Random Walks for Image Segmentation

Leo Grady, *Member*, *IEEE*

Abstract—A novel method is proposed for performing multilabel, interactive image segmentation. Given a small number of pixels with user-defined (or predefined) labels, one can analytically and quickly determine the probability that a random walker starting at each unlabeled pixel will first reach one of the prelabeled pixels. By assigning each pixel to the label for which the greatest probability is calculated, a high-quality image segmentation may be obtained. Theoretical properties of this algorithm are developed along with the corresponding connections to discrete potential theory and electrical circuits. This algorithm is formulated in discrete space (i.e., on a graph) using combinatorial analogues of standard operators and principles from continuous potential theory, allowing it to be applied in arbitrary dimension on arbitrary graphs.

Index Terms—Image segmentation, interactive segmentation, graph theory, random walks, combinatorial Dirichlet problem, harmonic functions, Laplace equation, graph cuts, boundary completion.

1 Introduction

Image segmentation has often been defined as the problem of localizing regions of an image relative to content (e.g., image homogeneity). However, recent image segmentation approaches have provided interactive methods that implicitly define the segmentation problem relative to a particular *task* of content localization. This approach to image segmentation requires user (or preprocessor) guidance of the segmentation algorithm to define the desired content to be extracted.

A practical interactive segmentation algorithm must provide four qualities:

1. fast computation,
2. fast editing,
3. an ability to produce an arbitrary segmentation with enough interaction, and
4. intuitive segmentations.

The random walker algorithm introduced here exhibits all of these desired qualities. We note that this algorithm was first presented in a shortened form as a conference paper [1]. The random walker algorithm requires the solution of a sparse, symmetric positive-definite system of linear equations which may be solved quickly through a variety of methods. The algorithm may perform fast editing by using the previous solution as the initialization of an iterative matrix solver. An arbitrary segmentation may also be achieved through enough user interaction.

In this paper, we present a novel approach to K-way image segmentation given user-defined **seeds** indicating regions of the image belonging to K objects. Each seed specifies a location with a user-defined label. The algorithm labels an unseeded pixel by resolving the question: Given a random walker starting at this location, what is the probability that it first reaches each of the K seed points? It will be shown that this calculation may be performed exactly without the simulation of a random walk. By performing this calculation, we assign a K-tuple vector to each pixel that specifies the probability that a random walker starting from each unseeded pixel will first reach each of the K seed points. A final segmentation may be derived from these K-tuples by selecting for each pixel the most probable seed destination for a random walker. By biasing the random walker to avoid crossing sharp intensity gradients, a quality segmentation is obtained that respects object boundaries (including weak boundaries). In a uniform image (e.g., all black) or, as will be proved in Section 4, an image of pure noise, a segmentation will be obtained that roughly corresponds to Voronoi cells for each set of seed points. We term this segmentation the neutral segmentation since the image is **neutral** (i.e., devoid of meaningful content).

In our approach, we treat an image (or volume) as a purely discrete object—a graph with a fixed number of vertices and edges. Each edge is assigned a real-valued weight corresponding to the likelihood that a random walker will cross that edge (e.g., a weight of zero means that the walker may not move along that edge). The advantage of formulating the problem on a graph is that purely combinatorial operators may be used that require no discretization and therefore incur no discretization errors or ambiguities. Formulation of the algorithm on a graph also allows the application of the algorithm to surface meshes or space-variant images [2], [3]. Regardless of the dimensions of the data, we will use the term *pixel* throughout this paper to refer to the basic picture element in the context of its intensity values. In contrast, the term *node* will be used in the context of a graph-theoretical discussion.

Although the present algorithm is motivated in terms of random walks, an adequate sampling from this distribution would be completely infeasible for segmentation problems of interest. Fortunately, it has been previously established [4], [5] that the probability a random walker first reaches a seed point exactly equals the solution to the Dirichlet

• *The author is with Siemens Corporate Research, Department of Imaging and Visualization, 755 College Road East, Princeton, NJ 08540. E-mail: Leo.Grady@siemens.com.*

Manuscript received 19 May 2005; revised 27 Feb. 2006; accepted 1 Mar. 2006; published online 14 Sept. 2006.
Recommended for acceptance by J. Buhmann.
For information on obtaining reprints of this article, please send e-mail to: tpami@computer.org, and reference IEEECS Log Number TPAMI-0259-0505.

problem [6] with boundary conditions at the locations of the seed points and the seed point in question fixed to unity while the others are set to zero. For a popular account of this connection, see [7]. The development of a fully discrete calculus [8] has allowed for the connection between random walks on graphs [9] and discrete potential theory [10] to be made completely explicit [5]. The solution to the *combinatorial* Dirichlet problem on an arbitrary graph is given exactly by the distribution of electric potentials on the nodes of an electrical circuit with resistors representing the inverse of the weights (i.e., the weights represent *conductance*) and the "boundary conditions" given by voltage sources fixing the electric potential at the "boundary nodes."

In light of the connection between random walks on graphs and discrete potential theory, one may calculate the probability, x_i^s, that a random walker starting at pixel v_i first reaches a seed with label s, by solving the circuit theory problem that corresponds to a combinatorial analog of the Dirichlet problem [5]. Ground (i.e., fix the potential to zero) all seed points belonging to labels other than s and establish a unit voltage source with ground that fixes the s-labeled seeds to have a unit potential. The electric potentials established at each unseeded node provide the probabilities that a walker originating from that node will first reach the seed with label s. These electric potentials may be calculated through the solution of a system of sparse linear equations, as described in Section 3.7. The full K-tuple may be calculated by finding the potentials established through switching "on" (providing a unit voltage source to) each labeled collection of nodes and "off" (grounding) the remaining labeled nodes. Therefore, $K-1$ systems of linear equations must be solved. By linearity (i.e., the principle of superposition in circuit theory), the potentials so calculated must sum to unity. This allows us to avoid solving for one of the systems by subtracting the sum of the calculated potentials from unity to find the last entry in the full K-tuple. A function that solves the Dirichlet problem for a given set of boundary conditions is known as **harmonic**. Fig. 1 illustrates the harmonic functions (and subsequent segmentation) obtained for a 4×4 graph with unit weights in the presence of three seeds with different labels.

Additional properties of our approach that will be established in Section 4.3 include:

1. Each segment is guaranteed to be connected to seed points with the same label, i.e., there are no isolated regions of a particular label that contain no seed points.
2. The K-tuple of probabilities at each pixel is equal to the weighted average of the K-tuples of neighboring pixels, with the weights given by walker biases.
3. The solution for the potentials is unique.
4. The expected segmentation for an image of pure noise, given by independent, equal-mean, random variables, is equal to that obtained in the neutral segmentation.

A rich tradition of work in image segmentation has focused on the establishment of appropriate image (object) models and the development of algorithms focused on finding the parameters for these models (e.g., [11]). For example, the FRAME model of [12] provides a method for both synthesis and analysis of image textures. A different line of research in computer vision has first established the desired behavior of an algorithm and then set out to identify a PDE or other physical process that exhibits the desired behavior. In such approaches, an image is typically viewed as a domain with material properties (metric) induced by the image content upon which the PDE or other physical process is simulated. Notable examples of research along this second line of work include anisotropic diffusion for image filtering [13] and normalized cuts for image segmentation [14]. In such approaches, the primary focus is typically on the characteristic behavior of the process and the manner in which the image content induces a metric is left as a task-specific question (e.g., this information may come from intensity gradients, color gradients, or texture gradients, as appropriate to the particular problem). The present random walker approach follows from this second tradition in computer vision in which desirable behavioral properties of an interactive segmentation algorithm are identified and a particular physical process is proposed that exhibits the required characteristics. In this case, the characteristics that we try to capture in an interactive segmentation algorithm are:

1. location of weak (or missing) boundaries,
2. noise robustness,
3. ability to identify multiple objects simultaneously,
4. fast computation (and editing), and
5. avoidance of small/trivial solutions (i.e., an avoidance of a "small cut" phenomenon).

This paper is organized as follows: Section 2 reviews the relationship of this work to previous approaches. Section 3 gives a simple weighting function, derives the set of linear equations that must be solved and provides implementation details. Section 4 establishes theoretical properties and Section 5 examines behavioral properties of the algorithm. Section 6 provides segmentation results and we conclude in Section 7 with a summary of the algorithm presented and a discussion of future work.

2 Prior Work

Image segmentation is a vast topic. Therefore, we limit our review to supervised and/or graph-based algorithms. Additional work on random walks and combinatorial harmonic functions will also be discussed.

2.1 Supervised Segmentation

Supervised segmentation algorithms typically operate under one of two paradigms for guidance: 1) Specification of pieces of the boundary of the desired object or a nearby complete boundary that evolves to the desired boundary. 2) Specification of a small set of pixels belonging to the desired object and (possibly) a set of pixels belonging to the background. We note also that any of the automatic segmentation algorithms might be considered supervised by subsequent user selection of the desired segment. However, if the desired object is not a complete segment, a secondary clustering/segmentation algorithm must be employed to split or merge the automatic segments.

The intelligent scissors algorithm [15] treats the image as a graph where each pixel is associated with a node and a connectivity structure is imposed. This technique requires the user to place points along the boundary of the desired object.

