Noname manuscript No.
(will be inserted by the editor)

Spectral Image Segmentation using Image Decomposition and Inner Product-based metric

Wallace Casaca · Afonso Paiva · Erick Gomez-Nieto
Paulo Joia · Luis Gustavo Nonato

Received: date / Accepted: date

Abstract Image segmentation is an indispensable tool in computer vision applications, such as recognition, detection and tracking. In this work, we introduce a novel user-assisted image segmentation technique which combines image decomposition, inner product-based similarity metric, and spectral graph theory into a concise and unified framework. First, we perform an image decomposition to split the image into texture and cartoon components. Then, an affinity graph is generated and the weights are assigned to its edges according to a gradient-based inner-product function. From the eigenstructure of the affinity graph, the image is partitioned through the spectral cut of the underlying graph. The computational effort of our framework is alleviated by an image coarsening process, which reduces the graph size considerably. Moreover, the image partitioning can be improved by interactively changing the graph weights by sketching. Finally, a coarse-to-fine interpolation is applied in order to assemble the partition back onto the original image. The efficiency of the proposed methodology is attested by comparisons with state-of-art spectral segmentation methods through a qualitative and quantitative analysis of the results.

Keywords spectral cut · image segmentation · similarity graph · cartoon-texture decomposition · harmonic analysis.

Instituto de Ciências Matemáticas e de Computação – ICMC
Universidade de São Paulo – USP
São Carlos – São Paulo – Brazil
E-mail: wallace,apneto,egomezn,pjoia,gnonato@icmc.usp.br

1 Introduction

Image segmentation is no doubt one of the most important tasks in digital image processing and computer vision. The wide range of important applications that rely on image segmentation such as image coding, content-based image retrieval, and pattern recognition, have motivated the development of an enormous quantity of techniques for segmenting images. In particular, graph-based techniques figure among the most effective methods, mainly due to the flexibility it provides to handle color, texture, noise and specific features [2,16,18,31, 33] into a unified framework.

The strength of graph-based approaches derive from the solid mathematical foundation it relies on, since most of the well-established graph theory [5] can be directly used to handle the image segmentation problem. For instance, spectral graph theory [12] has been the basic tool for the so-called spectral cuts method [33,26, 29,36], which exploits the eigenstructures of the image affinity graph so as to accomplish the image 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 techniques, turning out spectral cuts an attractive image segmentation approach.

Despite their pliability and powerful, the methods inspired on spectral cuts present some aspects that must be observed in order to ensure the success of the segmentation process. For example, the accuracy in detecting the boundaries between image regions is highly dependent on the weights assigned to the edges of the graph. Although automatic schemes have been proposed to accurately compute those weights [33,26,13,39,7],

it is well-known that user intervention is essential in many cases to correctly define the object boundaries [22]. 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 [24].

In this paper, we present a new framework for image segmentation that relies on spectral cuts while addressing the issues raised above in an innovative manner. The proposed framework can be summarized in the following steps. Firstly, we decompose the target image into two new images: the smooth and texture components. This mechanism is based on a cartoon-texture image decomposition scheme (Section 3.1) that facilitates the identification of the different features contained in the image. Next, we provide a novel mechanism to assign weights to the edges of the affinity graph (Section 3.3) that results in accurate segmentation in most cases. In contrast to other spectral cut-based approaches, our technique allows for user intervention, enabling to automatically modify weights according to the user perception (Section 3.5). Moreover, we show how to build the similarity graph from a coarse representation of the input image without degrading segmentation results. Building the graph in a coarser resolution reduces the size of the graph, thus lessening the computational effort during the eigendecomposition, which permits to handle large images. Our results show (Section 4) that the proposed approach outperforms classical spectral image segmentation techniques in aspects such as accuracy and robustness (refer to [11] to a less detailed and presentation of the some results discussed here).

Contributions. We can summarize the main contributions of this work as:

- an image segmentation technique that combines cartoon-texture decomposition and spectral cuts;
- a novel method to compute and assign weights to the edges of a similarity graph using the cartoon component of the image;
- a new strategy to modify the weights of the graph according to user interaction, taking into account the texture component of the image.

2 Related Works and Basic Concepts

The literature on image segmentation is huge and a comprehensive overview about this theme is beyond the scope of this paper. In order to contextualize this work, we focus our discussion on graph-based clustering techniques applied to image segmentation task.

Spectral methods. Given an image $\mathcal{I}$ it can be modeled as a weighted graph $G = (V, E, W)$, also called *pixel-affinity graph*, where each node $v_i \in V$ addresses a pixel $P_i \in \mathcal{I}$, each edge $e_{ij} = v_i v_j \in E$ connects a pair of neighbor pixels and the weight $w_{ij} = w(e_{ij})$ is defined in terms of specific attributes of P_i and P_j, such as luminance, position and image gradient. Fiedler, in his seminal work in graph theory [19], proposed a graph partitioning scheme based on spectral properties of the graph Laplacian matrix $L = (l_{ij})$, as follows:

$$l_{ij} = \begin{cases} -w_{ij}, & \text{if} \quad e_{ij} \in E \\ \sum_j w_{ij}, & \text{if} \quad v_i = v_j \\ 0, & \text{otherwise} \end{cases}. \tag{1}$$

Since the matrix L is symmetric and positive semi-definite, the solution of the eigenproblem

$$L\mathbf{x} = \lambda\mathbf{x} \tag{2}$$

is given by non-negative real eigenvalues with respective real eigenvectors. The eigenvector associated to the second smallest eigenvalue of L is the so-called *Fiedler vector*. According to *Courant Nodal Domain theorem* [4], the zero-set of the Fiedler vector splits the graph G into two disjoint graphs, thus the recursive computation of the Fiedler vector partitions the graph hierarchically as a binary tree. This strategy has been used successfully in several approaches to segment images via average cut [23,32,35] and spectral cut with non-normalized Laplacian) [33], providing a simple and efficient methodology for many other problems involving graph clustering.

Normalized cut methods. Shi and Malik [33] introduced the concept of *Normalized Cut* (NCut), a graph optimization problem that aims to subdivide a graph so as to minimize the normalize sum of weights in the resulting graphs. They show that finding the optimal NCut is an NP-hard problem, but good solutions can be reached from the Fiedler vector computed by solving a generalized eigenvalue problem.

