

Interactive Image Segmentation Via Minimization of Quadratic Energies on Directed Graphs*

Dheeraj Singaraju
Johns Hopkins University
Baltimore, MD
dheeraj@cis.jhu.edu

Leo Grady
Siemens Corporate Research
Princeton, NJ
leo.grady@siemens.com

René Vidal
Johns Hopkins University
Baltimore, MD
rvidal@cis.jhu.edu

Abstract

We propose a scheme to introduce directionality in the Random Walker algorithm for image segmentation. In particular, we extend the optimization framework of this algorithm to combinatorial graphs with directed edges. Our scheme is interactive and requires the user to label a few pixels that are representative of a foreground object and of the background. These labeled pixels are used to learn intensity models for the object and the background, which allow us to automatically set the weights of the directed edges. These weights are chosen so that they bias the direction of the object boundary gradients to flow from regions that agree well with the learned object intensity model to regions that do not agree well. We use these weights to define an energy function that associates asymmetric quadratic penalties with the edges in the graph. We show that this energy function is convex, hence it has a unique minimizer. We propose a provably convergent iterative algorithm for minimizing this energy function. We also describe the construction of an equivalent electrical network with diodes and resistors that solves the same segmentation problem as our framework. Finally, our experiments on a database of 69 images show that the use of directional information does improve the segmenting power of the Random Walker algorithm.

1. Introduction

Image segmentation refers to the problem of dividing an image into a number of disjoint regions such that the features of each region are consistent with each other. Since images generally contain a lot of objects that are further surrounded by clutter, it is often not possible to define a unique

segmentation. In other words, the segmentation problem can be ill-posed when working in an unsupervised framework. Interactive algorithms allow the user to label a few pixels as either *object* or *background*, thereby making the segmentation problem well posed.

Traditional variational methods approach the segmentation problem by minimizing energy functionals that favor alignment of the object boundary with regions of high intensity gradient (e.g., [9, 2]). Recently, Vasilevskiy and Siddiqi [11] noted that the *direction* of the intensity gradient contains valuable information that can improve the quality of the resulting segmentation. More specifically, this directional information is used to define a vector field and the segmentation problem is subsequently posed as the estimation of the object boundary that maximizes the *flux* of this vector field. It was experimentally shown in [11] and [5] that the use of such directional information can greatly benefit the segmentation of elongated objects.

Kolmogorov and Boykov adopted the use of directional gradients in discrete optimization. They solved the image segmentation problem via max-flow/min-cut on a combinatorial graph with directed edges [6]. The authors analyzed the use of Finsler metrics to compute the boundary length of the segmented object and subsequently showed that the use of directed edges is conceptually related to the flux of vector fields defined on the graph. This work additionally corroborated the ability of directional gradient information (i.e., graphs with directed edges) to aid the segmentation of thin structures. Later, Boykov and Funka-Lea demonstrated that the introduction of directional information also improves the ability of Graph Cuts to segment general objects rather than just elongated structures [1]. Subsequent work has shown that the formulation of such directionality in Graph Cuts also has utility in applications other than segmentation [8].

In this work, we focus our interest on Grady’s Random Walker algorithm. This algorithm estimates the segmentation as the solution to the Dirichlet problem formulated on a combinatorial graph with boundary conditions [4]. It

*A majority of this work was done at Siemens Corporate Research (Princeton, NJ). This work is also supported by JHU startup funds, by grants NSF CAREER 0447739, NSF EHS-0509101, and ONR N00014-05-1-083, and by contract JHU APL “Information Fusion & Localization in Distributed Sensor Systems.”

was shown in [4], that the Random Walker algorithm has some advantages with respect to Graph Cuts. For instance it avoids the “shrinking bias” and metrication errors. Also, it easily generalizes to the case of more than two labels. However, a salient feature of the Random Walker algorithm is that it employs symmetric penalties for detecting the image gradients. Motivated by the recent success in using the direction of the image gradients to improve segmentation quality in both continuous and combinatorial algorithms, we propose an approach that incorporates directional information in the Random Walker algorithm.

