solution, an important trait not always present in seed-based image segmentation algorithms. Other fundamental characteristic of Laplacian Coordinates is that the minimizer of the cost function is given by the solution of a constrained system of linear equations, making the algorithm quite simple to be used and coded.

The Laplacian Coordinates pipeline is very simple and comprises four main steps: *Definition of Seeds*, *Affinity Graph Building*, *Energy Functional Construction and Solution*, and *Assignment of Labels*. Figure 19 illustrates the use of the Laplacian Coordinates approach. In summary, the main contributions of the proposed technique are:

1) A novel and easy-to-implement formulation for seed-based image segmentation, which we call *Laplacian Coordinates* [3].

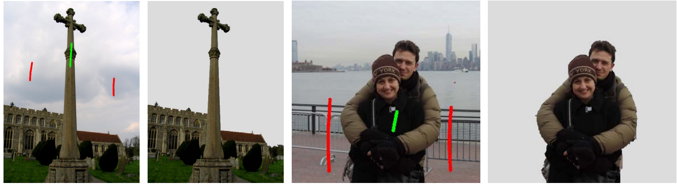


Fig. 19. Segmentations produced by the Laplacian Coordinates approach, where red and green scribes indicate the seedings.

2) Laplacian Coordinates bears several important properties such as boundary fitting, anisotropy and unique solution for the minimizer.
3) The segmentation simply consists of solving a constrained sparse linear system of equations.
4) A comprehensive set of quantitative and qualitative comparisons against state-of-art algorithms that shows the effectiveness of Laplacian Coordinates.

B. Computing the Laplacian Coordinates Energy on Graphs

Let I be a color or grayscale image. For a color image, we denote the RGB vector by $I_i = (R_i, G_i, B_i)$, which indicates the luminance of *red*, *green* and *blue* channels at pixel $P_i \in I$. For a grayscale image, I_i is the pixel intensity. As a basic tool to proceed with the segmentation task, we construct a weighted graph $G = (V, E, W_E)$, where V is the set of nodes $i \in V$ corresponding to the pixel $P_i \in I$, the edge set E corresponds to pairs of pixels locally connected in an 8-neighborhood, and $w_{ij} \in W_E$ is the weight assigned to edge (i, j) of the graph. The set $N(i) = \{j : (i, j) \in E\}$ represents the indices of the pixels P_j that share an edge with pixel P_i and the quantity

$$d_i = \sum_{j \in N(i)} w_{ij} \tag{12}$$

measures the weighted valency associated to pixel P_i.

1) Graph Weight Setup: There are many different ways to define the set of weights W_E, many of them relying on pixel intensity, gradient, scalability, saliency and contour information [34], [35], [79], [2], [42], [80]. Aiming at keeping our approach as simple as possible, we only consider pixel intensities to define the weights. More precisely, the weight $w_{ij} = w(P_i, P_j)$ assigned to each pixel pair (P_i, P_j) is computed as follows:

$$w_{ij} = \exp\left(-\frac{\beta||I_i - I_j\|_\infty^2}{\sigma}\right), \quad \sigma = \max_{(i,j)\in E} ||I_i - I_j||_\infty, \tag{13}$$

where β is a tuning constant. Notice that the weights are positive and symmetric in the sense that $w_{ij} = w_{ji}$. In terms of usage, a small constant $\epsilon = 10^{-6}$ is added into Equation (13) to avoid null weights, as suggested by [61]. Sophisticated weighting functions can also be employed to reach more refined segmentations such as the inner-product similarity metric (4).

2) Laplacian Coordinates Energy Functional: Given the affinity matrix $\mathbf{W}$ computed as in Equation (13), the set of background B and foreground F seeded pixels and their corresponding labels x_B and x_F, the following energy functional $E(\mathbf{x})$ is minimized with respect to vector $\mathbf{x}$:

$$\underbrace{\sum_{i\in B}\|x_i - x_B\|_2^2}_{\text{fidelity term}} + \underbrace{\sum_{i\in F}\|x_i - x_F\|_2^2}_{\text{fidelity term}} + \underbrace{\sum_{i\in V}\left\|d_i x_i - \sum_{j\in N(i)} w_{ij}x_j\right\|_2^2}_{\text{LC term}} \tag{14}$$

where $\mathbf{x} = (x_1, x_2, ..., x_n)^t$ is the sought solution, that is, the scalar values assigned to the pixels $(P_1, P_2, ..., P_n)$ so as to minimize the functional $E(\mathbf{x})$, n is the number of pixels and w_{ij} is computed as in Eq. (13). Without loss of generality, assume that $x_B > x_F$. Once the Energy (14) is minimized, the segmentation is accomplished by assigning background and foreground labels $y_i \in \{x_B, x_F\}, i \in V$, as follows:

$$y_i = \begin{cases} x_B, & \text{if} \quad x_i \geq \dfrac{x_B + x_F}{2} \\ x_F, & \text{otherwise} \end{cases} . \tag{15}$$

Laplacian Coordinates energy functional (14) is made up of two main components, one accounting for the 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 notation, LC energy term can be rewritten as follows:

$$\sum_{i\in V}\left\|d_i x_i - \sum_{j\in N(i)} w_{ij}x_j\right\|_2^2 = \sum_{i\in V}\left(\sum_{j\in N(i)} w_{ij}(x_i - x_j)\right)^2 =$$
$$\underbrace{\left(\sum_{j\in N(1)} w_{1j}(x_1 - x_j), \ldots, \sum_{j\in N(n)} w_{nj}(x_n - x_j)\right)}_{\mathbf{v}^t} \cdot \mathbf{v} = (\mathbf{Lx})^t(\mathbf{Lx})$$

Thus,

$$\sum_{i\in V}\left\|d_i x_i - \sum_{j\in N(i)} w_{ij}x_j\right\|_2^2 = (\mathbf{Lx})^t(\mathbf{Lx}) = \|\mathbf{Lx}\|_2^2\,, \tag{16}$$

where $\mathbf{L} = \mathbf{D} - \mathbf{W}$ is the *graph Laplacian matrix*, $\mathbf{D}$ is the diagonal matrix where $D_{ii} = d_i$ (see Equation (12)) and $\mathbf{W}$ denotes the weighted adjacency matrix of the graph, that is,

$$W_{ij} = \begin{cases} w_{ij}, & \text{if} \quad (i,j) \in E \\ 0, & \text{otherwise} \end{cases} . \tag{17}$$

Notice that the i-th component of $\mathbf{Lx}$ corresponds to the *differential/average operator*:

$$\delta_i = x_i - \frac{1}{d_i}\sum_{j\in N(i)} w_{ij}x_j, \text{ that is, } (\mathbf{Lx})_i = d_i\delta_i\,. \tag{18}$$

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

3) Minimizing the Energy Functional: Energy (14) can be modeled in a more general matricial form, that is,

$$\begin{aligned} E(\mathbf{x}) &= \sum_{i\in B}(x_i - x_B)^2 + \sum_{i\in F}(x_i - x_F)^2 + \|\mathbf{Lx}\|_2^2 \\ &= \sum_{i\in S} x_i^2 - 2\left(\sum_{i\in B} x_i x_B + \sum_{i\in F} x_i x_F\right) + c + (\mathbf{Lx})^t(\mathbf{Lx}) \\ &= \mathbf{x^t L^2 x} + \mathbf{x^t I_S x} - 2\mathbf{x^t b} + c \end{aligned}$$

$$E(\mathbf{x}) = \mathbf{x}^{\mathbf{t}}(\mathbf{I_S} + \mathbf{L^2})\mathbf{x} - 2\mathbf{x}^{\mathbf{t}}\mathbf{b} + c, \tag{19}$$

where $\mathbf{I_S}$ is a diagonal matrix such that $I_S(i,i) = 1, i \in S = B \cup F$, and zero, otherwise, $\mathbf{b}$ is the vector where $b(i) = x_B, i \in B$; $b(i) = x_F, i \in F$, and zero otherwise, and c is a constant. Quadratic form (19) has a unique minimizer since $(\mathbf{I_S} + \mathbf{L^2})$ is a symmetric and positive definite matrix. Moreover, its minimizer vector $\mathbf{x}$ is the solution of the following linear system [81]:

$$(\mathbf{I_S} + \mathbf{L^2})\mathbf{x} = \mathbf{b}. \tag{20}$$

Therefore, minimizing $E(\mathbf{x})$ is equivalent to solving the linear system (20), which, in turn, holds quite attractive properties such as symmetry, positive definiteness and sparsity. After solving Eq. (20), the segmentation is then performed by assigning a label to each image pixel according to Eq. (15).

4) Laplacian Coordinates Properties: Besides being computationally efficient, easy-to-implement and ensuring unique solution, the proposed method has additional properties that render it attractive to segment images, as discussed below.

Boundary and Constraint Fitting. The main characteristic that differs Laplacian Coordinates from other seed-based approaches is its capability to better propagate the seeds (constraint information). Figure 20 illustrates this by comparing Laplacian Coordinates against the Random Walker approach in one-dimensional case. First row of Figure 20 shows a 1D graph with 500 nodes ordered from left to right. Second row in Figure 20 shows two different distributions of edge weights: on the left, unitary weights are assigned to edges, except for edges in the middle of the graph, where weights have a distribution that decreases and gets close to zero increasing back to 1. On the right, weights are distributed similarly, but now with two picks isometrically arranged. Constraints (seeds) are imposed in the yellow and purple nodes. As one can easily see on the third row of Figure 20, Laplacian Coordinates spread the constraint information in a smoother way, taking longer to diffuse the constraint information when compared to Random Walker approach. For the sake of illustration, last row in Figure 20 presents the result of applying Laplacian Coordinates and Random Walker when weights are set equal 1.

Solution in Terms of Extended Neighborhood. An interesting interpretation of the solution of Laplacian Coordinates is that each pixel x_i is written not in terms of the first-order neighbors, but considering distant neighborhoods, instead. In mathematical terms, at an unconstrained pixel P_i, we have:

$$(\mathbf{L}\mathbf{x})_i = \frac{1}{d_i} \sum_{j \in N(i)} w_{ij} (\mathbf{L}\mathbf{x})_j \tag{21}$$

where $(\mathbf{L}\mathbf{x})_j$ is computed as in Equation (18). The solution x_i is then computed taking into account an extended neighborhood, mathematically expressed by the equation:

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

Therefore, information coming from the constraints takes longer to be diffused by the Laplacian Coordinates approach.

