Graph Laplacian for spectral clustering and
seeded image segmentation

Wallace Correa de Oliveira Casaca

SERVIÇO DE PÓS-GRADUAÇÃO DO ICMC-USP

Data de Depósito:

Assinatura:

Graph Laplacian for spectral clustering and seeded image segmentation

Wallace Correa de Oliveira Casaca

***Advisor:* Prof. Dr. Luis Gustavo Nonato**

***Co-Advisor:* Prof. Dr. Gabriel Taubin**

Doctoral dissertation submitted to the *Instituto de Ciências Matemáticas e de Computação* - ICMC-USP, in partial fulfillment of the requirements for the degree of the Doctorate Program in Computer Science and Computational Mathematics. *HP CN'XGTUQP0*

USP – São Carlos
February 2015

Ficha catalográfica elaborada pela Biblioteca Prof. Achille Bassi
e Seção Técnica de Informática, ICMC/USP,
com os dados fornecidos pelo(a) autor(a)

C334g Casaca, Wallace
 Graph Laplacian for spectral clustering and
 seeded image segmentation / Wallace Casaca;
 orientador Luis Gustavo Nonato; co-orientador
 Gabriel Taubin. -- São Carlos, 2015.
 127 p.

 Tese (Doutorado - Programa de Pós-Graduação em
 Ciências de Computação e Matemática Computacional) --
 Instituto de Ciências Matemáticas e de Computação,
 Universidade de São Paulo, 2015.

 1. image segmentation. 2. spectral graph theory.
 3. image colorization. 4. image inpainting. I.
 Nonato, Luis Gustavo, orient. II. Taubin, Gabriel,
 co-orient. III. Título.

Estudo do Laplaciano do [rafo para o droblema
de Musterização Yspectral e gegmentação
]nterativa de]magens

Wallace Correa de Oliveira Casaca

SERVIÇO DE PÓS-GRADUAÇÃO DO ICMC-USP

Data de Depósito:

Assinatura:

Estudo do Laplaciano do [rafo para o droblema de Wusterização Yspectral e gegmentação]nterativa de]magens

Wallace Correa de Oliveira Casaca

***Orientador:* Prof. Dr. Luis Gustavo Nonato**

***Co-orientador:* Prof. Dr. Gabriel Taubin**

Tese apresentada ao Instituto de Ciências Matemáticas e de Computação - ICMC-USP, como parte dos requisitos para obtenção do título de Doutor em Ciências - Ciências de Computação e Matemática Computacional. *VERSÃO FINAL*.

USP – São Carlos
Fevereiro de 2015

Aos meus amigos e familiares,

Dedico.

Acknowledgements

Dedico esta tese a todos aqueles que, de alguma forma, contribuíram nesta árdua jornada desde meu ingresso no curso de doutorado até seu término. De fato, inúmeros desafios foram superados ao longo desta caminhada, os quais necessitaram de empenho, foco, comprometimento e madrugadas a claro trabalhando no intuito de serem vencidos, entretanto, o maior benefício alcançado nesta fase não foi o produto final gerado - a tese e seus artigos decorrentes, mas sim toda a experiência vivenciada no decorrer deste percurso. Nesse sentido, tive a oportunidade de ter vivenciado uma prospera “vida acadêmica” em termos de compartilhar experiências acadêmicas com alunos e pesquisadores da USP e de promover colaborações bastante frutíferas com outros grupos de pesquisadores no exterior, em particular da Brown e do MIT.

Ainda com relação às experiências de pesquisa vivenciadas deste período, é importante ressaltar também que estive cercado de diversos amigos e familiares iluminados que, em etapas decisivas, me incentivaram a continuar mantendo o foco nos objetivos. Nesse sentido, reservo os parágrafos a seguir para agradecer:

À Deus, pela oportunidade concedida de aprender com a matemática, com a computação, com as ciências exatas e, principalmente, com as pessoas ao meu redor.

À minha querida mãe, Neuza, pelo apoio e amor incondicionais manifestados durante todo o desenvolvimento da tese.

À minha irmã, Aline, e meu cunhado, Cae, pelos momentos de descontração, incentivo e pelos conselhos sempre bem-vindos.

À minha “amorzoquinha”, Marilaine, pela convivência, amor infindável e pelo sorriso contínuo que ela sempre consegue estampar no meu rosto, mesmo diante das circunstâncias mais adversas.

Aos meus familiares, que me ensinaram que o aspecto humano deve ser valorizado acima de tudo. Em especial, agradeço à minha tia Shirley, ao meu tio Edinho, e à minha afilhada Dhara.

Aos amigos e também colaboradores, Marcos Proença, Erick Gomez-Nieto e Danilo Motta, pela amizade, pelas infinitas madrugadas em claro trabalhando e pelo engajamento na produção científica.

Ao amigos, Vinícius Malatesta, Eduardo Almeida, Carol Remondi, Vanessa Maria, Kárida Mateus, Juliana e Rafa EL, por toda a ajuda prestada na ocasião em que estive no exterior e pela valiosa amizade que lá foi construída.

Ao meu orientador, Prof. Luis Gustavo Nonato, o qual tenho como modelo de profissional competente e pesquisador comprometido com a pesquisa, além de ser uma pessoa extremamente empenhada com a formação de seus orientandos.

Ao Prof. Gabriel Taubin (Brown) e Prof. Ramesh Raskar (MIT), pela confiança, experiência acadêmica e pelas valiosas orientações durante a pesquisa conduzida no exterior.

Aos membros titulares de minha banca, Prof. Marcelo Siqueira (UFRN), Prof. Siome Goldenstein (IC-UNICAMP), Prof. Roberto Marcondes Cesar Junior (IME-USP), Prof. João Batista Neto (ICMC-USP), e também aos membros suplentes, Prof. Alexandre Falcão (IC-UNICAMP), Prof. Aparecido Nilceu Marana (UNESP), Prof. Odemir Bruno (IFSC-USP),

Prof. Adilson Gonzaga (EESC-USP) e Prof. Afonso Paiva Neto (ICMC-USP), pelas valiosas sugestões e contribuições.

A todos meus amigos de longa data, pela amizade incondicional ao longo dos anos e por me apoiarem principalmente nos momentos mais conturbados do curso.

Aos professores do ICMC, Antonio Castelo Filho e Maria Cristina Ferreira de Oliveira, e do IBILCE-UNESP, Alagacone Sri Ranga, Maurílio Boaventura e Cleonice Bracciali que, diretamente, contribuíram no enriquecimento de minha formação acadêmica e intelectual.

Aos amigos de pós-graduação e do LMACC-USP, pelos momentos de descontração, estudo e convívio acadêmico.

Aos colegas de graduação dos cursos de Bacharelado em Matemática Aplicada e Computacional, Matemática Pura e Licenciatura, pela amizade e apoio. Também agradeço à minha professora Clécia, pela ajuda e dedicação prestadas.

Às agências de fomento CAPES, FAPESP e seus acessores, pela oportunidade concedida e pelo financiamento durante toda a pesquisa de doutorado e, também, durante minha estadia no exterior.

Finalmente, finalizo minhas considerações vislumbrando a possibilidade de aprender cada vez mais com o exercício da docência e da pesquisa científica, as quais me motivam a prosseguir trabalhando cada vez mais com mais afinco.

*“I’m a great believer in luck,
and I find the harder I work
the more I have of it.”*

Coleman Cox

Abstract

*I*mage segmentation is an essential tool to enhance the ability of computer systems to efficiently perform elementary cognitive tasks such as detection, recognition and tracking. In this thesis we concentrate on the investigation of two fundamental topics in the context of image segmentation: spectral clustering and seeded image segmentation. We introduce two new algorithms for those topics that, in summary, rely on Laplacian-based operators, spectral graph theory, and minimization of energy functionals. The effectiveness of both segmentation algorithms is verified by visually evaluating the resulting partitions against state-of-the-art methods as well as through a variety of quantitative measures typically employed as benchmark by the image segmentation community.

Our spectral-based segmentation algorithm combines image decomposition, similarity metrics, and spectral graph theory into a concise and powerful framework. An image decomposition is performed to split the input image into texture and cartoon components. Then, an affinity graph is generated and weights are assigned to the edges of the graph 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. Moreover, the image partitioning can be improved by changing the graph weights by sketching interactively. Visual and numerical evaluation were conducted against representative spectral-based segmentation techniques using boundary and partition quality measures in the well-known BSDS dataset.

Unlike most existing seed-based methods that rely on complex mathematical formulations that typically do not guarantee unique solution for the segmentation problem while still being prone to be trapped in local minima, our segmentation approach is mathematically simple to formulate, easy-to-implement, and it guarantees to produce a unique solution. Moreover, the formulation holds an anisotropic behavior, that is, pixels sharing similar attributes are preserved closer to each other while big discontinuities are naturally imposed on the boundary between image regions, thus ensuring better fitting on object boundaries. We show that the proposed approach significantly outperforms competing techniques both quantitatively as well as qualitatively, using the classical “GrabCut” dataset from Microsoft as a benchmark.

While most of this research concentrates on the particular problem of segmenting an image, we also develop two new techniques to address the problem of image inpainting and photo colorization. Both methods couple the developed segmentation tools with other computer vision approaches in order to operate properly.

Keywords: image segmentation, spectral graph theory, seeded image segmentation, optimization techniques, laplacian operator, image inpainting, image colorization, computer vision applications

Resumo

Segmentar uma image é visto nos dias de hoje como uma prerrogativa para melhorar a capacidade de sistemas de computador para realizar tarefas complexas de natureza cognitiva tais como detecção de objetos, reconhecimento de padrões e monitoramento de alvos. Esta pesquisa de doutorado visa estudar dois temas de fundamental importância no contexto de segmentação de imagens: clusterização espectral e segmentação interativa de imagens. Foram propostos dois novos algoritmos de segmentação dentro das linhas supracitadas, os quais se baseiam em operadores do Laplaciano, teoria espectral de grafos e na minimização de funcionais de energia. A eficácia de ambos os algoritmos pode ser constatada através de avaliações visuais das segmentações originadas, como também através de medidas quantitativas computadas com base nos resultados obtidos por técnicas do estado-da-arte em segmentação de imagens.

Nosso primeiro algoritmo de segmentação, o qual é baseado na teoria espectral de grafos, combina técnicas de decomposição de imagens e medidas de similaridade em grafos em uma única e robusta ferramenta computacional. Primeiramente, um método de decomposição de imagens é aplicado para dividir a imagem alvo em duas componentes: textura e cartoon. Em seguida, um grafo de afinidade é gerado e pesos são atribuídos às suas arestas de acordo com uma função escalar proveniente de um operador de produto interno. Com base no grafo de afinidade, a imagem é então subdividida por meio do processo de corte espectral.

Além disso, o resultado da segmentação pode ser refinado de forma interativa, mudando-se, desta forma, os pesos do grafo base. Experimentos visuais e numéricos foram conduzidos tomando-se por base métodos representativos do estado-da-arte e a clássica base de dados BSDS a fim de averiguar a eficiência da metodologia proposta.

Ao contrário de grande parte dos métodos existentes de segmentação interativa, os quais são modelados por formulações matemáticas complexas que normalmente não garantem solução única para o problema de segmentação, nossa segunda metodologia aqui proposta é matematicamente simples de ser interpretada, fácil de implementar e ainda garante unicidade de solução. Além disso, o método proposto possui um comportamento anisotrópico, ou seja, pixels semelhantes são preservados mais próximos uns dos outros enquanto descon- tinuidades bruscas são impostas entre regiões da imagem onde as bordas são mais salientes. Como no caso anterior, foram realizadas diversas avaliações qualitativas e quantitativas en- volvendo nossa técnica e métodos do estado-da-arte, tomando-se como referência a base de dados “GrabCut” da Microsoft.

Enquanto a maior parte desta pesquisa de doutorado concentra-se no problema específico de segmentar imagens, como conteúdo complementar de pesquisa foram propostas duas no- vas técnicas para tratar o problema de retoque digital e colorização de imagens.

Keywords: segmentação de imagens, teoria espectral de grafos, segmentação interativa de imagens, técnicas de otimização, operador laplaciano, retoque digital, colorização de ima- gens, aplicações em visão computacional.

Contents

Abstract	vii
Resumo	ix
Summary	xi
List of Figures	xv
List of Tables	xxiii
1 Introduction	1
1.1 Contributions	3
1.2 Scientific Dissemination and Awards	6
1.3 Thesis Outline	9

2	Graph Laplacian: Review and Fundamental Concepts	11
2.1	The Discrete Laplacian Operator	12
2.2	Laplacian Operator and Spectral Segmentation	14
2.3	Graph Laplacian-based Energy Functionals	17
3	Spectral Image Segmentation	23
3.1	Spectral Image Segmentation Methods	23
3.1.1	Normalized Cut Segmentation	25
3.1.2	Multiscale-based Segmentation	27
3.2	Spectral Image Segmentation via Cartoon-Texture Decomposition and Inner Product-based Metric	28
3.2.1	Pipeline Overview	30
3.2.2	Results, Comparisons and Evaluation	40
3.2.3	Discussion and Limitations	49
4	Seeded Image Segmentation	51
4.1	Seed-based Image Segmentation Methods	51
4.1.1	Graph Cut-based Approach	54
4.1.2	Random Walker-based Approach	57
4.1.3	Watersheds/Maximum Spanning Forest	58

4.1.4	Shortest Path Forest	59
4.2	Laplacian Coordinates for Image Segmentation	60
4.2.1	Minimizing the Laplacian Coordinates Energy Functional on Graphs	63
4.2.2	Results, Comparisons and Evaluation	74
4.2.3	Discussion and Limitations	81
5	Laplacian Coordinates as an Effective Tool for Computer Vision Applications	83
5.1	User-Friendly Interactive Image Inpainting with Laplacian Coordinates	84
5.1.1	Pipeline Overview	86
5.1.2	User-steered Selection of the Inpainting Domain	87
5.1.3	Filling Order Assignment	87
5.1.4	Dynamic Processing of the Sampling Region	90
5.1.5	Block-based pixel replication	91
5.1.6	Experimental Results and Comparisons	93
5.2	Interactive Image Colorization using Laplacian Coordinates	99
5.2.1	Pipeline Overview	101
5.2.2	Prior Segmentation	101
5.2.3	Color Assignment	101
5.2.4	Interactive Manipulation of the Colorization Result	103
5.2.5	Results and Comparisons	104

6 Conclusion	107
References	127

List of Figures

1.1	Some results obtained during the PhD research period.	4
1.2	Dissemination of the technologies originated during the PhD to the wider public.	7
1.3	Some visual results obtained by the awarded papers.	8
2.1	A trivial connected graph with five vertices and its combinatorial Laplacian matrix $\mathbf{K}$	15
2.2	The partitioning of the graph from Fig. 2.1 using the signs (“zero-set” criterion) of the Fiedler vector (second column) taken from the matrix of eigenvectors $\mathcal{V}$	16
2.3	Simple illustrative graph with six vertices.	16
2.4	Graph partition obtained by the Fiedler vector.	16
2.5	Graph partition obtained by the 3rd order eigenvector.	16

2.6	Graph partition obtained by the 4th order eigenvector.	16
2.7	Graph partition obtained by the 5th order eigenvector.	17
2.8	Graph partition obtained by the 6th order eigenvector.	17
3.1	(a) Original image, (b) ground truth, (c) the result obtained by the NCut technique [Shi and Malik, 2000] and (d) NCut multiscale [Cour et al., 2005].	25
3.2	Optimal cut (in green) obtained by NCut algorithm for a trivial connected graph.	27
3.3	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.	30
3.4	Pipeline of the proposed image segmentation framework.	31
3.5	Image decomposition into a cartoon and texture component.	32
3.6	Fine-to-coarse step illustration obtained from [Shuai et al., 2007].	33
3.7	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). . . .	35
3.8	Spectral cut pipeline to partition the image from the zero-set of the Fiedler vector.	36
3.9	Hierarchical segmentation by recursively computing the spectral decomposition for multiple parts of the image.	36
3.10	Automatic result with the proposed framework.	37

3.11	Improving segmentation of the noise-textured image from user's strokes. . .	39
3.12	A simple stroke (greenish region between the two owls) is sufficient to improve the segmentation.	39
3.13	The influence of the user intervention in comparison with static approaches.	41
3.14	The result of applying NCut, MS-NCut, and the proposed approach (in automatic mode) in a fingerprint image.	42
3.15	Our approach (in automatic mode) produces smoother segmentation curves when compared to NCut and MS-NCut.	42
3.16	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 [Estrada and Jepson, 2009].	45
3.17	From left to right: The input image, the segmentations resulting from NCut, MS-NCut, our approach and the human-drawn ground-truth.	46
3.18	From left to right: The input image, the segmentations resulting from NCut, MS-NCut, our approach and the human-drawn ground-truth.	47
3.19	Segmentation driven by user produced by our technique (top row), RWS-EP (middle row) and RWS-EPP (bottom row). Fig. 3.19(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. 3.19(e) and 3.19(i))	48
4.1	Segmentations taking as input the seeded image (a). (b) Depicts the ground truth, (c) the result obtained by the Graph-Cut technique [Boykov and Funkalea, 2006], (d) Power Watershed [Coupric et al., 2011], (e) Geodesic-based [Anh et al., 2012] and (f) Random Walker [Grady, 2006].	54

4.2	Application of the binary graph cut model for a very simple connected graph. (a) Shows the input graph where the seeded links (in red) are managed by the term D_i in Eq. (4.2) and the quantity $V_{i,j}$ manages the connections (in black) between graph nodes. (b) Shows the partitioning into two disjunct sets.	55
4.3	Segmentation from a user-provided box proposed in [Rother et al., 2004]. (a) Presents in red the selection performed by the user and (b) shows the segmentation result.	56
4.4	Interpretation of the Random Walker approach in terms of probability of a random walker $\mathbf{P}$ reaching a seeded point ($\mathbf{F}$ or $\mathbf{B}$).	58
4.5	(a)-(b) Segmentations obtained by the Livewire-based approach (extracted from the paper [Hamarneh et al., 2005]).	60
4.6	Segmentations produced by the Laplacian Coordinates approach, where red and green scribes indicate the seedings.	62
4.7	Geometric interpretation of the differential operator δ_i at vertex $i = 5$. It measures the deviation between x_5 and its center of mass $\frac{1}{d_5} \sum_{j \in N(5)} w_{ij} x_j$.	66
4.8	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.	68
4.9	Random Walker and Laplacian Coordinates image boundary fitting capability. (First and Third rows) Seeded images and the segmentation results obtained from Random Walker and Laplacian Coordinates, respectively, and (Second and Fourth rows) Graphs of the solution associated to First and Third rows, respectively.	70