The construction introduced by Kolmogorov and Boykov for using gradient directionality on a graph involved replacing undirected edges between neighboring nodes with *directed* edges. Kolmogorov and Boykov then used the results from [7] to show that these directed edges may be equivalently converted back to undirected edges with source (“t-link”) terms introduced. A natural approach to introducing directionality (i.e., directed edges) into the Random Walker algorithm might be to apply a similar transformation. Unfortunately, the transformation applied in the Graph Cuts case is only possible because of the linear dependence of each pairwise energy term on the difference of potentials between nodes. In contrast, the Random Walker algorithm employs a quadratic penalty (see [10]), which makes such a construction unsuitable.

Doyle and Snell showed that one can construct an equivalent electrical network with purely resistive impedances, such that the network solves the Dirichlet problem with boundary conditions [3]. Therefore, the optimization of the Random Walker algorithm has an equivalent circuit theory formulation. In this paper, we show that this original circuit theory formulation can be modified in order to interpret the directed edges as *circuit branches with diodes and resistors*. In particular, this modification ensures that the circuit branches exhibit different impedances based on the direction of the current flow through the branches. It is of interest to note that the equivalent circuit theory and random walk interpretations of the Random Walker algorithm of [4] are no longer equivalent when asymmetric penalties are employed in the graph with directed edges.

The paper is organized as follows. In Section 2, we review the classical Random Walker algorithm and describe the construction of the equivalent electrical network that solves the same optimization problem. In Section 3, we propose our modification to this network that serves to extend the Random Walker algorithm to the realm of graphs with directed edges. This formulation helps us pose the segmentation problem as a constrained optimization problem. In Section 4, we establish the existence and uniqueness of the solution to this optimization problem and outline an algorithm for the numerical estimation of this solution. We also discuss the details of another contribution of this paper,

where we provide the machinery to automatically choose the weights of the directed edges in the graph. Finally, in Section 5 we present qualitative and quantitative results that show how our proposed framework helps improve the segmentation results given by the Random Walker algorithm.

2. Image Segmentation Using the Random Walker Algorithm

In this section, we present a short review of the classical Random Walker algorithm for image segmentation. This will consequently help us appreciate why its framework is restricted to the realm of graphs with undirected edges.

Recall that the Random Walker algorithm poses image segmentation as an optimization problem on a combinatorial graph. Therefore, before we proceed, we shall formally define the graph that we work with. The same notation shall be used for the rest of the paper.

A graph \mathcal{G} consists of a pair $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with nodes $v_i \in \mathcal{V}$ and edges $e_{ij} \in \mathcal{E} \subset \mathcal{V} \times \mathcal{V}$. The nodes on the graph typically correspond to pixels in the image. An edge that spans two vertices v_i and v_j is denoted by e_{ij} , and we follow the convention that e_{ij} is oriented from v_i to v_j . The neighborhood of a node v_i , denoted by $\mathcal{N}(v_i)$, is given by the set of all nodes v_j that share an edge with v_i . We construct a weighted graph by assigning to each edge e_{ij} a non-negative value w_{ij} that is referred to as its weight. For graphs with undirected edges, we note that for all edges $e_{ij} \in \mathcal{E}$, we have $w_{ij} = w_{ji}$. This equality need not hold for graphs with directed edges.

The Random Walker algorithm requires the user to interact with it and mark representative *seed* nodes for the object and the background in the image. These seeds embed membership constraints in the graph and are subsequently used to predict the memberships of the remaining unmarked nodes. The set $\mathcal{M} \subset \mathcal{V}$ contains the locations of the nodes marked as seeds and the set $\mathcal{U} \subset \mathcal{V}$ contains the locations of the unmarked nodes. By construction $\mathcal{M} \cap \mathcal{U} = \emptyset$ and $\mathcal{M} \cup \mathcal{U} = \mathcal{V}$. We further split the set \mathcal{M} into the sets $\mathcal{O} \subset \mathcal{M}$ and $\mathcal{B} \subset \mathcal{M}$ that contain the locations of the seeds for the object and the background, respectively. By construction, we have $\mathcal{O} \cap \mathcal{B} = \emptyset$ and $\mathcal{O} \cup \mathcal{B} = \mathcal{M}$.