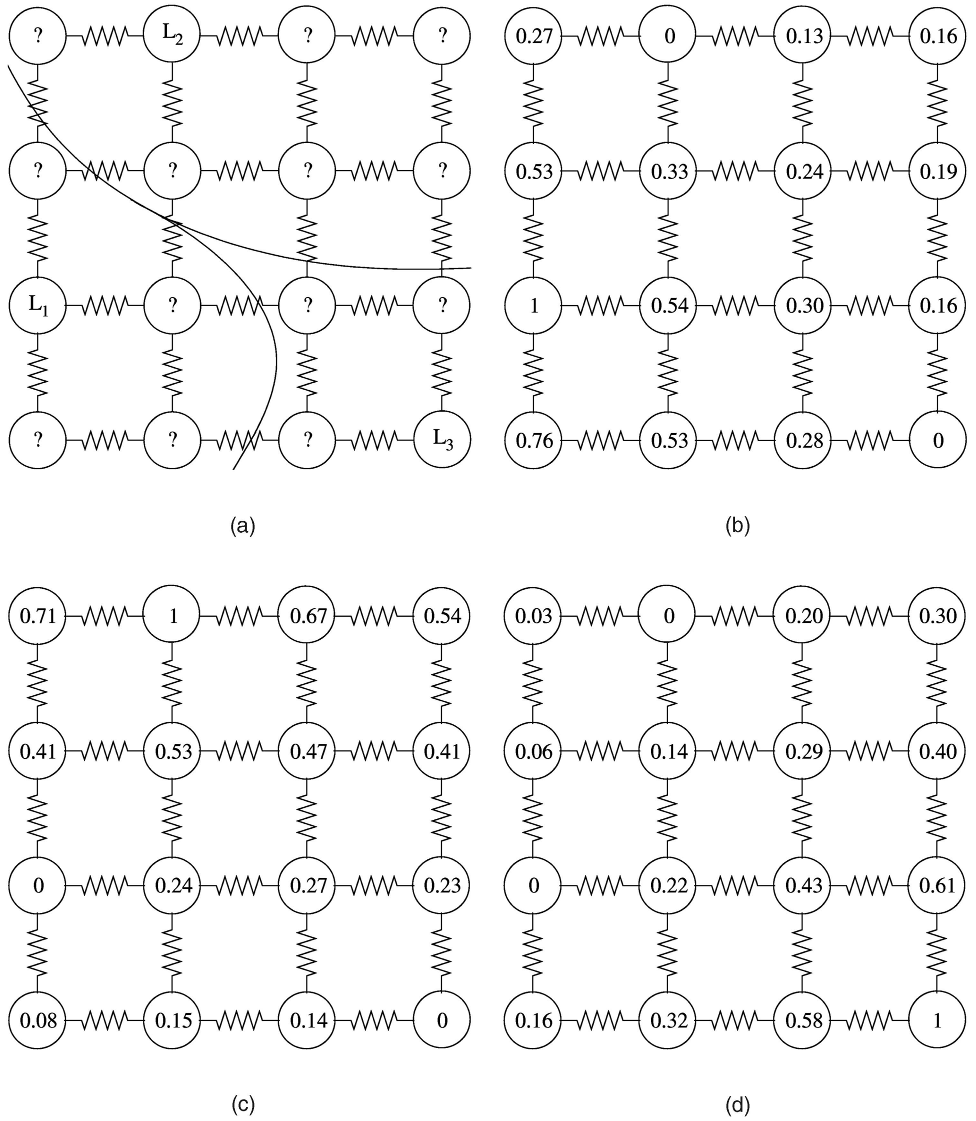


Fig. 1. Illustration of the approach to segmentation. With three seed points representing three different labels (denoted L_1, L_2, L_3), alternately fix the potential of each label to unity (i.e., with a voltage source tied to ground) and set to zero (i.e., ground) the remaining nodes. The electric potentials calculated represent the probability that a random walker starting at each node first reaches the seed point currently set to unity. Fig. 1a shows the initial seed points and the segmentation resulting from assigning each node the label that corresponds to its greatest probability. For illustration, all the weights (resistors) were set to unity. In the case of an image, these resistors would be a function of the intensity gradient. The reader can verify that the probabilities at each node sum to unity (up to rounding). (a) Seed points with segmentation. (b) Probability that a random walker starting from each node first reaches seed L_1. (c) Probability that a random walker starting from each node first reaches seed L_2. (d) Probability that a random walker starting from each node first reaches seed L_3.

Dijkstra's algorithm is then used to compute the shortest path between the user-defined points and this path is treated as the object boundary. The algorithm is simple to implement, very fast, and may be used to obtain an arbitrary boundary with enough points. Unfortunately, a low-contrast or noisy boundary may require the specification of many points and the algorithm is inapplicable to 3D boundaries.

Although the family of active contours and level sets is large [16], a user is generally asked to place a contour near the desired boundary and the algorithm evolves the boundary to a local energy minimum. Many different terms in the energy functional may be used to achieve different effects or employ domain knowledge for the problem. The main problems with level set methods are difficulty of implementation (often requiring specification of several free parameters) and difficulty in fixing an incorrect solution, especially if the desired contour does not correspond to a local energy minimum. Although the early paper by Kass et al. [17] incorporated user interaction, the active contours/level sets community appears to have trended away from this aspect. From a theoretical standpoint, these methods are defined in the continuum and achieve a local energy minimum, leading to difficulties in trying to theoretically predict or understand the properties of a practical solution.

The graph cuts [18], [19] technique has been developed as a method for interactive, seeded, segmentation. As with intelligent scissors, graph cuts views the image as a graph, weighted to reflect intensity changes. A user marks some nodes as foreground and others as background and the algorithm performs a max-flow/min-cut analysis to find the minimum-weight cut between the source and the sink. A feature of this algorithm is that an arbitrary segmentation may be obtained with enough user interaction and it generalizes easily to 3D and beyond. However, although performing well in many situations, there are a few concerns associated with this technique. For example, since the algorithm returns the smallest cut separating the seeds, the algorithm will often return the cut that minimally separates the seeds from the rest of the image, if a small number of seeds are used. Therefore, a user may need to continue placing seeds in order to overcome this "small cut" problem. Additionally, the K-way graph cuts problem is NP-Hard, requiring use of a heuristic to obtain a solution. Although one may find a solution within a bound of the optimal multiway cut [20], the problem becomes more difficult and one cannot be sure that the optimal cut is achieved. Finally, multiple "smallest cuts" may exist in the image that are quite different from each other. Therefore, a small amount of noise (adjusting even a single pixel) could cause the contour returned by the algorithm to change drastically. Mathematically, we note that the present algorithm may be considered as a *relaxation* of the binary values of the potential function in graph cuts. Although this may appear to constitute a minor modification of graph cuts, in fact, the motivation, theoretical properties, practical behavior, and method of solution are all quite different. The graph cuts approach of [18] differs from the present work by including a priors term on the intensity of the foreground and background (with a consequent additional parameter). Although we will not further discuss it here, such a modification to the random walker algorithm may also be achieved [21].

The graph cuts segmentation algorithm has been extended in two different directions in order to address issues of speed, color images, and the user interaction. The first type of extension to the graph cuts algorithm has focused on speed increases by coarsening the graph before applying the graph cuts algorithm. This coarsening has been accomplished in two manners: 1) by applying a standard multilevel approach and solving subsequent, smaller graph cuts problems in a fixed band to produce the final, full-resolution segmentation [22] and 2) by applying a watershed algorithm to the image and treating each watershed basin as a "supernode" in a coarse graph to which graph cuts in applied [23]. We note that the Lazy Snapping approach of [23] additionally proposes interactive tools for dividing watershed basins that may have incorrectly merged the foreground and background regions. The primary goal of these two approaches is to increase the computational speed of graph cuts by intelligently reducing the number of nodes in the graph. As stated in [22], the objective is to produce the same segmentation result as regular graph cuts by introducing a heuristic that greatly speeds the computation. Therefore, the benefits and difficulties of the graph cuts algorithm listed above also apply to these approaches, with an added uncertainty about the role of the coarsening operator in the final result (i.e., the final segmentation is no longer guaranteed to be the minimum cut). Additionally, both approaches to increasing the computational speed of graph cuts could equally be applied to the present algorithm with similar computational gains.

The second direction of extension to the graph cuts algorithm followed from the iterative estimation of a color model with the graph cuts algorithm [24]. This iterative color model was later coupled with an alteration of the user interface to create the GrabCuts algorithm [25]. The GrabCuts approach asks the user to draw a box around the object to be segmented and employs the color model as priors ("t-links") to obviate the need for explicit specification of foreground seeds. The added color model is of clear value in the application of color image segmentation and the "box-interface" requires less user interaction. Although the approach does perform well in the domain of color image segmentation, the iterative nature of the algorithm does increase the computational burden of the algorithm (requiring a solution to the max-flow/min-cut problem on each iteration) and there is no longer a guarantee of optimality (the algorithm is terminated when the iterations stagnate). For gray-scale images, the GrabCuts system essentially becomes standard graph cuts with a changed user interface. However, it appears that the "box-interface" is not always sufficient to capture the desired object, since further editing of the results with standard graph cuts is often required. As with the multilevel extensions described above, it would be possible to merge the novel aspects of the GrabCuts system (the iterative color image model and "box-interface") with the random walker algorithm described here. Since the graph cuts algorithm of [18] forms the heart of the GrabCuts system, and fulfills the same role as the present approach, we will focus on the relative strengths and weaknesses of these two algorithms.

2.2 Graph-Based Methods of Image Segmentation

Early papers of Zahn [26] and Wu and Leahy [27] are among the first approaches to apply graph theory to problems in image analysis. However, recent interest largely appears to have been spurred by Shi and Malik's introduction of the normalized cuts algorithm [14]. Most subsequent algorithms have focused on the spectral properties of the graph (e.g., [28], [29]), although the isoperimetric algorithm [30] and the Swendsen-Wang algorithm [31] are notable exceptions.

2.3 Random Walks and Combinatorial Harmonic Functions

Harmonic functions defined on graphs with given Dirichlet boundary conditions have seen recent interest in many applications, including image filtering [32], image colorization [33], and machine learning [34]. Although purely combinatorial harmonic functions were studied as early as 1945 by Eckmann [35], the earliest use of combinatorial harmonic functions that the author is aware of was an application to circuit layout given by Kodres [36]. Combinatorial harmonic functions were also famously employed by Tutte for graph drawing [37]. For an excellent collection of current knowledge on combinatorial harmonic functions, see [10].

Random walks first appeared in computer vision in the early work of Wechsler and Kidode for texture discrimination [38]. More recently, the average hitting time of a random walk from an object boundary has been studied as a measure to characterize object shape [39]. The isoperimetric graph partitioning algorithm introduced in [40] was shown to have an interpretation in terms of random walks in the sense that hitting times are computed from all nodes to a designated node and these values are thresholded to produce a partition that has various beneficial theoretical properties. This approach was recently applied to automatic image segmentation [30] by choosing the designated node randomly and recursively partitioning until a measure of partition quality is violated.

Recently, various steady-state properties of random walks have also been used to define automatic clustering algorithms. Harel and Koren [41] employ the notion of escape probabilities on subgraphs to iteratively weaken graph edges and eventually break the graph into disconnected components. Yen et al. [42] use the notions of average first-passage time and average commute time to replace traditional shortest-path distances between nodes in a graph and show that standard clustering algorithms (e.g., K-means) produce better results when applied to these reweighted graphs. Both of these methods represent automatic clustering algorithms (as opposed to the seeded method here) and require either extensive computations to produce pairwise random walk quantities for each pair of nodes, or employ a heuristic method of employing subgraphs to restrict the computation. The advantage of examining the probabilities that random walkers first arrive at predefined traps (given by the seed points) considered here is that the probabilities may be computed quickly and the various properties of noise robustness and harmonic functions (e.g., mean-value theorem, etc.) examined in Section 4.3 may be used to characterize the algorithm's behavior. Furthermore, these approaches require the specification of additional free parameters beyond what are necessary in the present approach.

Newman uses concepts from random walks to introduce a notion of "betweenness" on the nodes on a graph by considering a node's "betweenness" measure to be equal to how often a random walk starting at any pair of nodes passes through the node, averaged across all pairs [43]. Such a measure is shown to offer more intuitive behavior over other methods of "betweenness" computation at the cost of an expensive matrix inversion.

3 Exposition of the Algorithm

Although the random walker algorithm was motivated in Section 1 by placing random walkers at pixels and noting which seeds they first arrive at, such a method of computation would be completely impractical. Fortunately, established connections between random walks and potential theory (or circuit theory, on a graph) provide us with a simple, convenient method for analytically computing the desired probabilities. This section describes three aspects of the algorithm: generating the graph weights, establishing the system of equations to solve the problem, and the practical details of implementation.

We begin by defining a precise notion for a graph. A **graph** [44] consists of a pair $G = (V, E)$ with **vertices (nodes)** $v \in V$ and **edges** $e \in E \subseteq V \times V$. An edge, e, spanning two vertices, v_i and v_j, is denoted by e_{ij}. A **weighted graph** assigns a value to each edge called a weight. The weight of an edge, e_{ij}, is denoted by $w(e_{ij})$ or w_{ij}. The **degree** of a vertex is $d_i = \sum w(e_{ij})$ for all edges e_{ij} incident on v_i. In order to interpret w_{ij} as the bias affecting a random walker's choice, we require that $w_{ij} > 0$. The following will also assume that our graph is connected and undirected (i.e., $w_{ij} = w_{ji}$).

3.1 Edge Weights

In order to represent the image structure (given at the pixels) by random walker biases (i.e., edge weights), one must define a function that maps a change in image intensities to edge weights. This is a common feature of graph-based algorithms for image analysis and several weighting functions are commonly used in the literature [14], [20], [45]. Additionally, it was proposed in [46] to use a function that maximizes the entropy of the resulting weights. In this work, we have preferred (for empirical reasons) the typical Gaussian weighting function given by

$$w_{ij} = \exp\left(-\beta(g_i - g_j)^2\right), \tag{1}$$

where g_i indicates the image intensity at pixel i. The value of β represents the only free parameter in this algorithm. We have found it useful to normalize the square gradients $(g_i - g_j)^2 \; \forall e_{ij} \in E$ before application of (1). Of course, (1) could be modified to handle color or general vector-valued data by replacing $(g_i - g_j)^2$ with $||g_i - g_j||^2$ for a vector-valued g_i. Additionally, for problem-specific domains, (1) could be modified to apply to texture information, filter coefficients or other image features.