4.10	Geometric sketch showing the solution x_i ($i = 7$) in terms of its neighborhood pixels. In the example, the purple circle illustrates the points used for calculating the differential coordinate δ_i at pixel $i = 11$	71
4.11	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. . .	72
4.12	Extension of the Laplacian Coordinates (4.9) 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.	73
4.13	The "Grabcut" dataset benchmark. (a) Depicts the original image, (b) the input tri-map seeds, (c) the ground truth and (d) the unsegmented region. . .	74
4.14	Comparison of six seed-based segmentation methods regarding to RI and GCE quality metrics. In both cases, the proposed Laplacian Coordinates framework has presented better results then all other five evaluated techniques.	77
4.15	Comparison of six seed-based segmentation methods regarding to VoI and BDE quality metrics. In general, Laplacian Coordinates outperforms all other methods.	78
4.16	F -score quality metric. Laplacian Coordinates is considerably better than other methods when parameter R increases.	78
4.17	From left to right: Ground-truth, the tri-map images (seeds and the unknown region) provided by the Grabcut dataset, the segmentations resulting from GC, MSFK, MSKP and LC approach.	79

4.18	From left to right: Ground-truth, the tri-map images (seeds and the unknown region) provided by the Grabcut dataset, and segmentations resulting from PWS, RW and LC approach.	80
5.1	Image inpainting and colorization using the proposed frameworks.	84
5.2	Pipeline of our interactive inpainting framework.	86
5.3	Illustration of our interactive image inpainting framework.	87
5.4	Representation of the gradient field in an illustrative image.	88
5.5	Illustration of the priority filling order mechanism (Equation (5.2)).	90
5.6	The use of the dynamic sampling scheme to deal with huge photography for a given processing time	91
5.7	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.	92
5.8	Comparison with existing inpainting methods. [Efros and Leung, 1999] (<i>pure texture synthesis</i>), [Bertalmío et al., 2003] (<i>cartoon/texture inpainting</i>) and [Criminisi et al., 2004] (<i>exemplar-based inpainting</i>).	95
5.9	Comparison with existing inpainting methods. [Cao et al., 2011] (<i>skeleton-guided inpainting</i>) and [Kawai et al., 2009] (<i>optimization-based inpainting</i>).	96

5.10 Comparison with sparse representation-based inpainting. (a) Input images (with missing block in black), (b) inpainted by Guleryuz [2006], (c)-(d) inpainted by Elad et al. [2005] and Fadili et al. [2009], (e) inpainted by Xu and Sun [2010], (f) inpainted by Li [2011], (g) inpainted by our method, respectively from top to down and (h) the ground-truth images.	98
5.11 Pipeline of the proposed colorization framework.	101
5.12 Illustration of our interactive image colorization framework.	102
5.13 The use of our colorization framework when allowing for user intervention.	103
5.14 Comparison between our framework and Casaca et al. [2012b] technique.	105
5.15 Comparison between our framework and Levin et al. [2004] technique.	105
5.16 Comparison between our framework against Levin et al. [2004], Yao et al. [2011], Yatziv and Sapiro [2006] methods.	106

List of Tables

3.1	Comparative timing table (in seconds) with respect to experiment involving user autonomy (Fig. 3.19, image dimension: 256×256). The timing +4 means the time taken for the inclusion of seeds.	49
4.1	Variation of the parameters p and q in Equation (4.1) versus classical segmentation algorithms [Couprie et al., 2011].	53
5.1	Quantitative evaluation using PSNR (in dB) for all comparative images from Figure 5.10.	98

Introduction

The use of the Laplacian operator has been extensively exploited in a variety of scientific applications where smoothness of the resulting data is a characteristic of paramount importance. The mathematical community, for instance, employs the Laplacian operator when investigating fluid dynamics, graph theory and spectral analysis of signals. The Laplacian operator also appears as an indispensable tool in the context of *image segmentation*, *pattern recognition* and *clustering*. Indeed, Laplacian-based mechanisms have turned out quite effective for processing information, assisting a great deal of sophisticated applications, specially those that rely on image segmentation as an intermediate step to high-level tasks and computer vision systems such as object recognition and tracking.

In fact, the capability of handling images as *graphs* [[Diaz et al., 2002](#), [Tolliver and Miller, 2006](#)] rendered Laplacian operators an effective and flexible segmentation procedure, as different choices can be made towards defining the connectivity of pixels as well as edge weights of the graph. Such pliability as to graph construction and edge weighting can be

successfully combined with the human capability of distinguishing patterns, structures and shapes, making Laplacian-based methods even more versatile.

Inspired mainly by visual human skill and the solid mathematical foundation of graph theory [Bollobás, 1998, Spielman, 2007], *semi-supervised image segmentation techniques* have become a trend, allowing users to interact with images in order to properly settle the Laplacian operator and further improve the segmentation. The user-assisted mechanism often defines the Laplacian operator in a similarity/affinity graph matrix which encodes image information such as colors, textures, gradients and other inherent image patterns. Moreover, it involves a cost function defined on the graph [Boykov and Funka-Lea, 2006, Casaca et al., 2014b, Cousty et al., 2009b, Grady, 2006] or solving a spectral-cut problem [Casaca et al., 2013b, Maji et al., 2011, Shi and Malik, 2000, Taylor, 2013]. However, as pointed out in [Peng et al., 2013, Vicente et al., 2008, Yi and Moon, 2012], 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 great computational effort, specially for processing high resolution images, such those obtained nowadays by mobile devices, thus undermining user interactivity.
4. They are highly sensitive to the adjustment of the edge weights in the graph.

This thesis focus on the use of Laplacian operator for the task of image segmentation and also to support general computer vision applications such as *image inpainting* [Bertalmío et al., 2003, Casaca et al., 2014a, Xu and Sun, 2010] and *image colorization* [Casaca et al., 2012b, Irony et al., 2005, Luan et al., 2007]. We mainly focus our attention on the study of

spectral cuts and *segmentation from energy functional minimization*, in an effort to address the issues raised above. Moreover, we overview the Laplacian operator in the context of graph theory, spectral analysis and energy optimization on graphs. As a result, the notion of the Laplacian operator can be understood from different perspectives enabling a comprehensive practical and theoretical review about its use in the field of image segmentation and image enhancement.

1.1 Contributions

The main contribution of this thesis are two new computer-assisted image partitioning techniques, one addressing the particular topic of spectral segmentation and, another focused on seeded image segmentation. The technique derived from spectral clustering family (reported in Chapter 3) combines image decomposition, similarity metrics, and spectral graph theory into a unified and powerful framework. The proposed algorithm holds attractive properties such as awareness of noise and texture data, accuracy in detecting image edges, low computational cost when performing the spectral decomposition and it enables human intervention. The novel seed-based segmentation method, described in Chapter 4, was designated to propagate a scalar field throughout the image graph so that pixels sharing similar attributes are kept closer to each other in terms of the assigned scalars while big jumps take place the boundary between distinct regions of the image. Furthermore, the algorithm is mathematically simple, easy-to-code, and guaranteed to produce a unique solution. The effectiveness of both techniques are assessed by comparisons with state-of-the-art methods using a variety of measures and datasets typically employed as benchmark by the community of computer vision.

As additional contribution, we also propose two new algorithms to address the problem of image inpainting and colorization, which are described in details in Chapter 5. The methods rely on the proposed segmentation apparatus and other computer vision methodologies to operate properly. Figure 1.1 illustrates a few results obtained during the doctoral research.

Figure 1.1: Some results obtained during the PhD research period.

The list below provides a summary of the main publications originated during the development of this thesis:

Publications Directly Related to the Thesis

- [[Casaca et al., 2011b](#)] Casaca, W., Paiva, A., Nonato, L.G., Spectral Segmentation using Cartoon-Texture Decomposition and Inner Product-based Metric, *24th Conference on Graphics, Patterns and Images (SIBGRAPI'2011)*, IEEE Computer Society, pp. 266-273, 2011.
- [[Casaca et al., 2012b](#)] Casaca, W., Gomez-Nieto, E., Ferreira, C. de O., Tavares, G., Pagliosa, P., Paulovich, F., Nonato, L.G., Paiva, A., Colorization by Multidimensional Projection, *25th Conference on Graphics, Patterns and Images (SIBGRAPI'2012)*, IEEE Computer Society, pp. 32-38, 2012.
- [[Casaca et al., 2013b](#)] Casaca, W., Paiva, A., Gomez-Nieto, E., Joia, P., Nonato, L.G., Spectral Image Segmentation using Image Decomposition and Inner Product-based Metric, *Journal of Mathematical Imaging and Vision (JMIV)*, Springer US, Volume 45(3), pp. 227-238, 2013.
- [[Casaca et al., 2014a](#)] Casaca, W., Almeida, M.P., Boaventura, M., Nonato, L.G., Combining Anisotropic Diffusion, Transport Equation and Texture Synthesis for Inpainting Textured Images, *Pattern Recognition Letters (PRL)*, Volume 36, pp. 36-45, 2014.
- [[Casaca et al., 2014b](#)] Casaca, W., Nonato, L.G., Taubin, G., Laplacian Coordinates for Seeded Image Segmentation, *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE Computer Society, pp. 384-391, 2014.

Other Publications

- [[Casaca et al., 2011a](#)] Casaca, W., Boaventura, M., Almeida, M.P., Classifying Texture and Free-Textured Content from Images using Nonlinear Anisotropic Diffusion, *5th Conference on Mathematical Analysis*, Volume 5, pp. 210-214, 2011.

- [Joia et al., 2012] Joia, P., Gomez-Nieto, E., Batista Neto, J., Casaca, W., Botelho, G., Paiva, A., Nonato, L.G., Class-specific Metrics for Multidimensional Data Projection applied to CBIR, *The Visual Computer (TVC)*, Springer Berlin, Volume 28(10), pp. 1027-1037, 2012.
- [Casaca et al., 2013a] Casaca, W., Almeida, M.P., Boaventura, M., Denoising Textured Images via Regularized Anisotropic Diffusion, *Chapter Book: An Introductory Guide to the Emerging Areas of Digital Image Processing* (available at amazon.com), editors: Akshaya Mishra, Zafir Nawaz, Zafir Shahid, 1st Edition, iConcept Press (Australia), pp. 48-71, 2013.
- [Gomez-Nieto et al., 2013] Gomez-Nieto, E., Casaca, W., Nonato, L.G., Taubin, G., Mixed Integer Optimization for Layout Arrangement, *26th Conference on Graphics, Patterns and Images (SIBGRAPI'2013)*, IEEE Computer Society, pp. 115-122, 2013.
- [Gomez-Nieto et al., 2014] Gomez-Nieto, E., Roman, F.S., Pagliosa, P., Casaca, W., Helou, E.S., Oliveira, M.C.F., Nonato, L.G., Similarity Preserving Snippet-Based Visualization of Web Search Results, *IEEE Transactions on Visualization and Computer Graphics (TVCG)*, IEEE Computer Society, Volume 20(3), pp. 457-470, 2014.
- [Casaca et al., 2015] Casaca, W., Gomez-Nieto, E., Motta, D., Hartmann, I., Taubin, G., Nonato, L.G., Dealing with Multiple Requirements in Geometric Arrangements, *IEEE Transactions on Visualization and Computer Graphics (TVCG)*, 13 pgs, IEEE Computer Society, 2015 (accepted for publication).

1.2 Scientific Dissemination and Awards

This section summarizes the dissemination of the technologies developed during the PhD studies towards making them closer and available to the general community (for an illustration, see Figure 1.2). We also list in this section the papers awarded in recent years.

Figure 1.2: Dissemination of the technologies originated during the PhD to the wider public.

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

Figure 1.3: Some visual results obtained by the awarded papers.

The following works have been awarded “Best Paper” or selected for “Honorable Mention” during the doctoral period (see Figure 1.3 for an illustration).

- **Best Paper Award** in *Image Processing and Computer Vision* at SIBGRAPI’2011: Spectral Segmentation using Cartoon-Texture Decomposition and Inner Product-based Metric, 24th Conference on Graphics, Patterns and Images, IEEE Computer Society, pp. 266-273, 2011.

- **Best Paper Award** in *Graphics and Visualization* at SIBGRAPI'2013: Mixed Integer Optimization for Layout Arrangement, 26th Conference on Graphics, Patterns and Images, IEEE Computer Society, pp. 266-273, 2011.
- **Honorable Mention** in *Poster/Demo Section* at ICCP'2013: Image Colorization based on Multidimensional Projection, 5th IEEE Conference on Computational Photography, IEEE Computer Society, 2013.

1.3 Thesis Outline

This thesis is structured in six chapters as summarized below:

Chapter 2 [Graph Laplacian: Review and Fundamental Concepts]: covers the studies about Laplacian operator on graphs and its use in image segmentation and graph clustering.

Chapter 3 [Spectral Image Segmentation]: presents the proposed algorithm for spectral image segmentation and clustering analysis. A review about the topic as well as a full evaluation against competing methods are also provided.

Chapter 4 [Seeded Image Segmentation]: describes the proposed technique designated to segment images by minimizing a novel functional of energy. Existing seed-based segmentation methods were also revisited and compared against our methodology.

Chapter 5 [Laplacian Coordinates as an Effective Tool for Computer Vision Applications]: introduces two novel frameworks that tackle the problem of image inpainting and photo colorization. Comparison with representative state-of-the-art methods are also provided.

Chapter 6 [Conclusion]: summarizes the thesis and its main contributions.

Graph Laplacian: Review and Fundamental Concepts

The Laplacian operator has appeared in a multitude of theoretical and applied scenarios leveraging new technologies and improving the effectiveness of existing ones. The possibility of defining the Laplacian operator on graphs has attracted great amount of research in important scientific fields such as *graph theory* [[Diaz et al., 2002](#), [Mohar, 1997](#)], *computer vision* [[Casaca et al., 2011b](#), [Milyaev and Barinova, 2013](#)], *computer graphics* [[Petronetto et al., 2013](#), [Zhang et al., 2007](#)] and *data exploration and visualization* [[Gomez-Nieto et al., 2013, 2014](#)]. In particular, the Laplacian operator plays an important role in clustering graphs, specially due to its good mathematical properties and its capability in promoting visual analysis.

In this chapter, we focus particularly on the use of graph Laplacian for graph clustering problems. From the graph Laplacian, one can obtain important information about the

topology and geometry of the graph, which allow to perform partitions by simply processing spectral content. In order to clarify those aspects and better contextualize our work, we review key concepts about *discrete Laplacian operator*, *Laplacian operator for spectral segmentation* and *graph Laplacian-based energy functionals*.

2.1 The Discrete Laplacian Operator

Let $G = (V, E, W_E)$ be a undirected graph. There are many ways to define the discrete Laplacian operator, which varies mainly in terms of its numerical applicability as well as the branch of mathematics under study [Godsil and Royle, 2001, Merris, 1994, Wardetzky et al., 2007, Zhang, 2004]. In this work, the general Laplacian operator is locally formulated through the weighted average of a discrete function $\mathbf{f} = (f_i)$ defined at vertex $i \in V$, that is,

$$(\mathbf{L}\mathbf{f})_i = (b_i)^{-1} \sum_{j \in N(i)} w_{ij}(f_i - f_j), \quad (2.1)$$

where w_{ij} represents the weight assigned to the edge $(i, j) \in E$, b_i is a tuning parameter and $N(i)$ indicates the set of vertices connected to i . Operator (2.1) is a *first order Laplacian* since it covers only first order neighbors. Furthermore, Laplacian operator (2.1) can be rewritten in terms of the following matrix factorization:

$$\mathbf{L} = \mathbf{B}^{-1}\mathbf{S}, \quad (2.2)$$

where $\mathbf{B}^{-1}$ is a diagonal matrix built from the coefficients $(b_i)^{-1}$ and $\mathbf{S}$ is a symmetric matrix with entries given by:

$$S_{ij} = \begin{cases} -w_{ij}, & \text{if } e_{ij} \in E \\ \sum_{k \in N(i)} w_{ik}, & \text{if } i = j \\ 0, & \text{otherwise} \end{cases}, \quad (2.3)$$

A fully symmetry matrix $\mathbf{L}$ can be obtained by using the auxiliary symmetric matrix $\mathbf{L}_{\text{sym}} = \mathbf{B}^{-1/2}\mathbf{S}\mathbf{B}^{-1/2}$ as follows:

$$\mathbf{L} = \mathbf{B}^{-1}\mathbf{S} = \mathbf{B}^{-1/2}\mathbf{B}^{-1/2}\mathbf{S}\mathbf{B}^{-1/2}\mathbf{B}^{1/2} = \mathbf{B}^{-1/2}\mathbf{L}_{\text{sym}}\mathbf{B}^{1/2}. \quad (2.4)$$

Equation (2.4) ensures that $\mathbf{L}$ and $\mathbf{L}_{\text{sym}}$ have the same eigenvalues which are all non-complex numbers. Moreover, this decomposition states that $\mathbf{v}_i$ is an eigenvector of $\mathbf{L}_{\text{sym}}$ and λ_i is its corresponding eigenvalue, iff the eigenvector of $\mathbf{L}$ that matches λ_i is given by $\mathbf{u}_i = \mathbf{B}^{-1/2}\mathbf{v}_i$.

The eigenvectors of $\mathbf{L}_{\text{sym}}$ in Equation (2.4) are orthogonal, which may not be true for the Laplacian matrix $\mathbf{L}$, however, one can make the eigenvectors of $\mathbf{L}$ orthogonal using the biased inner product:

$$\langle \mathbf{u}_i, \mathbf{u}_j \rangle_B = \mathbf{u}_i^T \mathbf{B} \mathbf{u}_j = \mathbf{v}_i^T \mathbf{v}_j = \delta_{ij}, \quad (2.5)$$

being $\mathbf{L}$ positive semi-definite if $w_{ij} \geq 0$. In other words:

$$\langle \mathbf{f}, \mathbf{L}\mathbf{f} \rangle_B = \mathbf{f}^T \mathbf{B} (\mathbf{B}^{-1}\mathbf{S}) \mathbf{f} = \mathbf{f}^T \mathbf{S} \mathbf{f} = \frac{1}{2} \sum_{i,j=1}^n w_{ij} (f_i - f_j)^2 \geq 0. \quad (2.6)$$

The Laplacian operator (2.1) holds other attractive properties such as the zero-sum property which ensures that the sum of entries in each row/column is equal to zero. Indeed, taking $\mathbf{f}$ as a constant vector (all its entries are formed by a unique real number), one realizes that $\mathbf{L}\mathbf{f} = \mathbf{0}$, that is, $\mathbf{f}$ is the eigenvector that corresponds to the zero eigenvalue. From this fact, important outcomes can be mathematically proved for the Laplacian operator. For instance, the multiplicity of the zero eigenvalue is equal to the number of connected components of the graph [Mohar, 1997].

2.2 Laplacian Operator and Spectral Segmentation

In this section we describe how to easily generate partitions in a graph by exploiting spectral properties of the Laplacian operator.