The Random Walker algorithm proceeds by first defining the weight of edge e_{ij} as $w_{ij} = e^{-\beta \|I_i - I_j\|^2}$, $\beta > 0$, where I_i refers to the intensity of the i^{th} pixel. The user defined seeds provide the boundary conditions on the graph. The seeds corresponding to the object and background are assigned values $x = 1$ and $x = 0$, respectively. The estimation of the memberships $\{x_i\}_{v_i \in \mathcal{U}}$ of the unmarked nodes is then posed as the problem of estimating a harmonic function that satisfies the boundary constraints given by the seeds. This framework is also referred to as the Dirichlet problem

with boundary conditions, the solution of which is given as

$$\begin{aligned} \{\mathbf{x}_i\}_{\mathbf{v}_i \in \mathcal{U}} &= \operatorname{argmin}_{\mathbf{x}} \sum_{e_{ij} \in \mathcal{E}} w_{ij} (\mathbf{x}_i - \mathbf{x}_j)^2, \\ \text{s.t. } \mathbf{x}_i &= 1, \text{ if } \mathbf{v}_i \in \mathcal{O} \text{ and } \mathbf{x}_i = 0, \text{ if } \mathbf{v}_i \in \mathcal{B}. \end{aligned} \quad (1)$$

The maximum principle for harmonic functions states that such functions achieve their maximum and minimum value at the boundary. Hence, the value of \mathbf{x}_i at any unmarked node \mathbf{v}_i is constrained to lie in $[0, 1]$ [3]. Moreover, \mathbf{x}_i is the probability that a random walker starting from node \mathbf{v}_i reaches the object seeds before the background seeds. Therefore, by placing a decision boundary of $\mathbf{x} = 0.5$ (for the two label case) on the values at the unmarked seeds, the algorithm obtains a segmentation of the image.

The setup in (1) has the interesting property that there exists an equivalent electrical network that solves the same optimization problem. The network is represented by the same graph constructed by the Random Walker algorithm. Every node in the graph corresponds to a node in the network. The edge weights correspond to the conductance values of resistors connecting neighboring nodes, *i.e.* $\frac{1}{R_{ij}} = w_{ij}$. The background seeds provide the network's ground, while the object seeds act as unit voltage sources with respect to the ground. Therefore, each background seed $\mathbf{v}_i \in \mathcal{B}$ is at potential $\mathbf{x}_i = 0V$ and each object seed $\mathbf{v}_i \in \mathcal{O}$ is at potential $\mathbf{x}_i = 1V$, where all measurements are with respect to the ground. From network theory, we know that the potentials at the unmarked nodes distribute themselves such that they satisfy Kirchoff's current and voltage laws. It can be shown that these potentials minimize the energy dissipated by the network, which is given by the expression in (1) [3].

3. Electrical Networks for Asymmetric Quadratic Penalty Schemes

In this section, we discuss how the the numerical framework of the classical Random Walker algorithm can be extended to graphs with directed edges. We introduce the optimization problem that needs to be solved for this purpose, and describe the construction of equivalent electrical networks that solve the same optimization problem. These networks help provide an intuition for the behavior and properties of our proposed framework.

The construction of the electrical network described in Section 2 helps us appreciate why the Random Walker algorithm is designed for undirected edges. The resistor is a symmetric network element and we note from the expression in (1) that a transition from \mathbf{v}_i to \mathbf{v}_j is penalized the same as a transition from \mathbf{v}_j to \mathbf{v}_i . To introduce directionality, we would like the penalties to depend on the direction of the current flow between a pair of nodes. More specifically, the desired behavior is as described in Table 1. Consequently, we modify the energy function minimized by

Type of edges	Penalty for $\mathbf{x}_i \geq \mathbf{x}_j$	Penalty for $\mathbf{x}_i < \mathbf{x}_j$
Undirected ($w_{ij} = w_{ji}$)	$w_{ij}(\mathbf{x}_i - \mathbf{x}_j)^2$	$w_{ij}(\mathbf{x}_j - \mathbf{x}_i)^2$
Directed ($w_{ij} \neq w_{ji}$)	$w_{ij}(\mathbf{x}_i - \mathbf{x}_j)^2$	$w_{ji}(\mathbf{x}_j - \mathbf{x}_i)^2$

Table 1. Different penalty schemes for graphs with undirected edges and graphs with directed edges.