More specifically, the NCut method builds a graph G from a given image $\mathcal{I}$ by considering each pixel as a

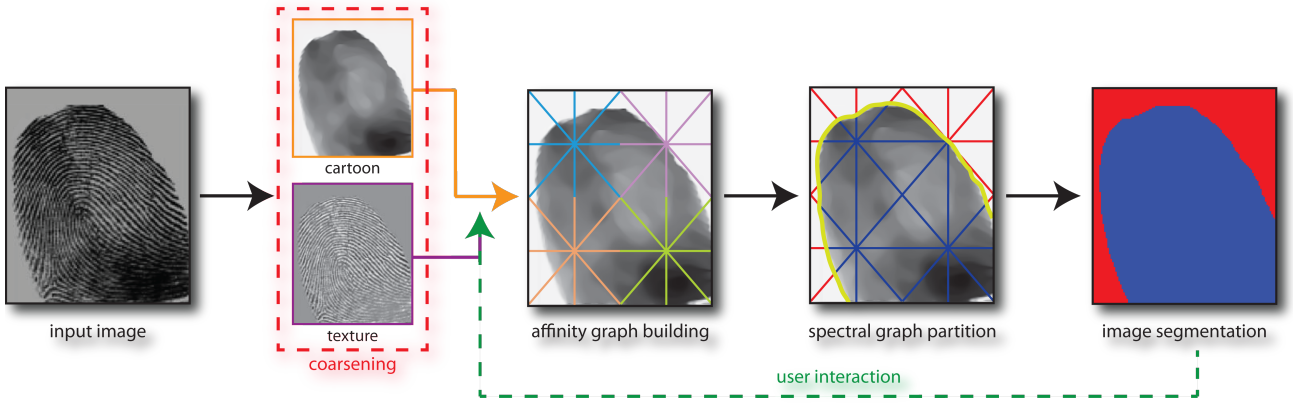


Fig. 1 Pipeline of our image segmentation framework.

node of the graph and connecting two nodes v_i and v_j with an edge e_{ij} if

$$\|P_i - P_j\|_2 < r, \tag{3}$$

where P_i and P_j are the pixels associated to the nodes v_i and v_j while r is a parameter defining how local the edges should be. The weights $w_{ij} = w(e_{ij})$ assigned to the edges are computed from the following formula:

$$w_{ij} = \exp\left(-\frac{\|P_i - P_j\|_2^2}{\sigma_P^2} - \frac{\|I_i - I_j\|_2^2}{\sigma_I^2}\right), \tag{4}$$

where I_i and I_j account for intensity values in P_i and P_j. The pair (σ_P, σ_I) are tuning parameters used to control the scope of each term (position and intensity) in the edge weights.

Many variants of the NCut method have been appeared in the literature, most of which proposing alternative graph construction and weight assignment. For instance, nodes of the graph and weights may be defined from watershed segmentation [38,8], quadtree decomposition [9], Markov random fields [6], texture descriptors [25] and biased normalized cuts [26], just to cite a few.

Multiscale methods. Aiming at achieving multiscale image segmentation, some authors have proposed the multilevel representation of the underlying graph, varying progressively the number of nodes and edges. Multiscale approaches aim to capture both the local and global geometric relations among the structures of the image. The multilevel representation is accomplished using distinct multiscale approaches such as a combination of graph compression and cross-scale constraint [13], shape information with PCA [7], and texture descriptors [37].

Other graph-based methods. Techniques that relies on graph structures while avoiding spectral decomposition have also been proposed in the literature. The *Image Foresting Transform* (IFT) [18,3] is a good example of graph-derived image segmentation technique that does not make use of spectral analysis. In short words, the IFT accomplishes the image partitioning by finding paths of minimum cost between seed nodes. IFT also allows user intervention to tune edge weights towards improving the segmentation results. Another technique that brings good results is the isoperimetric regions-based approach [21], which aims to compute the shape with minimal perimeter taking into account the resolution of the problem.

Methods based on random walks for interactive image segmentation have also become popular in the computer vision community. An interesting example of this kind of approach was proposed by Grady and Sinop [22], who used normalized cuts to estimate random walk probabilities. Their methodology was further improved in [1], where a priori models [20] were incorporated to speed up the process.

The technique described in this paper proposes a new gradient-based weight computation which is only possible due to the cartoon-texture decomposition of the input image. Moreover, the proposed approach allows for user interaction so as to tune weights and thus improve the segmentation.

3 Pipeline overview

The proposed approach is comprised of five main steps, as illustrated in Fig. 1. The first step, *Cartoon-Texture Decomposition* decomposes 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 in $\mathcal{C}$ and $\mathcal{T}$ so as to alleviate the third step of the pipeline, namely the *affinity graph construction*. Besides speeding up the spectral decomposition, the reduced number of edges also lessen the computational burden during the weights assignment stage. 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 in order to incorporate the high-level oscillatory information into the spectral cut processing. Details about each step are presented below.

3.1 Cartoon-Texture Image Decomposition

Cartoon-Texture Decomposition (CTD) separates 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-piece of $\mathcal{I}$ while the texture component contains textures, oscillating patterns, fine details and noise. This decomposition satisfies the important relationship $\mathcal{I} = \mathcal{C} + \mathcal{T}$ (see [40,41] and the underlying mathematical theory in [28]). Similar to [40,41], where a functional minimization problem has been formulated and solved through a system of partial differential equations, here both cartoon $\mathcal{C}$ and texture $\mathcal{T}$ components are obtained by solving the following system of equations:

$$\begin{cases} \mathcal{C} = \mathcal{I} - \partial_x g_1 - \partial_y g_2 + \dfrac{1}{2\lambda}\,\mathrm{div}\left(\dfrac{\nabla \mathcal{C}}{|\nabla \mathcal{C}|}\right) \\ \mu \dfrac{g_1}{\sqrt{g_1^2 + g_2^2}} = 2\lambda\left[\dfrac{\partial}{\partial x}(\mathcal{C} - \mathcal{I}) + \partial^2_{xx} g_1 + \partial^2_{xy} g_2\right] \\ \mu \dfrac{g_2}{\sqrt{g_1^2 + g_2^2}} = 2\lambda\left[\dfrac{\partial}{\partial y}(\mathcal{C} - \mathcal{I}) + \partial^2_{xy} g_1 + \partial^2_{yy} g_2\right] \end{cases} \quad (5)$$

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

$$\begin{cases} \dfrac{\nabla \mathcal{C}}{|\nabla \mathcal{C}|} \cdot (n_x, n_y) = 0 \\ (\mathcal{I} - \mathcal{C} - \partial_x g_1 - \partial_y g_2) \cdot nx = 0 \\ (\mathcal{I} - \mathcal{C} - \partial_x g_1 - \partial_y g_2) \cdot ny = 0 \end{cases}. \quad (6)$$

From the bounded variation functional $\overrightarrow{g} = (g_1, g_2) \in L^2(\mathbb{R}^2)$ one can obtain the texture component as $\mathcal{T} = \mathrm{div}(\overrightarrow{g})$. The constants $\lambda, \mu > 0$ are tuning parameters.