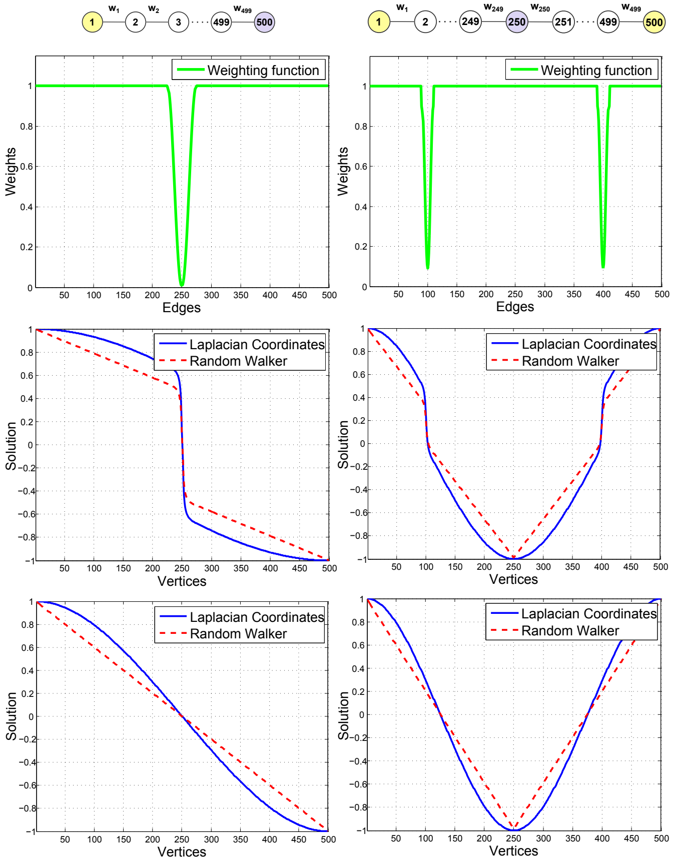


Fig. 20. Comparison between the solution obtained from Laplacian Coordinates and the classical Random Walker algorithm under the same initial conditions. Line graphs are shown in the top row with seeded vertices in yellow and purple while the corresponding edge weights are shown in the second row. The solution with and without the mentioned weights are given in the third and fourth rows.

Seeding Flexibility and Adaptability. Figure 21 shows the robustness of Laplacian Coordinates in producing different segmentations by just selecting distinct targets in the image. Notice from the two initial configurations (left and middle columns) of Fig. 21 that both objects (the boys) are accurately segmented, attesting the accuracy of the proposed approach. In fact, an even more general solution can be obtained by simultaneously seeding the two targets of the image, as depicted in the last column of Fig. 21.

Multiple-Region Segmentation. Laplacian Coordinates can also easily be extended to segment an image into several parts. This extension is carried out by simply solving $(N-1)$ system of linear equations similar to Equation (20):

$$(\mathbf{I_S} + \mathbf{L^2})\mathbf{x^{(j)}} = \mathbf{b^{(j)}}, \tag{22}$$

but setting $I_S(i,i) = 1$ for all seeded pixels in the image and specifying different $b^{(j)}$ for each one of the given labels $K_j \in K = \{K_1, K_2, ..., K_N\}$, $1 \leq j \leq (N-1)$. 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 $\mathbf{x^{(j)}}$ are bounded by $[-C, C]$, the last scalar map $\mathbf{x^{(N)}}$ is then obtained as follows:

$$x_i^{(N)} = C - \max_{1 \leq j \leq (N-1)} \{x_i^{(j)}\} . \tag{23}$$

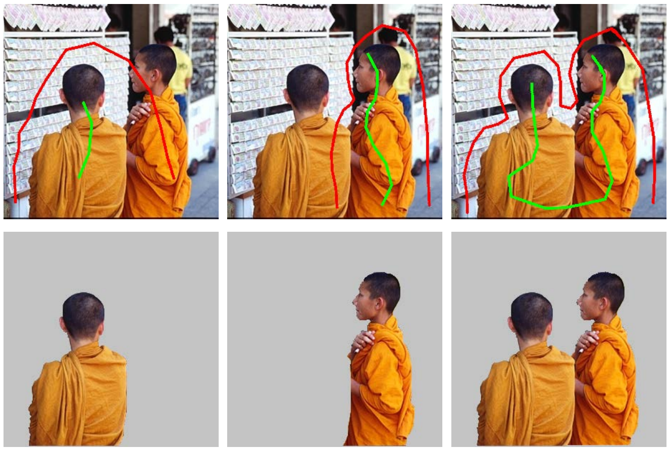


Fig. 21. Selecting different objects from the image by exploiting the seed sensitivity of the Laplacian Coordinates. First row: multiple selections are given as input to the method and Second row: the corresponding segmentations.

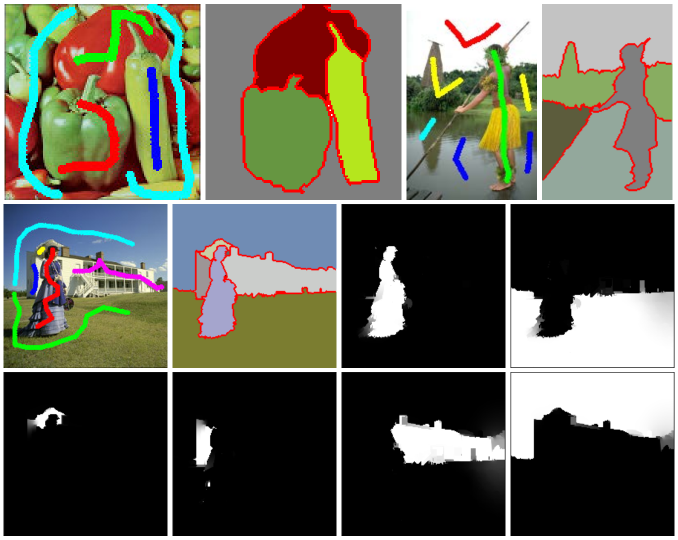


Fig. 22. Extension of the Laplacian Coordinates (14) for multiple segmentation. First row: multiple seeds are sketched as colored strokes, from which Laplacian Coordinates produced the multiple segmented regions. Middle and bottom row: sketched seeds, the final segmentation and the six solution vectors $\mathbf{x^{(j)}}$ that give rise to the multiple segmentation.

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

$$y^{(j)} = \bigcap_{\substack{p=1,\ldots,N \\ p\neq j}} \left(x^{(j)} > x^{(p)}\right), \tag{24}$$

where $>$ is computed for all pixels of the image. Figure 22 depicts the result of Laplacian Coordinates to segment multiple regions. Color strokes mark the objects (strokes with the same color correspond to the same region), from which Laplacian Coordinates generates the segmentation in multiple regions.

C. Results, Comparisons and Evaluation

In this section we provide a comprehensive experimental evaluation of Laplacian Coordinates against competing state-of-the-art techniques. We evaluate each method using multiple measures traditionally employed by the image segmentation community.

In order to perform the experimental analysis, we use one of the most popular benchmark dataset for seeded-based segmentation: the *"Grabcut" dataset* [57]. This dataset contains 50 prototype images, their ground-truth (obtained from manual human segmentation), and seeded maps marking foreground and background regions of the images. We make use of this benchmark dataset to compare the Laplacian Coordinates approach (LC) [3] against five traditional seed-based segmentation algorithms:

- Graph Cuts algorithm (GC) [57], [4];
- Power Watershed algorithm (PWS) [82], [65];
- Maximum Spanning Forest with Kruskal's algorithm (MSFK) [6], [65];
- Maximum Spanning Forest with Prim's algorithm (MSFP) [6], [65];
- Random Walker algorithm (RW) [5].

Quantitative evaluations are performed comparing the output quality in terms of object/region detection as well as the accuracy in preserving ground-truth boundaries.

1) Region Quality Metrics: We employ three distinct region quality metrics to gauge the quality of Laplacian Coordinates against others, namely

Rand Index (RI): measures the closeness between the output segmentation S and the ground-truth G by counting the number of pixel pairs that have the same label [83]. More formally, it computes the sum of the pixel pairs that share a common label in S and G and those that share distinct labels in both images, divided by the total number of pixel pairs.

Global Consistency Error (GCE): computes how much the segmentation can be interpreted as a refinement of other, forcing the local refinements to be in the same orientation [44]. In fact, GCE does not take into account the scale of images to proceed with the measurements, ensuring consistency even when comparing different scales. Lower values are better.

Variation of Information (VoI): quantifies the distance between ground-truth and segmentation in terms of their relative entropies [84]. More precisely, it quantifies the amount of randomness information in changing from one clustering to another. Moreover, VoI is properly a metric in the sense of the theory of linear algebra satisfying the positivity, symmetry and the triangle inequality [85], [86]. Values close to 0 are better.

2) Boundary Quality Metrics: We also consider in our experiments two popular metrics to measure the quality of the boundaries detected by the segmentations, that is,

Boundary Displacement Error (BDE): quantifies the average of the displacement error taking into account the amount of boundary pixels from the segmentation result that correspond to the closest boundary pixels from the ground-truth [87].

The Harmonic Average Score (F-Score): summarizes the *Recall* and *Precision* baseline metrics [55], [56], which measure how much the segmentation matches the ground-truth boundaries (see Section II-C). The matching is established in terms of the boundary pixel proximity for different values of radius R, as proposed in [55].