the Random Walker algorithm to estimate the memberships $\{\mathbf{x}_i\}_{\mathbf{v}_i \in \mathcal{U}}$ of the unmarked nodes as the minimizer of

$$E(\mathbf{x}) = \sum_{e_{ij} \in \mathcal{E}} \left[w_{ij} \mathcal{I}(\mathbf{x}_i \geq \mathbf{x}_j) + w_{ji} \mathcal{I}(\mathbf{x}_i < \mathbf{x}_j) \right] (\mathbf{x}_i - \mathbf{x}_j)^2, \quad (2)$$

subject to the constraints $\mathbf{x}_i = 1$, if $\mathbf{v}_i \in \mathcal{O}$ and $\mathbf{x}_i = 0$, if $\mathbf{v}_i \in \mathcal{B}$. We use $\mathcal{I}(A)$ as an indicator function of the event A , such that $\mathcal{I}(A) = 1$, if the event A is true and $\mathcal{I}(A) = 0$ otherwise. Note that the energy minimized by the Random Walker algorithm is a special case of $E(\mathbf{x})$ where $\forall e_{ij} \in \mathcal{E}, w_{ij} = w_{ji}$. A simple check by plugging $w_{ij} = w_{ji}$ in our subsequent analysis will show that the conclusions of our general framework reduce to the standard results known about the Random Walker algorithm.

It is interesting to look at what implications this modification has in the domain of electrical networks. The values \mathbf{x}_i have the same interpretation as in the classical random walker setup, *i.e.* they represent the potentials at the nodes in the network. The seeds for the object and background provide the voltage sources and the ground. The modification is in the construction of the network elements connecting neighboring nodes. Directionality is introduced by connecting every pair of neighboring nodes with diodes and resistors as shown in Figure 1. If the diodes are ideal, they are active only when the potential drop across them is positive. Therefore, if \mathbf{x}_i is greater than \mathbf{x}_j , the diode D_{ij} is active and the diode D_{ji} is off. The current flows from \mathbf{v}_i to \mathbf{v}_j and the value of the resistance between the nodes is equal to $R_{ij} = \frac{1}{w_{ij}}$. By a similar argument one can see that if \mathbf{x}_j is greater than \mathbf{x}_i , the current flows from \mathbf{v}_j to \mathbf{v}_i and the resistance between the nodes is $R_{ji} = \frac{1}{w_{ji}}$.

It is of interest to note that while the circuit theory and random walk interpretations of the Random Walker algorithm of [4] are equivalent in the case of an undirected

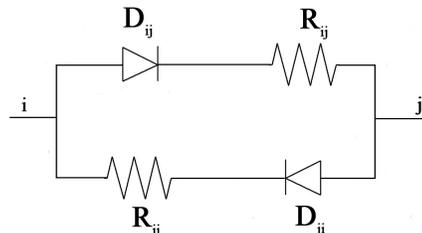


Figure 1. Resistor-diode configuration between a pair of nodes.

graph, the two interpretations are no longer equivalent when directed edges are employed in the graph. Although it would have been possible to formulate a Random Walker algorithm using a directed walk, it can be verified that such an algorithm would still penalize the boundary gradients in a symmetric fashion. However, the circuit that we have formulated clearly employs asymmetric penalty schemes. Therefore, the incorporation of asymmetric penalties into the Random Walker algorithm corresponds to a generalization of the energy dissipated by the network and is *not* equivalent to a random walk on a directed graph.

4. Numerical Minimization of Asymmetric Quadratic Energies on Directed Graphs

In this section, we shall establish that the proposed energy function $E(\mathbf{x})$ is guaranteed to have a unique minimizer. As we shall elaborate, the optimization required for estimating this minimizer is not as simple as that of the Random Walker algorithm. Recall that the constructed network contains diodes which are essentially non-linear devices. As a result, the potentials cannot be obtained by merely solving a linear system. To this effect, we shall outline an iterative algorithm for numerically estimating the minimizer.

4.1. Uniqueness and Existence of Solution

The fact that $E(\mathbf{x})$ has a minimizer follows naturally from the construction of the network. Note that if we are provided information about all the active diodes, one can treat the network as having purely resistive impedances, the values of which are governed by the activity of the diodes. Since the impedance for each edge can take on one out of two values, there is a finite number ($\leq 2^{|\mathcal{E}|}$) of such purely resistive networks. For each of these networks, recall that the Random Walker algorithm guarantees the existence of a unique minimizer of the energy dissipated by the network. Therefore, one can go through the entire list of these resistive networks, in order to find the potentials of the unmarked nodes that minimize $E(\mathbf{x})$. Hence, our optimization problem is guaranteed to have a solution.