3.2 Combinatorial Dirichlet Problem

In the introduction, we noted that the combinatorial Dirichlet problem has the same solution as the desired random walker probabilities [4], [5], [10]. In this section, we review the combinatorial Dirichlet problem and show how to find its solution.

The **Dirichlet integral** may be defined as

$$D[u] = \frac{1}{2}\int_\Omega |\nabla u|^2 d\Omega, \tag{2}$$

for a field u and region Ω [6]. This integral arises in many physical situations, including heat transfer, electrostatics, and random walks.

A **harmonic function** is a function that satisfies the **Laplace equation**

$$\nabla^2 u = 0. \tag{3}$$

The problem of finding a harmonic function subject to its boundary values is called the **Dirichlet problem**. The harmonic function that satisfies the boundary conditions minimizes the Dirichlet integral since the Laplace equation is the Euler-Lagrange equation for the Dirichlet integral [6].

Define the combinatorial Laplacian matrix [47] as

$$L_{ij} = \begin{cases} d_i & \text{if } i = j, \\ -w_{ij} & \text{if } v_i \text{ and } v_j \text{ are adjacent nodes,} \\ 0 & \text{otherwise,} \end{cases} \tag{4}$$

where L_{ij} is indexed by vertices v_i and v_j.

Define the $m \times n$ edge-node **incidence matrix** as

$$A_{e_{ij}v_k} = \begin{cases} +1 & \text{if } i = k, \\ -1 & \text{if } j = k, \\ 0 & \text{otherwise,} \end{cases} \tag{5}$$

for every vertex v_k and edge e_{ij}, where each e_{ij} has been arbitrarily assigned an orientation. As with the Laplacian matrix above, $A_{e_{ij}v_k}$ is used to indicate that the incidence matrix is indexed by edge e_{ij} and node v_k. As an operator, A may be interpreted as a combinatorial gradient operator and A^T as a combinatorial divergence [48], [8] by virtue of the equivalent role of A and Grad as the coboundary operator on the space of 0-cochains or 0-forms, respectively (see [49] for more information).

We define the $m \times m$ **constitutive matrix**, C, as the diagonal matrix with the weights of each edge along the diagonal. As in the continuum setting, the isotropic combinatorial Laplacian is the composition of the combinatorial divergence operator with the combinatorial gradient operator, $L = A^T A$. The constitutive matrix may be interpreted as representing a metric in the sense that it defines a weighted inner product on the vector space of 1-cochains (i.e., functions defined on the edge set). In this sense, the combinatorial Laplacian generalizes to the combinatorial Laplace-Beltrami operator [50] via $L = A^T CA$. The case of a trivial metric, (i.e., equally weighted, unit valued, edges) reduces to $C = I$ and $L = A^T A$.

With these definitions in place, we can determine how to solve for the harmonic function that finds probabilities/potentials on unseeded nodes, while keeping the seed nodes fixed. A combinatorial formulation of the Dirichlet integral (2) is

$$D[x] = \frac{1}{2}(Ax)^T C(Ax) = \frac{1}{2}x^T Lx = \frac{1}{2}\sum_{e_{ij}\in E} w_{ij}(x_i - x_j)^2, \quad (6)$$

and a combinatorial harmonic is a function x that minimizes (6). Since L is positive semidefinite, the only critical points of $D[x]$ will be minima.

Partition the vertices into two sets, V_M (marked/seed nodes) and V_U (unseeded nodes) such that $V_M \cup V_U = V$ and $V_M \cap V_U = \emptyset$. Note that V_M contains all seed points, regardless of their label. We may assume without loss of generality that the nodes in L and x are ordered such that seed nodes are first and unseeded nodes are second. Therefore, we may decompose (6) into

$$\begin{aligned} D[x_U] &= \frac{1}{2}\left[x_M^T x_U^T\right]\begin{bmatrix} L_M & B \\ B^T & L_U \end{bmatrix}\begin{bmatrix} x_M \\ x_U \end{bmatrix} \\ &= \frac{1}{2}\left(x_M^T L_M x_M + 2x_U^T B^T x_M + x_U^T L_U x_U\right), \end{aligned} \quad (7)$$

where x_B and x_U correspond to the potentials of the seeded and unseeded nodes, respectively. Differentiating $D[x_U]$ with respect to x_U and finding the critical point yields

$$L_U x_U = -B^T x_M, \quad (8)$$

which is a system of linear equations with $|V_U|$ unknowns. If the graph is connected, or if every connected component contains a seed, then (8) will be nonsingular [51].

Denote the probability (potential) assumed at node, v_i, for each label, s, by x_i^s. Define the set of labels for the seed points as a function $Q(v_j) = s,\ \forall v_j \in V_M$, where $s \in \mathbb{Z}, 0 < s \leq K$. Define the $|V_M| \times 1$ vector (where $|\cdot|$ denotes cardinality) for each label, s, at node $v_j \in V_M$ as

$$m_j^s = \begin{cases} 1 & \text{if } Q(v_j) = s, \\ 0 & \text{if } Q(v_j) \neq s. \end{cases} \quad (9)$$

Therefore, for label s, the solution to the combinatorial Dirichlet problem may be found by solving

$$L_U x^s = -B^T m^s, \quad (10)$$

for one label or

$$L_U X = -B^T M, \quad (11)$$

for all labels, where X has K columns taken by each x^s and M has columns given by each m^s. Since the probabilities at any node will sum to unity, i.e.,

$$\sum_s x_i^s = 1, \forall v_i \in V, \quad (12)$$

only $K-1$ sparse linear systems must be solved, where K is the total number of labels.

3.3 Circuit Analogy

Although the algorithm was motivated in terms of random walks, it is well-known that there are many equivalences between random walks and electrical circuits [5]. Specifically, as illustrated in Fig. 1, the solution to (10) may be interpreted as a circuit simulation. Consider the three fundamental equations of circuit theory (Kirchhoff's current and voltage law and Ohm's law), which may be written in the above notation as

$$\begin{aligned} A^T z &= f && \text{(Kirchhoff's Current Law)}, && (13) \\ Cp &= z && \text{(Ohm's Law)}, && (14) \\ p &= Ax + b && \text{(Kirchhoff's Voltage Law)}, && (15) \end{aligned}$$

for a vector of branch currents, z, current sources, f, voltage sources, b, and potential drops (voltages), p. These three equations may be combined into the linear system

$$\begin{aligned} A^T CAx + A^T Cb &= f, && (16) \\ Lx &= f - A^T Cb, && (17) \end{aligned}$$

which is equivalent to (10), with $f = 0$ (no current sources) and the role of the voltage sources taken by the user-defined seeds. We note that (6) may also be interpreted as *power* in the circuit theory context and (17) represents the resulting minimization performed by the physical world.

3.4 Relationship to Diffusion

Since diffusion processes have such a significant history in computer vision and such a process may be described by a random walk (i.e., Brownian motion), it is useful to examine the relationship between a diffusion process and the present approach.

The fundamental difference between a diffusion equation and the Laplace equation of (28) is that diffusion represents a transient process occurring in time, while a Laplace equation describes a steady-state distribution. This straightforward relationship is illustrated by examining the equations together:

$$\begin{aligned} \frac{du}{dt} &= \nabla^2 u && \text{(Diffusion equation)}, && (18) \\ 0 &= \nabla^2 u && \text{(Laplace equation)}. && (19) \end{aligned}$$

In fact, a circuit analogy of the diffusion process also appears in Perona and Malik's classic paper [13]. The two circuit formulations differ in that the voltage sources (used to define the steady-state potentials) are replaced by capacitors charged to values representing an initial condition (used to define the transient potentials after a predefined amount of time has passed). In the case of two labels (i.e., a single source/ground pair) and infinite time the two formalisms can be made to give the same results (up to a shift and scale) if one seed is taken as an infinite source of random walkers (diffusive particles) and the other seed as an infinite sink of random walkers (diffusive particles).

Despite the mathematical similarities between the Laplace and diffusion equations, these algorithms are very different. Specifically, diffusion is typically employed as an image enhancement algorithm in which the original gray-scale values are taken as initial conditions and the solution is stopped after a predetermined amount of time. In contrast, we describe a seeded segmentation algorithm that makes no use of initial conditions and examines the steady-state distribution of potentials in order to define segmentation boundaries.

3.5 Image Model

In contrast to several popular image segmentation algorithms (e.g., [11]), the random walker segmentation approach presented here is not derived explicitly from an image model. However, an implicit image model exists in the approach and it is therefore useful to examine the algorithm from this standpoint.

Piecewise constant image models have existed from the earliest days of computer vision. In such a model, each object in the image is expected to be of constant value (e.g., intensity, color, and texture). Although simplistic, such models remain popular and surprisingly effective. However, three problems immediately present themselves:

1. The image may be corrupted with noise.
2. Neighboring (touching) objects may have the *same* value, resulting in low-contrast or absent boundaries.
3. Ambiguity exists when there are more piecewise constant regions than seed groups (labels) in the image.

The random walker algorithm may be viewed as a proposal to address these issues. Almost any image segmentation approach (even region growing or thresholding) may be used to localize correct segments in a piecewise constant image that does not suffer from the above problems. Clearly, in such an image, the random walker algorithm introduced here would also produce the correct segmentation. However, the behavior of the random walker algorithm in the presence of the three difficulties outlined above distinguishes it from other approaches. The behavior of the algorithm in response to these three confounding factors is detailed in Section 4.

The weighting function of (1) implies that the image has piecewise constant intensity. Although such a simple model is reasonable in many gray-scale images, other models such as a piecewise constant texture or color may be used to define the affinities in place of (1) where appropriate.

3.6 Numerical Practicalities

Many good sources exist on the solution to large, sparse, symmetric, linear systems of equations (e.g., [52]). A direct method, such as LU decomposition with partial pivoting has the advantage that the computation necessary to solve (11) is only negligibly increased over the amount of work required to solve (10). Unfortunately, current medical data volumes frequently exceed $256 \times 256 \times 256 \approx 16e^6$ voxels and, hence, require the solution of an equal number of equations. Furthermore, there is no reason to believe that the resolution will not continue to increase. The memory capabilities of most contemporary computers do not have enough memory to allow an LU decomposition with such a large number of equations.

The standard alternative to the class of direct solvers for large, sparse systems is the class of iterative solvers [53]. These solvers have the advantages of a small memory requirement and the ability to represent the matrix-vector multiplication as a function. For a lattice, the matrix L_U has a circulant nonzero structure (although the coefficients are changing), one may avoid storing the matrix entirely. Instead, a vector of weights may be stored (or computed on the fly, if memory is at a premium) and the operation $L_U x_U^s$ may be performed very cheaply. Furthermore, sparse matrix operations (like those required for conjugate gradients) may be efficiently parallelized [54], [55], e.g., for use on a GPU [56], [57]. Because of the relationship of (10) to a finite differences approach to solving the Dirichlet problem on a hypercube domain, the techniques of numerical solution to PDEs may also be applied. Most notably, the algebraic multigrid method [58], [59] achieves near-optimal performance for the solution to equations like (10). Additionally, use of a small world topology [60] might significantly improve the computation speed.