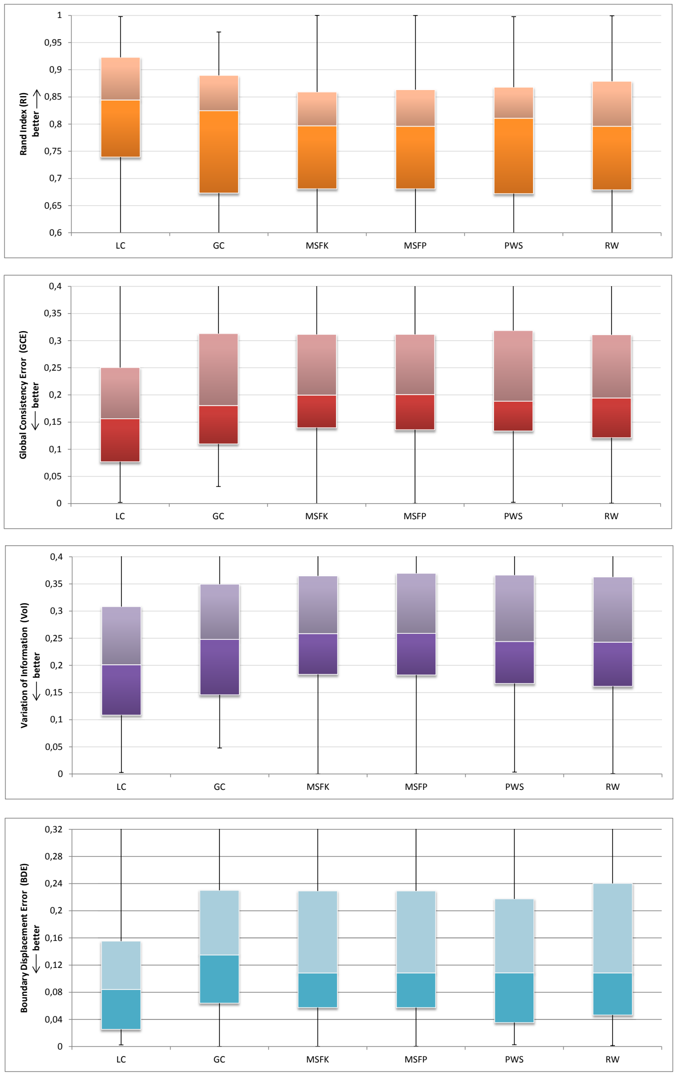


Fig. 23. Comparison of six seed-based segmentation methods regarding to RI, GCE, VoI and BDE quality metrics. In all cases, the proposed Laplacian Coordinates framework outperforms all other five evaluated techniques.

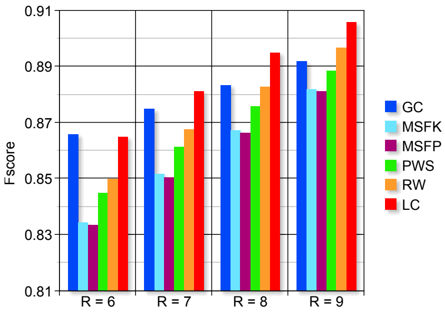


Fig. 24. *F-score* quality metric. Laplacian Coordinates is considerably better than other methods when parameter R increases.

Figure 23 summarize the quantitative results of the metrics RI, GCE, VoI and BDE for the Microsoft "Grabcut" dataset. Notice that the Laplacian Coordinates clearly outperforms the other five methods in all quality metrics, being also fairly stable to those metrics.

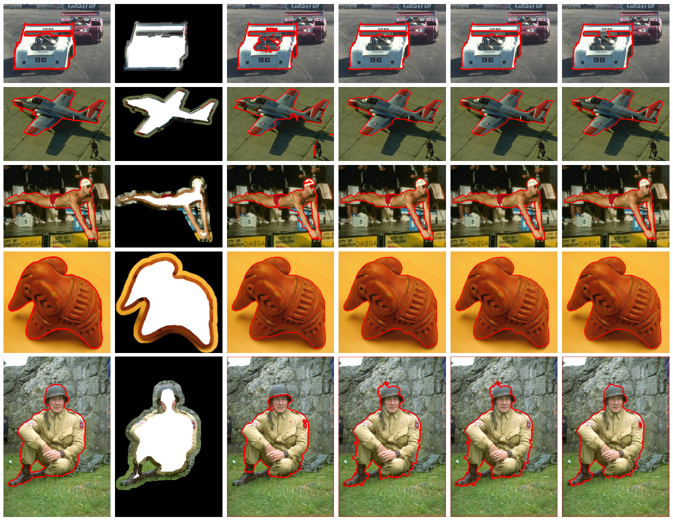


Fig. 25. From left to right: Ground-truth, the tri-map images (seeds and unknown region) provided from Grabcut dataset, and the segmentation results from GC, MSFK, MSKP and LC approach.

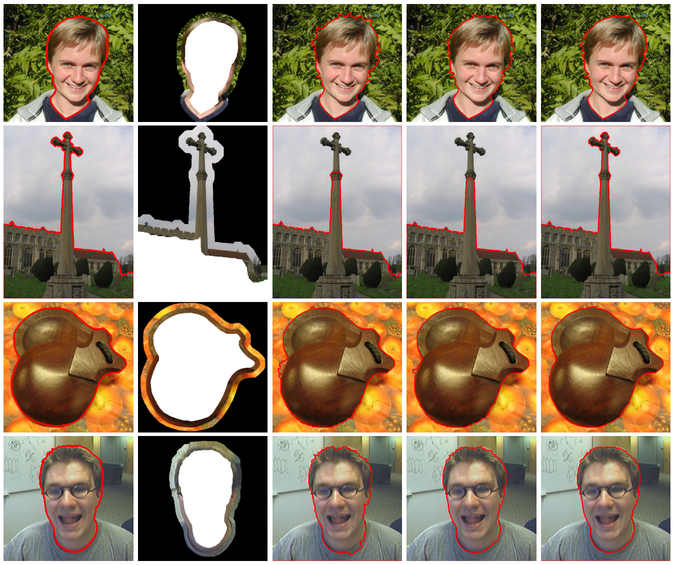


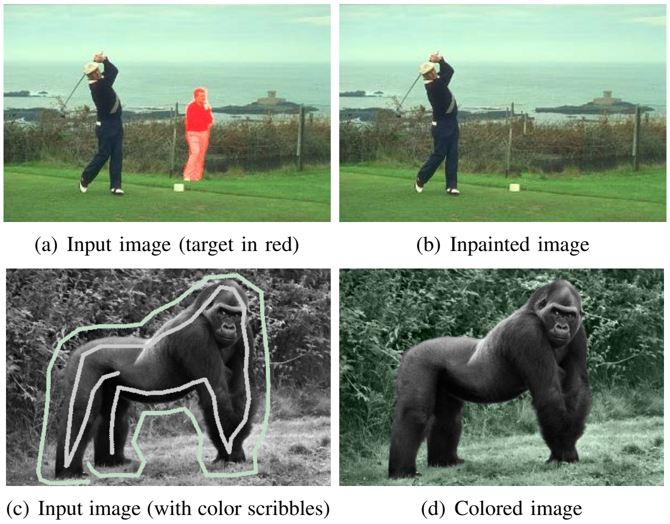
Fig. 26. From left to right: Ground-truth, tri-map images (seeds and the unknown region), and segmentations from PWS, RW and LC approach.

Regarding *F-score*, the LC approach also presents very good performance, specially when the parameter R increases. As one can see from Figure 24, the proposed approach shows a better *F-score* than other techniques, outperforming all for R equal or bigger than 7. These quantitative results show the effectiveness of Laplacian Coordinates as a seeded image segmentation method.

Figures 25 and 26 present qualitative results comparing GC, MSFK, MSFP, PWS and RW against Laplacian Coordinates. One can see that, besides accurately capturing boundaries, Laplacian Coordinates tends to simultaneously generate smoother and better fit boundary curves, a characteristic not present in any other approach, which are less accurate while still producing more jagged boundary curves.

IV. Image Inpainting and Photo Colorization

This section presents the use of the proposed segmentation approaches in two problems typically studied by the computer vision and engineering communities: *image inpainting* and *image colorization*. Image inpainting seeks to recover the natural aspect of an image where data has been partially damaged or occluded by undesired objects. Photo colorization is a computer-assisted process by which grayscale images or black-and-white films are properly colored (see Figure 27 for an illustrative example).


(a) Input image (target in red)

(b) Inpainted image

(c) Input image (with color scribbles)

(d) Colored image

Fig. 27. Image inpainting and colorization using the proposed frameworks.

Both inpainting and colorization applications share the common underlying idea that how accurately the objects to be filled or colorized are segmented. Therefore, build a good segmentation scheme into those applications is essential to ensure the success of those tasks. In this spirit, we introduce two new techniques that address the problem of image inpainting and photo colorization which make use of our segmentation tools as starting point to reach their final goal. We also compare the effectiveness of those techniques against representative state-of-the-art methods for a variety of synthetic and real images.

A. Interactive Image Inpainting using Laplacian Coordinates

Image inpainting is a modern research topic that has received a great deal of attention in recent years. It focuses on studying restoration and disocclusion processes for damaged digital images and artistic edition finalities. Methods devoted to perform inpainting can be arranged in several groups, as suggested by the surveys [88], [89], [15]. In short words, existing approaches differ in terms of pixel propagation, sensibility when synthesizing textures, and filling order criterion [88].

Although techniques for performing image inpainting vary in many fundamental aspects, a common drawback not covered by most inpainting systems is that they require the user to manually "carving" the targets to be edited. Selecting those targets consists of a meticulous process that demands great effort from users to precisely separate the targets from the image background. In fact, there are a few methods that address the problem of selecting the regions to be inpainted in an automatic or semi-supervised way. The techniques proposed by [90], [91], [92] automatically detect defects easily to identify visually but difficult to be segmented by hand. However, those algorithms can only handle a limited class of defects, thus constraining their application to specific problems. A method that covers a broader number of cases was proposed by [93], which introduces user knowledge into the pipeline of inpainting. The authors provide a user interface that allows users to guide the restoration by drawing straight lines on target regions. A similar user-steered interface was proposed by [94], which relies on the nearest neighbor correspondences among parts of the image. Their algorithm has recently been introduced in the *Adobe Photoshop Engine* as an interactive tool to perform image retouching. Since a great amount of user intervention is required to drive the whole inpainting process, the time for generating a pleasant result is considerable, specially when object to be inpainted is large.

Aiming at overcoming the issues mentioned above while providing a friendly and intuitive interface to select the regions to be recovered, we propose a novel framework (fist reported in [21]) that generates pleasant results with a reduced number of user interventions. Moreover, the non-interactive version of our technique, firstly reported in [15], outperforms existing inpainting methods in several aspects, as we show hereafter.

1) Pipeline Overview: In order to combine the accuracy and the high-adherence on image contours of the Laplacian Coordinates approach with the efficiency of the proposed inpainting technique, we design a new tool that allows users to easily brush the objects to be inpainted. As illustrated in Figure 28, our pipeline comprises four basic steps: *User-steered Selection of the Inpainting Domain*, *Pixel Filling Order Assignment*, and *Patch-based Pixel Replication*.