From the discussion above, we know that the solution to our optimization problem is obtained by applying the Random Walker algorithm to a resistive network, the impedances of which are governed by the diodes' activity. Recall from Section 2 that the potentials at the unmarked nodes, as estimated by the Random Walker algorithm, are constrained to take values between 0 and 1. Consequently, we know that any solution to our optimization problem must satisfy (a) $x_i = 1$, if $i \in \mathcal{O}$, (a) $x_i = 0$, if $i \in \mathcal{B}$, and (c) $x_i \in [0, 1]$, if $i \in \mathcal{U}$. It is easy to verify that this set (say \mathcal{X}) of possible solutions is convex. Therefore, it is sufficient to establish that $E(\mathbf{x})$ is a strictly convex function on \mathcal{X} , in order to establish uniqueness of the minimizer.

In particular, consider any arbitrary scalar $\lambda \in [0, 1]$ and two arbitrary vectors $\mathbf{x}, \mathbf{y} \in \mathcal{X}$ such that $\mathbf{x} \neq \mathbf{y}$. Define the vector $\mathbf{z} \in \mathcal{X}$ as $\mathbf{z} = \lambda\mathbf{x} + (1 - \lambda)\mathbf{y}$. To establish strict convexity of $E(\cdot)$, it is sufficient to prove that

$$\lambda E(\mathbf{x}) + (1 - \lambda)E(\mathbf{y}) \geq E(\mathbf{z}), \quad (3)$$

with equality occurring iff $\lambda = 0$ or $\lambda = 1$. In what follows, we shall prove (3) by showing that for each edge $e_{ij} \in \mathcal{E}$,

$$\begin{aligned} & \lambda \left[w_{ij} \mathcal{I}(\mathbf{x}_i \geq \mathbf{x}_j) + w_{ji} \mathcal{I}(\mathbf{x}_i < \mathbf{x}_j) \right] (\mathbf{x}_i - \mathbf{x}_j)^2 \\ & + (1 - \lambda) \left[w_{ij} \mathcal{I}(\mathbf{y}_i \geq \mathbf{y}_j) + w_{ji} \mathcal{I}(\mathbf{y}_i < \mathbf{y}_j) \right] (\mathbf{y}_i - \mathbf{y}_j)^2 \\ & \geq \left[w_{ij} \mathcal{I}(\mathbf{z}_i \geq \mathbf{z}_j) + w_{ji} \mathcal{I}(\mathbf{z}_i < \mathbf{z}_j) \right] (\mathbf{z}_i - \mathbf{z}_j)^2, \end{aligned} \quad (4)$$

with equality occurring iff $\lambda = 0$ or $\lambda = 1$. Notice that this would essentially require us to verify (4) for 6 possible cases, based on the sign of the potential drop across an edge. More specifically, we would have to verify (4) for the following cases.

1. $\mathbf{x}_i \geq \mathbf{x}_j, \mathbf{y}_i \geq \mathbf{y}_j, \mathbf{z}_i \geq \mathbf{z}_j$
2. $\mathbf{x}_i \geq \mathbf{x}_j, \mathbf{y}_i < \mathbf{y}_j, \mathbf{z}_i \geq \mathbf{z}_j$
3. $\mathbf{x}_i \geq \mathbf{x}_j, \mathbf{y}_i < \mathbf{y}_j, \mathbf{z}_i < \mathbf{z}_j$
4. $\mathbf{x}_i < \mathbf{x}_j, \mathbf{y}_i \geq \mathbf{y}_j, \mathbf{z}_i \geq \mathbf{z}_j$
5. $\mathbf{x}_i < \mathbf{x}_j, \mathbf{y}_i \geq \mathbf{y}_j, \mathbf{z}_i < \mathbf{z}_j$
6. $\mathbf{x}_i < \mathbf{x}_j, \mathbf{y}_i < \mathbf{y}_j, \mathbf{z}_i < \mathbf{z}_j$

We shall verify (4) for cases 1 and 2 only. The proof for case 1 is similar to the proof for case 6. Similarly, the proof for case 2 can be used to construct the proofs for cases 3-5.