The Graph Analysis Toolbox [61] for MATLAB may be used to easily build weighted image graphs and solve the requisite system of linear equations. Specialty code to perform the random walker segmentation will be made available upon publication on the author's Web page. Although MATLAB has efficient, C++ (MEX), direct solvers for sparse linear systems, the preconditioned conjugate gradient method is written in highly inefficient MATLAB code. Therefore, for research purposes, we recommend using the MATLAB code provided (sufficient for 512×512 images, on present-day technology). A more industrial use will require implementation of conjugate gradients or multigrid code in C++. Fortunately, good references exist for these methods (with source code) [52] that allow for a straightforward implementation. Using MATLAB's direct solver, solution of (10) for a 256×256 image with two randomly placed seed points required 2.5 seconds on an Intel Xeon 2.40GHz processor with 1GB of RAM.

3.7 Algorithm Summary

To summarize, the steps of the algorithm are:

1. Using (1), map the image intensities to edge weights in the lattice.
2. Obtain a set, V_M, of marked (labeled) pixels with K labels, either interactively or automatically.

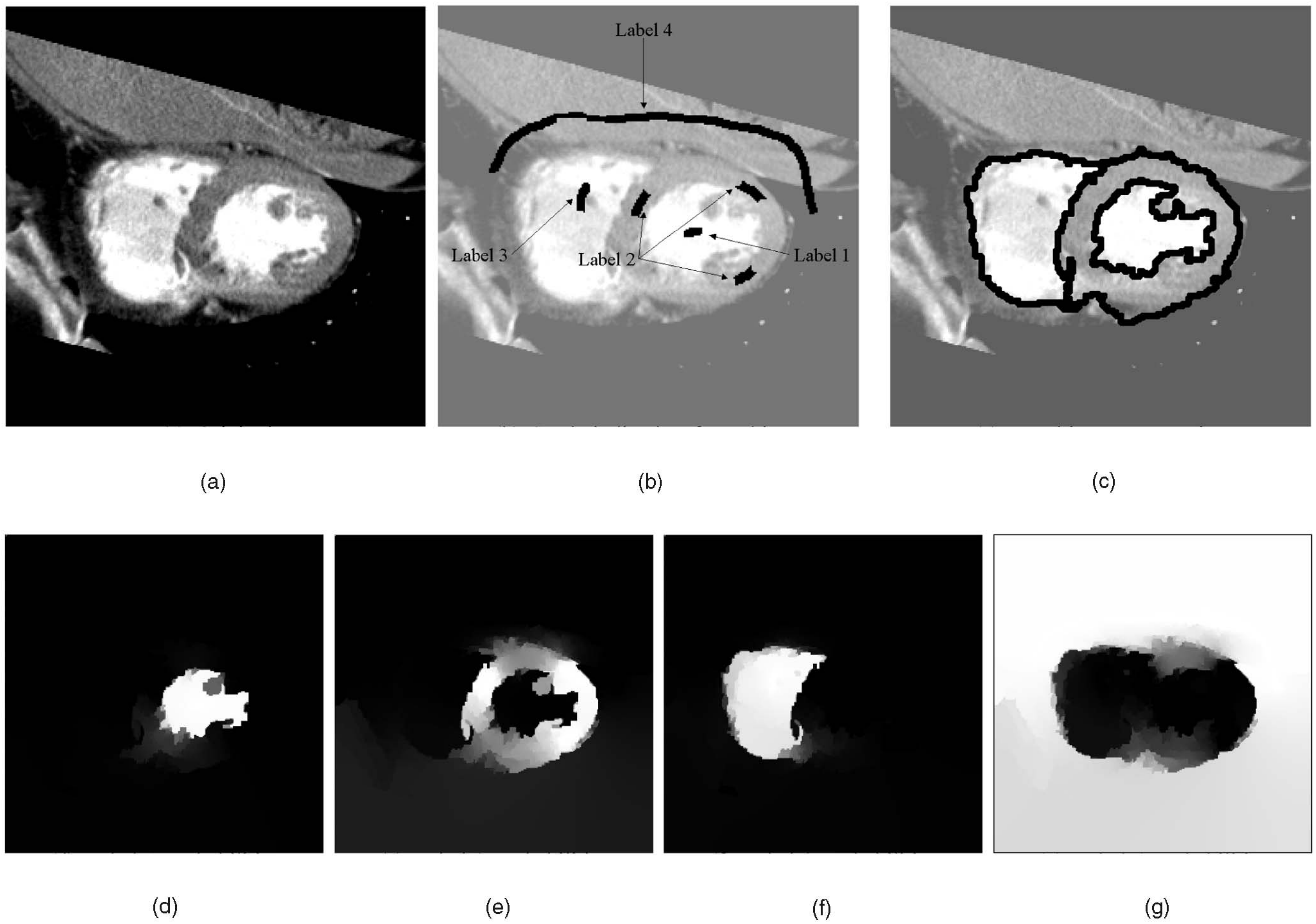


Fig. 2. Overview of segmentation computation. (a) Original image to be segmented. (b) User-placed seeds indicating a desired segmentation into four objects. (d), (e), (f), and (g) Probabilities (potentials) obtained by solving (10) for each label. (c) Segmentation obtained by assigning each pixel to the label for which a random walker is most likely to reach first. Each system required less than three seconds to solve using MATLAB. (a) Original. (b) Seeds indicating four objects. (c) Resulting segmentation. (d) Label 1 probabilities. (e) Label 2 probabilities. (f) Label 3 probabilities. (g) Label 4 probabilities.

3. Solve (11) outright for the potentials or solve (10) for each label except the final one, f (for computational efficiency). Set $x_i^f = 1 - \sum_{s<f} x_i^s$.
4. Obtain a final segmentation by assigning to each node, v_i, the label corresponding to $\max_s (x_i^s)$.

Code is available (in MATLAB) on the author's Web page at: http://www.cns.bu.edu/~lgrady/random_walker_matlab_code.zip.

We note that other options might be explored for assigning a label to each pixel based on the potentials (e.g., applying a clustering algorithm to the K-dimensional vectors at each node). Fig. 2 displays all of the steps in this process from seed acquisition to calculation of the potentials (probabilities) and the resulting segmentation.

If interactive editing of the segmentation were needed (i.e., through the addition/deletion of seeds), one could start at Step 2 in the above procedure with the new seed set and use the previous solution as the starting point for an iterative matrix solver for the new system (10). In general, the previous solution will be "close" to the desired solution, requiring much less time to compute.

4 Theoretical Properties of the Algorithm

Although a new technique was presented for interactive image segmentation, it is necessary to explore what may be predicted about its behavior, both analytically and practically. In this section, we examine theoretical properties of the algorithm, exploring connectedness and the expected behavior in the presence of noise. Specifically, we will show that the segments will be connected and that one can generally expect the algorithm to behave robustly in the presence of noise. We begin by detailing four mathematically equivalent ways of viewing how the algorithm assigns labels to each unseeded pixel and then employ the most convenient analogy to prove the desired theoretical propositions.

4.1 Analogies

There are four mathematically equivalent ways of viewing how the random walker algorithm assigns an unseeded pixel to a label, given a weighted graph:

1. If a random walker leaving the pixel is most likely to first reach a seed bearing label s, assign the pixel to label s.
2. If the seeds are alternately replaced by grounds/unit voltage sources, assign the pixel to the label for

which its seeds being "on" produces the greatest electrical potential.

3. Assign the pixel to the label for which its seeds have the largest effective conductance (i.e., smallest effective resistance) with the pixel.
4. If a 2-tree is drawn randomly from the graph (with probability given by the product of weights in the 2-tree), assign the pixel to the label for which the pixel is most likely to remain connected to. See Section 4.2 for definition of a 2-tree.

The first way of viewing the algorithm provides the motivation and the second provides the implementation, as illustrated in Fig. 1. We examine the third and fourth analogies in the next section.

4.2 Effective Conductance and 2-Trees

The **effective conductance** between nodes v_i and v_j, $\rho(i,j)$, equals the current flow between nodes v_i and v_j when a unit voltage is applied across nodes v_i, v_j. Alternately, the Dirichlet integral of (6) equals the effective conductance between nodes labeled "1" (i.e., "on") and those labeled "0" (i.e., "off") [10]. Therefore, given a solution to (8) with nodes v_i, v_j used as the source/sink, the effective conductance between v_i and v_j, $\rho(i,j)$, may be computed conveniently by $D[x] = \rho(i,j) = x^T L x$, where x is intended to include both x_M and x_U from (7).

It was shown in [10] that the effective conductance between two nodes, v_i, v_j is given by

$$\rho(i,j) = \frac{\tau}{\chi(i,j)}, \tag{20}$$

where τ is a constant for the graph defined as

$$\tau = \sum_{\text{All trees}} \prod_{e \in T} w(e), \tag{21}$$

where T is a set of edges defining a connected tree and the sum is over all possible trees in the graph. Note that τ does not depend on the choice of nodes v_i, v_j. The term $\chi(i,j)$ is defined as

$$\chi(i,j) = \sum_{\text{All TT(i,j)}} \prod_{e \in TT(i,j)} w(e), \tag{22}$$

where $TT(i,j)$ is used to represent the set of edges defining a 2-tree, such that node v_i is in one component and v_j is in another. A **2-tree** is defined to be a tree with one edge removed. It should not be surprising that there exists an analogy with a tree algorithm, since trees have been a major part of circuit theory dating all the way back to Kirchhoff [62].

In addition to solution by (8), it is known [10] that the potential of a node v_t, given $\{0,1\}$ labels at nodes v_i and v_j, respectively, may also be computed (albeit impractically) via

$$x_t^j = \frac{\rho(i,j)\chi^*(i,j,t)}{\tau}, \tag{23}$$

where

$$\chi^*(i,j,t) = \sum_{\text{All } TT(i,j,t)} \prod_{e \in TT(i,j)} w(e) \tag{24}$$

is taken over the sum of all 2-trees such that v_i and v_j are in different components and v_t is in the same component as v_j. Therefore, we note that $\chi^*(i,j,t) = \chi(i,j) - \chi(j,t)$. For a fixed v_i, v_j, it is clear that τ, $\chi(i,j)$, and $\rho(i,j)$ are constants, regardless of whether or not it is v_i or v_j that are "on" or "off." Denoting x_t^i and x_t^j as the probabilities obtained (e.g., via solution to (8)) for v_i set to unity and v_j set to unity, respectively (while the other node is set to zero), then the above equations yield that the following expressions are equivalent

$$\chi(j,t) > \chi(i,t), \tag{25}$$

$$x_t^i > x_t^j, \tag{26}$$

$$\rho(i,t) > \rho(j,t). \tag{27}$$

Since the segmentation is computed from the potentials by assigning the pixel to the label for which it has greatest potential (probability), the equivalence of (26) with (25) and (27) show that these two quantities are also sufficient to define the same segmentation. In other words, the third and fourth analogies given in Section 4.1 are shown to be true. In the following sections, we use all of these viewpoints to theoretically examine the behavior of the algorithm.

4.3 Properties

We begin by giving two properties that are combinatorial analogues of properties of continuous harmonic functions [6] and may be seen directly by viewing the solution to the combinatorial Dirichlet problem as a solution to the combinatorial Laplace equation (with Dirichlet boundary conditions), where the potential of each unseeded node must satisfy

$$x_i^s = \frac{1}{d_i} \sum_{e_{ij} \in E} w(e_{ij}) x_j^s, \tag{28}$$

where the $x_j^s \in V$ (i.e., may include seed points).

1. A potential $0 \le x_i^s \le 1$, $\forall\, i, s$ (maximum/minimum principle).
2. The potential of each unseeded node assumes the weighted average of its neighboring nodes (the mean-value theorem).

We use these properties to first examine the connectedness of the segmentation and show, intuitively, that the segments are always connected to a seed. This property demonstrates that we can expect the segmentation to avoid the noisy or fragmented segmentations that sometimes result from application of other algorithms.