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

Fig. 2 Image decomposition into a cartoon and texture components.

Equations (5) are usually discretized by a semi-implicit finite difference schemes and solved using an iterative algorithm based on fixed point iteration (more details about the numerical aspects can be found in [40]). Fig. 2 shows the result of the CTD scheme applied to a digital image.

In our context, 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 $\mathcal{C}$ [40]. 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.

3.2 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 version $\tilde{\mathcal{C}}$ of $\mathcal{C}$. Such a transformation is accomplished using the bicubic interpolation method described in [34], which minimizes the blurring effect while still preserving gradients in the coarse image.

Our experiments showed that coarsening the image to one-fourth of its original resolution is a good trade-off between computational times and accuracy, speeding up the processing up to 6 times. This performance gain can also be seen in the comparative evaluation shown in Table 1.

3.3 Building the affinity graph

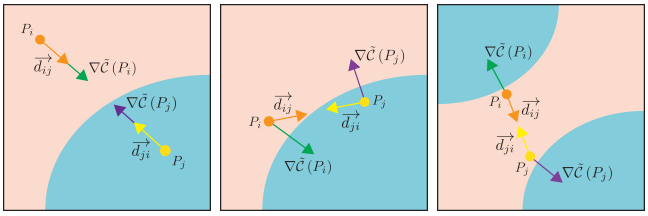
The affinity graph G is built by associating each pixel from $\tilde{\mathcal{C}}$ to a node of the graph, connecting the nodes according to the distance between corresponding pixels (Eq. (3) with the supremum norm instead of euclidian). The weight assigned to each edge of G is derived from the proposed inner product-based metric. In contrast to the original NCut, which takes into consideration

only spatial positions and pixel intensities (Eq. (4)), the inner product-based 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:

$$w_{ij} = \frac{1}{1+\eta g_{ij}^2}, \; g_{ij} = \max\left\{\frac{\partial\tilde{\mathcal{C}}(P_i)}{\partial\overrightarrow{d_{ij}}}, \frac{\partial\tilde{\mathcal{C}}(P_j)}{\partial\overrightarrow{d_{ji}}}, 0\right\}, \quad (7)$$

$$\frac{\partial\tilde{\mathcal{C}}(x)}{\partial\overrightarrow{d_{ij}}} = \nabla\tilde{\mathcal{C}}(x)\,\overrightarrow{d_{ij}}, \quad \text{with} \quad \overrightarrow{d_{ij}} = \frac{\overrightarrow{P_iP_j}}{|\overrightarrow{P_iP_j}|}. \quad (8)$$

The left term in Eq. (8) 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. In other words, similarly to Eq. (4), our formulation includes the intensity and geometric information to define the weights into a unique measure: the inner-product in the edges direction. Fig. 3 provides a geometric interpretation of the proposed metric.



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

Fig. 3 Geometric interpretation of the inner product-based metric. Maximum weights occurs 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) produces minimum weights (zero).

The effective weights w_{ij} are chosen from Eq. (7) rather then using the exponential measure usually employed by other authors [33,9,37]. The scheme proposed in Eq. (7) 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. (7) is indeed derived from the Malik-Perona diffusivity term [30], which was originally used for establishing the notion of anisotropy in the heat equation. Moreover, the inner product-based similarity metric (7) holds the property $w_{ij} = w_{ji}$, which ensures symmetry for the graph Laplacian matrix L. This fact is of paramount importance to guarantee that the eigenstructure of L is made up of only real numbers.

3.4 Spectral cutting and coarse-to-fine

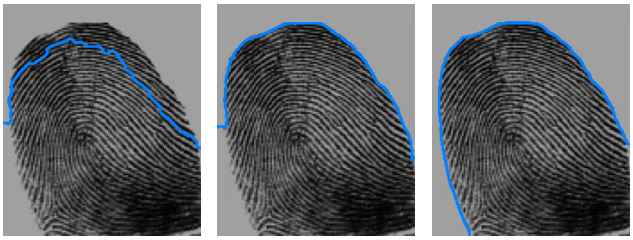
Given the affinity graph G built from $\tilde{\mathcal{C}}$ and the number of partitions initially defined by the user we carry out the spectral decomposition using the same methodology proposed in [33]. More specifically, we first decompose the graph Laplacian matrix as $L = D - W$ where D and W contain the diagonal and off-diagonal elements of L (Eq. (1)). Then, the Fiedler vector f is obtained by solving the generalized eigenvalue problem

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

getting f as the eigenvector associated to the smallest non-zero eigenvalue.

The Fiedler vector splits $\tilde{\mathcal{C}}$ into two subsets, one containing the pixels corresponding to nodes of the graph where the entries of f are positive and other containing the pixels with negative values of f. Therefore, the zero-set of f is a curve that separates the regions with different signs. The partitioning created in $\tilde{\mathcal{C}}$ is brought back to $\mathcal{C}$ using bicubic interpolation from f. By recursively computing the spectral decomposition for each part of the image, one can produce finer segmentation. The recursive process may be driven by the user, who can specify the highest level of recursion, in addition to brushing defined by himself in any pieces of the image during each one of recursion steps (Sec. 3.5).