Proof for the case $\mathbf{x}_i \geq \mathbf{x}_j, \mathbf{y}_i \geq \mathbf{y}_j, \mathbf{z}_i \geq \mathbf{z}_j$

The proof in this case is very simple. Essentially, we need to verify that (4) holds true, by checking the fact that

$$\lambda w_{ij} (\mathbf{x}_i - \mathbf{x}_j)^2 + (1 - \lambda) w_{ij} (\mathbf{y}_i - \mathbf{y}_j)^2 - w_{ij} (\mathbf{z}_i - \mathbf{z}_j)^2 \geq 0. \quad (5)$$

After substituting $\mathbf{z}_i = \lambda\mathbf{x}_i + (1 - \lambda)\mathbf{y}_i$ and $\mathbf{z}_j = \lambda\mathbf{x}_j + (1 - \lambda)\mathbf{y}_j$ into (5) and rearranging the resulting terms, we see that the expression in the left hand side of (5) can be written as $\lambda(1 - \lambda)w_{ij} [(\mathbf{x}_i - \mathbf{x}_j) - (\mathbf{y}_i - \mathbf{y}_j)]^2$.

This expression is non-negative and is identically equal to zero if and only if (a) $\lambda = 0$, (b) $\lambda = 1$, or (c) $\mathbf{x}_i - \mathbf{x}_j = \mathbf{y}_i - \mathbf{y}_j$. Note that although we get an extra undesirable condition (c) for the left hand side of (5) being identically equal to zero, we shall show later that this does not affect our proof for the strict convexity of $E(\cdot)$.

Proof for the case $\mathbf{x}_i \geq \mathbf{x}_j, \mathbf{y}_i < \mathbf{y}_j, \mathbf{z}_i \geq \mathbf{z}_j$

In this case, we need to verify (4) by checking the fact that

$$\lambda w_{ij}(\mathbf{x}_i - \mathbf{x}_j)^2 + (1 - \lambda)w_{ji}(\mathbf{y}_i - \mathbf{y}_j)^2 - w_{ij}(\mathbf{z}_i - \mathbf{z}_j)^2 \geq 0. \quad (6)$$

We use a result from the proof for case 1, to observe that the expression in the left hand side of (6) can be rewritten as $(1 - \lambda)[\lambda w_{ij}[(\mathbf{x}_i - \mathbf{x}_j) - (\mathbf{y}_i - \mathbf{y}_j)]^2 + (w_{ji} - w_{ij})(\mathbf{y}_i - \mathbf{y}_j)^2]$.

Now, note that although λ is allowed to take values in $[0, 1]$ in general, it is actually constrained to a smaller set in this case. It can take only those values in $[0, 1]$ that ensure that the hypothesis of this case is satisfied, *i.e.* $\mathbf{z}_i \geq \mathbf{z}_j$. We therefore see that we have the result

$$\begin{aligned} \lambda \mathbf{x}_i + (1 - \lambda)\mathbf{y}_i &\geq \lambda \mathbf{x}_j + (1 - \lambda)\mathbf{y}_j \\ \implies \lambda &\geq \frac{\mathbf{y}_j - \mathbf{y}_i}{(\mathbf{x}_i - \mathbf{x}_j) - (\mathbf{y}_i - \mathbf{y}_j)} > 0. \end{aligned} \quad (7)$$

Therefore, λ is constrained to take values in $(\frac{\mathbf{y}_j - \mathbf{y}_i}{(\mathbf{x}_i - \mathbf{x}_j) - (\mathbf{y}_i - \mathbf{y}_j)}, 1]$. We use this constraint to derive a lower bound for $\lambda w_{ij}[(\mathbf{x}_i - \mathbf{x}_j) - (\mathbf{y}_i - \mathbf{y}_j)]^2$ as

$$\begin{aligned} &\lambda w_{ij}[(\mathbf{x}_i - \mathbf{x}_j) - (\mathbf{y}_i - \mathbf{y}_j)]^2 \\ &\geq \lambda^2 w_{ij}[(\mathbf{x}_i - \mathbf{x}_j) - (\mathbf{y}_i - \mathbf{y}_j)]^2 \quad (\text{because } \lambda \leq 1) \\ &> w_{ij}(\mathbf{y}_i - \mathbf{y}_j)^2. \quad (\text{because } \lambda > \frac{\mathbf{y}_j - \mathbf{y}_i}{(\mathbf{x}_i - \mathbf{x}_j) - (\mathbf{y}_i - \mathbf{y}_j)}) \end{aligned} \quad (8)$$