Proposition 1. *If the final segmentation is determined from the potentials using the rule: node v_i is assigned to segment, s, only if $x_i^s > x_i^f \; \forall f \neq s$, then each node assigned to segment s is connected through a path of nodes also assigned to segment s to at least one of the seed points with label s.*

A restatement of this proposition is that the connected components generated by the final segmentation must contain at least one seed point bearing that label.

Proof. Note, this proof is similar to that given in [3] for connectedness using the isoperimetric algorithm.

The result follows if it can be shown that any connected subset, $P \subseteq V_U$, assigned to segment s must be connected to at least one node that is also labeled s.

A block matrix form of (28) may be written

$$L_P x_P^s = -R_P x_{\overline{P}}^s, \tag{29}$$

where $x^s = [x_P^s, x_{\overline{P}}^s]^T$, L has been decomposed into the block form

$$L = \begin{bmatrix} L_P & R_P \\ R_P^T & L_{\overline{P}} \end{bmatrix}, \tag{30}$$

and $\overline{P}$ denotes the set complement of P in V. For example, in the case of $P = \{v_i\}$ in (28), $L_P = d_i$ and

$$-R_P x_{\overline{P}}^s = \sum_{e_{ij} \in E} w(e_{ij}) x_j^s.$$

If $x_P^s > x_P^f \;\; \forall f \neq s$, then $x_P^s - x_P^f > 0$ and

$$-L_P^{-1} R_P \left(x_{\overline{P}}^s - x_{\overline{P}}^f \right) > 0.$$

The entries of R_P are nonpositive by definition of L. Since L is an M-matrix, any block diagonal submatrix of an M-matrix is also an M-matrix, and the inverse of an M-matrix has nonnegative entries (see [63] for the previous three facts), then $-L_P^{-1}R$ has nonnegative entries and, therefore, some $x_i^s \in \overline{P}$ must be greater than $x_i^f \in \overline{P}$. Furthermore, since the entries of R_P are zero for nodes not connected to P, the nodes in $\overline{P}$ satisfying the inequality must be connected to a node in P. □

We note that it is possible, although it almost never occurs in practice, that $x_i^s = x_i^f$, i.e., the potentials for two labels are equal at a node. In this case, one may enforce the continuity property by assigning a connected component of isopotential nodes to a label taken by a neighbor of the set. As demonstrated in the proof of Proposition 1, the *set* of (seedless) isopotential nodes must have at least one neighbor with a potential both greater and lesser than the isopotential nodes.

In the original conference paper, proofs of several propositions concerning noise were given that rested on the proof of a lemma concerning the ratio of random variables. It has subsequently been determined that a flaw exists in the original proof of this lemma,[1] rendering the subsequent proofs invalid. Here, we use the equivalences of (25), (26), and (27) to provide similar statements.

If the graph weights are uniform (i.e., obtained from a uniform image), we term the resulting segmentation the **neutral** segmentation. For simplicity, we take $w(e_{ij}) = 1, \forall e_{ij} \in E$ since multiplication of all weights by a constant does not affect the resulting solution, as may be seen by (8). By, (23), we know that

$$\eta_t^i > \eta_t^j \iff |TT(j,t)| = \chi(j,t) > \chi(i,t) = |TT(i,t)|, \tag{31}$$

where $|TT(i,t)|$ indicates the number of 2-trees with v_i in one component and v_t in the other and η_t^i represents the potential for the neutral segmentation at node v_t with v_i set "on." In the following propositions, boldface will be used to indicate random variables.

1. The author would like to thank Dr. Kurt Majewski for bringing this to his attention.

Proposition 2. *If the set of weights, $\boldsymbol{w_{ij}}$, are independent random variables with equal mean, μ, then $\mathrm{E}[\boldsymbol{\chi}(\boldsymbol{j},\boldsymbol{t})] > \mathrm{E}[\boldsymbol{\chi}(\boldsymbol{i},\boldsymbol{t})]$ if and only if $\eta_t^i > \eta_t^j$.*

Proof. The variable $\boldsymbol{\chi}(\boldsymbol{i},\boldsymbol{t})$ defines a sum of the product of $N-2$ equal-mean, independent variables (i.e., for the $N-2$ edges in a 2-tree). Therefore,

$$\mathrm{E}[\boldsymbol{\chi}(\boldsymbol{j},\boldsymbol{t})] = \mu^{(N-2)} |TT(j,t)|, \tag{32}$$

$$\mathrm{E}[\boldsymbol{\chi}(\boldsymbol{i},\boldsymbol{t})] = \mu^{(N-2)} |TT(i,t)|. \tag{33}$$

Consequently,

$$\mathrm{E}[\boldsymbol{\chi}(\boldsymbol{j},\boldsymbol{t})] > \mathrm{E}[\boldsymbol{\chi}(\boldsymbol{i},\boldsymbol{t})], \tag{34}$$

holds if and only if

$$|TT(j,t)| > |TT(i,t)|, \tag{35}$$

which is known to hold for the neutral segmentation by (31). □

Consequently, in the expected case, the segmentation will be the same as for the neutral segmentation. Since the same technique as above may be used to verify the following two propositions, the proofs are left to the reader.

Proposition 3. *If the set of weights, $\boldsymbol{w_{ij}}$, are independent random variables with corresponding means μ_{ij}, then $\mathrm{E}[\boldsymbol{\chi}(\boldsymbol{j},\boldsymbol{t})] > \mathrm{E}[\boldsymbol{\chi}(\boldsymbol{i},\boldsymbol{t})]$ if and only if $x_t^i > x_t^j$ when the weights are set to $w_{ij} = \mu_{ij}$.*

Proposition 4. *If $w_{ij} = k_{ij}\boldsymbol{y_{ij}}$, where the k_{ij} are (not necessarily equal) constants and $\boldsymbol{y_{ij}}$ are independent random variables, such that $\boldsymbol{y_{ij}} > 0$ and $\mathrm{E}[\boldsymbol{y_{ij}}] = \mu, \forall e_{ij} \in E$, then $\mathrm{E}[\boldsymbol{\chi}(\boldsymbol{j},\boldsymbol{t})] > \mathrm{E}[\boldsymbol{\chi}(\boldsymbol{i},\boldsymbol{t})]$ if and only if $x_t^i > x_t^j$ when the weights are set to $w_{ij} = k_{ij}$.*

Proposition 3 suggests that the means of random weights provide an indicator of the expected segmentation and Proposition 4 indicates that equal-mean, multiplicative noise (of the weights) is not expected to disrupt the solution. We also consider use of the means of random weights to provide an initial guess for a problem with noise.

Proposition 5. *Given a solution to $L_U x^s = -Bm^s$ with $w_{ij} = k_{ij}$ for some, not necessarily equal, constants k, this solution provides an expected residual of zero for the graph where the weights are random variables, $\boldsymbol{y_{ij}}$, with means $\mathrm{E}[\boldsymbol{y_{ij}}] = k_{ij}$.*

Proof. Denote the terms of (8) obtained by setting $w_{ij} = \mu_{ij}$ as $L_U x = -Bm$ (where the label s has been ignored since it is assumed to be fixed) and the terms of (8) in the randomized case as $\boldsymbol{L_U x} = -\boldsymbol{B}m$. Then,

$$\boldsymbol{L_U}(x - \boldsymbol{x}) = (-\boldsymbol{B}m) - \boldsymbol{L_U}x = \boldsymbol{r}, \tag{36}$$

where $\boldsymbol{r}$ represents the residual. Since neither m nor x are random, $\mathrm{E}[\boldsymbol{B}m] = -Bm$, and $\mathrm{E}[\boldsymbol{L_U}x] = L_U x = -Bm$. Consequently, $\mathrm{E}[\boldsymbol{r}] = 0$ and the proposition holds. □

We note that Proposition 5 applies to arbitrary random variables. Although it is well-known that a small residual does not necessarily indicate a small error [52], it is usually a reasonable indicator and, more importantly, forms the stopping criterion for many iterative solvers. Therefore, we can conclude that, if one has obtained a nonrandom

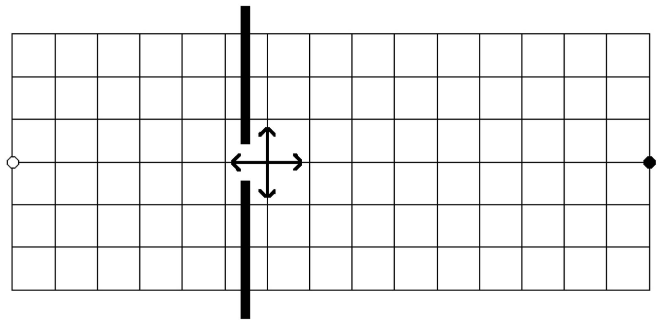


Fig. 3. Illustration of why the segmentation obeys weak image boundaries. Consider the 16×7 image consisting of just one hard boundary with a hole, represented by the thick black line, and two seed points placed at the white and black circles at the far ends of the image. A random walker starting at the pixel next to the weakness in the boundary (the center of the arrows) has three out of four chances on its initial step to enter into the region that is likely to be labeled as belonging to the black circle. Since the same holds true on the other side of the weak boundary, there will be a sharp drop in the probabilities and, consequently, the segmentation will respect the boundary, even though it is weak.

solution to (8), it will generally provide a good starting point for solving the system after noise has been added.

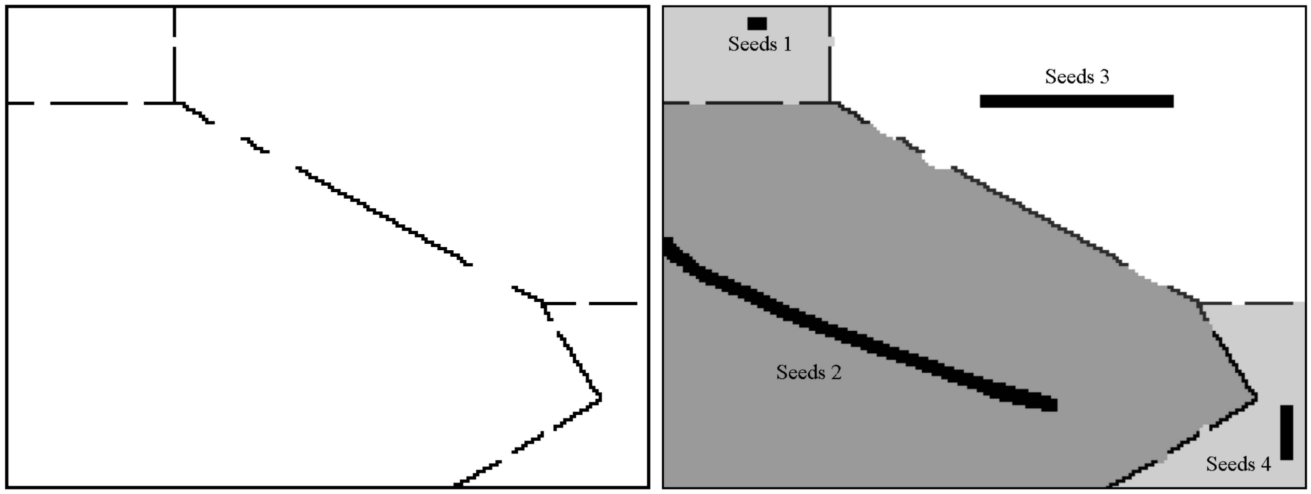
5 Behavioral Properties

In this section, we demonstrate three pragmatic properties of the random walker algorithm—weak boundary detection, noise robustness, and the assignment of ambiguous regions.