User-steered Selection of the Inpainting Domain. Let f be the target image, $\Omega \subset \mathbf{R}^2$ the region to be inpainted and $\partial\Omega$ its boundary. In our approach, entire objects can be easily segmented so as to avoid the meticulous election of the boundary pixels employed by tradicional approaches. The user selects a target region Ω by brushing on the object of interest and the marked pixels are used as seeds for the segmentation process. The image background must also be roughly marked to properly settle optimization constraints for the Laplacian Coordinates. Managing multiple regions is also allowed, since Laplacian Coordinates enables multiple segmentations simultaneously. Moreover, user can recursively steer the resulting partition towards reaching a higher quality result. Figure 28 (left) illustrates the above-mentioned scheme.

Filling Order Assignment. Our inpainting procedure starts by computing a mapping image u (cartoon image) from f using the anisotropic diffusion equation proposed in [95]. Similar to [54], [18], component u is obtained by numerically solving the following partial differential equation:

$$\frac{\partial f^{(t)}}{\partial t} = g|\nabla f^{(t)}|div\left(\frac{\nabla f^{(t)}}{|\nabla f^{(t)}|}\right) - (1-g)(f^{(t)} - f), \quad (25)$$

where $f^{(t)}$ is the scaled version of f, $g = g(|\nabla G_\sigma * f^{(t)}|)$ is an edge detection function, G_σ represents the gaussian function and σ is a tuning parameter. Figure 29 shows the original image, its cartoon version and the gradient fields computed from the rectangular samples highlighted in both images. As one can observe, the field derived from the cartoon image is better behaved than the one computed directly from f.

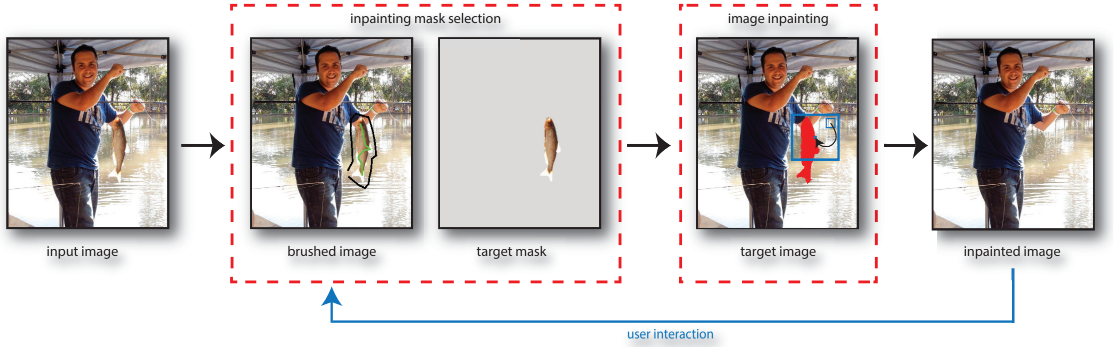
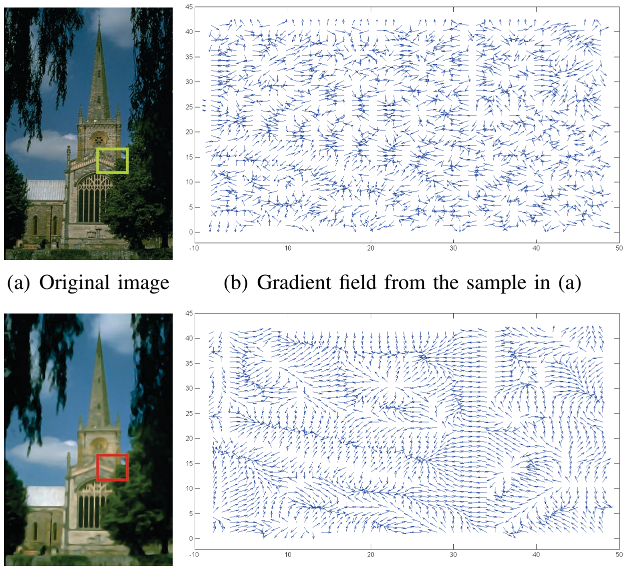


Fig. 28. Pipeline of our interactive inpainting framework.



(a) Original image (b) Gradient field from the sample in (a)

(c) Carton image (d) Gradient field from the sample in (b)

Fig. 29. Representation of the gradient field for an illustrative image.

The use of the cartoon image u allows us to embed image isophotes as well as image structures into the mechanism that computes the filling order of the damaged pixels. This mechanism is computed in terms of a new priority measure:

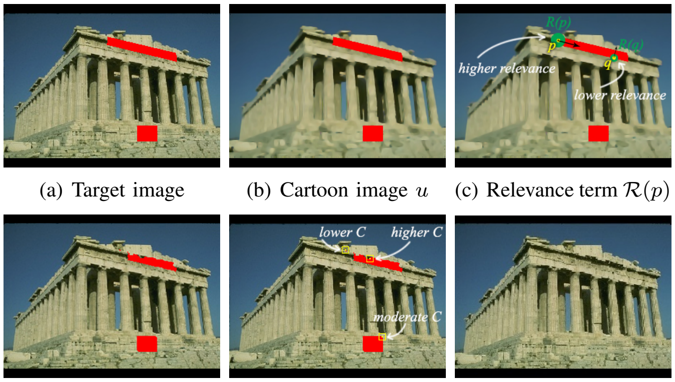
$$\mathcal{P}(p) = \mathcal{R}(p) \cdot \mathcal{C}(p), \quad p \in \partial\Omega, \tag{26}$$

where $\mathcal{R}(p)$ and $\mathcal{C}(p)$ represent the *Relevance* and *Biased Confidence* terms:

$$\mathcal{R}(p) = |\nabla(\Delta u_p) \cdot \overrightarrow{d}_p|, \qquad \overrightarrow{d_p} = \frac{\nabla^{\perp} u_p}{|\nabla^{\perp} u_p|}, \tag{27}$$

$$\mathcal{C}(p) = \left(\frac{\sum\limits_{q \in H_m(p) \cap (D-\Omega)} C(q)^{\frac{1}{k}}}{|H_m(p)|} \right)^k, \tag{28}$$

with $|H_m(p)|$ denoting the size of a squared block $m \times m$ centered at pixel p, and $k > 0$ is the bias parameter. Relevance term $\mathcal{R}$ computes the isophote direction from $\partial\Omega$. In fact, Expression (27) is similar to the transport equation proposed in [96], since it takes into consideration the directional derivative of a nontextured image for isophote propagation: $\mathcal{R}(p) = \left|\frac{\partial(\Delta u_p)}{\partial \overrightarrow{d_p}}\right|$. Biased confidence term (28) allows us to balance the filling order mechanism according to k parameter. For an illustration of $\mathcal{R}$ and $\mathcal{C}$, see Figure 30.



(a) Target image (b) Cartoon image u (c) Relevance term $\mathcal{R}(p)$

(d) Isophote completion (e) Conf. term $\mathcal{C}(p)$ (f) Inpainted image

Fig. 30. Illustration of the priority filling order mechanism (Equation (26)).

Block-based pixel replication. In this stage we are able to allocate the most suitable patch of pixels from the sampling region $\Lambda\Omega_p$ to the neighborhood of $p \in \partial\Omega$. Given a pixel $p \in \partial\Omega$, we define the region $H_L(p)$ from $\Lambda\Omega_p$ (see Fig.31(a)). Our algorithm then makes use of a cartoon-based metric to compare the fixed patch $H_n(p)$ with all candidate patches $H_n(q)$ inside $\Lambda\Omega_p$. More precisely, the optimal patch $H_n(\widehat{q})$ is the one which minimizes the distance between $H_n(p)$ and $H_n(q)$ w.r.t. a given metric. A smaller patch $H_m(\widehat{q})$ is then selected from $H_n(\widehat{q})$ and its valid pixels $H(\widehat{q})$ are placed in corresponding pixels in the neighborhood of p, namely $H(p)$ (see Fig.31(b)-(c)).

Let $\mathbf{p} = (f_{p_1}, f_{p_2}, ..., f_{p_l})$, $\mathbf{q} = (f_{q_1}, f_{q_2}, ..., f_{q_l})$ be the column vectors in $\mathbb{R}^l$, $l < n^2$, containing the intensities of the given pixels on $H_n(p)$ and in corresponding positions on $H_n(q)$. Aiming at measuring the distance between $H_n(p)$ (target) and $H_n(q)$ (candidate) blocks, we propose the following metric:

$$d(\mathbf{p},\mathbf{q}) = \frac{||\mathbf{p}-\mathbf{q}||_{\Delta U}}{\sqrt{||\mathbf{p}||^2_{\Delta U} + ||\mathbf{q}||^2_{\Delta U}}}, \; ||\mathbf{p}||_{\Delta U} := \sqrt{\mathbf{p}^T \Delta U \mathbf{p}}, \tag{29}$$

with ΔU being a diagonal matrix defined by the Laplacian of cartoon u: $\Delta U_{ii} = \Delta u_{p_i}, p_i \in H_n(p) \cap \Lambda\Omega_p$. Metric

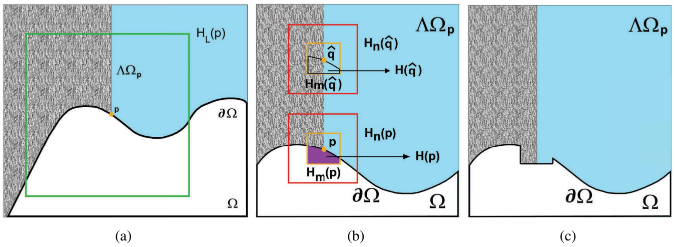


Fig. 31. Illustrative sketch of the dynamic sampling and the completion process. (a) $\Lambda\Omega_p$ (gray and blue parts) is the region inside $H_L(p)$ (green square) which provides candidate pixels. (b) Comparison between content of patches $H_n(p)$ and $H_n(\hat{q})$ (optimal patch) and (c) result after copying the information of interest.

(29) assigns higher weights for pixels located on the edges of the Laplacian operator of u. The weights of the valid pixels in Equation (29) are defined from the Laplacian of u, Δu, and then embedded into the distance computation of the pixel blocks.

2) Experimental Results and Comparisons: In this section we present a comparative study against state-of-the-art inpainting methods when taking the inpainting domain as input data. Best parameters were tuned according to original papers as well as the author's implementation available at their personal websites for the following methodologies: *Texture synthesis* [97], *Cartoon/texture decomposition-based inpainting* [98], *Exemplar-based inpainting* [99], *Optimization-based inpainting* [100] and *Sparsity-based inpainting* [101], [102], [103], [14], [104].