The *combinatorial graph Laplacian*, denoted by $\mathbf{K} = K(G)$, is the matrix built as follows:

$$K_{ij} = \begin{cases} -1, & \text{if } e_{ij} \in E \\ d_i, & \text{if } i = j \\ 0, & \text{otherwise} \end{cases}, \quad (2.7)$$

where $d_i = \#N(i)$ (the valency of vertex i). Figure 2.1 depicts a simple graph and its combinatorial Laplacian matrix $\mathbf{K}$. Matrix (2.7) establishes a relationship between the spectra and the topology of the graph [Diaz et al., 2002]. Notice that expression (2.7) can be generated by taking $b_i^{-1} = 1$ and $w_{ij} = 1$ in the general Laplacian operator (2.2). Indeed, if weights w_{ij} are set to the graph edges e_{ij} , then the so-called *graph Laplacian* $\mathbf{L}$ assumes the form of Matrix (2.3), that is:

$$\mathbf{L} = \mathbf{S}. \quad (2.8)$$

Matrix (2.8) accounts for the similarity between the nodes of the graph, since it embeds into its structure the edge weights w_{ij} . By employing the graph Laplacian $\mathbf{L}$, one can partition the set of nodes V into disjoint sets $V_1, V_2, \dots, V_n$ so that the similarity among nodes in V_i is high and similarity across V_i and V_j is low. In fact, a typical procedure to partition a graph is to compute the eigenvector associated to the second smallest eigenvalue of $\mathbf{L}$, namely the *Fiedler vector* [Fiedler, 1973]. Since Matrix (2.8) is symmetric and positive semi-definite, the solution of its associated eigenproblem

Figure 2.1: A trivial connected graph with five vertices and its combinatorial Laplacian matrix K .

$$Lx = \lambda x, \quad (2.9)$$

is given by non-negative real eigenvalues with respective real eigenvectors. According to *Courant Nodal Domain Theorem* [Biyikoglu et al., 2007], 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. Figure 2.2 shows an illustrative example of the Fiedler vector and the provided partitioning. This strategy has been successfully used in several approaches to cluster graphs as [Jameson et al., 2008, Sarkar and Soundararajan, 2000, Soundararajan and Sarkar, 2001] (*Average Cut*) and [Shi and Malik, 2000, Taylor, 2013] (*Normalized Spectral Cut*). Similarly to the traditional Fiedler vector, other eigenvectors can be also employed to clustering the graph as shown in the trivial example in Figs. 2.4-2.8.

Despite the usefulness and simplicity of the Fiedler vector and other independent eigenvectors in performing cuts in a graph, it does not allow for multiple scale partitions. Integrating multiscale analysis into the segmentation process is of paramount importance to establish well behaved partitions as pointed out in [Arbeláez et al., 2011, Cour et al., 2005]. In fact, techniques that preform multiscale decomposition from the Laplacian spectra usually produce satisfactory results when operating in general graphs, however, for example, they

Figure 2.2: The partitioning of the graph from Fig. 2.1 using the signs (“zero-set” criterium) of the Fiedler vector (second column) taken from the matrix of eigenvectors V .

Figure 2.3: Simple illustrative graph with six vertices.

Figure 2.4: Graph partition obtained by the Fiedler vector.

Figure 2.5: Graph partition obtained by the 3rd order eigenvector.

Figure 2.6: Graph partition obtained by the 4th order eigenvector.

Figure 2.7: Graph partition obtained by the 5th order eigenvector.

Figure 2.8: Graph partition obtained by the 6th order eigenvector.

typically demand intricate post-processing steps when segmenting images, since boundaries are not captured accurately.

2.3 Graph Laplacian-based Energy Functionals

Graph clustering problems are usually modeled in terms of discrete formulations: their solutions are partitions of a finite set, the set of graph vertices. However, methods that rely on minimization of energy functionals face this problem by employing continuous tools, instead. This is a trend observed in many complex discrete problems in recent years, specially those in the field of computer sciences.

In this spirit, energy minimization-based approaches have been proposed in a variety of practical scenarios. For instance, the computer graphics community has combined graph Laplacian energy functional minimization and concepts of riemannian manifolds to perform mesh partitioning and geometry processing [Chartrand et al., 2014, Liu et al., 2007]. In fact, Mohar [1997] has proved that the Laplacian matrix (2.8) can be interpreted as a discrete version of the Laplace-Beltrami operator defined on a riemannian manifold.

In the image segmentation context, instead of riemannian manifolds, the image affinity graph is taken as input and image features are conveniently combined and rearranged to tune weights of the graph. The well-known image segmentation framework proposed by Shi

and Malik [2000] is a good example of technique that relies on the minimization of a graph energy functional to proceed with the segmentation. Their energy cost function is built in terms of the classical minimum-cut measure:

$$cut(A, B) = \sum_{i \in A, j \in B} w_{ij}, \quad (2.10)$$

where A and B are disjoint partitions of the set of vertices V . Minimizing the $cut(A, B)$ will give a partition with the maximum disassociation w.r.t. edge weights. The partitioning provided by Equation (2.10) is not unique and the minimum cuts of the graph can be computed in polynomial time, as pointed out in [Nagamochi et al., 1997]. Despite its simplicity, the Minimum-Cut (2.10) will often choose a cut with one small cluster, thus constraining it for a reduced number of applications [Shi and Malik, 2000].

To avoid the “small cluster” bias, Shi and Malik [2000] have proposed to minimize the normalized version of Equation (2.10), that is:

$$NCut(A, B) = \frac{cut(A, B)}{vol(A)} + \frac{cut(A, B)}{vol(B)}, \quad (2.11)$$

where $vol(A) = \sum_{i \in A, j \in V} w_{ij}$. Letting $\mathbf{D}$ be a diagonal matrix such that $D_{ii} = \sum_{k \in N(i)} w_{ik}$. Equation (2.11) can be algebraically rearranged as follows:

$$NCut(A, B) = \frac{\mathbf{y}^t \mathbf{L} \mathbf{y}}{\mathbf{y}^t \mathbf{D} \mathbf{y}}, \quad (2.12)$$

subject to $\mathbf{y}^t \mathbf{D} \mathbf{1} = 0$ and $\mathbf{y} \in \mathcal{X} = \{(y_1, y_2, \dots, y_n) \mid y_i \in \{a, b\}, a, b \in \mathbf{R}^*, \mathbf{y}^t \mathbf{D} \mathbf{1} = 0\}$. The hard constraint $\mathbf{y}^t \mathbf{D} \mathbf{1} = 0$ is imposed to get rid of the denominator in the cost function, since Equations (2.12), $\mathbf{y}^t \mathbf{D} \mathbf{1} = 0$ and the elements in the solution space $\mathcal{X}$ are invariant to scale. Also, one can define $\mathbf{y} = (y_i)$ such that

$$y_i = \begin{cases} a, & \text{if } v_i \in A \\ b, & \text{if } v_i \notin A \end{cases}, \quad (2.13)$$

where a and b are real numbers, $a \neq b$. So, from Equations (2.12) and (2.13), we have that

$$NCut(A, B) = \frac{\mathbf{y}^t \mathbf{L} \mathbf{y}}{\mathbf{y}^t \mathbf{D} \mathbf{y}},$$

if and only if

$$a\alpha + b(d - \alpha) = 0,$$

with $d = \mathbf{1}^t \mathbf{D} \mathbf{1}$ and $\alpha = vol(A)$. On the other hand, one may easily note that $\mathbf{y}^t \mathbf{D} \mathbf{1} = a\alpha + b(d - \alpha) = 0$. Moreover, if a and b are nonzero scalars, $a\alpha + b(d - \alpha) = 0$ results in

$$b = -\frac{\alpha}{(d - \alpha)} a.$$

Computing the NCut Energy (2.12) so that Equation (2.13) is held is a NP-Complete problem, so an approximation is needed [Shi and Malik, 2000]. To overcome this problem, [Shi and Malik, 2000] make use of the Courant-Fischer Formula, which states the following:

Theorem 1. *Let $\mathbf{Q}$ be a $n \times n$ symmetric matrix with eigenvalues*

$$\lambda_n \geq \dots \geq \lambda_2 \geq \lambda_1 \geq 0.$$

If $\{\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_n\}$ is an orthogonal basis of eigenvectors for $\mathbf{Q}$, where $\mathbf{u}_i$ is the unitary eigenvector associated to λ_i , then:

$$\lambda_1 = \min_{\mathbf{x} \neq 0} \frac{\mathbf{x}^t \mathbf{Q} \mathbf{x}}{\mathbf{x}^t \mathbf{x}}$$

and

$$\lambda_i = \min_{\mathbf{x} \neq 0, \mathbf{x} \in \{\mathbf{u}_1, \dots, \mathbf{u}_{i-1}\}^\perp} \frac{\mathbf{x}^t \mathbf{Q} \mathbf{x}}{\mathbf{x}^t \mathbf{x}},$$

$$2 \leq i \leq n.$$

By using the Theorem (1) while relaxing Equation (2.12) to take real numbers, one may write $\mathbf{x} = \mathbf{D}^{\frac{1}{2}}\mathbf{y} \iff \mathbf{y} = \mathbf{D}^{-\frac{1}{2}}\mathbf{x}$ so that

$$\begin{aligned} \mathbf{y}^t \mathbf{D} \mathbf{y} &= 1 \iff \mathbf{x}^t \mathbf{x} = 1, \\ \mathbf{y}^t \mathbf{D} \mathbf{1} &= 0 \iff \mathbf{x}^t \mathbf{D}^{\frac{1}{2}} \mathbf{1} = 0, \\ \mathbf{y}^t \mathbf{L} \mathbf{y} &\iff \mathbf{x}^t \mathbf{D}^{-\frac{1}{2}} \mathbf{L} \mathbf{D}^{-\frac{1}{2}} \mathbf{x}. \end{aligned}$$

Finally, Equation (2.12) is simplified to the following quadratic energy function:

$$E(\mathbf{x}) = \mathbf{x}^T \underbrace{\mathbf{D}^{-1/2} \mathbf{L} \mathbf{D}^{-1/2}}_{\mathbf{L}_n} \mathbf{x}, \quad (2.14)$$

subject to the orthogonal penalty $\mathbf{x} \perp \mathbf{D}^{1/2} \mathbf{1}$, being as the approximation of the minimal solution the Fiedler vector from the following generalized eigenvector problem:

$$\mathbf{L} \mathbf{x} = \lambda \mathbf{D} \mathbf{x}. \quad (2.15)$$

Similar to the Expression (2.15), the technique advocated by Grady [2006] also makes use of the Laplacian operator to produce segmentations. Grady [2006] solves the classical Random Walk-based problem for the unlabeled vertex vector $\mathbf{x}$ subject to given labels provided by the user:

$$\mathbf{L} \mathbf{x} = 0 \iff (\mathbf{D} - \mathbf{W}) \mathbf{x} = 0 \iff \mathbf{P} \mathbf{x} = \mathbf{x}, \quad (2.16)$$

where $\mathbf{W}$ is the matrix of edge weights and $\mathbf{P} = \mathbf{D}^{-1} \mathbf{W}$ represents the *Tuttle Laplacian* which was firstly used by Taubin [1995] to design mesh fairing. The Tuttle Laplacian

can be obtained by taking $b_i = d_i$ when assuming the general form of Laplacian operator (2.1). Moreover, one can prove that $\mathbf{P}$ is similar to $\mathbf{L}_n$ from Equation (2.14), that is, $\mathbf{P} = \mathbf{D}^{-1/2}\mathbf{L}_n\mathbf{D}^{1/2}$.

Rewriting Equation (2.16) in terms of a minimization problem, the following quadratic energy is derived:

$$E(\mathbf{x}) = \mathbf{x}^T \mathbf{L} \mathbf{x}, \quad (2.17)$$

being mathematically constrained to the graph labels A and B provided by the user, that is, $x_A = 1$, $x_B = 0$. The linear algebra formulation of Equation (2.16) allows a unique minimizer for the energy functional (2.17), as described in [Grady, 2006].

Notice that the energy derived in Equation (2.14) is comparable to the cost function in Equation (2.17). In fact, they are quadratic forms of the Laplacian matrix (2.8), differing each other in terms of constraints imposed to the optimization problem as well as the minimization procedure.

Spectral Image Segmentation

In this chapter we introduce a novel user-assisted image segmentation technique, first reported in [Casaca et al., 2011b, 2013b], which combines three classical approaches usually exploited by the computer vision community: *image decomposition*, *similarity metrics*, and *spectral graph theory*. As part of the related work in the image segmentation field, we also briefly discuss a variety of existing methods that rely on spectral clustering on graphs. Finally, we end this chapter by conducting experimental evaluations against state-of-the-art spectral segmentation methods through a qualitative and quantitative analysis of the results.

3.1 Spectral Image Segmentation Methods

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* [Jiang et al., 2013, Wu, 1992], *content-based image retrieval* [Bruno et al., 2008, Ozden and Polat, 2007] and *object recognition* [Li et al., 2007, Tu et al., 2005], 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 specific attributes of the image [Diaz et al., 2002, Falcão et al., 2004, Noma et al., 2012, Pratt, 2008, Shi and Malik, 2000].

The strength of graph-based approaches comes from the solid mathematical foundation it relies on, since most of the well-established graph theory [Bollobás, 1998] can be directly used to manipulate the image segmentation problem. For instance, spectral graph theory [Chung, 1997] has been the basic tool for the so-called spectral cut method [Maji et al., 2011, Mohar, 1997, Shi and Malik, 2000, Spielman, 2007], 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 approach.

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 algorithms* [Carvalho et al., 2010a,b, Ma et al., 2008, Maji et al., 2011, Sáez et al., 2014, Shi and Malik, 2000, Tao et al., 2007, Taylor, 2013];
2. *Multiscale Segmentation-based algorithms* [Cai and Chung, 2006, Cour et al., 2005, Maire and Yu, 2013, Sun and He, 2009, Yu, 2004];
3. *Random Walker-based algorithms* [Andrews et al., 2009, Grady, 2005, 2006, Grady and Sinop, 2008, Sinop and Grady, 2007].

Figure 3.1: (a) Original image, (b) ground truth, (c) the result obtained by the NCut technique [Shi and Malik, 2000] and (d) NCut multiscale [Cour et al., 2005].

Figure 3.1 presents some results obtained from the approaches listed above. In the following, we briefly describe the first two groups (spectral-computed based) while the third one (energy functional based) is detailed in the Section 4.1.

3.1.1 Normalized Cut Segmentation

Shi and Malik [2000] introduced the concept of *Normalized Cut* (NCut), a graph optimization problem that aims to subdivide a graph so as to minimize the normalized sum of weights of the graph edges. They show that finding the optimal normalized cut is an NP-Hard problem, but feasible solutions can be reached from the Fiedler vector (see Section 2 for a remark) computed by solving a generalized eigenvalue problem.

More specifically, the NCut methodology builds a graph $G = (V, E, W_E)$ from the given image $\mathcal{I}$ by associating each pixel $P_i \in I$ with a node of the graph and connecting two pixels P_i and P_j with an edge $e_{ij} \in E$, if Equation (3.1) holds:

$$\|P_i - P_j\|_2 < r, \quad (3.1)$$

where P_i and P_j are the corresponding pixels to the nodes $i \in V$ and $j \in V$ while r is a parameter that measures how local the edges should be. The weights $w_{ij} = w(e_{ij}) \in W_E$ assigned to the edges are computed from the following formula:

$$w_{ij} = \exp \left(-\frac{\|P_i - P_j\|_2^2}{\sigma_P^2} - \frac{\|I_i - I_j\|_2^2}{\sigma_I^2} \right), \quad (3.2)$$

where I_i and I_j account for the 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 Equation (3.2). Variations of the pairwise function (3.2) have also been proposed in the literature of spectral cut which rely on learning local grouping cues from the image spectrum [Kim et al., 2013] or inferring graph node connections through random forest computation [Zhu et al., 2014].

The optimal cut separating the sets A and B such that $A \cap B = \emptyset$ and $A \cup B = V$ is performed by minimizing the normalized cut penalty function:

$$NCut(A, B) = \frac{cut(A, B)}{vol(A)} + \frac{cut(A, B)}{vol(B)}, \quad (3.3)$$

being $cut(A, B) = \sum_{i \in A, j \in B} w_{ij}$ and $vol(A) = \sum_{i \in A, k \in V} w_{ik}$. For an illustration, see Figure 3.2. In practice, Equation (3.3) is relaxed and solved by computing the eigenvectors associated to the graph Laplacian matrix $\mathbf{L}$ (see Section 2 for details) so that the generated solution is used to label the image pixels [Shi and Malik, 2000].

For instance, the multiscale technique proposed by [Cour et al. \[2005\]](#) attempts to decompose the affinity graph in multiple scales so that the local grouping cues are better propagated along image segments when the radius of the graph neighborhood increases. Given an initial radius r , an affinity matrix $\mathbf{W}$ is built from a sequence of subgraph matrices $\mathbf{W}_s$ at scale s , where the scale reflects pixels sampled at $(2r + 1)^{s-1}$ distance:

$$\mathbf{W} = \mathbf{W}_1 + \mathbf{W}_2 + \dots + \mathbf{W}_s. \quad (3.4)$$

Performing the spectral clustering from Matrix (3.4) allows for capturing coarse as well as fine details of the image, as pointed out in [\[Cour et al., 2005\]](#). A drawback of this algorithm is that to reach a good level of clustering refinement, the value of r must be large, rendering the method not so feasible for segmenting moderate or large size images.

Multiscale-based techniques for image partition have also appeared in the context of multidimensional projection [\[Cai and Chung, 2006\]](#), quad-tree decomposition [\[Carvalho et al., 2010b\]](#) and texture descriptors [\[Sun and He, 2009\]](#).

3.2 Spectral Image Segmentation via Cartoon-Texture Decomposition and Inner Product-based Metric

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 is highly dependent on the weights assigned to the edges of the graph. Although automatic schemes have been proposed to accurately compute those weights [\[Cai and Chung, 2006, Cour et al., 2005, Kim et al., 2013, Shi and Malik, 2000, Tolliver and Miller, 2006\]](#), it is well-known that user intervention is essential in many cases to correctly define object boundaries [\[Grady and Sinop, 2008\]](#). 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 pragmatic 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 [Koutis et al., 2009].

In this chapter we introduce a new methodology for image segmentation that relies on spectral cuts but addresses the issues raised above. The proposed methodology, first reported in [Casaca et al., 2011b, 2013b], can be summarized in the following steps:

- 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 [Vese and Osher, 2003, 2006] that facilitates the identification of different features contained in the image.
- Next, we provide a novel mechanism to assign weights to the edges of the affinity graph that results in accurate segmentation in most cases.
- In the third step, we describe how to build the similarity graph from a coarse representation of the input image without degrading the quality of the segmentation.
- In contrast to other spectral cut-based approaches, our technique allows for user intervention in critical situations, enabling users to interactively modify weights according to their perception.

We show in Section 3.2.2 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 [Martin et al., 2001]. Figure 3.3 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 [Casaca et al., 2013b];

Figure 3.3: 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 scribbles) to improve the segmentation.

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.

3.2.1 Pipeline Overview

The proposed approach comprises five main steps, as illustrated in Figure 3.4. 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

Figure 3.4: Pipeline of the proposed image segmentation framework.

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 [Demanet and Ying, 2007a,b] in order to incorporate the high-level oscillatory information into the spectral cut process.