5.1 Weak Boundaries

We will prefer the term object *boundary* to the traditional computer vision term *edge* ("edge detection") to avoid confusion with the edge set of the graph (e.g., $e_{ij} \in E$). Unlike region growing approaches, one aspect of the random walker motivation for this algorithm is that weak object boundaries will be found when they are part of a consistent boundary. This behavior may be explained by considering Fig. 3. On a four-connected lattice, consider the walker staring its walk at the center of the four arrows in Fig. 3. This walker has three initial steps that keep it on one side of the boundary and only one step that crosses the boundary. Since other nodes on that side of the boundary are all very likely to reach seed one (filled circle), this walker is also very likely to first reach seed one. For the same reasons, a walker on the other side of the weak boundary is also very likely to first reach seed two (open circle). Consequently, the walker at the arrows finds the first seed (filled circle) and the walker on the opposite side of the boundary weakness finds seed two (open circle). This behavior may also be explained from a circuit perspective. Although the resistance in the boundary weakness is low, nearly all the current from one seed to the other must pass through the boundary weakness, resulting in a large voltage drop over the resistor (by Ohm's Law). Practical behavior of the algorithm in response to weak boundaries is displayed in Figs. 3 and 4. Fig. 4 shows the segmentation obtained for a synthetic image with four areas of varying sizes and convexity with missing boundary sections and few seeds. We note that no obvious "metrication artifacts" are present, despite the fact that these results were obtained using a 4-connected lattice as the underlying structure.

The graph cuts algorithm of [18] is also capable of finding weak boundaries. However, since graph cuts searches for the minimum cut, the graph cuts algorithm is more susceptible to the "small cuts" problem in the presence of weak (i.e., costly) boundaries. Fig. 5 compares the graph cuts and random walker algorithms in a simple, foreground/background segmentation with a weak boundary and small seeds. In contrast to graph cuts, the random walker algorithm also provides a "confidence" value of the segmentation in terms of the random walker probabilities, as Fig. 5 also illustrates.



(a) (b)

Fig. 4. Demonstration of the algorithm response to weak boundaries of different types, large/small regions and nonconvex regions on a synthetic image consisting of only black and white pixels. (a) A synthetic image was created to designate four areas of different size, shape, and convexity by drawing black lines. Sections of the line were then completely erased to remove all contrast at those locations. (b) Seeding and resulting segmentation (visualized by shaded regions). Despite missing boundary information, the algorithm accurately localizes the boundaries. Note that a 4-connected lattice was employed as the underlying graph. (a) Original. (b) Segmentation.

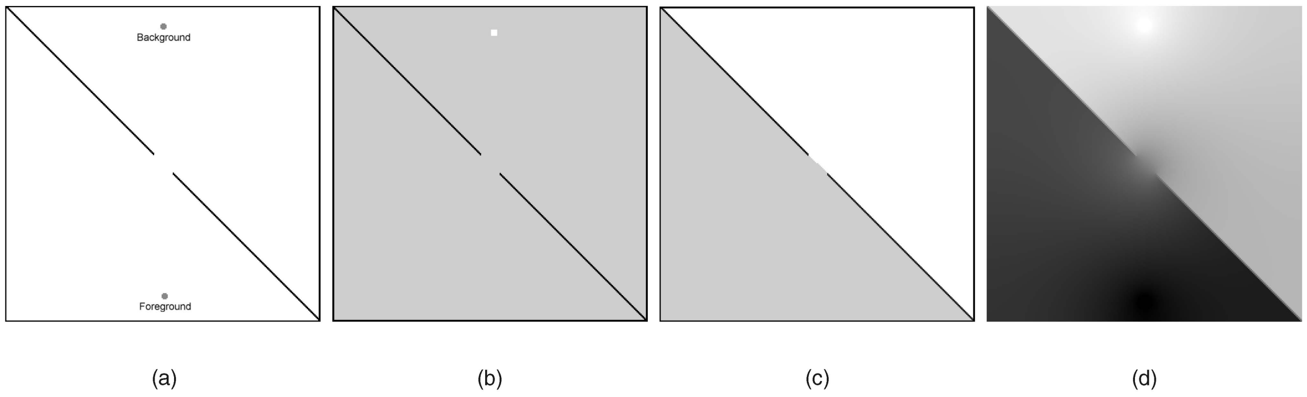


Fig. 5. Comparison of random walker algorithm to graph cuts for a weak boundary with small seeds. Note that a 4-connected graph was used in these experiments. (a) Original (synthetic) image created with a diagonal black line with a section completely erased. (b) Graph cuts solution—Since surface area of seeds is smaller than the weak boundary, the smallest cut minimally surrounds the seeds. (c) Random walker solution. (d) Probabilities associated with the random walker algorithm offer a notion of segmentation confidence at each pixel.

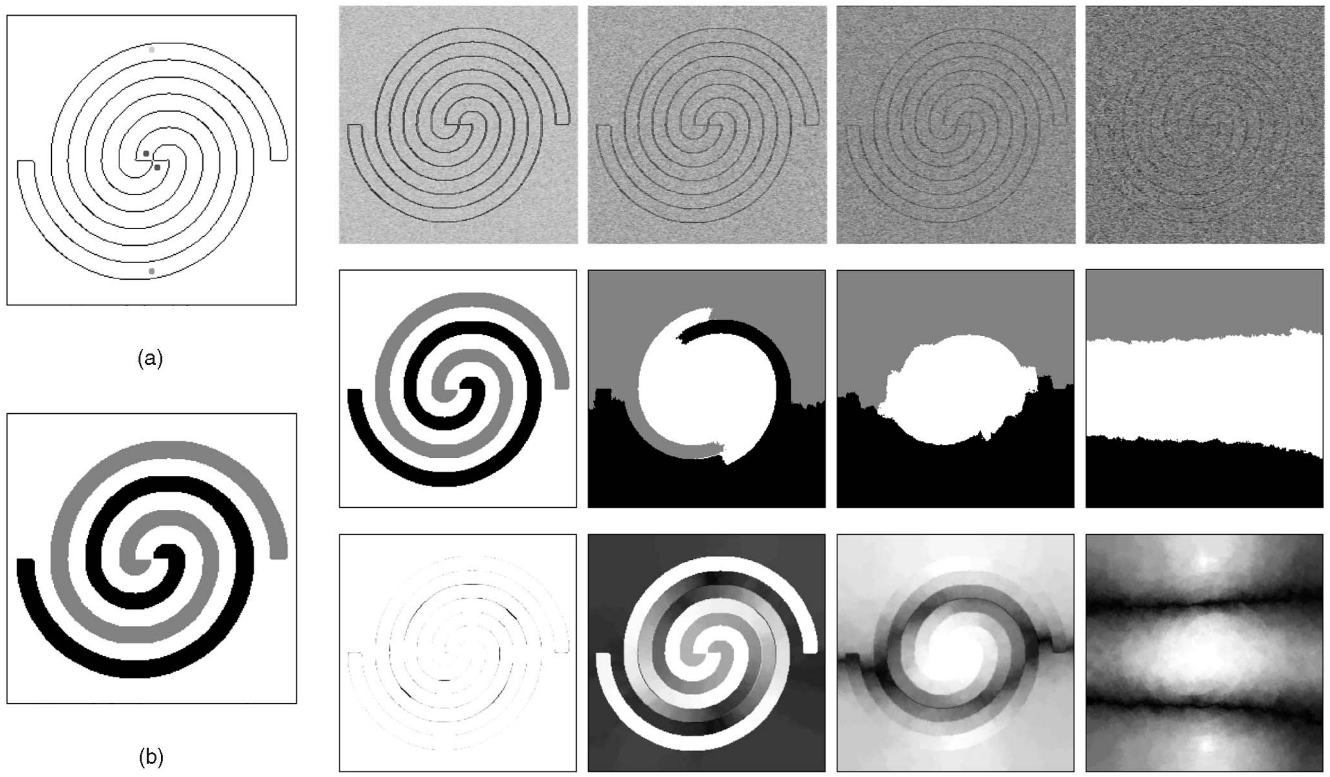


Fig. 6. This figure shows the characteristic response of the algorithm to noise. (a) Original image. An image consisting of two nested spirals was seeded with one seed in each spiral and two background seeds in the center (outside the spirals). (b) Original segmentation. The initial (correct) segmentation. Increasing amounts of additive noise were then introduced into the image and the response of the algorithm was tracked. For each noise level, 100 experiments were run in which the corrupted image was generated and the results were recorded. Top: A representative corrupted image, Middle: The "average" segmentation obtained from application of the algorithm. Bottom: The segmentation variability. An "average" segmentation was calculated by assigning the pixel to the label for which it was most often assigned over the 100 trials. Segmentation variability of a pixel was measured by calculating the percentage of the trials for which the pixel was assigned the label from the "average" segmentation, with high percentage mapped to high intensity (white) and low percentage mapped to low intensity (black).

5.2 Noise Robustness

The theoretical results of Section 4.3 detail how the expected probabilities (and hence, the resulting segmentation) should behave in response to i.i.d. randomness of the weights. Although the weighting function (1) does not translate i.i.d. randomness of the pixel values to i.i.d. randomness of the weights, the behavior of the segmentations empirically behaves as if the weights were i.i.d. This practical behavior might be explained by Proposition 5, which applies to arbitrary (e.g., nonindependent) random variables.

Fig. 6 characterizes the response of the algorithm to noise. In this experiment, an image consisting of two nested spirals was seeded with one seed in each spiral and background seeds placed in the center (outside the spirals). Increasing amounts of additive noise was then introduced into the image and the response of the algorithm was tracked. For each noise level, 100 experiments were run in which the

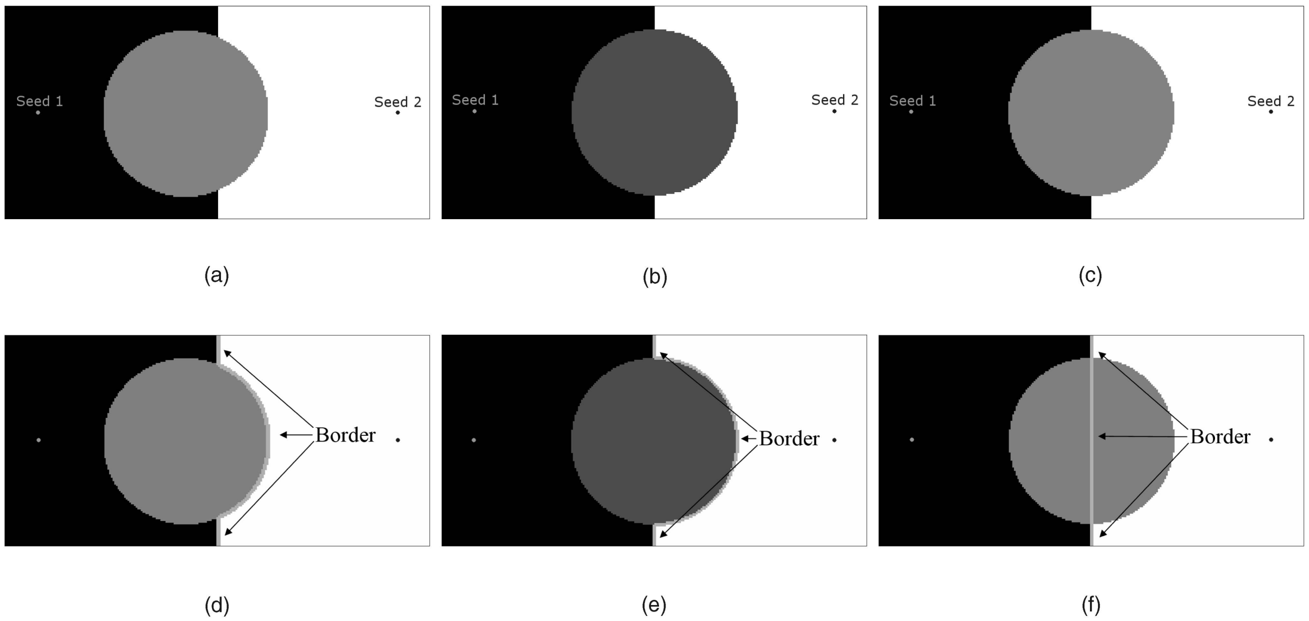


Fig. 7. This figure shows how ambiguous, unseeded regions are assigned to neighboring regions. (a) and (d) If the ambiguous region shares more surface area with one region, it is assigned to that region. (b) and (e) If the ambiguous region is closer in intensity to a neighboring region, it is assigned to that region. (c) and (f) If the ambiguous region is precisely centered between two regions with respect to both surface area and intensity, that region is divided in half. If more ambiguous regions are present (in a piecewise constant image), the ambiguous region will have the average probability of its neighbors, weighted by shared surface area and intensity difference.