Comparison with Pure Texture Synthesis. Figure 32(a) reproduces a synthetic image made up of six different groups of textures, in which the inpainting region comprises a large amount of pixels. The challenge is to precisely reconstruct the boundary regions between distinct textures. It is clear in Fig. 32(b)-(c) that our framework outperforms the texture synthesis-based approach [97].

Comparison with Cartoon-Texture Inpainting. The experiment depicted in Figure 32(d)-(f) presents comparative results with the cartoon-texture inpainting algorithm [98]. Fig. 32(d) shows the input image and Fig. 32(e) depicts the resulting inpainting by [98]. Notice that some linear artifacts (inside the water area) were generated during the progress of the restoration. Our framework (Fig.32(f)), however, better recovers the natural aspect of the image.

Comparison with Exemplar-based Inpainting. Figure 32(g) shows a high-quality image. The result obtained by [99] (Fig. 32(h)) exhibits some artifacts creating inconsistent color tonalities. Our technique (Fig. 32(i)), however, captures fine details such as the layers of the "human eye".

Comparison with Optimization-based Inpainting. Object removal problem is investigated in this experiment. Fig. 32(j) presents a classical example of occluded image, where the goal is to accurately recover the gate parts hidden by the statue (Fig. 32(k)) while maintaining the regular aspect of the image. One can see that the inpainting method by [100] (Fig. 32(l)) results in a non-realistic image, as some parts of the image are overly smoothed. The proposed method, however, presents a more natural outcome (Fig. 32(m)), reconstructing the broken lines in a more satisfactory manner.

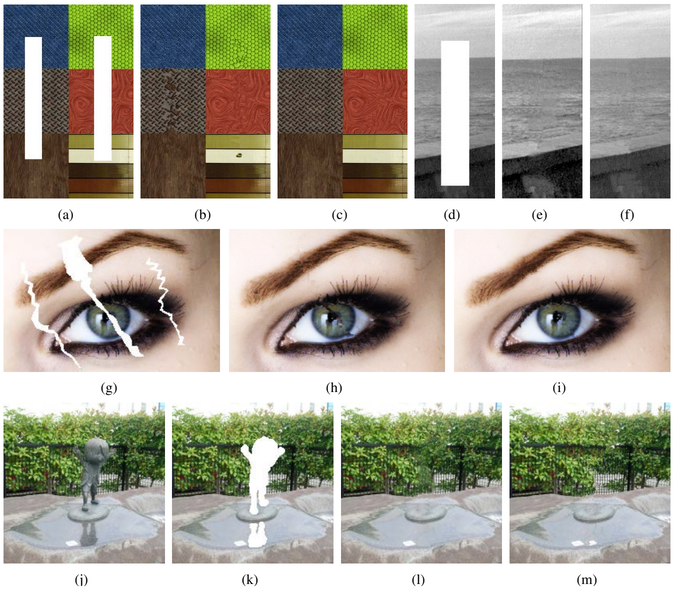


Fig. 32. Comparison with [97] (*pure texture synthesis*), [98] (*cartoon/texture inpainting*), [99] (*exemplar-based*) and [100] (*optimization-based*).

Comparison with Missing Block Completion Techniques. Finally, we finish this section performing both qualitative and quantitative evaluations against a variety of methods that address the problem of missing block completion. From left-to-right, we show in Figure 33 the input images and the inpainting results obtained by [101], [102], [103], [14], [104]. All examples are tricky to handle due to the predominance of structures and textures around the region to be filled. Notice that the algorithms [102] and [103] produce blurring effect on the images while the method [101] generates some artifacts in the outputs. The results reached by [14] as well as the method [104] are visually better than others but they still suffer from smoothing effect. In contrast, our technique leads to non-blurred completion and accurately recovers isophotes and pure texture regions, producing a pleasant and more realistic result.

For sake of quantitative comparison, PSNR (*Peak Signal-to-Noise Ratio*) between recovered and original images from Figure 33 have been considered (see Table III). Individual as well as the mean of PSNR values show that the proposed method quantitatively outperforms the others.

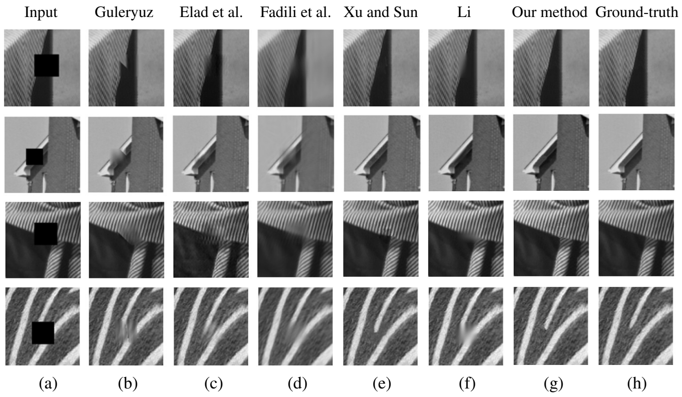


Fig. 33. Comparison with sparse representation-based inpainting methods.

TABLE III. QUANTITATIVE EVALUATION USING PSNR (IN dB) FOR ALL COMPARATIVE IMAGES FROM FIGURE 33.

Image	[102]	[101]	[103]	[14]	[104]	**Ours**
Tissue	20.41	22.43	22.16	23.53	22.21	25.02
Eaves	16.15	22.85	17.86	28.45	26.77	29.30
B.part	18.39	19.20	17.85	23.07	23.61	24.43
Fur	16.46	19.49	20.67	20.87	18.43	21.55
Mean	**17.85**	**20.99**	**19.64**	**23.98**	**22.75**	**25.08**

B. Interactive Photo Colorization via Laplacian Coordinates

Colorization is a computer-supervised process by which colors are imparted to grayscale images or to black-and-white films. It has been widely used in photo editing and scientific illustration, to modernize old films and to enhance the visual appear of an image. Traditionally, colorization is tedious, time consuming and requires artistic skills to precisely add appropriate colors to a grayscale image.

Aiming at making the colorization process simpler and less laborious, several computational systems have been proposed in the last decade, which can be roughly divided into two classes: *Example-based* [105], [106], [16], [17] and *Scribble-based* [107], [108], [109], [110], [111], [17]. Example-based methods accomplish the colorization process by matching the luminance of the grayscale image with the luminance of a reference color image used to drive the colorization. In scribble-based methods, the user guides the colorization by defining colored strokes onto the grayscale image. Due to the flexibility to operate arbitrary colorizations and the non-requirement for a reference image, scribble-based strategy has performed better than the example-based one in recent years [112]. This trend has been observed especially due to the simplicity of scribble-based methods which basically relies on an interactive interface in order to work.

The classical work by [107] is a good representative of scribble-based approach. Levin's method aims at optimizing the color of all image pixels using the scribbles as constraints. Although it shows good results for various types of images, Levin's method tends to propagate colors beyond the texture boundaries, thus resulting in unpleasant colorizations. The technique proposed by [108] employs adaptive edge detection so as to prevent colors from going beyond region boundaries. Further improvements have been proposed by [109], who present a faster scribble-based color optimization technique that relies on chrominance blending to perform the colorization. [110] and [111] employ texture continuity to colorize manga-cartoons and natural images, respectively. Despite good results, most existing scribble-based approach require intensive user involvement, especially when the image contains complex structures or has different texture patterns, which can demand lots of scribbles until acceptable outcomes are reached. In a more recent work, [17] has introduced an innovative user-based interface namely *ProjColor* that relies on a simple drag-and-drop manipulation of the badly colorized pixels using multidimensional projection as an recursive tool.

In this work we propose a new approach for colorizing grayscale images that relies on a scribble-based interface to replace the excessive provision of user strokes typically employed by existing scribble-driven methods. Moreover, the proposed approach holds the good segmentation properties derived from Laplacian Coordinates [3]. Since Laplacian Coordinates is used to precompute a prior segmentation of the input monochromatic image, our framework leads to pleasant results and requires just a few user interventions to colorize the image. As we shall show, by one-shot stroking the image, the user can colorize complex textured areas quite easily, preserving region edges and preventing the addition of new scribbles. In summary, the main contributions of this work are:

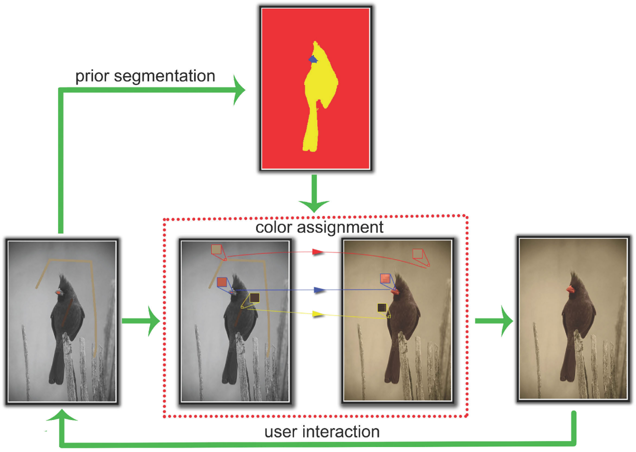


Fig. 34. Pipeline of our colorization framework.

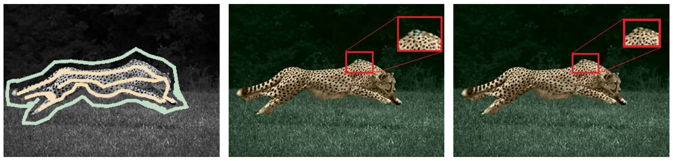
1) A novel interactive image colorization technique that combines the accuracy and the effectiveness of the Laplacian Coordinates approach with a fast color matching scheme to colorize images.
2) An efficient system that allows for recursively colorizing the image by reintroducing new seeds to reach more pleasant results.
3) The new method is easy-to-implement and it requires just a small amount of user intervention to quickly reach a good result.

1) Pipeline Overview: As illustrated in Figure 34, the proposed colorization pipeline comprises three main steps, namely, *User-driven Prior Segmentation*, *Color Assignment* and *Interactive Recolorization*. First, color scribbles given by the user are taken as constraints to the Laplacian Coordinates approach aiming at generating a prior segmentation. The partitioning obtained is then used to promote color transfer between input scribbles and image segments. Badly colored regions can be updated by interacting with the Laplacian Coordinates segmentation interface. Details of each stage of the pipeline are presented in the following sections.