Given this lower bound on $\lambda w_{ij}[(\mathbf{x}_i - \mathbf{x}_j) - (\mathbf{y}_i - \mathbf{y}_j)]^2$, we can obtain a lower bound for the expression in the left hand side of (6) as

$$\begin{aligned} &\lambda w_{ij}[(\mathbf{x}_i - \mathbf{x}_j) - (\mathbf{y}_i - \mathbf{y}_j)]^2 + (w_{ji} - w_{ij})(\mathbf{y}_i - \mathbf{y}_j)^2 \\ &> [w_{ij}(\mathbf{y}_i - \mathbf{y}_j)^2 + (w_{ji} - w_{ij})(\mathbf{y}_i - \mathbf{y}_j)^2] \\ &= w_{ji}(\mathbf{y}_i - \mathbf{y}_j)^2. \end{aligned} \quad (9)$$

We use this inequality to conclude that the expression in the left hand side of (6) is non-negative and is identically equal to zero if and only if $\lambda = 1$. The proof for case 2 is now complete.

Note that cases 2,3,4 and 5 always give a strict inequality if $\lambda \notin \{0, 1\}$. The only area of concern when $\lambda \notin \{0, 1\}$, is when all the edges satisfy either case 1 or case 6. This is due to the condition (c) as derived for case 1, which states that $\forall e_{ij} \in \mathcal{E}, (\mathbf{x}_i - \mathbf{x}_j) = (\mathbf{y}_i - \mathbf{y}_j)$. However, recall that the hypothesis states that the values of \mathbf{x}_i are fixed at all the nodes $\mathbf{v}_i \in \mathcal{M}$ that correspond to the seeds. Coupled with these fixed boundary conditions, condition (c) implies that $\forall \mathbf{v}_i \in \mathcal{V}, \mathbf{x}_i = \mathbf{y}_i$, which contradicts the hypothesis that $\mathbf{x} \neq \mathbf{y}$. Therefore, this extra condition does not affect our proof, and we conclude that $E(\cdot)$ is strictly convex.

4.2. Algorithm Details

As motivated earlier, the potentials cannot be estimated in a linear fashion. One could adopt a brute force approach and scan all the possible resistive networks to find the minimizer of $E(\mathbf{x})$. However, this approach is computationally very expensive. Alternatively, we can adopt a mesh analysis and estimate the mesh currents in the network. However, this estimation would require us to enforce the additional constraint that the current flowing across an edge is consistent with the direction of the active diode. This becomes a problem of quadratic programming, which can be computationally cumbersome.

In what follows, we shall outline the steps for numerically estimating the minimizer of the energy function $E(\mathbf{x})$. For the sake of simplicity, let us define a Laplacian matrix L^x whose entries depend on the graph's edge weights and the node potentials as

$$L_{ij}^x = \begin{cases} w_{ij}\mathcal{I}(\mathbf{x}_i \geq \mathbf{x}_j) + w_{ji}\mathcal{I}(\mathbf{x}_i < \mathbf{x}_j) & \text{if } e_{ij} \in \mathcal{E}, \\ -\sum_{k \in \mathcal{N}(\mathbf{v}_i)} L_{ik}^x & \text{if } i = j, \\ 0 & \text{otherwise.} \end{cases} \quad (10)$$

Consequently, the energy $E(\mathbf{x})$ that is to be minimized can be expressed in terms of the potentials \mathbf{x} as

$$\begin{aligned} E(\mathbf{x}) &= \mathbf{x}^\top L^x \mathbf{x} \\ &= [\mathbf{x}_U^\top \quad \mathbf{x}_M^\top] \begin{bmatrix} L_U^x & B \\ B^\top & L_M \end{bmatrix} \begin{bmatrix} \mathbf{x}_U \\ \mathbf{x}_M \end{bmatrix} \\ &= \mathbf{x}_U^\top L_U^x \mathbf{x}_U + 2\mathbf{x}_U^\top B \mathbf{x}_M + \mathbf{x}_M^\top L_M \mathbf{x}_M, \end{aligned} \quad (11)$$

where the vectors \mathbf{x}_M and \mathbf{x}_U refer to the potentials of the seeds and the unmarked points. Recall from Section 2, that the potentials at the seeds take values 0 or 1, while the potentials at the unmarked nodes always lie in $[0, 1]$. Therefore, the orientation of the diodes is fixed in the immediate neighborhood of the seeds. Consequently, we can drop the dependencies of the matrices of L_M and B on \mathbf{x} since they define the connections of the seeds with their neighboring nodes. Having formalized our definitions, we outline the steps of our proposed framework in Algorithm 1. Essentially, we estimate the desired solution by adopting an iterative scheme that descends the energy function $E(\mathbf{x})$.