Fig. 4 shows the result of applying our methodology to segment a fingerprint image. For the sake of comparison, we show in Fig. 4.a the result of computing weights (using Eq. (7)) directly from the original image $\mathcal{I}$, that is, skipping the CTD, while Fig. 4.b depicts the result using CTD and Eq. (4) instead of Eq.(7) to define the



(a) Without CTD (b) Without the proposed metric (c) Complete pipeline

Fig. 4 Automatic result with the proposed approach.

graph weights. Notice, from Fig. 4.c, how better the segmentation is when Eq. (7) and CTD are combined.

3.5 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 [15,14] called *wave atoms*. The wave atoms-based techniques have high directional anisotropy and sensitivity to noise, which makes them suited to identify 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, similarly to what is proposed in [10]. 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".

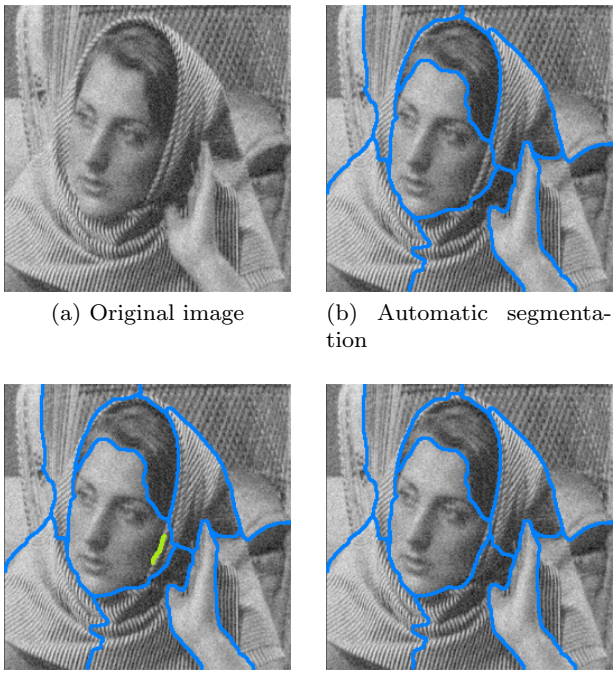
Starting from this premise, the weights of edges incident to pixels brushed by the user are modified as follows:

$$w_{ij} = \rho * \min_{e_{ij} \in E} w_{ij} * \max\{S(\tilde{\mathcal{T}}_i), S(\tilde{\mathcal{T}}_j)\}, \tag{9}$$

where the constant $\rho \in (0,1)$ is the smallest non-zero weight of the edges in G and $\tilde{\mathcal{T}}$ is the coarse version of $\mathcal{T}$, which is generated similarly to $\tilde{\mathcal{C}}$. The constant ρ enforces a more drastic change in the weights on the region brushed by the user, since the new weights will have the lowest possible non-zero value within the graph to be repartitioned.

4 Results and Comparisons

Now we present some results obtained with our framework and present a comparative study involving four other existing 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 (Section 3.1), the default parameters suggested in [34] for the bicubic interpolation (Section 3.2 and Section 3.4) and a hard threshold at 3σ (noise-to-signal ratio of the image) combined with cycle spinning [15] for the wave atom transform (Section 3.5). We set $r = 1$ and $\eta = 5$ in Eqs. (3)–(7), respectively. Finally, we have used parameters and implementations suggested by the authors for the methods used in our comparisons: *k-way Normalized Cuts* (NCut) [33], *Multiscale NCut* (MS-NCut) [13] (with radius 2, 3 and 7 for each scale, respectively), *Random Walk-based Segmentation with pure Eigenvector Precomputation* (RWS-EP) [22] and *Random Walk mixed with Eigenvector Precomputation and Prior scheme* (RWS-EPP) [1] (with 80 precomputed eigenvectors in both cases). All results were generated on a 1.80Hz AMD with 1GB of RAM.



(a) Original image (b) Automatic segmentation (c) Brush made by user (green) (d) Final result

Fig. 5 Improving segmentation of the noise-textured image from user's strokes.

User interaction. We start showing how user intervention can be used to fix imperfections in the segmentation process. Fig. 5 shows the result of segmenting Fig. 5.a using our method for 10 partitions. Notice from Fig. 5.b 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. 5.c and Fig. 5.d. This post-segmentation was only feasible because the texture descriptor used to accomplish this task is sensitive to noise.

Fig. 6 shows that is not necessary to perform a large number of user interventions to achieve the desired segmentation. The blue curve in Fig. 6.b is the result produced by our method without user interven-

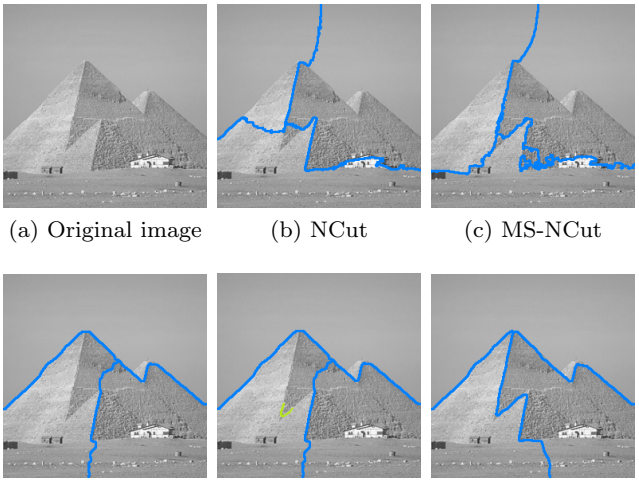
(a) Original image (b) Small stroke (c) Final result

Fig. 6 A simple stroke (greenish region between the two owls) is sufficient to improve the segmentation.

tion. The simple greenish stroke depicted on the texture region between the two owls and setting the partition number to 4 were enough to enforce a more satisfactory segmentation, as shown in Fig. 6.c.

Comparisons. In order to check the quality of the proposed method, we provide comparisons against two automatic (NCut and MS-NCut) and two user-assisted (RWS-EP and RWS-EPP) spectral-based techniques. Color images were converted to grayscale before processing.