Details about each step of the pipeline are provided below.

Cartoon-Texture Image Decomposition

The *Cartoon-Texture 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 [Casaca et al., 2011a, Vese and Osher, 2003] and the underlying mathematical theory proposed in [Meyer, 2001]). Similar to [Vese and Osher, 2003], where a functional minimization problem has been formulated and solved through a system of partial differential equations, both

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

$$\begin{cases} \mathcal{C} = \mathcal{I} - \partial_x g_1 - \partial_y g_2 + \frac{1}{2\lambda} \operatorname{div} \left(\frac{\nabla \mathcal{C}}{|\nabla \mathcal{C}|} \right) \\ \mu \frac{g_1}{\sqrt{g_1^2 + g_2^2}} = 2\lambda \left[\frac{\partial}{\partial x} (\mathcal{C} - \mathcal{I}) + \partial_{xx}^2 g_1 + \partial_{xy}^2 g_2 \right] \\ \mu \frac{g_2}{\sqrt{g_1^2 + g_2^2}} = 2\lambda \left[\frac{\partial}{\partial y} (\mathcal{C} - \mathcal{I}) + \partial_{xy}^2 g_1 + \partial_{yy}^2 g_2 \right] \end{cases}, \quad (3.5)$$

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

$$\begin{cases} \frac{\nabla \mathcal{C}}{|\nabla \mathcal{C}|} \cdot (n_x, n_y) = 0 \\ (\mathcal{I} - \mathcal{C} - \partial_x g_1 - \partial_y g_2) \cdot n_x = 0 \\ (\mathcal{I} - \mathcal{C} - \partial_x g_1 - \partial_y g_2) \cdot n_y = 0 \end{cases}. \quad (3.6)$$

Mathematically, the cartoon component $\mathcal{C}$ is a bounded variation function, $\vec{g} = (g_1, g_2) \in L^2(\mathbb{R}^2)$ where the texture component $\mathcal{T} = \operatorname{div}(\vec{g})$, and the constants $\lambda, \mu > 0$ are tuning parameters. Equations (3.5) 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 [Vese and Osher, 2003, 2006]). Figure 3.5 shows the result of the CTD scheme applied to a digital image.

Figure 3.5: Image decomposition into a cartoon and texture component.

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 [Vese and Osher, 2003]. 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.

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 [Shuai et al., 2007], which minimizes the blurring effect while still preserving gradients in the coarse image (see Figure 3.6 for an illustration). Other downsampling techniques such as [Zhang et al., 2011] 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. In fact, this performance gain can be seen in the evaluation shown in Table 3.1.

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

Figure 3.6: Fine-to-coarse step illustration obtained from [Shuai et al., 2007].

words,

$$\|P_i - P_j\|_\infty < r. \quad (3.7)$$

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. (3.2)), our 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:

$$w_{ij} = \frac{1}{1 + \eta h_{ij}^2}, \quad h_{ij} = \max \left\{ \frac{\partial \tilde{\mathcal{C}}(P_i)}{\partial \vec{d}_{ij}}, \frac{\partial \tilde{\mathcal{C}}(P_j)}{\partial \vec{d}_{ji}}, 0 \right\}, \quad (3.8)$$

$$\frac{\partial \tilde{\mathcal{C}}(x)}{\partial \vec{d}_{ij}} = \langle \nabla \tilde{\mathcal{C}}(x), \vec{d}_{ij} \rangle, \quad \text{with} \quad \vec{d}_{ij} = \frac{\overrightarrow{P_i P_j}}{|\overrightarrow{P_i P_j}|}. \quad (3.9)$$

The left most term in Equation (3.9) is the directional derivative of $\tilde{\mathcal{C}}$ in the direction $\vec{d}_{ij}$, 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 Equation (3.2), our formulation accounts for the intensity and geometric information to define the weights through the inner-product in the edge direction. Figure 3.7 provides a geometric interpretation of the proposed metric.

The effective weights w_{ij} are chosen from Eq. (3.8) rather than using the exponential measure usually employed by other authors [Carvalho et al., 2010b, Shi and Malik, 2000, Sun and He, 2009]. The scheme proposed in Eq. (3.8) 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. (3.8) is indeed derived from the Malik-Perona diffusivity term [Casaca and Boaventura, 2009, Perona and Malik, 1990],

Figure 3.7: 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).

which was originally used for establishing the notion of anisotropy in partial differential equations. Moreover, the inner product-based metric (3.8) 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.

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 [Shi and Malik, 2000]. 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

$$(\mathbf{D} - \mathbf{W})\mathbf{x} = \lambda\mathbf{D}\mathbf{x}, \quad (3.10)$$

getting $\mathbf{f}$ as the eigenvector associated to the smallest non-zero eigenvalue.

Figure 3.8: Spectral cut pipeline to partition the image from the zero-set of the Fiedler vector.

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. Figure 3.8 portrays the spectral cut procedure. The partitioning created in $\tilde{\mathcal{C}}$ is then brought back to $\mathcal{C}$ using bicubic interpolation from f .

Multiple partitions can also be reached by recursively computing the spectral decomposition for each part of the image before the interpolation process, as depicted in Figure 3.9. This recursive process, namely “two-way” cut, was employed rather than the simultaneous “k-way” procedure [Shi and Malik, 2000] due to the supervised step of our algorithm, which manages the computational burden of the spectral decomposition by only performing new cuts in the regions modified by the user. 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 3.10 shows the result of applying our methodology to segment a fingerprint im-

Figure 3.9: Hierarchical segmentation by recursively computing the spectral decomposition for multiple parts of the image.

Figure 3.10: Automatic result with the proposed framework.

age. For the sake of comparison, we show in Fig. 3.10(a) the result of computing weights (using Eq. (3.8)) directly from the original image $\mathcal{I}$, that is, skipping the CTD, while Fig. 3.10(b) depicts the result using CTD and Eq. (3.2) instead of Eq. (3.8) to define the graph weights. Notice, from Fig. 3.10(c), how better the segmentation is when Eq. (3.8) and CTD are combined. This combination allows our methodology to deal only with the cartoon part of the image in the first stage of the segmentation so that Metric (3.2) can operate more efficiently.

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* [Demanet and Ying, 2007a,b].

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 [Casaca and Boaventura, 2010] 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”.

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

$$w_{ij} = \rho \left(\min_{e_{ij} \in E, w_{ij} \neq 0} w_{ij} \right) \left(1 - \max\{S(\tilde{\mathcal{T}}_i), S(\tilde{\mathcal{T}}_j)\} \right), \quad (3.11)$$

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 graph to be repartitioned.

Figure 3.11 shows the result of segmenting the image in Fig. 3.11(a) using our method setting 10 partitions. Notice from Fig. 3.11(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. 3.11(c) and Fig. 3.11(d). This post-segmentation was only feasible because the texture mapping used to accomplish this task is sensitive to noise. Figure 3.12 shows that is not necessary to perform a large number of user interventions to achieve the desired segmentation. The red curve in Fig 3.12(b) is the result produced by our method without user intervention. The simple greenish stroke depicted on the texture region between the two owls (setting the partition number to 4) were enough to separate the birds, as depicted in Fig. 3.12(c).

Figure 3.11: Improving segmentation of the noise-textured image from user's strokes.

Figure 3.12: A simple stroke (greenish region between the two owls) is sufficient to improve the segmentation.

3.2.2 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 [Shuai et al., 2007] for the bicubic interpolation and a hard threshold at 3σ (noise-to-signal ratio of the image) combined with cycle spinning [Demanet and Ying, 2007b] for the wave atom transform. We set $r = 1$ and $\eta = 5$ in Equations (3.1)–(3.8), 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)¹ [Sáez et al., 2014, Shi and Malik, 2000];
- Multiscale Normalized Cuts method (MS-NCut)² [Cour et al., 2005] (with radius 2.3);
- Random Walker-based Segmentation with pure Eigenvector Precomputation (RWS-EP) [Grady and Sinop, 2008];
- Random Walker based on Eigenvector Precomputation and Prior scheme (MSFP)³ (with 80 precomputed eigenvectors) [Andrews et al., 2009];

¹available at <http://note.sonots.com/SciSoftware/NcutImageSegmentation.html>

²available at <http://www.cis.upenn.edu/~jshi/software/>

³available at <http://fastrw.cs.sfu.ca>

Figure 3.13: The influence of the user intervention in comparison with static approaches.

Comparison with Automatic Methods

The first experiment shown in Figure 3.13 presents a comparative analysis of our technique against the non-interactive NCut and MS-NCut approaches. We can see that both classical NCut (Fig. 3.13(b)) and MS-NCut (Fig. 3.13(c)) badly segment parts of the image. Our approach results in a better partitioning (Fig. 3.13(d)), although some regions are also segmented in an incorrect way. After user intervention, shown in Fig. 3.13(e), the result improves considerably (Fig. 3.13(f)).

The result of applying the three above-mentioned methods in a fingerprint image for two partitions is shown in Figure 3.14. Notice that the NCut (Fig. 3.14(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. 3.14(b) that the MS-NCut tends to produce a jagged segmentation curve while

Figure 3.14: The result of applying NCut, MS-NCut, and the proposed approach (in automatic mode) in a fingerprint image.

Figure 3.15: Our approach (in automatic mode) produces smoother segmentation curves when compared to NCut and MS-NCut.

our method results in a smoother curve, as shown in Fig. 3.14(c). It becomes clear from Figure 3.15 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.

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 [Estrada and Jepson, 2009] on the well-known *Berkeley Segmentation Dataset (BSDS)* [Arbeláez et al., 2011], 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

$$\text{Recall} = \frac{\text{Matched}(S_{\text{target}}, S_{\text{source}})}{|S_{\text{target}}|}, \quad (3.12)$$

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 (3.12) 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} .

- **Precision:** holds the opposite situation, that is,

$$\text{Precision} = \frac{\text{Matched}(S_{\text{source}}, S_{\text{target}})}{|S_{\text{source}}|}. \quad (3.13)$$

- **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:

$$F\text{-Score} = 2 \frac{\text{Precision Recall}}{\text{Precision} + \text{Recall}}. \quad (3.14)$$

Figure 3.16 presents the computation of Recall (3.12), Precision (3.13) and the F-Score (3.14) 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.

Figures 3.17-3.18 depict the partitioning produced by NCut, MS-NCut, and our approach when applied to illustrative images randomly extracted from the BSDS dataset [Arbeláez et al., 2011]. 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*, *horse*, *man*, and *surfer* were better captured by our technique.

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 [Grady and Sinop, 2008] and RWS-EPP [Andrews et al., 2009]), as depicted in Figure 3.19. In contrast with random walk algorithms 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-EPP, which require much more user interference to reach a pleasant result (Figs. 3.19(h) and 3.19(l)), our approach needed only two brushes to yield the desired result (Fig. 3.19(d)). Besides, our technique is considerably faster when compared against RWS-EP and RWS-EPP (see Table 3.1).

Figure 3.16: 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 [Estrada and Jepson, 2009].

Figure 3.17: From left to right: The input image, the segmentations resulting from NCut, MS-NCut, our approach and the human-drawn ground-truth.

Figure 3.18: From left to right: The input image, the segmentations resulting from NCut, MS-NCut, our approach and the human-drawn ground-truth.

Figure 3.19: Segmentation driven by user produced by our technique (top row), RWS-EP (middle row) and RWS-EPP (bottom row). Fig. 3.19(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. 3.19(e) and 3.19(i))

Processing stage	RWS-EP method	RWS-EPP method	Proposed Framework
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 3.1: Comparative timing table (in seconds) with respect to experiment involving user autonomy (Fig. 3.19, image dimension: 256×256). The timing +4 means the time taken for the inclusion of seeds.

3.2.3 Discussion and Limitations

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 3.2.2 clearly show the effectiveness of the proposed spectral cut segmentation approach, surpassing, in terms of accuracy, competing 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.

Seeded Image Segmentation

In this chapter we describe the *Laplacian Coordinates* proposed in [Casaca et al., 2014b] which tackles the problem of image segmentation from user-provided seeds. We also provide an overview of the main related research efforts focusing on techniques that rely on energy functionals defined on graphs. The proposed Laplacian Coordinates methodology is compared against the state-of-the-art approaches using a variety of quantitative and qualitative measures typically employed by the image segmentation community.

4.1 Seed-based Image Segmentation Methods

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 variation or gradients and they are used to properly 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 [Angelova and Zhu, 2013, Boykov and Funka-Lea, 2006, Ciesielski et al., 2012, Cousty et al., 2009b, Grady, 2006, 2012, Miranda and Falcão, 2009, Rother et al., 2004, Vicente et al., 2008, Wang and Tu, 2012], most of them making use of energy functional minimization on graphs. In terms of energy minimization formulation, Couprie et al. [2011], Cousty et al. [2009a] showed that most seed-based image segmentation methods can be understood as variations of a small group of basic techniques such as *Graph Cuts* [Boykov and Funka-Lea, 2006, Boykov and Jolly, 2001], *Random Walker* [Andrews et al., 2009, Grady, 2006, Grady and Sinop, 2008], *Optimal Shortest Paths/Geodesics* [Bai and Sapiro, 2009, Criminisi et al., 2008, Falcão et al., 2004] and *Watersheds* [Coustry et al., 2010, Cousty et al., 2009b, Monteiro and Campilho, 2008], 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 [Couprie et al., 2011]:

$$E_{PWS}(\mathbf{x}) = \sum_{i \in B} w_{Bi}^p |x_i - x_B|^q + \sum_{i \in F} w_{Fi}^p |x_i - x_F|^q + \frac{1}{2} \sum_{i \in V} \left(\sum_{j \in N(i)} w_{ij}^p |x_i - x_j|^q \right), \quad (4.1)$$

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 (4.1) assumes the form of specific image segmentation algorithms when p and q vary as shown in Table 4.1:

Table 4.1: Variation of the parameters p and q in Equation (4.1) versus classical segmentation algorithms [Couprie et al., 2011].

$q \backslash p$	0	finite	∞
1	Collapse to the seeds	<i>Graph Cuts</i>	<i>Power Watershed</i> ($q = 1$)
2	l_2 norm Voronoi	<i>Random Walker</i>	<i>Power Watershed</i> ($q = 2$)
∞	l_1 norm Voronoi	l_1 norm Voronoi	<i>Shortest Path Forest</i>

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

1. *Graph Cuts-based algorithms* [Boykov and Funka-Lea, 2006, Boykov and Jolly, 2001, Hower et al., 2009, Rother et al., 2004, Vicente et al., 2008];
2. *Random Walker-based algorithms* [Andrews et al., 2009, Grady, 2006, Grady and Sinop, 2008, Sinop and Grady, 2007];
3. *Watersheds-based algorithms* [Couprie et al., 2011, Coustry et al., 2010, Coustry et al., 2009a,b, Monteiro and Campilho, 2008];
4. *Shortest Path Forest/Geodesic-based methods* [Anh et al., 2012, Bai and Sapiro, 2009, Bergo et al., 2007, Criminisi et al., 2008, Falcão et al., 2004, Rocha et al., 2009].

Figure 4.1 shows segmentations obtained by methods from each class described above.

The literature of interactive image segmentation is huge and a general overview about graph-based segmentation has been already provided in Chapter 3. In order to contextualize this topic, we mainly focus our discussion on the four groups listed above. A comprehensive survey that addresses aspects not covered here can be found in the works by [Peng et al., 2013, Yi and Moon, 2012].

Figure 4.1: Segmentations taking as input the seeded image (a). (b) Depicts the ground truth, (c) the result obtained by the Graph-Cut technique [Boykov and Funka-Lea, 2006], (d) Power Watershed [Couprie et al., 2011], (e) Geodesic-based [Anh et al., 2012] and (f) Random Walker [Grady, 2006].

4.1.1 Graph Cut-based Approach

The well-known *GraphCut framework* (GC) has been introduced by Boykov and Funka-Lea [2006], Boykov and Jolly [2001] to address the problem of interactively clustering N-dimensional data from digital images.

The basic concept of the GC framework outlined in [Boykov and Funka-Lea, 2006] relies on the minimization of a cost function E_{GC} computed on a set of discrete variables taken from an affinity graph built from the input image. E_{GC} is defined so as to encode image features such as boundary and region information as well as the clues provided by

Figure 4.2: Application of the binary graph cut model for a very simple connected graph. (a) Shows the input graph where the seeded links (in red) are managed by the term D_i in Eq. (4.2) and the quantity $V_{i,j}$ manages the connections (in black) between graph nodes. (b) Shows the partitioning into two disjunct sets.

the user. For instance, if one attempts to partition an image into foreground (the object) and background, the following energy cost function must be minimized:

$$E_{GC}(\mathbf{x}) = \sum_{i \in F \cup B} D_i(x_i) + \lambda \sum_{(i,j) \in E} V_{i,j}(x_i, x_j), \quad (4.2)$$

where x_i is the unknown scalar value of the image pixel, D_i represents the data term (region-based term), $V_{i,j}$ is the regularization term (boundary/smoothness term), E is the set of neighborhoods of the affinity graph and $B \cup F$ represents the union of the background and foreground image labels. Quantity D_i measures the cost of assigning the label x_i to the pixel i while $V_{i,j}$ imposes spatial smoothness to the segmentation result (see Figure 4.2 for an illustration). $V_{i,j}$ attempts to ensure that pixels nearby the border of the object assume different labels so as to avoid overpenalize the labelings. Moreover, $V_{i,j}$ must be a nonconvex function of $|x_i - x_j|$, which demands the use of computationally costly optimization tools in order to produce an appropriated clustering. This issue is also faced by *Markov Random Fields* (MRF) formulation on general graph, specially those that deal with data partitioning [Kolmogorov and Zabih, 2004, Li, 1995].

For practical purposes, Equation (4.2) is often reformulated to incorporate the absolute difference $|x_i - x_j|$ as well as the weights w_{ij} of pairwise pixels $(i, j) \in E$ such that the sum of the following absolute difference is minimized:

Figure 4.3: Segmentation from a user-provided box proposed in [Rother et al., 2004]. (a) Presents in red the selection performed by the user and (b) shows the segmentation result.

$$E_{GC}(\mathbf{x}) = \sum_{i \in F} |x_i - x_F| + \sum_{i \in B} |x_i - x_B| + \sum_{(i,j) \in E} w_{ij} |x_i - x_j|. \quad (4.3)$$

The minimal cut on the graph is then obtained by solving the Background/Foreground Energy (4.3), which is computed by the classical combinatorial *Min-cut/Max-flow algorithm* [Boykov and Kolmogorov, 2004, Juan and Boykov, 2006]. If the energy involves multiple labels, minimization of Equation (4.2) is NP-Hard [DeLong, 2011], thus an approximation of the global optimal is required. Traditionally, the community of computer vision has made use of the augmenting-path scheme proposed by [Boykov and Kolmogorov, 2004] to reach near optimal solutions.