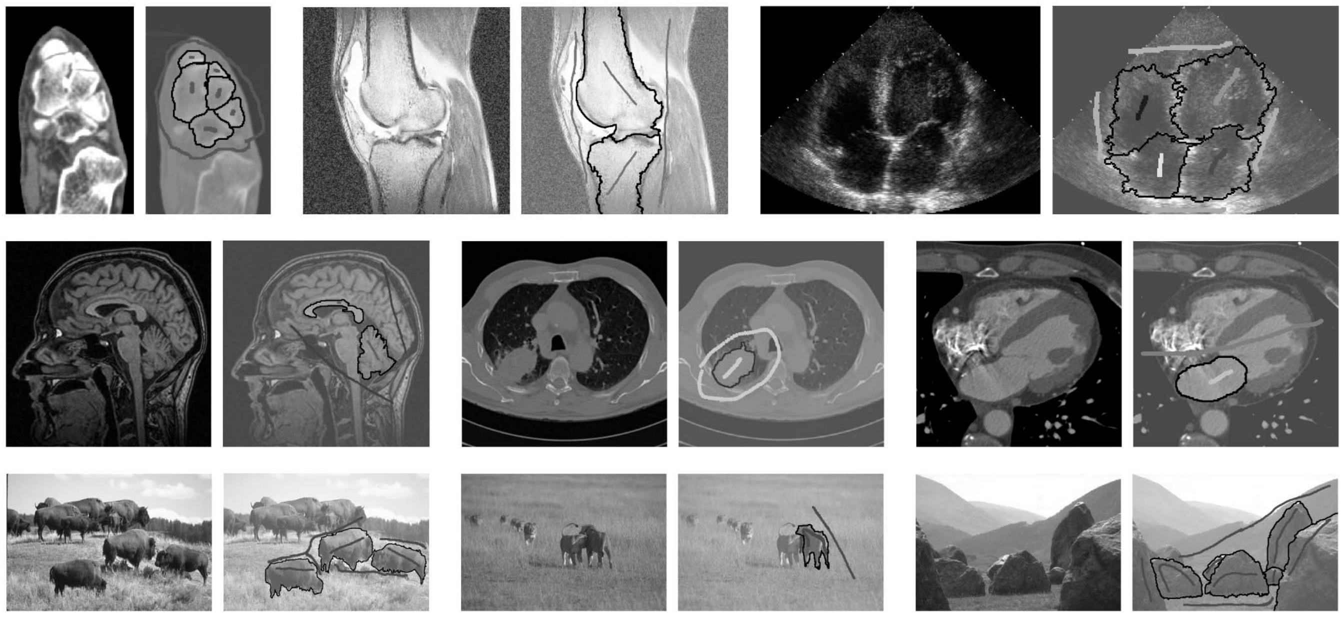
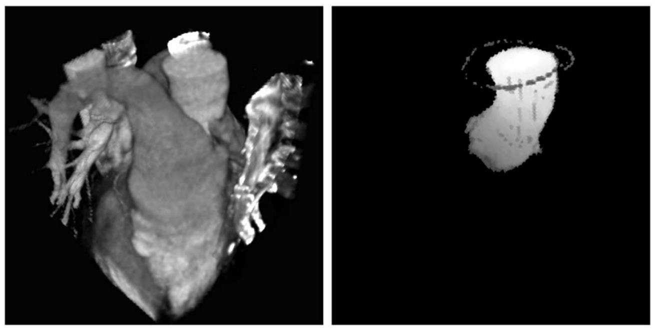


Fig. 8. Examples of segmentations on medical and nonmedical images. We stress that the edge weights are based upon intensity differences alone, although more advanced intensity/texture analysis could be used in a particular problem domain. The thick, gray lines (chosen to maximize contrast) represent the seed points and the thick black lines represent the segment boundaries. Note that the processed images were whitened in order to accentuate the seeds and the segmentation boundary. The same parameter value ($\beta = 900$) and 4-connected lattice topology were used for all segmentations.

corrupted image was generated and the results were recorded. Fig. 6 shows three images for each noise level: 1) A representative corrupted image, 2) the "average" segmentation obtained from application of the algorithm, and 3) the segmentation variability. An "average" segmentation was calculated by assigning the pixel to the label for which it was most often assigned over the 100 trials. Segmentation variability of a pixel was measured by calculating the percentage of the trials for which the pixel was assigned the label from the "average" segmentation, with high percentage mapped to high intensity (white) and low percentage mapped to low intensity (black).

5.3 Ambiguous Unseeded Regions

The analytical properties of the random walker algorithm may be used to examine its behavior in deciding ambiguous cases in which the number of piecewise constant regions exceeds the number of seed/label groups (i.e., unseeded



(a) (b)

Fig. 9. Segmentation on a 3D, 6-connected, lattice without any modification of the algorithm. Seeds were placed on a single slice, consisting of a mark inside the aorta and a ring of background seeds around the outside. (a) Original cardiac CT volume. (b) Segmentation of the aorta is shown in high-intensity pixels with the foreground/background seeds shown in black and white, respectively.

piecewise constant regions must be each assigned a label). In the case where no piecewise constant region has multiple seed/label groups, we may make the simplifying (and reasonable, e.g., Fig. 2) assumption that all pixels in a piecewise constant region may be treated as having the same potential/probability in the solution to (8). By the mean-value property of harmonic functions, the potential/probability inside an unseeded region will be the average of its neighboring regions, weighted by the contrast between the regions and the level of shared surface area. Therefore, an ambiguous region that shares an equal surface area with two seeded regions will be assigned to the region for which it has a lower contrast. However, if an ambiguous region has the same contrast with two seeded regions, it will be assigned to the seeded region with which it shares a greater boundary. If the contrast and the shared surface area of an ambiguous region with two seeded regions are equal, the simplifying assumption of an equipotential within the region breaks down. In such a case, the ambiguous region would be divided in half with respect to the two labels. Fig. 7 illustrates the behavior of the random walker algorithm in these three scenarios.

6 Algorithmic Results

6.1 Segmentation of Real Images

Fig. 8 shows the segmentation results on several medical and nonmedical images. Only gray-scale images were considered here for ease of publication clarity, but color images could be easily handled by modifying (1) to reflect color changes instead of intensity changes. The images and seeds were chosen to demonstrate the general applicability of the interactive segmentation approach on objects of varying uniformity, size, shape, and contrast. In each segmentation, the value of the one free parameter, β in (1), was kept constant, despite the different characteristics of the images. Fig. 9 shows the results of applying the segmentation algorithm to a 3D cardiac CT data set.

We note that a systematic study of the sensitivity of the segmentation to the seed locations/quantities was undertaken in a recent conference paper [64]. The overall result of this study was that the segmentation results were generally stable to perturbations of the seed locations/quantities. Intuitively, perturbations had a greater effect when the image "seemed difficult" to segment (e.g., the image included a large amount of noise and/or had missing or low-contrast boundaries) and showed a lesser effect when the segmentation task "seemed straightforward" to segment (i.e., the image exhibited greater conformity to the piecewise constant model). However, even for images that "seemed difficult" to segment, the algorithm exhibited generally stable behavior to seed locations/quantities.

7 Conclusion

We have presented a novel algorithm for general image segmentation based on a small set of prelabeled pixels. These prelabeled pixels may be given either interactively or generated automatically for a particular purpose. The algorithm functions by assigning each unseeded pixel to the label of the seed point that a random walker starting from that pixel would be most likely to reach first, given that it is biased to avoid crossing object boundaries (i.e., intensity gradients). Since the algorithm is formulated on a general graph and produces segmentations based on the separation of quantities defined at the nodes (i.e., potentials), the graph (lattice) may represent any dimension or topology.

We have demonstrated this approach on real images and shown that it provides a unique, quality, solution that is robust to weak object boundaries and that the solution respects the user's prelabeling choices. Furthermore, there is only a single free parameter, β in (1), and all of the segmentations shown in this paper were produced with the same choice of that parameter. Of course, this approach could also be combined with prefilters (e.g., median) or postfilters (e.g., clustering the probabilities) to produce enhanced, problem-specific results. Finally, the algorithm simply requires solution to a sparse, symmetric, positive-definite system of equations, which is straightforward to implement and performs efficiently. Additionally, interactive editing of the segmentation generally results in even faster computation time since the previous solution may be used as an initial solution for an iterative matrix solver.

The connections between random walks, combinatorial potential theory, Trees, and electric circuits allowed us to prove that the segments are guaranteed to be connected (i.e., unfragmented), and that noise robustness may be expected. Furthermore, the direct correspondence with analog electric circuits opens the possibility for a hardware (e.g., VLSI) implementation of the algorithm, where the physics of the circuit perform the same "computation" as the standard CPU, except at the extremely fast speed of the physical world. Finally, since our variational problem is formulated on a graph, there are no concerns about discretization errors or variations in implementation that sometimes cause problems for other variational approaches.

Future work will concentrate on a specialty solver, user validation, the use of prior information in the segmentation, and leveraging the theoretical results to produce a more effective weighting function.

Acknowledgments

The author would like to thank Gareth Funka-Lea and Yuri Boykov for their advice, criticism, and encouragement during the development of this algorithm. This work is dedicated to the author's father, Dr. Leo J. Grady, who's advice, criticism, and encouragement extended far beyond the development of this algorithm.