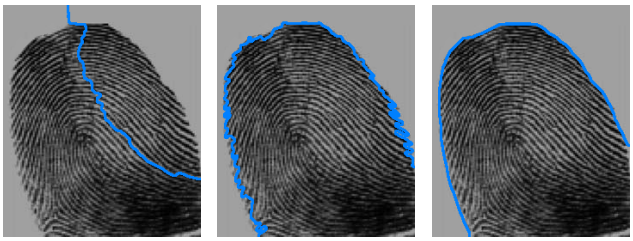
a. Comparison with automatic methods. The first experiment shown in Fig. 7 presents a comparative analysis of our technique against non-interactive NCut and MS-NCut approaches where user intervention is needed to improve the segmentation. We can see that both classical NCut (Fig. 7.b) and MS-NCut (Fig. 7.c) badly segment parts of the image. Our method results in a better partitioning (Fig. 7.d), although some regions are also segmented in an incorrect way. After user intervention (Fig. 7.e), the result improves considerably (Fig. 7.f).



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

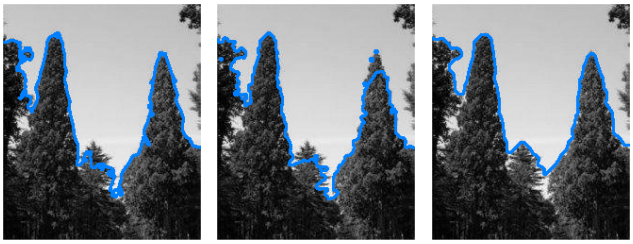
(d) Automatic result (e) With sketching (green) (f) Final result

Fig. 7 The influence of the user intervention in comparison with static approaches.



(a) NCut (b) MS-NCut (c) Ours

Fig. 8 The result of applying NCut, MS-NCut, and our approach in automatic mode in a fingerprint image.



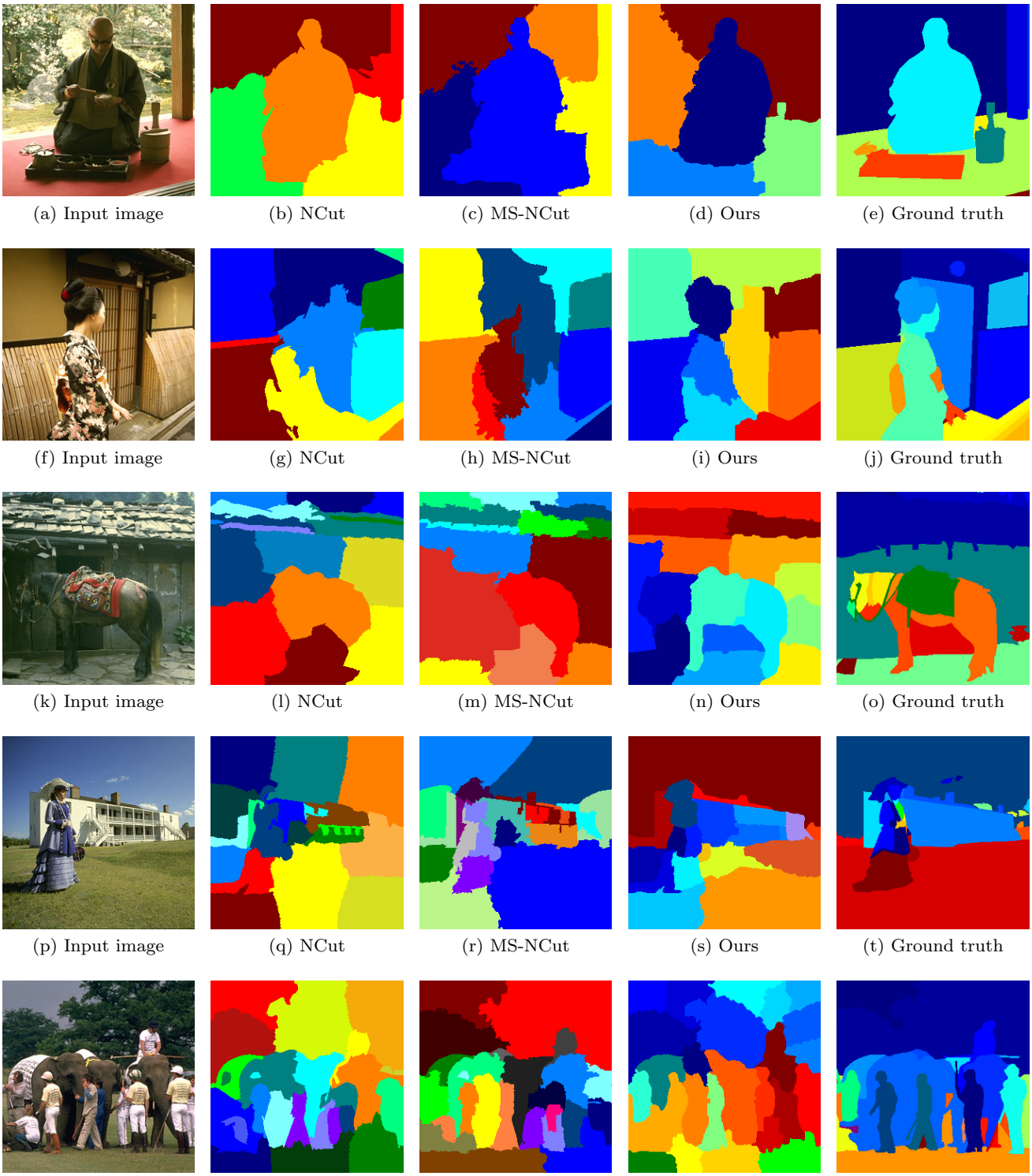
(a) NCut (b) MS-NCut (c) Ours

Fig. 9 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 using two levels of partition is shown in Fig. 8. Notice that the NCut (Fig. 8.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. 8.b that the MS-NCut tends to produce a jagged segmentation curve while our method results in a smoother curve, as shown in Fig. 8.c.