Many extensions of GC-based approach have been proposed in the literature, which differ mainly in terms of user interfaces such as [Lempitsky et al., 2009, Rother et al., 2004]. The methodology by Rother et al. [2004] enables an intuitive mechanism to provide prior clues to the segmentation system. The user surround the image object with a rectangle, and the algorithm is triggered only on the given box (see Figure 4.3 for an example). Significant improvements of the primary approach [Rother et al., 2004] have also been proposed [Chen et al., 2008, Lempitsky et al., 2009].

4.1.2 Random Walker-based Approach

The classical *Random Walk methodology* (RW) [Grady, 2006] is a powerful and easy-to-implement approach that relies on the standard graph Laplacian formulation. Given the affinity graph $G = (V, E, W_E)$ built from a target image, the Random Walker approach optimizes the following energy functional on G :

$$E_{RW}(\mathbf{x}) = \sum_{(i,j) \in E} w_{ij} (x_i - x_j)^2, \quad (4.4)$$

constrained to the input seeded vertices, that is, $x_i = x_B, i \in B$ and $x_i = x_F, i \in F$, where B and F represent the set of two distinct seed types. For instance, assume that $x_B = 1$, $x_F = 0$ and $\bar{\mathbf{B}}$ and $\bar{\mathbf{F}}$ are diagonal matrices with entries filled with 1's for seeded indices. One can impose soft constraints in Equation (4.4) so that the Random Walker Energy (4.4) is modeled as follows:

$$E_{RW}(\mathbf{x}) = \mathbf{x}^T \mathbf{L} \mathbf{x} + (1 - \mathbf{x})^T \bar{\mathbf{B}} (1 - \mathbf{x}) + \mathbf{x}^T \bar{\mathbf{F}} \mathbf{x}. \quad (4.5)$$

where $\mathbf{L} = (\mathbf{D} - \mathbf{W})$ is the graph Laplacian matrix built from the matrix of edge weights $\mathbf{W}$ and from the diagonal weighted valency matrix $\mathbf{D}$.

In [Grady, 2006], the author presents an interesting interpretation of the Laplacian-based formulation (4.5) by associating to each unseeded pixel the probability of a random walker starting on it to reach a background seed (assuming background seeds are labeled 1). The segmentation is then performed by assigning a background label to a pixel if the probability is greater than 0.5, otherwise the pixel is assigned to a foreground label (see Figure 4.4).

In terms of usage, instead of solving Equation (4.4), the RW algorithm solves the equation

Figure 4.4: Interpretation of the Random Walker approach in terms of probability of a random walker P reaching a seeded point (F or B).

$$D^{-1}Wx = x, \quad (4.6)$$

subject to input seeds provided by the user. The constrained Equation (4.6) ensures uniqueness of solution. Moreover, the solution is obtained by solving a linear system.

Other strategies that attempt to quickly compute the solution of the Random Walker Energy have been proposed in recent years. For instance, [Andrews et al. \[2009\]](#), [Grady and Sinop \[2008\]](#) optimize the runtime of the method by solving a spectral decomposition problem. Despite the good properties and satisfactory results, the Random Walker algorithm does not have an anisotropic behavior, meaning that the technique is prone to produce “flat” solutions in many situations, as shown in [\[Zhang et al., 2010\]](#).

4.1.3 Watersheds/Maximum Spanning Forest

The idea behind *Watershed/Maximum Spanning Forest algorithms* (MSF) [\[Cousty et al., 2007, 2009b\]](#) is to represent image objects as “catchment basins”, performing the segmentation by identifying the “basins” and their watershed lines (points equally likely to assume more than one minimum).

From the seeded image, the gradient of the image is usually taken as a relief map and, instead of minima, seeds are provided by the user or estimated automatically to perform the segmentation into desired regions. Trees are computed from the connected seed components

by spanning the nodes of the weighted graph so that the set of trees (a forest) is maximum w.r.t. weight intensities. As a result, the resulting segmentation is called a Watershed if the seeds correspond to the maxima. In practice, optimal spanning forests can be efficiently computed by the classical *Kruskal's* [Kruskal, 1956] or *Prim's* [Prim, 1957] algorithms.

In [Couprie et al., 2011], a robust seeded watershed algorithm namely *Power Watershed* (PWS) is proposed, where the outcome relies on the computation of an MSF algorithm. Although watershed is a very popular approach in the computer vision literature, they are not quite efficient in fitting objects where the gradient is locally irregular (see the first and second images in Figure 4.18).

4.1.4 Shortest Path Forest

Many seeded image segmentation methods that rely on finding the shortest path on a weighted graphs have appeared in the literature in recent years.

The rationale behind shortest path algorithms is to minimize the sum of edge weights of the path connecting two vertices. In fact, the problem of searching the best segmentation is modeled as the minimum cost path between two vertices, allowing to find the optimal solution from the given seeds. For instance, the *Shortest Path/Geodesic-based algorithm* (SP) [Bai and Sapiro, 2009] finds pixel labels by computing the shortest weighted path using geodesic distance from a target pixel to the foreground or background seeds. This method is attractive in terms of computational performance but it strongly depends on the position that seeds are sown.

An interesting approach (namely *Livewire*) that relies on computation of graph paths has been proposed in [Falcao et al., 1996, Falcão et al., 1998], that enables the user to interact with image boundaries to reach a feasible solution. Multiples points are sequentially placed on the object boundary so that the shortest path connecting them fits the object as the cursor runs. Small gaps can be nicely filled while the Livewire segment fits the whole boundary. However, the technique requires that the seeds be accurately placed on image boundaries.

Figure 4.5: (a)-(b) Segmentations obtained by the Livewire-based approach (extracted from the paper [Hamarneh et al., 2005]).

Another drawback is that the Livewire performs a search over the whole graph to find shortest paths, which is time consuming when dealing with large images.

In [Falcão et al., 2000], the computational burden of the original Livewire is alleviated by exploiting intrinsic properties of the graph in order to avoid unnecessary searches while computing optimal paths. Generalizations of this methodology has particularly appeared in the context of 3D medical imaging [Falcão and Udupa, 2000, Hamarneh et al., 2005]. Figure 4.5 depicts an example of a medical application of the Livewire technique.

The method presented in [Anh et al., 2012] performs well with respect to insertion of seeds, that is, it requires just a small amount of user guidance to achieve the segmentation but the algorithm relies on a pre-segmentation in order to work properly. Another technique that produces good results while allowing for user intervention to tune edge weights is the *Image Foresting Transform* (IFT) [Bergo et al., 2007, Falcão et al., 2004]. This technique accomplishes the image partitioning by finding paths of minimum cost between seeded nodes.

4.2 Laplacian Coordinates for Image Segmentation

Cousty et al. [2009a] and Couprie et al. [2011] proved that the set of methods originated by CG, MSF, PWS, RW and SP 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.

Aiming at dealing with the issues raised above, in this section we present a new methodology for seeded image segmentation, first reported in [Casaca et al., 2014b], which relies on the minimization of a novel quadratic energy functional defined from 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 [Sorkine, 2006, Xu et al., 2009] 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.

The Laplacian Coordinates pipeline is very simple and comprises four main steps:

- *Definition of Seeds;*
- *Affinity Graph Building;*
- *Energy Functional Construction and Solution;*
- *Assignment of Labels.*

Figure 4.6: Segmentations produced by the Laplacian Coordinates approach, where red and green scribes indicate the seedings.

An important 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. In the Section 4.2.2, we qualitatively and quantitatively evaluate the effectiveness of the Laplacian Coordinates segmentation comparing it against several state-of-the-art methods using the public “Grabcut” dataset provided by Microsoft [Boykov and Funka-Lea, 2006]. Figure 4.6 illustrates 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* [Casaca et al., 2014b].
2. Laplacian Coordinates bears several important properties such as boundary fitting, anisotropy, average neighborhood preservation 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.

Next, we describe the proposed technique and present some of its properties for interactive image segmentation.

4.2.1 Minimizing the Laplacian Coordinates Energy Functional 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

$$d_i = \sum_{j \in N(i)} w_{ij} \quad (4.7)$$

measures the weighted valency associated to pixel P_i .

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 [Cai and Chung, 2006, Casaca et al., 2013b, Cour et al., 2005, Kim et al., 2013, Yoon and Kweon, 2005, Zhu et al., 2014]. 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:

$$w_{ij} = \exp\left(-\frac{\beta \|I_i - I_j\|_\infty^2}{\sigma}\right), \quad \sigma = \max_{(i,j) \in E} \|I_i - I_j\|_\infty, \quad (4.8)$$

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 (4.8) to avoid null weights, as suggested by Grady [2012]. Sophisticated weighting functions can also be employed to reach more refined segmentations such as the inner-product similarity metric (3.8).

Neighborhood Average Preserving Energy Functional

Given the affinity matrix $\mathbf{W}$ computed as in Equation (4.8), the set of background B and foreground F seeded pixels and their corresponding labels x_B and x_F , the following energy functional is minimized with respect to vector $\mathbf{x}$:

$$E(\mathbf{x}) = k_1 \underbrace{\sum_{i \in B} \|x_i - x_B\|_2^2}_{\text{fidelity term}} + k_2 \underbrace{\sum_{i \in F} \|x_i - x_F\|_2^2}_{\text{fidelity term}} + k_3 \underbrace{\sum_{i \in V} \left\| d_i x_i - \sum_{j \in N(i)} w_{ij} x_j \right\|_2^2}_{\text{LC term}} \quad (4.9)$$

where $\mathbf{x} = (x_1, x_2, \dots, x_n)^t$ is the sought solution, that is, the scalar values assigned to the pixels $(P_1, P_2, \dots, P_n)$ so as to minimize the functional $E(\mathbf{x})$, n is the number of pixels, w_{ij} is computed as in Eq. (4.8) and k_1 , k_2 and k_3 are positive constants.

Without loss of generality, assume that $x_B > x_F$. Once the Energy (4.9) is minimized, the segmentation is accomplished by assigning background and foreground labels $y_i \in \{x_B, x_F\}$, $i \in V$, as follows:

$$y_i = \begin{cases} x_B, & \text{if } x_i \geq \frac{x_B + x_F}{2} \\ x_F, & \text{otherwise} \end{cases} \quad (4.10)$$

Laplacian Coordinates energy functional (4.9) 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*

(*Laplacian Coordinates*) *energy term*. In matricial notation, LC energy term can be rewritten as follows:

$$\begin{aligned}
\sum_{i \in V} \left\| d_i x_i - \sum_{j \in N(i)} w_{ij} x_j \right\|_2^2 &= \sum_{i \in V} \left(\sum_{j \in N(i)} w_{ij} x_i - \sum_{j \in N(i)} w_{ij} x_j \right)^2 = \sum_{i \in V} \left(\sum_{j \in N(i)} w_{ij} (x_i - x_j) \right)^2 \\
&= \left(\sum_{j \in N(1)} w_{1j} (x_1 - x_j) \right)^2 + \left(\sum_{j \in N(2)} w_{2j} (x_2 - x_j) \right)^2 + \dots + \left(\sum_{j \in N(n)} w_{nj} (x_n - x_j) \right)^2 = \\
&= \underbrace{\left(\sum_{j \in N(1)} w_{1j} (x_1 - x_j), \sum_{j \in N(2)} w_{2j} (x_2 - x_j), \dots, \sum_{j \in N(n)} w_{nj} (x_n - x_j) \right)}_{\mathbf{v}^t} \cdot \mathbf{v} = (\mathbf{L}\mathbf{x})^t (\mathbf{L}\mathbf{x})
\end{aligned}$$

Thus,

$$\sum_{i \in V} \left\| d_i x_i - \sum_{j \in N(i)} w_{ij} x_j \right\|_2^2 = (\mathbf{L}\mathbf{x})^t (\mathbf{L}\mathbf{x}) = \|\mathbf{L}\mathbf{x}\|_2^2, \quad (4.11)$$

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 (4.7)) and $\mathbf{W}$ denotes the weighted adjacency matrix of the graph, that is,

$$W_{ij} = \begin{cases} w_{ij}, & \text{if } (i, j) \in E \\ 0, & \text{otherwise} \end{cases}. \quad (4.12)$$

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

$$\delta_i = x_i - \frac{1}{d_i} \sum_{j \in N(i)} w_{ij} x_j, \quad (4.13)$$

in other words,

$$(\mathbf{L}\mathbf{x})_i = d_i \delta_i. \quad (4.14)$$

Figure 4.7: Geometric interpretation of the differential operator δ_i at vertex $i = 5$. It measures the deviation between x_5 and its center of mass $\frac{1}{d_5} \sum_{j \in N(5)} w_{ij} x_j$.

In less mathematical terms, quantity δ_i measures how much each node deviates from the weighted average of its neighbors (see Figure 4.7 for an illustration).

Minimizing the Energy Functional

Without loss of generality, let the tuning parameters in the energy functional $E(\mathbf{x})$ (4.9) assume unitary values, that is, $k_i = 1, i = 1, 2, 3$. Energy (4.9) can be modeled in a more general matricial form as follows:

$$\begin{aligned}
 E(\mathbf{x}) &= \sum_{i \in B} \|x_i - x_B\|_2^2 + \sum_{i \in F} \|x_i - x_F\|_2^2 + \sum_{i \in V} \left\| d_i x_i - \sum_{j \in N(i)} w_{ij} x_j \right\|_2^2 \\
 &= \sum_{i \in B} (x_i - x_B)^2 + \sum_{i \in F} (x_i - x_F)^2 + \|\mathbf{L}\mathbf{x}\|_2^2 \\
 &= \sum_{i \in B} (x_i^2 - 2x_i x_B + x_B^2) + \sum_{i \in F} (x_i^2 - 2x_i x_F + x_F^2) + (\mathbf{L}\mathbf{x})^t (\mathbf{L}\mathbf{x}) \\
 &= \sum_{i \in S} x_i^2 - 2 \left(\sum_{i \in B} x_i x_B + \sum_{i \in F} x_i x_F \right) + \#B x_B^2 + \#F x_F^2 + \mathbf{x}^t (\mathbf{L}^t \mathbf{L}) \mathbf{x} \\
 &= \mathbf{x}^t \mathbf{L}^2 \mathbf{x} + \mathbf{x}^t \mathbf{I}_S \mathbf{x} - 2\mathbf{x}^t \mathbf{b} + (\#B x_B^2 + \#F x_F^2)
 \end{aligned} \tag{4.15}$$

That is,

$$E(\mathbf{x}) = \mathbf{x}^t(\mathbf{I}_S + \mathbf{L}^2)\mathbf{x} - 2\mathbf{x}^t\mathbf{b} + c, \quad (4.16)$$

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 (4.16) 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 [Jankovic, 2005]:

$$(\mathbf{I}_S + \mathbf{L}^2)\mathbf{x} = \mathbf{b}. \quad (4.17)$$

Therefore, minimizing $E(\mathbf{x})$ is equivalent to solving the linear system (4.17), which, in turn, holds quite attractive properties such as symmetry, positive definiteness and sparsity. Indeed, Equation (4.17) can be solved efficiently using the supernodal sparse Cholesky factorization algorithm such as the one implemented in *Cholmod* [Davis and Hager, 2009] or classical MATLAB solvers [MATLAB, 2014].

After solving Equation (4.17), the segmentation is then performed by trivially assigning a foreground or background label to each pixel of the image according to Expression (4.10).

Laplacian Coordinates: Properties

Besides being computationally efficient, easy-to-implement and ensuring unique solution, the proposed methodology has additional properties that render it quite 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 4.8

Figure 4.8: 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.

illustrates this by comparing Laplacian Coordinates against the Random Walker approach in one-dimensional case. First row of Figure 4.8 shows a 1D graph with 500 nodes ordered from left to right. Second row in Figure 4.8 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 4.8, 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 4.8 presents the result of applying Laplacian Coordinates and Random Walker when weights are set equal 1.

The better preservation of labels can also be observed in Figure 4.9. Second column of Fig. 4.9 shows that the Random Walker approach was not able to properly capture the object contained in the image while Laplacian Coordinates has accurately identified the objects, as depicted in the third column Fig. 4.9. The reason for the better performance of Laplacian Coordinates is that labels tend to be preserved in homogeneous regions while Random Walker “diffuses” labels quickly according to the distance from the seeds, as shown in the second experiment of Fig. 4.9.

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:

$$(\mathbf{Lx})_i = \frac{1}{d_i} \sum_{j \in N(i)} w_{ij} (\mathbf{Lx})_j \quad (4.18)$$

where $(\mathbf{Lx})_j$ is computed as in Equation (4.14). The solution x_i is then computed taking into account an extended neighborhood, mathematically expressed by (see Fig.4.10 for an

Figure 4.9: Random Walker and Laplacian Coordinates image boundary fitting capability. (First and Third rows) Seeded images and the segmentation results obtained from Random Walker and Laplacian Coordinates, respectively, and (Second and Fourth rows) Graphs of the solution associated to First and Third rows, respectively.

Figure 4.10: Geometric sketch showing the solution x_i ($i = 7$) in terms of its neighborhood pixels. In the example, the purple circle illustrates the points used for calculating the differential coordinate δ_i at pixel $i = 11$.

illustration) the equation:

$$x_i = \frac{1}{d_i} \sum_{j \in N(i)} w_{ij} \left(x_j + \frac{\delta_j}{d_i} \right) = \frac{1}{d_i} \sum_{j \in N(i)} w_{ij} x_j + \frac{1}{d_i^2} \sum_{j \in N(i)} w_{ij} \left(\sum_{p \in N(j)} w_{jp} (x_j - x_p) \right) \quad (4.19)$$

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

Seeding Flexibility and Adaptability

Figure 4.11 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. 4.11 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. 4.11.

Figure 4.11: 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.

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 (4.17):

$$(\mathbf{I}_S + \mathbf{L}^2)\mathbf{x}^{(j)} = \mathbf{b}^{(j)}, \quad (4.20)$$

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 $K_j \in K = \{K_1, K_2, \dots, K_N\}$, $1 \leq j \leq (N - 1)$. Let C be a positive constant. We set $b_i^{(j)} = C$, $i \in K_j$, $b_i^{(j)} = -C$, $i \in (K \setminus K_j)$, 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:

$$x_i^{(N)} = C - \max_{1 \leq j \leq (N-1)} \{x_i^{(j)}\}. \quad (4.21)$$

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

$$y^{(j)} = \bigcap_{\substack{p=1, \dots, N \\ p \neq j}} (x^{(j)} > x^{(p)}), \quad (4.22)$$

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

Figure 4.12 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.

Figure 4.12: Extension of the Laplacian Coordinates (4.9) 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 $x^{(j)}$ that give rise to the multiple segmentation.

4.2.2 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 [Rother et al., 2004]. 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. Figure 4.13 shows all the information available, for each image, in the “Grabcut” dataset [Rother et al., 2004]. The benchmark dataset is available on the Microsoft Cambridge website and also includes 20 images from the classical *Berkeley Image Segmentation Benchmark Database* [Arbeláez et al., 2011].

We make use of this benchmark dataset to compare the Laplacian Coordinates approach (LC) [Casaca et al., 2014b] against five traditional seed-based segmentation algorithms already described in Section 4.1:

Figure 4.13: The “Grabcut” dataset benchmark. (a) Depicts the original image, (b) the input tri-map seeds, (c) the ground truth and (d) the unsegmented region.