User-driven Prior Segmentation. In this stage we use Laplacian Coordinates to assist the colorization process by fragmenting the image into multiple regions. Color labels are manually chosen by the user and freely spread inside representative image regions. Those labels are then designed to condition the linear system of equations formed by the Laplacian Coordinates approach. After solving the constrained linear system, the partitions obtained are used to support the next stage, *Color Assignment*. Figure 34 (up step) illustrates the procedure described above.

Color Assignment. The color assignment modulus is responsible for propagating the colors chosen by the user to the partitions generated by the LC segmentation. The propagation mechanism is accomplished as follows: given the set of color labels provided during the segmentation stage, we first convert those labels to $L\alpha\beta$ color system. Next, coordinates α and β are then copied to the uncolored pixels in the specified image segment. Similar procedure is performed until colorizing the remaining partitions (see the middle step in Figure 34).

Interactive Manipulation of the Colorization Result. One of the main contributions of the proposed framework is to exploit the flexibility provided by the LC scheme to interactively modify the colorization. Laplacian Coordinates enables an interactive tool that allows for repartitioning data by inserting new seeded pixels. In fact, if the result is not satisfactory, the user can select badly colored pixels, turning them into a different color label that can be reintroduced into the LC linear system to partition the image and, thereby, improve the resulting colorization. Figure 35 illustrates the need for user intervention. Notice that a few pixels close to the edge between the panther and the background grass were not colored properly as highlighted in Fig. 35(b). User can then provide an additional color scribble to the region with badly colorized pixels creating new constraints for the Laplacian Coordinates and, thus generating a better result as shown in Fig. 35(c).



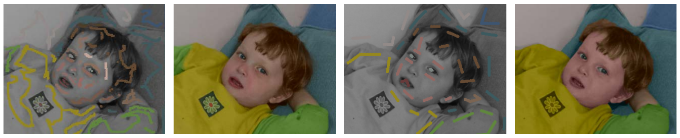
(a) Initial scribbles (b) The resulting colorization (c) Colorization after user interaction

Fig. 35. The use of our framework when allowing for user intervention.

2) Results and Comparisons: In order to confirm the effectiveness of our methodology, we provide comparisons against the well-established scribble techniques [107], [109], [112].

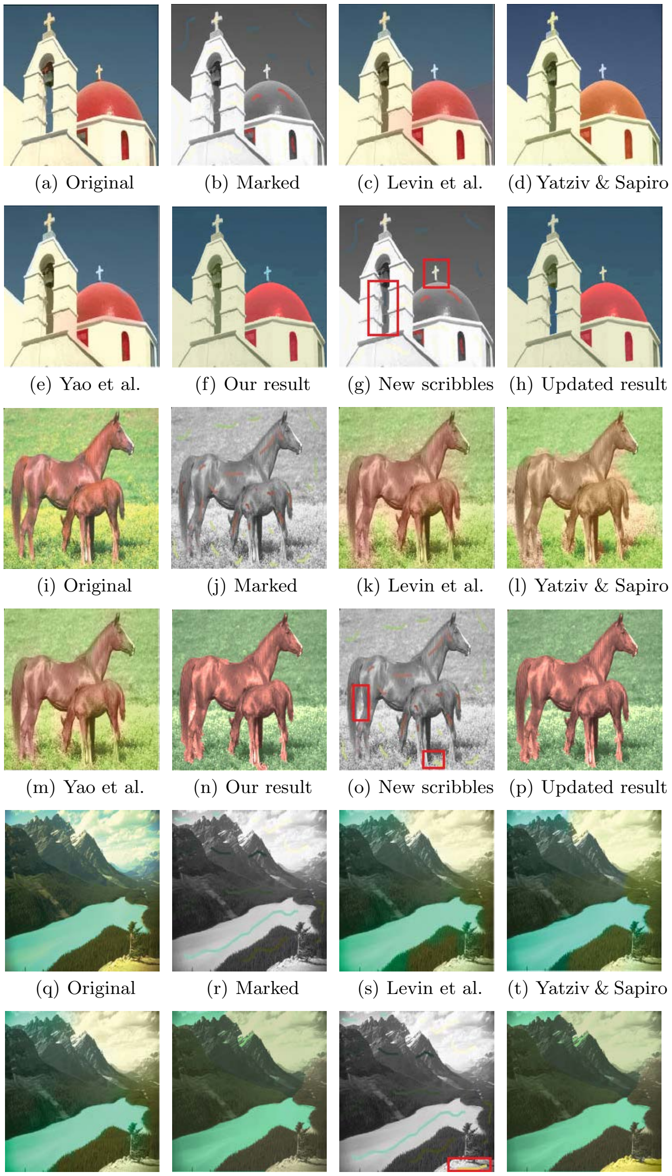
Figure 36 illustrates the capability of our method to perform colorization from a few user intervention. The seeding interface provided by Laplacian Coordinates approach is simpler than the traditional scribble-based employed by [107], as the user does not need to spread an excessive number of scribbles in the whole image to reach a reasonable result. In addition to its simplicity and ease of use, the colorization mechanism based on the Laplacian Coordinates produces pleasant results.

The experiment presented in Figure 37 establishes comparisons between the proposed technique and scribble-based methods [107], [109], [112]. Colorizations produced by [107] and [109] smoothed the images considerably almost all cases while the outcomes obtained by [112] and our technique have produced more refined results. By reintroducing just a small amount of seeds in the reference images in Figs.(g)-(o)-(w), one can see that our approach is quite flexible in capturing intrinsic details of the image such as pieces surrounded by image segments, a characteristic not present in the algorithm [112]. For sake of quantitative comparison, PSNR between the colorizations and original images in Fig. 37 were computed and summarized in Table IV. Notice from the average PSNR in the last row of Table IV that our approach outperforms others.



(a) Scribbles used by Levin's method (b) Levin's result from (a) (c) Scribbles used by our method (d) Our result from (c)

Fig. 36. Comparison between [107] and our framework.



(a) Original (b) Marked (c) Levin et al. (d) Yatziv & Sapiro

(e) Yao et al. (f) Our result (g) New scribbles (h) Updated result

(i) Original (j) Marked (k) Levin et al. (l) Yatziv & Sapiro

(m) Yao et al. (n) Our result (o) New scribbles (p) Updated result

(q) Original (r) Marked (s) Levin et al. (t) Yatziv & Sapiro

(u) Yao et al. (v) Our result (w) New scribbles (x) Updated result

Fig. 37. Comparison with [107], [109], [112] and our method.

TABLE IV. PSNR COMPUTATION FOR IMAGES FROM FIG. 37.

Image	[107]	[109]	[112]	**Ours**	**Updated**
Church	35.06	34.38	35.54	37.00	37.43
Horse	29.67	28.62	30.52	31.84	31.87
River	31.82	31.48	32.18	32.95	33.44
Average	**32.18**	**31.49**	**32.75**	**33.93**	**34.25**

V. Conclusion

This thesis has addressed the fundamental problem of image segmentation from different perspectives. More specifically, we focus our attention on studying spectral methods in the context of graph-based image representation. We have also presented a novel Laplacian-based energy minimization for image segmentation and two new algorithms that depend on segmentation to perform image inpainting and colorization.

The combination cartoon-texture decomposition and spectral cut turned out to be a quite efficient methodology for image segmentation. Moreover, the proposed inner product-based weight assignment mechanism has produced more accurate results than the exponential weighting function used by other spectral clustering methods. The qualitative and quantitative analysis clearly showed the effectiveness of the proposed technique, surpassing, in terms of accuracy and smoothness, representative spectral cut-based methods. Furthermore, the flexibility as to user intervention is an important trait of our method, which enables the user to fix the segmentation locally.

The Laplacian Coordinates introduced in Section III is a novel seed-based image segmentation technique which has several advantages when compared with other methods. Besides its simple mathematical formulation, Laplacian Coordinates is easy to implement, guarantees a unique solution, and outperforms existing methods with respect to well established quantitative measures popularly used in the context of image segmentation. Laplacian Coordinates also holds high accuracy in terms of image boundary fitting capability, rendering it an interesting and compelling seed-based segmentation method.

The new inpainting methodology presented in the last Section operates by copy-to-paste blocks to recover real and synthetic images containing a large variety of textures and structure parts. The combination between interactive image segmentation, image decomposition and transport equation were rearranged so as to provide a robust interactive interface that allows for user involvement while managing the hierarchy of the restoration task. Images of different complexity levels were evaluated against state-of-the-art methods with the purpose of assessing the efficiency of the proposed approach. Our segmentation apparatus has also been successfully used as a basic tool for the photo colorization application. In fact, besides enabling a local modification of badly colored regions, the proposed method turned out to be robust when dealing with real-world images. In summary, flexibility and effectiveness render the proposed method one of the most attractive alternatives in the context of image colorization.

Acknowledgment

Special thanks for Prof. Luis Gustavo Nonato (advisor, ICMC-USP), Prof. Gabriel Taubin (co-adivor, Brown University) and Prof. Alex J. Cuadros Vargas (UCSP) for their valuable comments and ideas to improve the quality of this PhD work. This research has been funded by FAPESP (grants #09/17801-0 and #13/07375-0), CNPq and NSF-USA.

References

[1] F. Chung, *Spectral Graph Theory*. CBMS, Reg. Conf. S. in Mathematics, Ameciran Mathematical Society, 1997.

[2] W. Casaca, A. Paiva, E. Gomez-Nieto, P. Joia, and L. G. Nonato, "Spectral image segmentation using image decomposition and inner product-based metric," *Journal of Mathematical Imaging and Vision*, vol. 45, no. 3, pp. 227–238, 2013.

[3] W. Casaca, L. G. Nonato, and G. Taubin, "Laplacian coordinates for seeded image segmentation," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014, pp. 384–391.

[4] Y. Boykov and G. Funka-Lea, "Graph cuts and efficient n-d image segmentation," *International Journal of Computer Vision*, vol. 70, no. 7, pp. 109–131, 2006.

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

[6] 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, 2009.

[7] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, pp. 888–905, 2000.

[8] S. Maji, N. Vishnhoi, and J. Malik, "Biased normalized cuts," in *IEEE Conference on Computer Vision and Patter Recognition (CVPR)*, 2011, pp. 2057–2064.