It becomes clear from Fig. 9 that the smoothness of the result produced by our approach also help to increase robustness. While NCut and MS-NCut tend to generate a segmentation curve with many artifacts and misfitted regions, our approach produces more consistent and pleasant results.

b. Evaluation using ground-truth images. Fig. 10 shows the partitioning produced by NCut, MS-NCut, and our approach in "automatic mode" (without user intervention) when applied to the images in the first column. These experimental images have been randomly extracted from the *Berkeley Image Database* [27]. The first column in Fig. 10 show the input images. Notice that the MS-NCut and our method produce 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 segmented regions and it clusters the image into slightly wider regions, two characteristic also presents in the ground truth images. More-



(a) Input image (b) NCut (c) MS-NCut (d) Ours (e) Ground truth

(f) Input image (g) NCut (h) MS-NCut (i) Ours (j) Ground truth

(k) Input image (l) NCut (m) MS-NCut (n) Ours (o) Ground truth

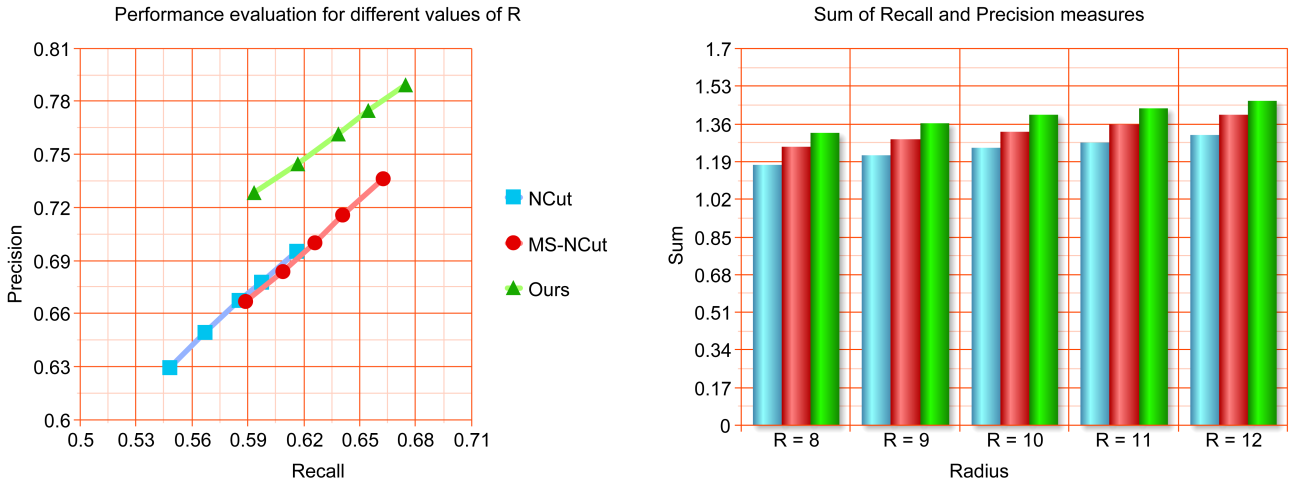
(p) Input image (q) NCut (r) MS-NCut (s) Ours (t) Ground truth

(u) Input image (v) NCut (w) MS-NCut (x) Ours (y) Ground truth

Fig. 10 Partitioning produced by NCut, MS-NCut, and our methodology. From top to bottom, the three methods partition the images in 5, 10, 15, 20 and 25 regions respectively.

Image	NCut				MS-NCut				Ours			
	Recall	Prec.	Sum	Time	Recall	Prec.	Sum	Time	Recall	Prec.	Sum	Time
Fig. 10.a	0.4803	0.7143	1.1946	87	0.4813	0.6434	1.1248	16	0.4814	0.7340	1.2154	8
Fig. 10.f	0.5867	0.8294	1.4161	119	0.6865	0.8513	1.5379	36	0.6710	0.9447	1.6158	14
Fig. 10.k	0.3917	0.4655	0.8572	204	0.4133	0.5559	0.9692	64	0.5602	0.6632	1.2234	22
Fig. 10.p	0.7124	0.5716	1.2840	147	0.8100	0.7097	1.5198	59	0.7220	0.7187	1.4407	21
Fig. 10.u	0.7424	0.7377	1.4407	226	0.7358	0.7417	1.4775	77	0.7429	0.7383	1.4812	23

Table 1 Comparative table containing computational timings (in seconds), recall and precision measures for all images from Fig. 10. For each image, both measures was obtained by computing the recall and precision average for $R = 8, 9, 10, 11$ and 12.



(a) Recall and precision for $R = 8, 9, ..., 12$.

(b) Sum of both measure per Radius.

Fig. 11 Quantitative comparisons. Each method is evaluated by taking the recall and precision average among all images from Fig. 10 for different value of R.

over, it can be seen that our method is more robust to identify objects and structures contained in the images. For instance, the input images monk (Fig. 10.d), geisha (Fig. 10.i), horse (Fig. 10.s) and woman (Fig. 10.x) were better captured by our technique.

All the results presented in Fig. 10 has also been evaluated by means of a quantitative analysis. To accomplish such quantitative analysis we use the *recall* and *precision* quantitative image segmentation measures proposed in [17]:

$$Recall = \frac{\text{Matched}(S_{target}, S_{source})}{|S_{target}|}, \tag{10}$$

$$Precision = \frac{\text{Matched}(S_{source}, S_{target})}{|S_{source}|}, \tag{11}$$

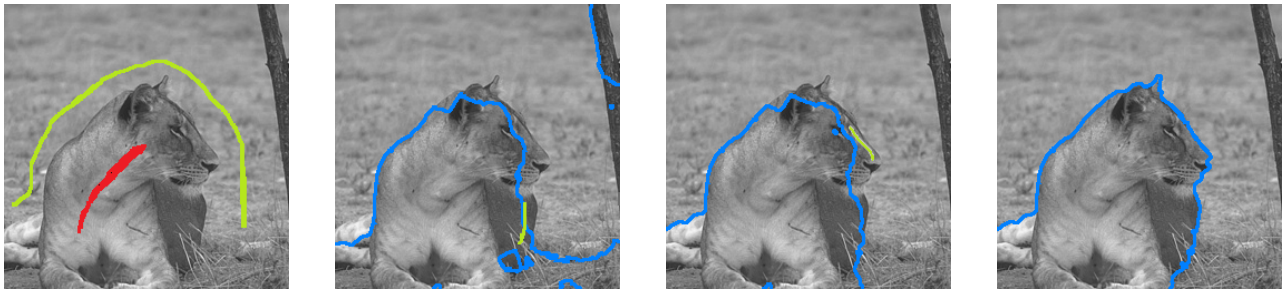
where S_{source} represents the ground truth segmentation, S_{target} the partitioning to be evaluated and $|\cdot|$ indicates the total number of boundary pixels in the current segmentation. Recall and precision measures (10)–(11) can be understood as the proportion of boundary pixels in S_{target} (resp. S_{source}) for which it is possible to find a matching boundary pixel in S_{source} (resp. S_{target}). To establish this matching we have used the same approach proposed in [17], which relies on boundary pixel proximity for different values of radius R.