- Graph Cuts algorithm (GC)¹ [Boykov and Funka-Lea, 2006, Rother et al., 2004];
- Power Watershed algorithm (PWS)² [Couprie et al., 2011, Cousty et al., 2007];
- Maximum Spanning Forest with Kruskal's algorithm (MSFK) [Couprie et al., 2011, Cousty et al., 2009b];
- Maximum Spanning Forest with Prim's algorithm (MSFP) [Couprie et al., 2011, Cousty et al., 2009b];
- Random Walker algorithm (RW)³ [Grady, 2006].

Quantitative evaluations are performed comparing the output quality in terms of object/region detection as well as the accuracy in preserving ground-truth boundaries. Input seeds are set as $x_B = 1$ and $x_F = -1$ in Equation (4.10) while in Equation (4.8), $\sigma = 0.1$.

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 [Unnikrishnan et al., 2007]. 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. Originally, this metric was devised for clustering evaluation finalities being modified by [Unnikrishnan et al., 2007, Yang et al., 2008] for dealing with multiple segmentations. The higher the value the better.

¹available at <http://grabcut.weebly.com/code.html>

²available at <http://powerwatershed.sourceforge.net>

³available at www.cns.bu.edu/~lgrady/

- **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 [Martin et al., 2001]. 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 [Meilă, 2005]. 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 [Meilă, 2007, Mignotte, 2014]. Values close to 0 are better.

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 consideration the amount of boundary pixels from the segmentation result that correspond to the closest boundary pixels from the ground-truth [Unnikrishnan et al., 2005]. Lower values are desired since BDE quantifies an error value.
- **The Harmonic Average Score (F-Score):** summarizes the *Recall* and *Precision* baseline metrics [Arbeláez et al., 2011, Estrada and Jepson, 2009], which measure how much the segmentation matches the ground-truth boundaries. Recall can be viewed as the proportion of boundary pixels in the resulting segmentation for which it is possible to find a matching boundary pixel in the ground-truth image. Precision holds the opposite situation (see Section 3.2.2). The matching is established in terms of the boundary pixel proximity for different values of radius R , as proposed in [Estrada and Jepson, 2009].

Figure 4.14: Comparison of six seed-based segmentation methods regarding to RI and GCE quality metrics. In both cases, the proposed Laplacian Coordinates framework has presented better results then all other five evaluated techniques.

Figures 4.14 and 4.15 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.

Regarding *F-score*, the LC approach also presents very good performance, specially when the parameter R increases. As one can see from Figure 4.16, 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.

Figure 4.15: Comparison of six seed-based segmentation methods regarding to VoI and BDE quality metrics. In general, Laplacian Coordinates outperforms all other methods.

Figure 4.16: F -score quality metric. Laplacian Coordinates is considerably better than other methods when parameter R increases.

Figure 4.17: From left to right: Ground-truth, the tri-map images (seeds and the unknown region) provided by the Grabcut dataset, the segmentations resulting from GC, MSFK, MSKP and LC approach.

Figures 4.17 and 4.18 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.

Figure 4.18: From left to right: Ground-truth, the tri-map images (seeds and the unknown region) provided by the Grabcut dataset, and segmentations resulting from PWS, RW and LC approach.

4.2.3 Discussion and Limitations

Laplacian Coordinates [[Casaca et al., 2014b](#)] is a new seed-based image segmentation technique that holds many attractive properties such as simple algebra formulation, easy to code, and uniqueness of solution.

The good results when dealing with complex images (e.g. third and second rows in Figs. [4.17-4.18](#)), multiple segmentations (e.g. Fig. [4.12](#)) and seeding flexibility (e.g. Fig. [4.11](#)) attest the robustness of the proposed methodology. In fact, our approach promotes significant improvements in treating the problem of image segmentation while ensuring high adherence on the contours of image segments.

Despite the good properties and results, our method also has limitations. For example, the approach can produce segmentations not totally satisfactory when applied to segment images with very thin pieces (e.g. the strap of the backpack in Fig. [4.17](#) - 5th row).

Laplacian Coordinates as an Effective Tool for Computer Vision Applications

This chapter presents the use of the Laplacian Coordinates segmentation tool 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. Image colorization is a computer-assisted process by which grayscale images or black-and-white films are properly colored (see Figure 5.1 for an illustrative example).

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

Figure 5.1: Image inpainting and colorization using the proposed frameworks.

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.

5.1 User-Friendly Interactive Image Inpainting with 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 for artistic edition finalities. Methods devoted to perform image inpainting can be arranged in several groups, as suggested by the surveys [Casaca, 2010, Casaca et al., 2014a, Subramanyam et al., 2010]. In short words, existing approaches differ in terms of

pixel propagation, sensibility when synthesizing textures, and the filling order criterion [Subramanyam et al., 2010].

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 Amano [2006], Chang et al. [2005], Padalkar et al. [2013], Tamaki et al. [2006] 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 Sun et al. [2005], 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 Barnes et al. [2009], 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 that generates pleasant outcomes with a reduced number of user interventions. Moreover, the initial non-interactive version of our technique, firstly reported in [Casaca et al., 2014a], outperforms existing inpainting methodologies in several aspects, as we show hereafter.

5.1.1 Pipeline Overview

In order to combine the accuracy and the high-adherence on image contours of the Laplacian Coordinates approach (Sec. 4) 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 5.2, our pipeline comprises four basic steps:

- *User-steered Selection of the Inpainting Domain*: consists of making use of the interactive segmentation tool [Casaca et al., 2014b] to initially select the objects that will be taken as input data to the inpainting process.
- *Pixel Filling Order Assignment*: assigns a label to unfilled pixels according to their level of priority in the filling front.
- *Construction of the Pixel Source Area*: defines the region for which pixels will be sampled to fill the inpainting area. This region is dynamically defined so as to accelerate the progress of the image completion.
- *Patch-based Pixel Replication*: allows the algorithm to properly copy-and-paste desirable pixels into the target unfilled area.

Figure 5.2: Pipeline of our interactive inpainting framework.

5.1.2 User-steered Selection of the Inpainting Domain

In our approach, entire objects can be easily segmented so as to avoid the meticulous election of the boundary pixels employed by traditional approaches. The user selects a target region 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 segmentation towards reaching a higher quality result. Figure 5.3 illustrates the above-mentioned scheme.

5.1.3 Filling Order Assignment

Let $f : D \subset \mathbb{R}^2 \rightarrow \mathbb{R}^l$ be the target image (where $l=1$: grayscale image and $l=3$: RGB color image), $\Omega \subset D$ the region to be inpainted (inpainting domain), $\partial\Omega$ its boundary and D a rectangular domain. Our inpainting algorithm starts by computing a mapping image u (cartoon image) from f using the anisotropic diffusion equation proposed in [Barcelos et al., 2003]. Image u is a non-textured function which contains geometric structures and isophote information taken from f . Figure 5.4 shows the original image, its cartoon version

Figure 5.3: Illustration of our interactive image inpainting framework.

Figure 5.4: Representation of the gradient field in an illustrative image.

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 the original image.

Similar to [Barcelos et al., 2003, Casaca et al., 2011a], component u is obtained by numerically solving the following anisotropic diffusion equation:

$$\frac{\partial f^{(t)}}{\partial t} = g|\nabla f^{(t)}| \operatorname{div} \left(\frac{\nabla f^{(t)}}{|\nabla f^{(t)}|} \right) - (1 - g)(f^{(t)} - f), \quad (5.1)$$

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.

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

$$\mathcal{P}(p) = \mathcal{R}(p) \cdot \mathcal{C}(p), \quad p \in \partial\Omega, \quad (5.2)$$

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

$$\mathcal{R}(p) = |\nabla(\Delta u_p) \cdot \vec{d}_p|, \quad \vec{d}_p = \frac{\nabla^\perp u_p}{|\nabla^\perp u_p|}, \quad (5.3)$$

$$\mathcal{C}(p) = \left(\frac{\sum_{q \in H_m(p) \cap (D - \Omega)} C(q)^{\frac{1}{k}}}{|H_m(p)|} \right)^k, \quad (5.4)$$

where $|H_m(p)|$ denotes the size of a squared block $m \times m$ centered at pixel p , $C(q)$ is initialized as one and $k > 0$ is the bias parameter.

Relevance term $\mathcal{R}$ computes the isophote direction from the boundary of the inpainting region, $\partial\Omega$. In fact, Expression (5.3) is similar to the transport equation proposed in [Bertalmio et al., 2000], since it takes into consideration the directional derivative of a non-textured image:

$$\mathcal{R}(p) = \left| \frac{\partial(\Delta u_p)}{\partial \vec{d}_p} \right|. \quad (5.5)$$

This means that in our approach, it is reasonable to approximate the input image f by the cartoon component u obtained from PDE (5.1) in order to better detect the image isophotes.

Balancing the priority mechanism while filling the damaged regions is also of paramount importance to guarantee the success of the restoration [Cheng et al., 2005, Criminisi et al.,

Figure 5.5: Illustration of the priority filling order mechanism (Equation (5.2)).

2004]. In this spirit, the Biased Confidence term (5.4) allows us to tune the priority mechanism as follows: if $k \ll 1$ then $\mathcal{C}$ assigns more significance in balancing Equation (5.2). Moderate significance occurs when $k \approx 1$ while $k \gg 1$ yields small significance. For an illustration of the measures $\mathcal{R}$ and $\mathcal{C}$, see Figure 5.5.

Finally, from Equation (5.2), we are able to recursively assign a label for each pixel p in the filling front, as we show in the next section.

5.1.4 Dynamic Processing of the Sampling Region

Let $\Lambda\Omega$ be the source region of the image that will provide the pixels to fill the inpainting domain. Unlike many existing inpainting approaches that take $\Lambda\Omega$ as the complement of the initial retouching domain, that is, Ω^c [Cao et al., 2011, Cheng et al., 2005, Criminisi et al., 2004, Efros and Leung, 1999], a pixel-dependent dynamic sampling mechanism is proposed in this work in an attempt to locally improve the quality of the inpainting as well as to reduce the computational cost.

Figure 5.6: The use of the dynamic sampling scheme to deal with huge photography for a given processing time

Region $\Lambda\Omega$ is dynamically built for each pixel $p \in \partial\Omega$ previously selected from the Priority Criterium (5.2), so we denote this region by $\Lambda\Omega_p$. It is defined in terms of the valid region on $H_L(p)$ (a $L \times L$ block of pixels centered at p). More specifically, for each pixel $p \in \partial\Omega$:

$$\Lambda\Omega_p = H_L(p) \cap \Omega^c. \quad (5.6)$$

For an illustration, see Figure 5.7(a). This strategy allows the algorithm to scan a reduced area of sample pixels only, instead of performing an exhaustive search into the whole image area as in previous approaches. For sake of comparison, Figure 5.6 shows the robustness of the dynamic sampling mechanism against [Criminisi et al., 2004] algorithm in terms of time consuming when recursively inpainting a huge photography (dimension: 1000×1300 , 51k damaged pixels).

5.1.5 Block-based pixel replication

In the final step of our framework, we are able to allocate the most suitable block of pixels from $\Lambda\Omega_p$ to the neighborhood of p . Given a pixel $p \in \partial\Omega$, we define the region $H_L(p)$ from $\Lambda\Omega_p$ (see Fig.5.7(a)).

Our algorithm 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 $H_n(\hat{q})$ is

Figure 5.7: 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.

the one which minimizes the distance between $H_n(p)$ and $H_n(q)$ w.r.t. a given metric. A smaller patch $H_m(\hat{q})$ is then selected from $H_n(\hat{q})$ and its valid pixels $H(\hat{q})$ are placed in corresponding pixels in the neighborhood of p , namely $H(p)$ (see Fig. 5.7(b)-(c)).

Let $\mathbf{p} = (f_{p_1}, f_{p_2}, \dots, f_{p_k})$, $\mathbf{q} = (f_{q_1}, f_{q_2}, \dots, f_{q_k})$ be the column vectors in $\mathbb{R}^k$, $k < n^2$, 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:

$$d(\mathbf{p}, \mathbf{q}) = \frac{\|\mathbf{p} - \mathbf{q}\|_{\Delta U}}{\sqrt{\|\mathbf{p}\|_{\Delta U}^2 + \|\mathbf{q}\|_{\Delta U}^2}}, \quad (5.7)$$

where,

$$\|\mathbf{p} - \mathbf{q}\|_{\Delta U} = \sqrt{(\mathbf{p} - \mathbf{q})^T \Delta U (\mathbf{p} - \mathbf{q})}, \|\mathbf{p}\|_{\Delta U} = \sqrt{\mathbf{p}^T \Delta U \mathbf{p}}, \|\mathbf{q}\|_{\Delta U} = \sqrt{\mathbf{q}^T \Delta U \mathbf{q}}, \quad (5.8)$$

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 (5.7) assigns higher weights for pixels located on the edges of the Laplacian operator of u . The weights of the pixels in Equation (5.7) are defined from the Laplacian of u and then embedded into the distance computation of the blocks.

5.1.6 Experimental Results and Comparisons

In this section we present a comparative study against state-of-the-art inpainting methodologies 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 methods:

- Texture synthesis-based approach [[Efros and Leung, 1999](#)];
- Cartoon/texture decomposition-based inpainting approach [[Bertalmío et al., 2003](#)];
- Exemplar-based inpainting approach [[Criminisi et al., 2004](#)];
- Skeleton-guided inpainting approach [[Cai et al., 2008](#)];
- Optimization-based inpainting approach [[Kawai et al., 2009](#)];
- Sparsity-based inpainting approach [[Elad et al., 2005](#), [Fadili et al., 2009](#), [Guleryuz, 2006](#), [Li, 2011](#), [Xu and Sun, 2010](#)];

Comparison with Pure Texture Synthesis

Figure 5.8(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. 5.8(b)-(c) that our framework outperforms the texture synthesis-based approach [[Efros and Leung, 1999](#)].

Comparison with Simultaneous Cartoon-Texture Inpainting

The experiment depicted in Figure 5.8(d)-(f) presents comparative results with the cartoon-texture inpainting algorithm [[Bertalmío et al., 2003](#)]. Fig. 5.8(d) shows the input image and

Fig. 5.8(e) depicts the resulting inpainting by Bertalmío et al. [2003]. Notice that some linear artifacts (inside the water area) were generated during the progress of the restoration. Our framework (Fig. 5.8(f)), however, better recovers the natural aspect of the image.

Comparison with Exemplar-based Inpainting

Figure 5.8(g) shows a high-quality image. We are concerned to reintroduce the missing parts of the image so as to produce a pleasant result while recovering its intrinsic details. The image obtained by Criminisi et al. [2004] (Fig. 5.8(h)) exhibits some artifacts creating inconsistent color tonalities. Our technique (Fig. 5.8(i)), however, captures fine details such as the layers of the "human eye".

The experiment shown in Fig. 5.8(j)-(l) presents a photograph with some large gaps intentionally made on complex structures of the image. Although the Criminisi et al. [2004] method (Fig. 5.8(k)) has generated a satisfactory result, some parts of the image were excessively blurred. In contrast, our framework has produced a more pleasant result as one can see in Fig. 5.8(l).

Comparison with Skeleton-guided Inpainting

A practical application in microscopic image processing is shown in Figure 5.9(a). The goal is to remove the white fibers created during the image acquisition process. Figs. 5.9(b)-(c) show the results produced by Cai et al. [2008] and by our technique, respectively. Despite both techniques show satisfactory results, Cai et al. [2008] blurs some important edges of the image, which does not happen with our approach. In fact, our method recovered the edges in a coherent way.

Comparison with Optimization-based Inpainting

Object removal problem is investigated in this experiment. Figure 5.9(d) presents a classical example of occluded image, where we want to remove the person on the center of

Figure 5.8: Comparison with existing inpainting methods. [Efros and Leung, 1999] (*pure texture synthesis*), [Bertalmio et al., 2003] (*cartoon/texture inpainting*) and [Criminisi et al., 2004] (*exemplar-based inpainting*).

Figure 5.9: Comparison with existing inpainting methods. [Cao et al., 2011] (*skeleton-guided inpainting*) and [Kawai et al., 2009] (*optimization-based inpainting*).

the photography. Since there are other important "objects" that share the same boundary region in the inpainting domain (Fig. 5.9(e)), the goal is to fill all those objects respecting their natural edges. Fig. 5.9(f) shows the result obtained with Kawai et al. [2009] method and Fig. 5.9(g) presents the result of our technique. Notice that in the first case, the inpainted region is blurred. Moreover, the *branch of the tree* was not properly filled. In contrast, our approach generates a pleasant image where the *tree* was totally completed.

Another example of object removal is shown in Figures 5.9(h)-(k), where the goal is to accurately recover the gate parts hidden by the statue (Fig. 5.9(i)) while maintaining the regular aspect of the image. One can see that the inpainting method by Kawai et al. [2009] (Fig. 5.9(j)) 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. 5.9(k)), reconstructing the broken lines in a more satisfactory manner.

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 5.10 the input images and the inpainting results obtained by [Elad et al., 2005, Fadili et al., 2009, Guleryuz, 2006, Li, 2011, Xu and Sun, 2010]. All examples are tricky to handle due to the predominance of structures and textures around the region to be filled. Notice that the Guleryuz [2006] and Fadili et al. [2009] algorithms produce blurring effect on the images while the Elad et al. [2005] method produces some artifacts in the outputs. The results reached by Xu and Sun [2010] as well as the Li [2011] method 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 5.10 have been considered (see Table 5.1). Individual as well as the mean of PSNR values show that the proposed method quantitatively outperforms the others.

Figure 5.10: Comparison with sparse representation-based inpainting. (a) Input images (with missing block in black), (b) inpainted by [Guleryuz \[2006\]](#), (c)-(d) inpainted by [Elad et al. \[2005\]](#) and [Fadili et al. \[2009\]](#), (e) inpainted by [Xu and Sun \[2010\]](#), (f) inpainted by [Li \[2011\]](#), (g) inpainted by our method, respectively from top to down and (h) the ground-truth images.

Table 5.1: Quantitative evaluation using PSNR (in dB) for all comparative images from Figure 5.10.

Image	Guleryuz	Elad et al.	Fadili et al.	Xu and Sun	Li	Our method
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
Barbara's 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

5.2 Interactive Image Colorization using 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* [[Casaca et al., 2012b](#), [Hertzmann et al., 2001](#), [Irony et al., 2005](#), [Liu et al., 2008](#), [Reinhard et al., 2001](#), [Welsh et al., 2002](#)] and *Scribble-based* [[Casaca et al., 2012a,b](#), [Huang et al., 2005](#), [Levin et al., 2004](#), [Luan et al., 2007](#), [Qu et al., 2006](#), [Yatziv and Sapiro, 2006](#)]. 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 to the non-requirement for an additional reference image, scribble-based strategy has performed better than the example-based one in recent years [[Yao et al., 2011](#)]. 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 [Levin et al. \[2004\]](#) 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 [Huang et al. \[2005\]](#) employs adaptive edge detection

so as to prevent colors from going beyond region boundaries. Further improvements have been proposed by [Yatziv and Sapiro \[2006\]](#), who present a faster scribble-based color optimization technique that relies on chrominance blending to perform the colorization. [Qu et al. \[2006\]](#) and [Luan et al. \[2007\]](#) 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, [Casaca et al. \[2012b\]](#) 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 [[Casaca et al., 2014b](#)]. 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 grayscale image, the user can colorize complex textured areas quite easily, preserving region edges and preventing the addition of new scribbles.