[9] C. J. Taylor, "Towards fast and accurate segmentation," in *IEEE Conf. on Computer Vision and Pattern Recognition*, 2013, pp. 1–8.

[10] W. Casaca, A. Paiva, and L. G. Nonato, "Spectral segmentation using cartoon-texture decomposition and inner product-based metric," in *24th Conference on Graphics, Patterns and Images (SIBGRAPI), Best paper award.* IEEE Computer Society, 2011, pp. 266–273.

[11] F. Yi and I. Moon, "Image segmentation: A survey of graph-cut methods," in *International Conference on Systems and Informatics (ICSAI)*, 2012, pp. 1936–1941.

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

[13] W. Casaca, "Graph laplacian for spectral clustering and seeded image segmentation," 161 pgs, PhD Thesis, University of São Paulo, São Carlos, Brazil, 2014.

[14] Z. Xu and J. Sun, "Image inpainting by patch propagation using patch sparsity," *IEEE Transactions on Image Processing*, vol. 19, no. 5, pp. 1153–1165, 2010.

[15] W. Casaca, M. P. Almeida, M. Boaventura, and L. G. Nonato, "Combining anisotropic diffuison, transport equation and texture synthesis for inpainting textured images," *Pattern Recognition Letters (PRL)*, vol. 36, pp. 36–45, 2014.

[16] R. Irony, D. Cohen-Or, and D. Lischinski, "Colorization by example," in *Proc. of the Eurographics Symp. on Rendering*, 2005, pp. 201–210.

[17] W. Casaca, E. Gomez-Nieto, C. de O.L. Ferreira, G. Tavares, P. Pagliosa, F. Paulovich, L. G. Nonato, and A. Paiva, "Colorization by multidimensional projection," in *25th Conference on Graphics, Patterns and Images (SIBGRAPI)*, 2012, pp. 32–38.

[18] W. Casaca, M. Boaventura, and M. P. de Almeida, "Classifying texture and free-textured content from images using nonlinear anisotropic diffusion," in *5th Conf. on Mathematical Analysis*, 2011, pp. 210–214.

[19] W. Casaca, M. P. Almeida, and M. Boaventura, "Denoising textured images via regularized anisotropic diffusion," in *An Introductory Guide to Image and Video Processing*, Z. N. Akshaya Mishra and Z. Shahid, Eds. Brisbane, Australia: Iconcept Press Ltd., 2013, pp. 48–71.

[20] W. Casaca, M. Colnago, and L. G. Nonato, "Interactive image colorization using laplacian coordinates," in *16th Int. Conf. on Computer Analysis of Images and Patterns (CAIP)*, 2015, pp. 1–12.

[21] W. Casaca, D. Motta, G. Taubin, and L. G. Nonato, "A user-friendly interactive image inpainting framework using laplacian coordinates," in *IEEE International Conference on Image Processing (ICIP) (accepted for publication).* IEEE Computer Society, 2015, pp. 1–5.

[22] P. Joia, E. Gomez-Nieto, J. Batista Neto, W. Casaca, G. Botelho, A. Paiva, and L. Gustavo Nonato, "Class-specific metrics for multidimensional data projection applied to cbir," *The Visual Computer*, vol. 28, no. 10, pp. 1027–1037, 2012.

[23] E. Gomez-Nieto, W. Casaca, L. G. Nonato, and G. Taubin, "Mixed integer optimization for layout arrangement," in *26th Conference on*

Graphics, Patterns and Images (SIBGRAPI), Best paper award in graphics and visualization, 2013, pp. 115–122.

[24] E. Gomez-Nieto, F. S. Roman, P. Pagliosa, W. Casaca, E. S. Helou, M. C. F. de Oliveira, and L. G. Nonato, "Similarity preserving snippet-based visualization of web search results," *IEEE Transactions on Visualization and Computer Graphics*, vol. 20, pp. 457–470, 2014.

[25] E. Gomez-Nieto, W. Casaca, I. Hartmman, and L. G. Nonato, "Understanding large legal datasets through visual analytics," in *28th Conference on Graphics, Patterns and Images (SIBGRAPI, WVis)*. IEEE Computer Society, 2015, pp. 1–4.

[26] B. Mohar, "Some applications of laplacian eigenvalues of graphs," *Graph Symetric: Algebraic Methods and Applications*, vol. 497, pp. 225–275, 1997.

[27] D. A. Spielman, "Spectral graph theory and its applications," in *Proc. of the 48th Annual IEEE Symposium on Foundations of Computer Science*, 2007, pp. 29–38.

[28] W. Tao, H. Jin, and Y. Zhang, "Color image segmentation based on mean shift and normalized cuts," *IEEE Systems, Man, and Cybernetics*, vol. 37, pp. 1382–1389, 2007.

[29] M. Carvalho, A. Costa, A. Ferreira, and R. C. Junior, "Image segmentation using component tree and normalized cut," in *23th Conference on Graphics, Patterns and Images (SIBGRAPI)*, 2010, pp. 317–322.

[30] M. Carvalho, A. Ferreira, and A. Costa, "Image segmentation using quadtree-based similarity graph and normalized cut," *Lectures Notes in Computer Science*, vol. 6419, pp. 329–337, 2010.

[31] X. Ma, W. Wan, and J. Yao, "Texture image segmentation on improved watershed and multiway spectral clustering," in *IEEE Int. Conf. on Audio, Language and Image Processing*, 2008, pp. 1693–1697.

[32] A. Sáez, C. Serrano, and B. Acha, "Normalized cut optimization based on color perception findings: A comparative study," *Machine Vision and Applications*, pp. 1–11, 2014.

[33] S. Yu, "Segmentation using multiscale cues," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2004, pp. I–247–I–254.

[34] T. Cour, F. Bénézit, and J. Shi, "Spectral segmentation with multiscale graph decomposition," in *IEEE Computer Vision and Pattern Recognition (CVPR)*, 2005, pp. 1124–1131.

[35] W. Cai and A. C. S. Chung, "Shape-based image segmentation using normalized cuts," in *Proc. of IEEE Int. Conference on Image Processing (ICIP)*, 2006, pp. 1101–1104.

[36] F. Sun and J.-P. He, "A normalized cuts based image segmentation method," in *International Conference on Information and Computing Science (ICIC)*, 2009, pp. 333–336.

[37] M. Maire and S. X. Yu, "Progressive multigrid eigensolvers for multiscale spectral segmentation," *IEEE International Conference on Computer Vision (ICCV)*, vol. 0, pp. 2184–2191, 2013.

[38] L. Grady, "Multilabel random walker image segmentation using prior models," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2005, pp. 763–770.

[39] L. Grady and A. K. Sinop, "Fast approximate random walker segmentation using eigenvector precomputation," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2008, pp. 1–8.

[40] S. Andrews, G. Hamarneh, and A. Saad, "Fast random walker with priors using precomputation for interactive medical image segmentation," in *Lecture Notes in Computer Science, Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, 2009, pp. 9–16.

[41] D. Tolliver and G. L. Miller, "Graph partitioning by spectral rounding: Applications in image segmentation and clustering," in *IEEE Conf. on Computer Vision and Pattern Recognition*, 2006, pp. 1053–1060.

[42] 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, 2013.

[43] I. Koutis, G. L. Miller, and D. Tolliver, "Combinatorial preconditioners and multilevel solvers for problems in computer vision and image processing," in *International Symposium on Visual Computing (ISVC)*, 2009, pp. 1067–1078.

[44] 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 *IEEE International Conference on Computer Vision (ICCV)*, 2001, pp. 416–423.

[45] L. Demanet and L. Ying, "Wave atoms and sparsity of oscillatory patterns," *Applied and Computational Harmonic Analysis*, vol. 23, pp. 368–387, 2007.

[46] ——, "Curvelets and wave atoms for mirror-extended images," in *Proc. of SPIE Wavelets XII*, vol. 6701, 2007, p. 67010J.

[47] L. Vese and S. Osher, "Modeling textures with total variation minimization and oscillating patters in image processing," *Journal of Scientific Computing*, vol. 19, pp. 553–572, 2003.

[48] Y. Meyer, *Oscillating Patterns in Image Processing and Nonlinear Evolution Equations*. AMS, 2001.

[49] L. Vese and S. Osher, "Color texture modeling and color image decomposition in a variational-pde approach," in *Proc. of the 8th Int. Symposium on Symbolic and Numeric Algorithms for Scientific Comp.*, 2006, pp. 103–110.

[50] Y. Shuai, A. Masahide, T. Akira, and K. Masayuki, "High accuracy bicubic interpolation using image local features," *Trans. Fundam. Electron. Commun. Comput. Sci.*, vol. 8, pp. 1611–1615, 2007.

[51] Y. Zhang, D. Zhao, J. Zhang, R. Xiong, and W. Gao, "Interpolation-dependent image downsampling." *IEEE Transactions on Image Processing*, vol. 20, no. 11, pp. 3291–3296, 2011.

[52] P. Perona and J. Malik, "Scale-space and edge detection using anisotropic diffusion," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 12, no. 7, pp. 629–639, 1990.

[53] W. Casaca and M. Boaventura, "A regularized nonlinear diffusion approach for texture image denoising," in *22th Conference on Computer Graphics and Image Processing (SIBGRAPI)*. IEEE Computer Society, 2009, pp. 164–171.

[54] ——, "A decomposition and noise removal method combining diffusion equation and wave atoms for textured images," *Mathematical Problems in Engineering*, vol. 2010, pp. 1–21, 2010.

[55] F. J. Estrada and A. D. Jepson, "Benchmarking image segmentation algorithms," *International Journal of Comput. Vision*, vol. 85, no. 2, pp. 167–181, 2009.

[56] 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, 2011.

[57] C. Rother, V. Kolmogorov, and A. Blake, "Grabcut: Interactive foreground extraction using iterated graph cuts," *ACM Transactions on Graphics*, vol. 23, no. 3, pp. 309–314, 2004.

[58] S. Vicente, V. Kolmogorov, and C. Rother, "Graph cut based image segmentation with connectivity priors," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2008, pp. 1–8.

[59] P. Miranda and A. Falcão, "Links between image segmentation based on optimum-path forest and minimum cut in graph," *Journal of Mathematical Imaging and Vision*, vol. 35, no. 2, pp. 128–142, 2009.