Table 1 shows the computation of recall, precision, sum of both and the processing time for all images from Fig. 10. As one can see, our technique is better than other methods for most of the cases. In fact, only the sum associated to Fig. 10r is better than our approach (Fig. 10.s). The good performance of the proposed framework can also be seen in the plots in Figure Fig. 11.a-b, which depicts the average of recall and precision quantities per technique for different values of radius R and their sums.

c. Comparison with interactive methods. The results provided from our technique considering user intervention against other two interactive spectral-based methods (RWS-EP and RWS-EPP) are displayed in Fig. 12. In contrast to the random walk techniques RWS-EP and RWS-EPP, our method does not require an initial user setup to produce the first segmentation result. Notice that differently from RWS-EP and RWS-

(a) Input image (b) 1st user interaction (c) 2nd user interaction (d) Final result (after two user brushes)

(e) Input image (f) 2nd user interaction (g) 4th user interaction (h) Final result (after six user brushes)



(i) Input image (j) 2nd user interaction (k) 6th user interaction (l) Final result (after ten user brushes)

Fig. 12 Segmentation driven by user produced by our technique (top row), RWS-EP (middle row) and RWS-EPP (bottom row). Fig. 12.a is required by our approach, while the target image and the initial setup with 2000 seeds (red and green pieces) must be provided by the user in random walker-based methods (Fig. 12.e and Fig. 12.i).

Processing stage	**RWS-EP**	**RWS-EPP**	**Ours**
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

Table 2 Computational times (in seconds) involved in the interactive segmentation process presented in Fig. 12. First row shows the times to produce the first segmentation; (+4) is the time to interactively set the seeds. Second row is the average time after each interaction while the bottom row is the total time to produce the final result shown in the last column in Fig. 12.

EPP, which require much more user intervention to reach a pleasant result (Fig. 12.h and Fig. 12.l), our approach demands only two brushes to yield the desired result (Fig. 12.d). Moreover, our technique is considerably faster when compared to RWS-EP and RWS-EPP (see Table 2).

5 Discussion and Limitation

The combination cartoon-texture decomposition and spectral cut turns 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 segmentation methods.

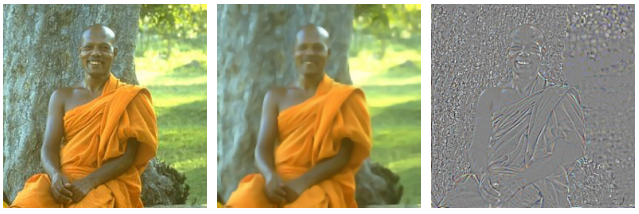
The qualitative and quantitative analysis presented in Section 4 clearly show the effectiveness of the proposed spectral cut segmentation method, surpassing, in terms of accuracy, the state-of-art methods. Moreover, the flexibility as to user intervention is an important trait of our method, which enables the user to fix the segmentation locally.

There are two aspects to be observed when using our technique. First, the segmentation may not behave as expected if the user changes the weights substantially by stroking many parts of the image. It is worth noticing the this is an extreme case, since our method tend to produces quite satisfactory results without any user intervention. Another aspect to be considered is that the fine-to-coarse process may miss small details of the image.

Finally, our approach can be extend to handle color images quite easily. To do that we first apply the CTD suited to color images (more details in [41]). Then we build the affinity graph replacing g_{ij} in Eq. (7) by:

$$g_{ij} = \max_{k \in \{R,G,B\}} \{g_{ij}^k\}, \tag{12}$$

where each g_{ij}^k is given in Eq. (8). The texture component used during the interaction stage can be handled similarly, thus allowing user intervention in the context of color images. Fig. 13 illustrates the version of our approach for color images.



(a) Original image (b) Cartoon (c) Texture

(d) Automatic result (e) Result after two user interactions (f) Ground-truth image (boundaries)

Fig. 13 Example of segmentation using the extension of our approach to RGB color images.

6 Conclusion and Future Works

In this work we proposed a new methodology for spectral cut-based image segmentation which relies on cartoon-texture decomposition. A new metric to measure the similarity between pixels and a new scheme to update weights of the affinity graph according to user intervention have also been presented. The evaluation we provided shows that our approach outperforms existing techniques in terms of accuracy and robustness, producing smoother segmentation curves.

As future work, the main idea is to extend the proposed framework to perform multiscale processing, in addition to evaluate the use of eigenvectors other than the first one. Moreover, we are also adapting this methodology to 3D images in the context of medical data and to incorporate the texture component in an automatic pipeline for 2D images.

Acknowledgment

We would like to thank the anonymous reviewers for their useful comments to improve the quality of this work and Shawn Andrews for kindly providing us with the implementation of RWS-EP and RWS-EPP. This research has been funded by FAPESP-Brazil, INCT-MACC and CNPq-Brazil.

References

1. Andrews, S., Hamarneh, G., Saad, A.: Fast random walker with priors using precomputation for interactive medical image segmentation. In: Lec. Notes in Comput. Sc., pp. 1–8 (2010)
2. Arbelaez, P., Maire, M., Fowlkes, C., Malik, J.: Contour detection and hierarchical image segmentation. IEEE Trans. Pattern Anal. Mach. Intell. **33**, 898–916 (2011)
3. Bergo, F.P., Falcão, A.X., Miranda, P.A., Rocha, L.M.: Automatic image segmentation by tree pruning. J. Math. Imaging Vis. **29**, 141–162 (2007)
4. Biyikoglu, T., Leydold, J., Stadler, P.F.: Laplacian eigenvectors of graphs: Perron-Frobenius and Faber-Krahn type theorems. Springer (2007)
5. Bollobás, B.: Modern Graph Theory. Springer (1998)
6. Boykov, Y., Funka-Lea, G.: Graph cuts and efficient n-d image segmentation. Int. J. Comput. Vision **70**, 109–131 (2006)
7. Cai, W., Chung, A.C.S.: Shape-based image segmentation using normalized cuts. In: Proc. of ICIP'06, pp. 1101–1104 (2006)
8. Carvalho, M., Costa, A., Ferreira, A., Junior, R.C.: Image segmentation using component tree and normalized cut. In: Proc. of Sibgrapi 2010 (23th Conference on Graphics, Patterns and Images), pp. 317–322 (2010)
9. Carvalho, M., Ferreira, A., Costa, A.: Image segmentation using quadtree-based similarity graph and normalized cut. Lec. Notes in Comput. Sc. **6419**, 329–337 (2010)