Contributions In summary, the main contributions of this work are:

- 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.
- An efficient system that allows for recursively colorizing the image by reintroducing new seeds to reach more pleasant results.
- The new method is easy-to-implement and it requires just a small amount of user intervention to quickly reach a good result.

Figure 5.11: Pipeline of the proposed colorization framework.

5.2.1 Pipeline Overview

As illustrated in Figure 5.11, 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.

5.2.2 Prior Segmentation

Similarly to the methodology described in Section 5.1, 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 method (Section 4). After solving the constrained linear system, the partitions obtained are used to support the next step of the pipeline, *Color Assignment*, which is described below. Figure 5.12 illustrates the steps presented above.

5.2.3 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

Figure 5.12: Illustration of our interactive image colorization framework.

accomplished as follows: given a grayscale pixel p_g and a user-provided set of colored pixels P_c , the $(l\alpha\beta)$ system color coordinates of p_g [Torres Mendez et al., 2009] is obtained by taking the α and β components from pixel $p_c \in P_c$ that minimizes the distance function d_l defined by:

$$d_l(l_c, l_g) = 0.5 (|l_c - l_g| + |std_c - std_g|), \quad (5.9)$$

where l_c and l_g are the luminance values of p_c and p_g , and the quantities std_c and std_g represent the standard deviations of l_c and l_g , respectively. After processing the pixels in $(l\alpha\beta)$ color system, pixel conversion to RGB system is then performed in order to produce the final color result. In our experiments we use a pixel neighborhood of size 5×5 to compute standard deviations of the pixel luminance.

5.2.4 Interactive Manipulation of the Colorization Result

One of the main contributions of the proposed framework is to exploit the flexibility provided by the Laplacian Coordinates 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 Laplacian Coordinates linear system to partition the image and, thereby, improve the resulting colorization.

Figure 5.13 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. 5.13(b). User can then provide an additional color scribble to the region with badly colored pixels (see Fig. 5.13(c)), creating new constraints for the Laplacian Coordinates and, thus generating a better result as shown in Fig. 5.13(d).

Figure 5.13: The use of our colorization framework when allowing for user intervention.

5.2.5 Results and Comparisons

In order to confirm the effectiveness of our methodology, we provide a few comparisons against the recent Projcolor technique [Casaca et al., 2012b] and the well-established scribble methods [Levin et al., 2004, Yao et al., 2011, Yatziv and Sapiro, 2006].

Figure 5.14 shows a comparison between our methodology and the Projcolor algorithm. Notice from Fig. 5.14(c) that our approach does not make use of any data exploratory interface as the one used by Projcolor approach to reach a natural-looking result (Fig. 5.14(b)).

Figure 5.15 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 Levin et al. [2004] method, 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.

Finally, the experiment presented in Figure 5.16 establishes comparisons between the proposed technique and scribble-based methods [Levin et al., 2004, Yao et al., 2011, Yatziv and Sapiro, 2006]. Colorizations produced by Levin et al. [2004] and Yatziv and Sapiro [2006] algorithms smoothed the images considerably almost all cases while the outcomes obtained by Yao et al. [2011] and our technique have produced more refined results. By reintroducing just a small amount of seeds in the reference images in Figs. 5.16(g)-(g), 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 Yao et al. [2011] method.

Figure 5.14: Comparison between our framework and [Casaca et al. \[2012b\]](#) technique.

Figure 5.15: Comparison between our framework and [Levin et al. \[2004\]](#) technique.

Figure 5.16: Comparison between our framework against [Levin et al. \[2004\]](#), [Yao et al. \[2011\]](#), [Yatziv and Sapiro \[2006\]](#) methods.

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. Both topics rely on the notion of the Laplacian operator and its mathematical properties to produce high-quality segmentations. Also, we present two methods 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 as shown in Chapter 3. 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 clearly showed the effectiveness of the proposed

spectral cut segmentation method, 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 Chapter 4 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. All those properties render Laplacian Coordinates an interesting and compelling seed-based image segmentation technique.

The new inpainting methodology presented in Chapter 5 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. Real and synthetic 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 image colorization application. In fact, besides enabling a local modification of badly colored regions, the proposed methodology 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.

Bibliography

- T. Amano. Correlation based image defect detection. In *IEEE International Conference on Pattern Recognition (ICPR)*, pages 163–166. IEEE Computer Society, 2006. (Cited on page [85](#)).
- 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)*, pages 9–16, 2009. (Cited on pages [24](#), [40](#), [44](#), [52](#), [53](#), and [58](#)).
- A. Angelova and S. Zhu. Efficient object detection and segmentation for fine-grained recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 811–818, 2013. (Cited on page [52](#)).
- N. T. N. Anh, J. Cai, J. Zhang, and J. Zheng. Robust interactive image segmentation using convex active contours. *IEEE Transactions on Image Processing (TIP)*, 21(8):3734–3743, 2012. (Cited on pages [xvii](#), [53](#), [54](#), and [60](#)).
- P. Arbeláez, M. Maire, C. Fowlkes, and J. Malik. Contour detection and hierarchical image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 33(5):898–916, 2011. (Cited on pages [15](#), [43](#), [44](#), [74](#), and [76](#)).

- X. Bai and G. Sapiro. Geodesic matting: A framework for fast interactive image and video segmentation and matting. *International Journal of Computer Vision (IJCV)*, 82(2):113–132, 2009. (Cited on pages 52, 53, and 59).
- 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 (TIP)*, 12(7):751–763, 2003. (Cited on pages 87 and 88).
- C. Barnes, E. Shechtman, A. Finkelstein, and D. B. Goldman. Patchmatch: A randomized correspondence algorithm for structural image editing. *ACM Transactions on Graphics (TOG)*, 28(3):24:1–24:11, 2009. (Cited on page 85).
- F. P. Bergo, A. X. Falcão, P. A. Miranda, and L. M. Rocha. Automatic image segmentation by tree pruning. *J. Math. Imaging Vis.*, 29:141–162, 2007. (Cited on pages 53 and 60).
- M. Bertalmio, G. Sapiro, V. Caselles, and C. Ballester. Image inpainting. In *Annual Conference on Computer Graphics (SIGGRAPH)*, pages 217–226, 2000. (Cited on page 89).
- M. Bertalmío, L. A. Vese, G. Sapiro, and S. Osher. Simultaneous structure and texture image inpainting. *IEEE Transactions on Image Processing (TIP)*, 12:882–889, 2003. (Cited on pages xx, 2, 93, 94, and 95).
- T. Biyikoglu, J. Leydold, and P. F. Stadler. *Laplacian Eigenvectors of Graphs: Perron-Frobenius and Faber-Krahn type Theorems*. Springer, 2007. (Cited on page 15).
- B. Bollobás. *Modern Graph Theory*. Springer, 1998. (Cited on pages 2 and 24).
- Y. Boykov and G. Funka-Lea. Graph cuts and efficient n-d image segmentation. *International Journal of Computer Vision*, 70(7):109–131, 2006. (Cited on pages xvii, 2, 52, 53, 54, 62, and 75).
- 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 (ICCV)*, volume 1, pages 105–112, 2001. (Cited on pages 52, 53, and 54).

- Y. Boykov and V. Kolmogorov. An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 26(9):1124–1137, 2004. ISSN 0162-8828. doi: 10.1109/TPAMI.2004.60. (Cited on page 56).
- O. M. Bruno, L. G. Nonato, M. A. Pazoti, and J. B. Neto. Topological multi-contour decomposition for image analysis and image retrieval. *Pattern Recognition Letters*, 29(11): 1675–1683, 2008. (Cited on page 24).
- J.-F. Cai, R. Chan, and Z. Shen. A framelet-based image inpainting algorithm. *Applied and Computational Harmonic Analysis*, 24:131–149, 2008. (Cited on pages 93 and 94).
- W. Cai and A. C. S. Chung. Shape-based image segmentation using normalized cuts. In *Proceedings of International Conference on Image Processing (ICIP)*, pages 1101–1104. IEEE Computer Society, 2006. (Cited on pages 24, 28, and 63).
- F. Cao, Y. Gousseau, S. Masnou, and P. Pérez. Geometrically guided exemplar-based inpainting. *SIAM Journal of Imaging Science*, 4:1143–1179, 2011. (Cited on pages xx, 90, and 96).
- 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 (SIB-GRAPI)*, pages 317–322, 2010a. (Cited on pages 24 and 27).
- M. Carvalho, A. Ferreira, and A. Costa. Image segmentation using quadtree-based similarity graph and normalized cut. *Lectures Notes in Computer Science*, 6419:329–337, 2010b. (Cited on pages 24, 28, and 34).
- 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. (Cited on page 84).
- W. Casaca and M. Boaventura. A regularized nonlinear diffusion approach for texture image

- denoising. In *22th Conference on Computer Graphics and Image Processing (SIBGRAPI)*, pages 164–171. IEEE Computer Society, 2009. (Cited on page 34).
- W. Casaca and M. Boaventura. A decomposition and noise removal method combining diffusion equation and wave atoms for textured images. *Mathematical Problems in Engineering*, 2010:1–21, 2010. (Cited on page 37).
- W. Casaca, M. Boaventura, and M. P. de Almeida. Classifying texture and free-textured content from images using nonlinear anisotropic diffusion. In *5th Conference on Mathematical Analysis (ENAMA)*, pages 210–214, 2011a. (Cited on pages 5, 31, and 88).
- 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*, pages 266–273. IEEE Computer Society, 2011b. (Cited on pages 5, 11, 23, 27, and 29).
- W. Casaca, E. Gomez-Nieto, C. de O.L. Ferreira, G. Tavares, P. Pagliosa, F. Paulovich, L. G. Nonato, and A. Paiva. Image colorization based on multidimensional projection. In *IEEE International Conference on Computational Photography (ICCP Posters)*, pages 32–38. IEEE Computer Society, 2012a. (Cited on page 99).
- 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)*, pages 32–38. IEEE Computer Society, 2012b. (Cited on pages xxi, 2, 5, 99, 100, 104, and 105).
- W. Casaca, M. P. Almeida, and M. Boaventura. Denoising textured images via regularized anisotropic diffusion. In Z. N. Akshaya Mishra and Z. Shahid, editors, *An Introductory Guide to Image and Video Processing*, pages 48–71. Iconcept Press Ltd., Brisbane, Australia, 2013a. (Cited on page 6).
- 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*, 45(3):227–238, 2013b. (Cited on pages 2, 5, 23, 29, and 63).

- W. Casaca, M. P. Almeida, M. Boaventura, and L. G. Nonato. Combining anisotropic diffusion, transport equation and texture synthesis for inpainting textured images. *Pattern Recognition Letters (PRL)*, 36:36–45, 2014a. (Cited on pages 2, 5, 84, and 85).
- W. Casaca, L. G. Nonato, and G. Taubin. Laplacian coordinates for seeded image segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 384–391. IEEE Computer Society, 2014b. (Cited on pages 2, 5, 51, 61, 62, 74, 81, 86, and 100).
- W. Casaca, E. Gomez-Nieto, D. Motta, I. Hartmann, G. Taubin, and L. G. Nonato. Dealing with multiple requirements in geometric arrangements. *IEEE Transactions on Visualization and Computer Graphics (TVCG) (accepted for publication)*, 1(1):1–13, 2015. (Cited on page 6).
- R.-C. Chang, Y.-L. Sie, S.-M. Chou, and T. K. Shih. Photo defect detection for image inpainting. In *IEEE International Symposium on Multimedia (ISM)*, pages 403–407. IEEE Computer Society, 2005. (Cited on page 85).
- G. Chartrand, T. Cresson, R. Chav, A. Gotra, A. Tang, and J. DeGuise. Semi-automated liver ct segmentation using laplacian meshes. In *Biomedical Imaging (ISBI), 2014 IEEE 11th International Symposium on*, pages 641–644, 2014. (Cited on page 17).
- D. Chen, B. Chan, G. Mamic, C. Fookes, and S. Sridharan. Improved grabcut segmentation via gmm optimisation. In *Proceedings of the Digital Image Computing: Techniques and Applications*, DICTA 2008, pages 39–45. IEEE Computer Society, 2008. (Cited on page 56).
- W.-H. Cheng, C.-W. Hsieh, S.-K. Lin, C.-W. Wang, and J.-L. Wu. Robust algorithm for exemplar-based image inpainting. In *Proc. of the International Conference on Computer Graphics, Imaging and Visualization*, pages 64–69, 2005. (Cited on pages 89 and 90).
- F. Chung. *Spectral Graph Theory*. CBMS, Reg. Conf. S. in Mathematics, Ameciran Mathematical Society, 1997. (Cited on page 24).

- 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*, 44(3):375–398, 2012. (Cited on page 52).
- C. Couprie, L. Grady, L. Najman, and H. Talbot. Power watershed: A unifying graph-based optimization framework. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 33(7):1384–1399, 2011. doi: 10.1109/TPAMI.2010.200. (Cited on pages xvii, xxiii, 52, 53, 54, 59, 60, and 75).
- T. Cour, F. Bénézit, and J. Shi. Spectral segmentation with multiscale graph decomposition. In *IEEE Computer Vision and Pattern Recognition (CVPR)*, pages 1124–1131, 2005. (Cited on pages xvi, 15, 24, 25, 27, 28, 40, and 63).
- J. Coustry, G. Bertrand, L. Najman, and M. Couprie. Watershed cuts: Thinnings, shortest path forests, and topological watersheds. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 32(5):925–939, 2010. ISSN 0162-8828. (Cited on pages 52 and 53).
- J. Cousty, G. Bertrand, L. Najman, and M. Couprie. Watershed cuts. In *Proceedings of Seventh International Symposium on Mathematical Morphology*, pages 301–312, 2007. (Cited on pages 58 and 75).
- 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)*, pages 731–738. IEEE Computer Society, 2009a. (Cited on pages 52, 53, and 60).
- J. Cousty, G. Bertrand, L. Najman, and M. Couprie. Watershed cuts: Minimum spanning forests and the drop of water principle. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 31(8):1362–1374, 2009b. (Cited on pages 2, 52, 53, 58, and 75).
- A. Criminisi, P. Perez, and K. Toyama. Region filling and object removal by exemplar-based image inpainting. *IEEE Transactions on Image Processing (TIP)*, 13:1200–1212, 2004. (Cited on pages xx, 89, 90, 91, 93, 94, and 95).

- A. Criminisi, T. Sharp, and A. Blake. Geos: Geodesic image segmentation. In *Proceedings of European Conference on Computer Vision: Part I (ECCV)*, pages 99–112, 2008. (Cited on pages [52](#) and [53](#)).
- T. A. Davis and W. W. Hager. Dynamic supernodes in sparse cholesky update/downdate and triangular solves. *ACM Trans. Math. Softw.*, 35(4), 2009. (Cited on page [67](#)).
- A. Delong. *Advances in Graph-Cut Optimizaiton: Multi-Surface Models, Label Costs, and Hierarchical Costs*. 135 pgs, phd thesis, University of Western Ontario, London, Ontario, Canada, 2011. (Cited on page [56](#)).
- L. Demanet and L. Ying. Curvelets and wave atoms for mirror-extended images. In *Proc. of SPIE Wavelets XII*, volume 6701, page 67010J, 2007a. (Cited on pages [31](#) and [37](#)).
- L. Demanet and L. Ying. Wave atoms and sparsity of oscillatory patterns. *Applied and Computational Harmonic Analysis*, 23:368–387, 2007b. (Cited on pages [31](#), [37](#), and [40](#)).
- J. Diaz, J. Petit, and M. Serna. A survey of graph layout problems. *ACM Computing Survey*, 34:313–356, 2002. (Cited on pages [1](#), [11](#), [14](#), and [24](#)).
- A. A. Efros and T. K. Leung. Texture synthesis by non-parametric sampling. In *IEEE International Conference on Computer Vision (ICCV)*, pages 1033–1038, 1999. (Cited on pages [xx](#), [90](#), [93](#), and [95](#)).
- M. Elad, J.-L. Starck, P. Querre, and D. Donoho. Simultaneous cartoon and texture image inpainting using morphological component analysis(mca). *Applied Computational Harmonic Analysis*, 19:340–358, 2005. (Cited on pages [xxi](#), [93](#), [97](#), and [98](#)).
- F. J. Estrada and A. D. Jepson. Benchmarking image segmentation algorithms. *International Journal of Comput. Vision*, 85(2):167–181, 2009. (Cited on pages [xvii](#), [43](#), [45](#), and [76](#)).
- M. Fadili, J.-L. Starck, and F. Murtagh. Inpainting and zooming using sparse representations. *The Computer Journal*, 52(1):64–79, 2009. (Cited on pages [xxi](#), [93](#), [97](#), and [98](#)).
- A. Falcao, J. Udupa, S. Samaresekera, and B. Hirsch. User steered image boundary segmentation. In *Proceedings of SPIE Medical Imaging*, volume 2710, pages 278–288, 1996. (Cited on page [59](#)).

- A. Falcão, J. Stolfi, and R. Lotufo. The image foresting transform: Theory, algorithms, and applications. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 26(1):19–29, 2004. (Cited on pages 24, 52, 53, and 60).
- A. Falcão and J. Udupa. A 3d generalization of user-steered live wire segmentation. *Medical Image Analysis*, 4(4):389–402, 2000. ISSN 1361-8415. (Cited on page 60).
- A. Falcão, J. Udupa, S. Samarasekara, and S. Sharma. User-steered image segmentation paradigms: Live wire and live lane. *Graphical Models and Image Processing*, 60:233–260, 1998. (Cited on page 59).
- A. Falcão, J. Udupa, and F. Miyazawa. An user-steered image segmentation paradigm: Live-wire-on-the-fly. *IEEE Transactions on Medical Imaging*, 19(1):55–62, 2000. ISSN 0278-0062. (Cited on page 60).
- M. Fiedler. Algebraic connectivity of graphs. *Czechoslovak Mathematical Journal*, 23(98): 298–305, 1973. (Cited on page 14).
- C. Godsil and G. Royle. *Algebraic Graph Theory*, volume 207 of *Graduate Texts in Mathematics*. Springer-Verlag, 2001. (Cited on page 12).
- 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*, pages 115–122, Arequipa, Peru, 2013. IEEE Computer Society. (Cited on pages 6 and 11).
- 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 (TVCG)*, 20(3):457–470, 2014. (Cited on pages 6 and 11).
- L. Grady. Multilabel random walker image segmentation using prior models. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 763–770, 2005. (Cited on page 24).