References

[1] L. Grady and G. Funka-Lea, "Multi-Label Image Segmentation for Medical Applications Based on Graph-Theoretic Electrical Potentials," *Proc. Workshop Computer Vision and Math. Methods in Medical and Biomedical Image Analysis,* pp. 230-245, May 2004.
[2] R. Wallace, P.-W. Ong, and E. Schwartz, "Space Variant Image Processing," *Int'l J. Computer Vision,* vol. 13, no. 1, pp. 71-90, Sept. 1994.
[3] L. Grady, "Space-Variant Computer Vision: A Graph-Theoretic Approach," PhD dissertation, Boston Univ., 2004.
[4] S. Kakutani, "Markov Processes and the Dirichlet Problem," *Proc. Japanese Academy,* vol. 21, pp. 227-233, 1945.
[5] P. Doyle and L. Snell, *Random Walks and Electric Networks,* no. 22, Washington, D.C.: Math. Assoc. of Am., 1984.
[6] R. Courant and D. Hilbert, *Methods of Math. Physics,* vol. 2. John Wiley and Sons, 1989.
[7] R. Hersh and R.J. Griego, "Brownian Motion and Potential Theory," *Scientific Am.,* vol. 220, pp. 67-74, 1969.
[8] F.H. Branin Jr., "The Algebraic-Topological Basis for Network Analogies and the Vector Calculus," *Proc. Conf. Generalized Networks,* pp. 453-491, Apr. 1966.
[9] P. Tetali, "Random Walks and the Effective Resistance of Networks," *J. Theoretical Probability,* vol. 4, no. 1, pp. 101-109, 1991.
[10] N. Biggs, "Algebraic Potential Theory on Graphs," *Bull. London Math. Soc.,* vol. 29, pp. 641-682, 1997.
[11] Z. Tu and S.-C. Zhu, "Image Segmentation by Data-Driven Markov Chain Monte Carlo," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 24, no. 5, pp. 657-673, May 2002.
[12] S.C. Zhu, Y. Wu, and D. Mumford, "Filters, Random Field and Maximum Entropy (FRAME)," *Int'l J. Computer Vision,* vol. 27, no. 2, pp. 1-20, Mar./Apr. 1998.
[13] P. Perona and J. Malik, "Scale-Space and Edge Detection Using Anisotropic Diffusion," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 12, no. 7, pp. 629-639, July 1990.
[14] J. Shi and J. Malik, "Normalized Cuts and Image Segmentation," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 22, no. 8, pp. 888-905, Aug. 2000.
[15] E. Mortensen and W. Barrett, "Interactive Segmentation with Intelligent Scissors," *Graphical Models in Image Processing,* vol. 60, no. 5, pp. 349-384, 1998.
[16] J.A. Sethian, *Level Set Methods and Fast Marching Methods.* Cambridge Univ. Press, 1999.
[17] M. Kass, A. Witkin, and D. Terzopoulos, "Snakes: Active Contour Models," *Int'l J. Computer Vision,* vol. 1, no. 4, pp. 321-331, 1987.
[18] Y. Boykov and M.-P. Jolly, "Interactive Graph Cuts for Optimal Boundary and Region Segmentation of Objects in N-D Images," *Proc. Int'l Conf. Computer Vision,* pp. 105-112, 2001.
[19] Y. Boykov and V. Kolmogorov, "An Experimental Comparison of Min-Cut/Max-Flow Algorithms for Energy Minimization in Vision," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 26, no. 9, pp. 1124-1137, Sept. 2004.
[20] Y. Boykov, O. Veksler, and R. Zabih, "A New Algorithm for Energy Minimization with Discontinuities," *Proc. Second Int'l Workshop Energy Minimization Methods in Computer Vision and Pattern Recognition,* pp. 205-220, July 1999.
[21] L. Grady, "Multilabel Random Walker Image Segmentation Using Prior Models," *Proc. 2005 IEEE CS Conf. Computer Vision and Pattern Recognition,* pp. 763-770, June 2005.
[22] H. Lombaert, Y. Sun, L. Grady, and C. Xu, "A Multilevel Banded Graph Cuts Method for Fast Image Segmentation," *Proc. Int'l Conf. Computer Vision 2005,* vol. 1, pp. 259-265, Oct. 2005.
[23] Y. Li, J. Sun, C. Tang, and H. Shum, "Lazy Snapping," *Proc. ACM SIGGRAPH Conf.,* pp. 303-308, Apr. 2004.
[24] A. Blake, C. Rother, M. Brown, P. Perez, and P. Torr, "Interactive Image Segmentation Using an Adaptic GMMRF Model," *Proc. European Conf. Computer Vision,* pp. 428-441, May 2004.
[25] C. Rother, V. Kolmogorov, and A. Blake, "'GrabCut'—Interactive Foreground Extraction Using Iterated Graph Cuts," *ACM Trans. Graphics,* vol. 23, no. 3, pp. 309-314, 2004.
[26] C. Zahn, "Graph Theoretical Methods for Detecting and Describing Gestalt Clusters," *IEEE Trans. Computers,* vol. 20, pp. 68-86, 1971.
[27] Z. Wu and R. Leahy, "An Optimal Graph Theoretic Approach to Data Clustering: Theory and Its Application to Image Segmentation," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 11, pp. 1101-1113, 1993.
[28] S. Sarkar and P. Soundararajan, "Supervised Learning of Large Perceptual Organization: Graph Spectral Partitioning and Learning Automata," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 22, no. 5, pp. 504-525, May 2000.
[29] P. Perona and W. Freeman, "A Factorization Approach to Grouping," *Proc. Fifth European Conf. Computer Vision,* pp. 655-670, June 1998.
[30] L. Grady and E.L. Schwartz, "Isoperimetric Graph Partitioning for Image Segmentation," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 28, no. 3, pp. 469-475, Mar. 2006.
[31] A. Barbu and S.-C. Zhu, "Graph Partition by Swendsen-Wang Cuts," *Proc. Ninth IEEE Conf. Computer Vision,* vol. 1, pp. 320-327, Oct. 2003.
[32] L. Grady and E. Schwartz, "Anisotropic Interpolation on Graphs: The Combinatorial Dirichlet Problem," Technical Report CAS/CNS-TR-03-014, Dept. of Cognitive and Neural Systems, Boston Univ., 2003.
[33] A. Levin, D. Lischinski, and Y. Weiss, "Colorization Using Optimization," *Proc. ACM SIGGRAPH Conf.,* pp. 689-694, Aug. 2004.
[34] X. Zhu, Z. Ghahramani, and J. Lafferty, "Semi-Supervised Learning Using Gaussian Fields and Harmonic Functions," *Proc. 20th Int'l Conf. Machine Learning,* pp. 912-919, 2003.
[35] V.B. Eckmann, "Harmonische Funktionen und Randwertaufgaben in einem Komplex," *Commentarii Math. Helvetici,* vol. 17, pp. 240-255, 1945.
[36] U.R. Kodres, "Geometrical Positioning of Circuit Elements in a Computer," *Proc. 1959 AIEE Fall General Meeting,* Oct. 1959.
[37] W.T. Tutte, "How to Draw a Graph," *Proc. London Math. Soc.,* vol. 13, no. 3, pp. 743-768, 1963.
[38] H. Wechsler and M. Kidode, "A Random Walk Procedure for Texture Discrimination," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 1, no. 3, pp. 272-280, 1979.
[39] L. Gorelick, M. Galun, E. Sharon, R. Basri, and A. Brandt, "Shape Representation and Classification Using the Poisson Equation," *Proc. 2004 IEEE CS Conf. Computer Vision and Pattern Recognition (CVPR '04),* vol. 2, pp. 61-67, July 2004.
[40] L. Grady and E.L. Schwartz, "Isoperimetric Partitioning: A New Algorithm for Graph Partitioning," *SIAM J. Scientific Computing,* vol. 27, no. 6, pp. 1844-1866, June 2006.
[41] D. Harel and Y. Koren, "On Clustering Using Random Walks," *Proc. 21st Conf. Foundations of Software Technology and Theoretical Computer Science,* vol. 2245, pp. 18-41, 2001.
[42] L. Yen, D. Vanvyve, F. Wouters, F. Fouss, M. Verleysen, and M. Saerens, "Clustering Using a Random-Walk Based Distance Measure," *Proc. 13th Symp. Artificial Neural Networks,* pp. 317-324, 2005.
[43] M.E.J. Newman, "A Measure of Betweenness Centrality Based on Random Walks," *Social Networks,* vol. 27, no. 1, pp. 39-54, Jan. 2005.
[44] F. Harary, *Graph Theory.* Addison-Wesley, 1994.
[45] M.J. Black, G. Sapiro, D.H. Marimont, and D. Heeger, "Robust Anisotropic Diffusion," *IEEE Trans. Image Processing,* vol. 7, no. 3, pp. 421-432, Mar. 1998.
[46] X. Zhu, J. Lafferty, and Z. Ghahramani, "Combining Active Learning and Semi-Supervised Learning Using Gaussian Fields and Harmonic Functions," *Proc. ICML 2003 Workshop Continuum from Labeled to Unlabeled Data in Machine Learning and Data Mining,* pp. 58-65, 2003.
[47] J. Dodziuk, "Difference Equations, Isoperimetric Inequality and the Transience of Certain Random Walks," *Trans. Am. Math. Soc.,* vol. 284, pp. 787-794, 1984.
[48] J. Roth, "An Application of Algebraic Topology to Numerical Analysis: On the Existence of a Solution to the Network Problem," *Proc. Nat'l Academy of Science of Am.,* vol. 41, pp. 518-521, 1955.
[49] C. Mattiussi, "The Finite Volume, Finite Element and Finite Difference Methods as Numerical Methods for Physical Field Problems," *Advances in Imaging and Electron Physics,* pp. 1-146, Apr. 2000.

[50] F.R.K. Chung, *Spectral Graph Theory,* no. 92, Providence, R.I.: Am. Math. Soc., 1997.

[51] N. Biggs, *Algebraic Graph Theory,* no. 67, Cambridge Univ. Press, 1974.

[52] G. Golub and C. Van Loan, *Matrix Computations,* third ed. The Johns Hopkins Univ. Press, 1996.

[53] W. Hackbusch, *Iterative Solution of Large Sparse Systems of Equations.* Springer-Verlag, 1994.

[54] J.J. Dongarra, I.S. Duff, D.C. Sorenson, and H.A. van der Vorst, *Solving Linear Systems on Vector and Shared Memory Computers.* Philadelphia: SIAM, 1991.

[55] K. Gremban, "Combinatorial Preconditioners for Sparse, Symmetric Diagonally Dominant Linear Systems," PhD dissertation, Carnegie Mellon Univ., Pittsburgh, Penn. Oct. 1996.

[56] J. Bolz, I. Farmer, E. Grinspun, and P. Schröder, "Sparse Matrix Solvers on the GPU: Conjugate Gradients and Multigrid," *ACM Trans. Graphics,* vol. 22, no. 3, pp. 917-924, July 2003.

[57] J. Krüger and R. Westermann, "Linear Algebra Operators for GPU Implementation of Numerical Algorithms," *ACM Trans. Graphics,* vol. 22, no. 3, pp. 908-916, July 2003.

[58] *Multigrid,* U. Trottenberg, C.W. Oosterlee, and A. Schuller, eds. San Diego, Calif.: Academic Press, 2000.

[59] J.E. Dendy, "Black Box Multigrid," *J. Computational Physics,* vol. 48, pp. 366-386, 1982.

[60] L. Grady and E.L. Schwartz, "Faster Graph-Theoretic Image Processing via Small-World and Quadtree Topologies," *Proc. 2004 IEEE CS Conf. Computer Vision and Pattern Recognition,* pp. 360-365, June-July 2004.

[61] L. Grady and E.L. Schwartz, "The Graph Analysis Toolbox: Image Processing on Arbitrary Graphs," Technical Report TR-03-021, Boston Univ., Aug. 2003.

[62] G. Kirchhoff, "Über die Auflösung der Gleichungen, auf Welche man bei der Untersuchung der Linearen Verteilung Galvanischer Ströme Geführt Wird," *Poggendorf's Annalen der Physik Chemie,* vol. 72, pp. 497-508, 1847.

[63] M. Fiedler, *Special Matrices and Their Applications in Numerical Mathematics.* Martinus Nijhoff Publishers, 1986.

[64] L. Grady, T. Schiwietz, S. Aharon, and R. Westermann, "Random Walks for Interactive Organ Segmentation in Two and Three Dimensions: Implementation and Validation," *Proc. Conf. Medical Image Computing and Computer-Assisted Intervention 2005 II,* pp. 773-780, Oct. 2005.

Leo Grady received the BSc degree in electrical engineering from the University of Vermont in 1999 and the PhD degree from the Cognitive and Neural Systems Department at Boston University in 2003. Since Autumn 2003, he has been a member of the technical staff at Siemens Corporate Research in the Imaging and Visualization Department, Princeton. His research focuses on image segmentation, data clustering, learning, and filtering using techniques from graph theory, combinatorial topology, and PDEs. Other interests include pattern/object recognition, applied mathematics, nonuniform data processing, image registration, cellular automata, machine learning, robotics, and emergent phenomena. He is a member of the IEEE.