10. Casaca, W., Boaventura, M.: A decomposition and noise removal method combining diffusion equation and wave atoms for textured images. Math. Probl. Eng. **2010**, 1–21 (2010)
11. Casaca, W., Paiva, A., Nonato, L.G.: Spectral segmentation using cartoon-texture decomposition and inner product-based metric. In: Proc. of Sibgrapi 2011 (24th Conference on Graphics, Patterns and Images), pp. 266–273 (2011)
12. Chung, F.R.K.: Spectral Graph Theory. AMS (1997)
13. Cour, T., Benezit, F., Shi, J.: Spectral segmentation with multiscale graph decomposition. In: Proc. of CVPR 2005, pp. 1124–1131 (2005)
14. Demanet, L., Ying, L.: Curvelets and wave atoms for mirror-extended images. In: Wavelets XII, Proc. of SPIE, vol. 6701, p. 67010J (2007)
15. Demanet, L., Ying, L.: Wave atoms and sparsity of oscillatory patterns. Appl. Comput. Harmon. Anal. **23**, 368–387 (2007)
16. Díaz, J., Petit, J., Serna, M.: A survey of graph layout problems. ACM Comput. Surv. **34**, 313–356 (2002)
17. Estrada, F.J., Jepson, A.D.: Benchmarking image segmentation algorithms. Int. J. of Comput. Vision **85**(2), 167–181 (2009)
18. Falcão, A., Stolfi, J., Lotufo, R.: The image foresting transform: Theory, algorithms, and applications. IEEE Trans. Pattern Anal. Mach. Intell. **26**, 19–29 (2004)
19. Fiedler, M.: Algebraic connectivity of graphs. Czech. Math. J. **23**, 298–305 (1973)
20. Grady, L.: Multilabel random walker image segmentation using prior models. In: Proc. of IEEE CVPR 2005, pp. 763–770 (2005)
21. Grady, L., Schwartz, E.L.: Isoperimetric graph partitioning for image segmentation. IEEE Trans. Pattern Anal. Mach. Intell. **28**, 469–475 (2006)
22. Grady, L., Sinop, A.K.: Fast approximate random walker segmentation using eigenvector precomputation. In: Proc. of IEEE CVPR 2008 (2008)
23. Jameson, C., Jujuunashvili, A., Knyazev, A.: Modern eigenvalue solvers for spectral image segmentation (2008). URL `http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.122.3624`
24. Koutis, I., Miller, G.L., Tolliver, D.: Combinatorial preconditioners and multilevel solvers for problems in computer vision and image processing. In: Int'l Symp. on Visual Computing, pp. 1067–1078 (2009)
25. Ma, X., Wan, W., Yao, J.: Texture image segmentation on improved watershed and multiway spectral clustering. In: Proc. of ICALIP'08, pp. 1693–1697 (2008)
26. Maji, S., Vishnhoi, N., Malik, J.: Biased normalized cuts. In: Proc. of CVPR 2011, pp. 2057–2064 (2011)
27. Martin, D., Fowlkes, C., Tal, D., Malik, J.: A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In: Proc. 8th Int'l Conf. Computer Vision, vol. 2, pp. 416–423 (2001)
28. Meyer, Y.: Oscillating Patterns in Image Processing and Nonlinear Evolution Equations. AMS (2001)
29. Mohar, B., Juvan, M.: Some applications of Laplace eigenvalues of graphs. In: Graph Symetric: Algebraic Methods and Applications, vol. 497, pp. 227–275 (1997)
30. Perona, P., Malik, J.: Scale-space and edge detection using anisotropic diffusion. IEEE Trans. Pattern Anal. Mach. Intell. **12**, 629–639 (1990)
31. Pratt, W.: Digital Image Processing. Wiley (2008)
32. Sarkar, S., Soundararajan, P.: Supervised learning of large perceptual organization: Graph spectral partitioning and learning automata. IEEE Trans. Pattern Anal. Mach. Intell. **22**, 504–525 (2000)
33. Shi, J., Malik, J.: Normalized cuts and image segmentation. IEEE Trans. Pattern Anal. Mach. Intell. **22**, 888–905 (2000)
34. Shuai, Y., Masahide, A., Akira, T., Masayuki, K.: High accuracy bicubic interpolation using image local features. IEICE Trans. Fundam. Electron. Commun. Comput. Sci. **E90-A**, 1611–1615 (2007)
35. Soundararajan, P., Sarkar, S.: Analysis of MinCut, average cut and normalized cut measures. In: Workshop on Perceptual Organization in Computer Vision (2001)
36. Spielman, D.A.: Spectral graph theory and its applications. In: Proc. of the 48th Annual IEEE Symposium on Foundations of Computer Science, pp. 29–38 (2007)
37. Sun, F., He, J.P.: A normalized cuts based image segmentation method. In: Proc. of ICIC'09, pp. 333–336 (2009)
38. Tao, W., Jin, H., Zhang, Y.: Color image segmentation based on mean shift and normalized cuts. IEEE Sys. Man and Cybern. **37**, 1382–1389 (2007)
39. Tolliver, D., Miller, G.L.: Graph partitioning by spectral rounding: Applications in image segmentation and clustering. In: Proc. of CVPR 2006, vol. 1, pp. 1053–1060 (2006)
40. Vese, L., Osher, S.: Modeling textures with total variation minimization and oscillating patters in image processing. J. Sci. Comput. **19**, 553–572 (2003)
41. Vese, L., Osher, S.: Color texture modeling and color image decomposition in a variational-PDE approach. In: Proc. of the 8th Int'l Symp. on Symbolic and Numeric Algorithms for Scientific Computing, pp. 103–110 (2006)