- L. Grady. Random walks for image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 28(11):1768–1783, 2006. (Cited on pages xvii, 2, 20, 21, 24, 52, 53, 54, 57, and 75).
- L. Grady. Targeted image segmentation using graph methods. In O. Lézoray and L. Grady, editors, *Image Processing and Analysis with Graphs*, pages 111–135. CRC Press, 2012. (Cited on pages 52 and 64).
- L. Grady and A. K. Sinop. Fast approximate random walker segmentation using eigenvector precomputation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 1, pages 1–8, 2008. (Cited on pages 24, 28, 40, 44, 52, 53, and 58).
- O. G. Guleryuz. Nonlinear approximation based image recovery using adaptive sparse reconstructions and iterated denoising-part ii: adaptive algorithms. *IEEE Transactions on Image Processing (TIP)*, 15(3):555–571, 2006. (Cited on pages xxi, 93, 97, and 98).
- G. Hamarneh, J. Yang, C. McIntosh, and M. Langille. 3d live-wire-based semi-automatic segmentation of medical images. In *Proceedings of SPIE Medical Imaging: Image Processing*, pages 1597–1603, 2005. (Cited on pages xviii and 60).
- A. Hertzmann, C. E. Jacobs, N. Oliver, B. Curless, and D. Salesin. Image analogies. In *ACM Transactions on Graphics (TOG)*, pages 327–340, 2001. (Cited on page 99).
- D. Hoyer, V. Singh, and S. C. Johnson. Label set perturbation for mrf based neuroimaging segmentation. In *IEEE International Conference on Computer Vision (ICCV)*, volume 0, pages 849–856. IEEE Computer Society, 2009. (Cited on page 53).
- 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*, pages 351–354, 2005. (Cited on page 99).
- R. Irony, D. Cohen-Or, and D. Lischinski. Colorization by example. In *Proc. of the Eurographics Symposium on Rendering*, pages 201–210, 2005. (Cited on pages 2 and 99).

- C. Jameson, A. Jujuunashvili, and A. Knyazev. Modern eigenvalue solvers for spectral image segmentation, 2008. URL <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.122.3624>. (Cited on page 15).
- V. Jankovic. Quadratic functions in several variables. *The Teaching of Mathematics*, VIII (1):53–60, 2005. (Cited on page 67).
- B. Jiang, A. Yang, C. Wang, and Z. Hou. Image coding based on image segmentation and shape adaptive all phase biorthogonal transform. In *IEEE International Conference on Imaging Systems and Techniques (IST)*, pages 95–100, 2013. (Cited on page 24).
- 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*, 28(10):1027–1037, 2012. (Cited on page 6).
- O. Juan and Y. Boykov. Active graph cuts. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 1, pages 1023–1029, 2006. (Cited on page 56).
- 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*, PSIVT '09, pages 271–282, 2009. (Cited on pages xx, 93, 96, and 97).
- T. H. Kim, K. M. Lee, and S. U. Lee. Learning full pairwise affinities for spectral segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 35(7):1690–1703, 2013. (Cited on pages 26, 28, and 63).
- V. Kolmogorov and R. Zabih. What energy functions can be minimized via graph cuts. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 26:65–81, 2004. (Cited on page 55).
- 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)*, pages 1067–1078, 2009. (Cited on page 29).

- J. Kruskal. On the shortest spanning tree of a graph and the traveling salesman problem. *Proceedings of the American Mathematical Society*, 7:48–50, 1956. (Cited on page 59).
- V. S. Lempitsky, P. Kohli, C. Rother, and T. Sharp. Image segmentation with a bounding box prior. In *IEEE International Conference on Computer Vision (ICCV)*, pages 277–284. IEEE Computer Society, 2009. (Cited on page 56).
- A. Levin, D. Lischinski, and Y. Weiss. Colorization using optimization. *ACM Transactions on Graphics (TOG)*, 23(3):689–694, 2004. (Cited on pages xxi, 99, 104, 105, and 106).
- S. Z. Li. *Markov Random Field Modeling in Computer Vision*. Springer, 1995. (Cited on page 55).
- X. Li. Image recovery via hybrid sparse representations: a deterministic annealing approach. *IEEE Journal of Selected Topics in Signal Processing*, 5(5):953–962, 2011. (Cited on pages xxi, 93, 97, and 98).
- Z. Li, S. Wachsmuth, J. Fritsch, and G. Sagerer. Manipulative action recognition for human-robot interaction. In G. Obinata and A. Dutta, editors, *Vision Systems: Segmentation and Pattern Recognition*. I-Tech Education and Publishing, 2007. (Cited on page 24).
- L. Liu, C.-L. Tai, Z. Ji, and G. Wang. Non-iterative approach for global mesh optimization. *Computer-Aided Design*, 39(9):772–782, 2007. (Cited on page 17).
- X. Liu, L. Wan, Y. Qu, T.-T. Wong, S. Lin, C.-S. Leung, and P.-A. Heng. Intrinsic colorization. *ACM Transactions on Graphics (TOG)*, 27(5):152:1–152:9, 2008. (Cited on page 99).
- 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*, pages 309–320, 2007. (Cited on pages 2, 99, and 100).
- U. Luxburg. A tutorial on spectral clustering. *Statistics and Computing*, 17(4):395–416, 2007. ISSN 0960-3174. (Cited on page 27).

- X. Ma, W. Wan, and J. Yao. Texture image segmentation on improved watershed and multi-way spectral clustering. In *IEE International Conference on Audio, Language and Image Processing (ICALIP)*, pages 1693–1697, 2008. (Cited on pages [24](#) and [27](#)).
- M. Maire and S. X. Yu. Progressive multigrid eigensolvers for multiscale spectral segmentation. *IEEE International Conference on Computer Vision (ICCV)*, 0:2184–2191, 2013. (Cited on pages [24](#) and [27](#)).
- S. Maji, N. Vishnhoi, and J. Malik. Biased normalized cuts. In *IEEE Conference on Computer Vision and Patter Recognition (CVPR)*, pages 2057–2064, 2011. (Cited on pages [2](#), [24](#), and [27](#)).
- 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)*, volume 2, pages 416–423, 2001. (Cited on pages [29](#) and [76](#)).
- MATLAB. *version 8.3 (R2014a)*. The MathWorks Inc., Natick, Massachusetts, 2014. (Cited on page [67](#)).
- M. Meilă. Comparing clusterings: An information based distance. *Journal of Multivariate Analysis*, 98(5):873–895, 2007. ISSN 0047-259X. (Cited on page [76](#)).
- M. Meilă. Comparing clusterings: an axiomatic view. In *Proceedings of the 22nd international Conference on Machine Learning, ICML '05*, pages 577–584, New York, NY, USA, 2005. ACM. ISBN 1-59593-180-5. (Cited on page [76](#)).
- R. Merris. Laplacian matrices of graphs: a survey. *Linear Algebra and its Applications*, 197-198:143–176, 1994. (Cited on page [12](#)).
- Y. Meyer. *Oscillating Patterns in Image Processing and Nonlinear Evolution Equations*. AMS, 2001. (Cited on page [31](#)).
- M. Mignotte. A label field fusion model with a variation of information estimator for image segmentation. *Information Fusion*, 20(1):7–20, 2014. ISSN 1566-2535. (Cited on page [76](#)).

- S. Milyaev and O. Barinova. Learning graph laplacian for image segmentation. In M. Gavrilova, C. Tan, and A. Konushin, editors, *Transactions on Computational Science XIX*, volume 7870 of *Lecture Notes in Computer Science*, pages 92–106. Springer Berlin Heidelberg, 2013. (Cited on page 11).
- 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*, 35(2):128–142, 2009. (Cited on page 52).
- B. Mohar. Some applications of laplacian eigenvalues of graphs. *Graph Symetric: Algebraic Methods and Applications*, 497:225–275, 1997. (Cited on pages 11, 13, 17, and 24).
- F. Monteiro and A. Campilho. Watershed framework to region-based image segmentation. In *Proceedings of International Conference on Pattern Recognition (ICPR)*, pages 1–4, 2008. (Cited on pages 52 and 53).
- H. Nagamochi, K. Nishimura, and T. Ibaraki. Computing all small cuts in an undirected network. *SIAM Journal on Discrete Mathematics*, 10:469–481, 1997. (Cited on page 18).
- A. Noma, A. B. V. Graciano, R. M. Cesar Jr, L. A. Consularo, and I. Bloch. Interactive image segmentation by matching attributed relational graphs. *Pattern Recognition*, 45(3): 1159–1179, 2012. ISSN 0031-3203. (Cited on page 24).
- M. Ozden and E. Polat. A color image segmentation approach for content-based image retrieval. *Pattern Recognition*, 40(4):1318–1325, 2007. ISSN 0031-3203. (Cited on page 24).
- 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*, volume 7729 of *Lecture Notes in Computer Science*, pages 61–71. Springer, 2013. (Cited on page 85).
- B. Peng, L. Zhang, and D. Zhang. A survey of graph theoretical approaches to image segmentation. *Pattern Recognition (PR Journal)*, 46(3):1020–1038, 2013. (Cited on pages 2 and 53).

- P. Perona and J. Malik. Scale-space and edge detection using anisotropic diffusion. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 12(7):629–639, 1990. (Cited on page [34](#)).
- F. Petronetto, A. Paiva, E. S. Helou, D. E. Stewart, and L. G. Nonato. Mesh-free discrete laplace-beltrami operator. *Computer Graphics Forum (CGF)*, 32(6):214–226, 2013. (Cited on page [11](#)).
- W. Pratt. *Digital Image Processing*. Wiley, 2008. (Cited on page [24](#)).
- R. Prim. Shortest connection newtworks and some generalizations. *Bell System Technical Journal*, 36:1389–1401, 1957. (Cited on page [59](#)).
- Y. Qu, T.-T. Wong, and P.-A. Heng. Manga colorization. *ACM Transactions on Graphics (TOG)*, 25(3):1214–1220, 2006. (Cited on pages [99](#) and [100](#)).
- E. Reinhard, M. Ashikhmin, B. Gooch, and P. Shirley. Color transfer between images. *IEEE Computer Graphics and Applications (CGA)*, 21(5):34–41, 2001. (Cited on page [99](#)).
- 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 - Contemporary*, 19(2):50–68, 2009. (Cited on page [53](#)).
- C. Rother, V. Kolmogorov, and A. Blake. Grabcut: Interactive foreground extraction using iterated graph cuts. *ACM Transactions on Graphics (SIGGRAPH 2004)*, 23(3):309–314, 2004. (Cited on pages [xviii](#), [52](#), [53](#), [56](#), [74](#), and [75](#)).
- S. Sarkar and P. Soundararajan. Supervised learning of large perceptual organization: Graph spectral partitioning and learning automata. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 22:504–525, 2000. (Cited on page [15](#)).
- A. Sáez, C. Serrano, and B. Acha. Normalized cut optimization based on color perception findings: A comparative study. *Machine Vision and Applications*, pages 1–11, 2014. (Cited on pages [24](#), [27](#), and [40](#)).

- J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22:888–905, 2000. (Cited on pages [xvi](#), [2](#), [15](#), [17](#), [18](#), [19](#), [24](#), [25](#), [26](#), [28](#), [34](#), [35](#), [36](#), and [40](#)).
- Y. Shuai, A. Masahide, T. Akira, and K. Masayuki. High accuracy bicubic interpolation using image local features. *Trans. Fundam. Electron. Commun. Comput. Sci.*, E90-A(8): 1611–1615, 2007. (Cited on pages [xvi](#), [33](#), and [40](#)).
- 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 (ICCV)*, volume 1, pages 1–8, 2007. (Cited on pages [24](#) and [53](#)).
- O. Sorkine. Differential representations for mesh processing. *Computer Graphics Forum (Eurographics)*, 25(4):789–807, 2006. (Cited on page [61](#)).
- P. Soundararajan and S. Sarkar. Analysis of MinCut, average cut and normalized cut measures. In *Workshop on Perceptual Organization in Computer Vision*, 2001. (Cited on page [15](#)).
- D. A. Spielman. Spectral graph theory and its applications. In *Proc. of the 48th Annual IEEE Symposium on Foundations of Computer Science*, pages 29–38, 2007. (Cited on pages [2](#) and [24](#)).
- 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*, 9(11):15–18, 2010. (Cited on pages [84](#) and [85](#)).
- F. Sun and J.-P. He. A normalized cuts based image segmentation method. In *International Conference on Information and Computing Science (ICIC)*, pages 333–336. IEEE Computer Society, 2009. (Cited on pages [24](#), [28](#), and [34](#)).
- J. Sun, L. Yuan, J. Jia, and H.-Y. Shum. Image completion with structure propagation. *ACM Transactions on Graphics (TOG)*, 24(3):861–868, 2005. (Cited on page [85](#)).

- T. Tamaki, H. Suzuki, and M. Yamamoto. String-like occluding region extraction for background restoration. In *IEEE International Conference on Pattern Recognition (ICPR)*, pages 615–618. IEEE Computer Society, 2006. (Cited on page 85).
- W. Tao, H. Jin, and Y. Zhang. Color image segmentation based on mean shift and normalized cuts. *IEEE Systems, Man, and Cybernetics*, 37:1382–1389, 2007. (Cited on pages 24 and 27).
- G. Taubin. A signal processing approach to fair surface design. In *Proceedings of ACM SIGGRAPH*, pages 351–358, 1995. (Cited on page 20).
- C. J. Taylor. Towards fast and accurate segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE Computer Society, 2013. (Cited on pages 2, 15, 24, and 27).
- D. Tolliver and G. L. Miller. Graph partitioning by spectral rounding: Applications in image segmentation and clustering. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 1, pages 1053–1060, 2006. (Cited on pages 1 and 28).
- L. Torres Mendez, C. Ramirez-Bejarano, G. Ortiz-Alvarado, and C. de Alba-Padilla. A fast color synthesis algorithm using the l-alpha-beta color space and a non-parametric mrf model. In *Eighth Mexican International Conference on Artificial Intelligence (MICA 2009)*, pages 53–58, 2009. (Cited on page 102).
- Z. Tu, X. Chen, A. L. Yuille, and S. C. Zhu. Image parsing: unifying segmentation, detection, and recognition. *International Journal of Computer Vision (IJCV)*, 63(2):113–140, 2005. (Cited on page 24).
- R. Unnikrishnan, C. Pantofaru, and M. Hebert. A measure for objective evaluation of image segmentation algorithms. In *Workshops of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR Workshops)*, CVPR '05, pages 34–39, Washington, DC, USA, 2005. IEEE Computer Society. ISBN 0-7695-2372-2-3. (Cited on page 76).

- R. Unnikrishnan, C. Pantofaru, and M. Hebert. Toward objective evaluation of image segmentation algorithms. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 29(6):929–944, 2007. ISSN 0162-8828. (Cited on page 75).
- L. Vese and S. Osher. Modeling textures with total variation minimization and oscillating patterns in image processing. *Journal of Scientific Computing*, 19:553–572, 2003. (Cited on pages 29, 31, 32, and 33).
- L. Vese and S. Osher. Color texture modeling and color image decomposition in a variational-pde approach. In *Proceedings of the 8th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing*, pages 103–110. IEEE Computer Society, 2006. (Cited on pages 29 and 32).
- 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)*, volume 0, pages 1–8. IEEE Computer Society, 2008. (Cited on pages 2, 52, and 53).
- 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)*, pages 2312–2319, 2012. (Cited on page 52).
- M. Wardetzky, S. Mathur, F. Kalberer, and E. Grinspun. Discrete laplace operators: No free lunch. In *Proceedings of the 15th Eurographics Symposium on Geometry Processing (SGP'07)*, pages 33–37, 2007. (Cited on page 12).
- T. Welsh, M. Ashikhmin, and K. Mueller. Transferring color to grayscale images. *ACM Transactions on Graphics (TOG)*, 21(3):277–280, 2002. (Cited on page 99).
- X. Wu. Image coding by adaptive tree-structured segmentation. *IEEE Transactions on Information Theory*, 38(6):1755–1767, 1992. (Cited on page 24).
- K. Xu, H. Zhang, D. Cohen-Or, and Y. Xiong. Dynamic harmonic fields for surface processing. *Computer and Graphics*, 33(3):391–398, 2009. (Cited on page 61).

- Z. Xu and J. Sun. Image inpainting by patch propagation using patch sparsity. *IEEE Transactions on Image Processing (TIP)*, 19(5):1153–1165, 2010. (Cited on pages [xxi](#), [2](#), [93](#), [97](#), and [98](#)).
- A. Y. Yang, J. Wright, Y. Ma, and S. S. Sastry. Unsupervised segmentation of natural images via lossy data compression. *Journal of Computer Vision and Image Understanding*, 110(2):212–225, 2008. ISSN 1077-3142. (Cited on page [75](#)).
- C. Yao, X. Yang, L. Chan, and Y. Xu. Image colorization using bayesian nonlocal inference. *Journal of Electronic Imaging*, 20(2):023008–023008–6, 2011. (Cited on pages [xxi](#), [99](#), [104](#), and [106](#)).
- L. Yatziv and G. Sapiro. Fast image and video colorization using chrominance blending. *IEEE Transactions on Image Processing (TIP)*, 15(5):1120–1129, 2006. (Cited on pages [xxi](#), [99](#), [100](#), [104](#), and [106](#)).
- F. Yi and I. Moon. Image segmentation: A survey of graph-cut methods. In *International Conference on Systems and Informatics (ICSAI)*, volume 1, pages 1936–1941. IEEE Computer Society, 2012. (Cited on pages [2](#) and [53](#)).
- K.-J. Yoon and I.-S. Kweon. Locally adaptive support-weight approach for visual correspondence search. In *IEEE Computer Vision and Pattern Recognition (CVPR)*, pages 924–931, 2005. (Cited on page [63](#)).
- S. Yu. Segmentation using multiscale cues. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 1, pages I–247–I–254. IEEE Computer Society, 2004. (Cited on page [24](#)).
- H. Zhang. Discrete combinatorial laplacian operators for digital geometry processing. In *SIAM Conference on Geometric Design*, pages 575–592, 2004. (Cited on page [12](#)).
- H. Zhang, O. van Kaick, and R. Dyer. Spectral methods for mesh processing and analysis. *Computer Graphics Forum (Proc. of Eurographics)*, pages 1–22, 2007. (Cited on page [11](#)).

- J. Zhang, J. Zheng, and J. Cai. A diffusion approach to seeded image segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2125–2132. IEEE Computer Society, 2010. (Cited on page [58](#)).
- Y. Zhang, D. Zhao, J. Zhang, R. Xiong, and W. Gao. Interpolation-dependent image downsampling. *IEEE Transactions on Image Processing (TIP)*, 20(11):3291–3296, 2011. (Cited on page [33](#)).
- Z. Zhu, C. C. Loy, and S. Gong. Constructing robust affinity graphs for spectral clustering. In *IEEE Computer Vision and Pattern Recognition (CVPR)*, pages 550–558, 2014. (Cited on pages [26](#) and [63](#)).