[60] K. C. Ciesielski, J. K. Udupa, A. X. Falcão, and P. Miranda, "Fuzzy connectedness image segmentation in graph cut formulation: A linear-time algorithm and a comparative analysis," *Journal of Mathematical Imaging and Vision*, vol. 44, no. 3, pp. 375–398, 2012.

[61] L. Grady, "Targeted image segmentation using graph methods," in *Image Processing and Analysis with Graphs*, O. Lézoray and L. Grady, Eds. CRC Press, 2012, pp. 111–135.

[62] B. Wang and Z. Tu, "Affinity learning via self-diffusion for image segmentation and clustering," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012, pp. 2312–2319.

[63] A. Angelova and S. Zhu, "Efficient object detection and segmentation for fine-grained recognition," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013, pp. 811–818.

[64] 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 *IEEE International Conference on Computer Vision (ICCV)*, 2009, pp. 731–738.

[65] 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, 2011.

[66] Y. Boykov and M.-P. Jolly, "Interactive graph cuts for optimal boundary & region segmentation of objects in n-d images," in *IEEE International Conference on Computer Vision*, 2001, pp. 105–112.

[67] D. Hower, V. Singh, and S. C. Johnson, “Label set perturbation for mrf based neuroimaging segmentation,” in *IEEE International Conference on Computer Vision (ICCV)*, 2009, pp. 849–856.

[68] A. K. Sinop and L. Grady, “A seeded image segmentation framework unifying graph cuts and random walker which yields a new algorithm,” in *IEEE International Conference on Computer Vision*, 2007, pp. 1–8.

[69] F. Monteiro and A. Campilho, “Watershed framework to region-based image segmentation,” in *Proceedings of International Conference on Pattern Recognition (ICPR)*, 2008, pp. 1–4.

[70] J. Coustry, 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, 2010.

[71] 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, 2004.

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

[73] A. Criminisi, T. Sharp, and A. Blake, “Geos: Geodesic image segmentation,” in *Proceddings of European Conference on Computer Vision: Part I (ECCV)*, 2008, pp. 99–112.

[74] L. M. Rocha, F. A. Cappabianco, and A. Falcão, “Data clustering as an optimum-path forest problem with applications in image analysis,” *International Journal of Imaging System and Technology*, vol. 19, no. 2, pp. 50–68, 2009.

[75] X. Bai and G. Sapiro, “Geodesic matting: A framework for fast interactive image and video segmentation and matting,” *International Journal of Computer Vision (IJCV)*, vol. 82, no. 2, pp. 113–132, 2009.

[76] N. T. N. Anh, J. Cai, J. Zhang, and J. Zheng, “Robust interactive image segmentation using convex active contours,” *IEEE Transactions on Image Processing*, vol. 21, no. 8, pp. 3734–3743, 2012.

[77] O. Sorkine, “Differential representations for mesh processing,” *Computer Graphics Forum*, vol. 25, no. 4, pp. 789–807, 2006.

[78] K. Xu, H. Zhang, D. Cohen-Or, and Y. Xiong, “Dynamic harmonic fields for surface processing,” *Computer and Graphics*, vol. 33, no. 3, pp. 391–398, 2009.

[79] K.-J. Yoon and I.-S. Kweon, “Locally adaptive support-weight approach for visual correspondence search,” in *IEEE Computer Vision and Pattern Recognition (CVPR)*, 2005, pp. 924–931.

[80] Z. Zhu, C. C. Loy, and S. Gong, “Constructing robust affinity graphs for spectral clustering,” in *IEEE Computer Vision and Pattern Recognition (CVPR)*, 2014, pp. 550–558.

[81] V. Jankovic, “Quadratic functions in several variables,” *The Teaching of Mathematics*, vol. VIII, no. 1, pp. 53–60, 2005.

[82] J. Cousty, G. Bertrand, L. Najman, and M. Couprie, “Watershed cuts,” in *Proceedings of Seventh International Symposium on Mathematical Morphology*, 2007, pp. 301–312.

[83] 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, 2007.

[84] M. Meilă, “Comparing clusterings: an axiomatic view,” in *Proceedings of the 22nd international Conference on Machine Learning*, ser. ICML ’05, 2005, pp. 577–584.

[85] M. Mignotte, “A label field fusion model with a variation of information estimator for image segmentation,” *Information Fusion*, vol. 20, no. 1, pp. 7–20, 2014.

[86] M. Meilă, “Comparing clusterings: An information based distance,” *Journal of Multivariate Analysis*, vol. 98, no. 5, pp. 873–895, 2007.

[87] R. Unnikrishnan, C. Pantofaru, and M. Hebert, “A measure for objective evaluation of image segmentation algorithms,” in *Workshops of the IEEE Conf. on Computer Vision and Pattern Recognition*, ser. CVPR ’05, 2005, pp. 34–39.

[88] M. Subramanyam, D. K. Nallaperumal, P. P., and D. S., “Analysis of image inpainting techniques with exemplar, poisson, successive elimination and 8 pixel neighborhood methods,” *International Journal of Computer Applications*, vol. 9, no. 11, pp. 15–18, 2010.

[89] W. Casaca, “Restauração de imagens digitais com texturas utilizando técnicas de decomposição e equações diferenciais parciais (in portuguese),” 192 pgs, Master Dissertation, São Paulo State University (UNESP), IBILCE, São José do Rio Preto, SP, Brazil, 2010.

[90] T. Tamaki, H. Suzuki, and M. Yamamoto, “String-like occluding region extraction for background restoration,” in *IEEE International Conference on Pattern Recognition (ICPR)*, 2006, pp. 615–618.

[91] T. Amano, “Correlation based image defect detection,” in *IEEE Int. Conf. on Pattern Recognition*, 2006, pp. 163–166.

[92] M. G. Padalkar, M. A. Zaveri, and M. V. Joshi, “Svd based automatic detection of target regions for image inpainting,” in *13th Asian Conference on Computer Vision*, ser. Lecture Notes in Computer Science, vol. 7729, 2013, pp. 61–71.

[93] J. Sun, L. Yuan, J. Jia, and H.-Y. Shum, “Image completion with structure propagation,” *ACM Transactions on Graphics*, vol. 24, no. 3, pp. 861–868, 2005.

[94] C. Barnes, E. Shechtman, A. Finkelstein, and D. B. Goldman, “Patchmatch: A randomized correspondence algorithm for structural image editing,” *ACM Trans. Graphics*, vol. 28, no. 3, pp. 24:1–24:11, 2009.

[95] C. A. Z. Barcelos, M. Boaventura, and E. C. S. Jr., “A well balanced flow equation for noise removal and edge detection,” *IEEE Transactions on Image Processing*, vol. 12, no. 7, pp. 751–763, 2003.

[96] M. Bertalmio, G. Sapiro, V. Caselles, and C. Ballester, “Image inpainting,” in *Annual Conf. on Comp. Graphics*, 2000, pp. 217–226.

[97] A. A. Efros and T. K. Leung, “Texture synthesis by non-parametric sampling,” in *IEEE International Conference on Computer Vision (ICCV)*, 1999, pp. 1033–1038.

[98] M. Bertalmío, L. A. Vese, G. Sapiro, and S. Osher, “Simultaneous structure and texture image inpainting,” *IEEE Transactions on Image Processing*, vol. 12, pp. 882–889, 2003.

[99] A. Criminisi, P. Perez, and K. Toyama, “Region filling and object removal by exemplar-based image inpainting,” *IEEE Transactions on Image Processing*, vol. 13, pp. 1200–1212, 2004.

[100] N. Kawai, T. Sato, and N. Yokoya, “Image inpainting considering brightness change and spatial locality of textures and its evaluation,” in *Proceedings of the 3rd Pacific Rim Symp. on Adv. in Img. and Video Technology*, ser. PSIVT ’09, 2009, pp. 271–282.

[101] M. Elad, J.-L. Starck, P. Querre, and D. Donoho, “Simultaneous cartoon and texture image inpainting using morphological component analysis,” *App. Comp. Harmonic Anal.*, vol. 19, pp. 340–358, 2005.

[102] O. G. Guleryuz, “Nonlinear approximation based image recovery using adaptive sparse reconstructions and iterated denoising-part ii: adaptive algorithms,” *IEEE Transactions on Image Processing*, vol. 15, no. 3, pp. 555–571, 2006.

[103] M. Fadili, J.-L. Starck, and F. Murtagh, “Inpainting and zooming using sparse representations,” *The Comp. J.*, vol. 52, no. 1, pp. 64–79, 2009.

[104] X. Li, “Image recovery via hybrid sparse representations: a deterministic annealing approach,” *IEEE Journal of Selected Topics in Signal Processing*, vol. 5, no. 5, pp. 953–962, 2011.

[105] A. Hertzmann, C. E. Jacobs, N. Oliver, B. Curless, and D. Salesin, “Image analogies,” in *ACM Trans. Graphics*, 2001, pp. 327–340.

[106] T. Welsh, M. Ashikhmin, and K. Mueller, “Transferring color to greyscale images,” *ACM Transactions on Graphics*, vol. 21, no. 3, pp. 277–280, 2002.

[107] A. Levin, D. Lischinski, and Y. Weiss, “Colorization using optimization,” *ACM Trans. Graphics*, vol. 23, no. 3, pp. 689–694, 2004.

[108] Y.-C. Huang, Y.-S. Tung, J.-C. Chen, S.-W. Wang, and J.-L. Wu, “An adaptive edge detection based colorization algorithm and its applications,” in *Proc. of the 13th ACM Intl. Conf. on Multimedia*, 2005, pp. 351–354.

[109] L. Yatziv and G. Sapiro, “Fast image and video colorization using chrominance blending,” *IEEE Transactions on Image Processing*, vol. 15, no. 5, pp. 1120–1129, 2006.

[110] Y. Qu, T.-T. Wong, and P.-A. Heng, “Manga colorization,” *ACM Transactions on Graphics*, vol. 25, no. 3, pp. 1214–1220, 2006.

[111] Q. Luan, F. Wen, D. Cohen-Or, L. Liang, Y.-Q. Xu, and H.-Y. Shum, “Natural image colorization,” in *Proc. of the Eurographics Symposium on Rendering*, 2007, pp. 309–320.

[112] C. Yao, X. Yang, L. Chan, and Y. Xu, “Image colorization using bayesian nonlocal inference,” *Journal of Electronic Imaing*, vol. 20, no. 2, pp. 023 008–023 008–6, 2011.