Note that $E(\mathbf{x})$ is differentiable. Moreover, $E(\mathbf{x})$ is also convex, hence one can use a gradient descent approach to estimate the minimizer of $E(\mathbf{x})$. However, such approaches are known to have slow convergence. Consequently, we may adopt a Newton descent approach, where the Hessian $\mathcal{H}(\mathbf{x}) = \frac{\partial^2 E(\mathbf{x})}{\partial \mathbf{x}_U^2}$ is used to estimate the search direction as $d = -\mathcal{H}^{-1}(\mathbf{x}) \frac{\partial E(\mathbf{x})}{\partial \mathbf{x}_U}$. Unfortunately, $E(\mathbf{x})$ is not C^2 and its Hessian cannot be evaluated for all $\mathbf{x}_U \in [0, 1]^{|\mathcal{U}|}$. In particular, whenever $\mathbf{x}_i = \mathbf{x}_j$, for some $e_{ij} \in \mathcal{E}$, the Hessian does not exist. Consequently, we operate in a Newton

Algorithm 1 (Numerical Scheme for Minimizing $E_\epsilon(\mathbf{x})$).

- Set the values of the potentials \mathbf{x}_M at the marked nodes as $\mathbf{x}_i = 1$, if $i \in \mathcal{O}$ and $\mathbf{x}_i = 0$, if $i \in \mathcal{B}$. Initialize the values of the potentials at the unmarked nodes by setting them to 0, *i.e.* $\mathbf{x}_U^{(0)} = \mathbf{0}$.
 - Set $i = 0$;
 - While $|\mathbf{x}_U^{(i+1)} - \mathbf{x}_U^{(i)}| > \delta$
 - Estimate $\mathbf{x}_U^{(i+1)}$ as the minimizer of $\mathbf{x}^\top L^{\mathbf{x}^{(i)}} \mathbf{x}$, which is given by $\mathbf{x}_U^{(i+1)} = -[L_U^{\mathbf{x}^{(i)}}]^{-1} B \mathbf{x}_M$
 - $i = i + 1$
 - end while
-

descent like framework, where the Hessian is approximated by the Laplacian $L^{\mathbf{x}}$, which is defined for all \mathbf{x} . In fact, it is interesting to observe that if $\forall e_{ij} \in \mathcal{E}, \mathbf{x}_i \neq \mathbf{x}_j$, then the Laplacian $L^{\mathbf{x}}$ is identically equal to $\mathcal{H}(\mathbf{x})$.

We make an interesting observation that when the graph is connected, the Laplacian matrix $L^{\mathbf{x}}$ has only one null vector given by the constant vector of 1s. In such a case $L_U^{\mathbf{x}}$ is positive definite, and this can be verified via a proof by contradiction. Assume that there exists some non zero \mathbf{x}_U such that $\mathbf{x}_U^\top L_U^{\mathbf{x}} \mathbf{x}_U \leq 0$. Now set $\mathbf{x}_M = \mathbf{0}_M$. It can be verified from (11) that $\mathbf{x}^\top L^{\mathbf{x}} \mathbf{x} = \mathbf{x}_U^\top L_U^{\mathbf{x}} \mathbf{x}_U \leq 0$. This contradicts the fact that the Laplacian is positive definite with the only null vector given by the constant vector of 1s.

Now, notice from Algorithm 1, that the the update of the unmarked potentials at the i^{th} step, is given by the vector $d\mathbf{x}^{(i)} = \mathbf{x}_U^{(i+1)} - \mathbf{x}_U^{(i)} = -\left(L_U^{\mathbf{x}^{(i)}}\right)^{-1} \left[L_U^{\mathbf{x}^{(i)}} \mathbf{x}_U^{(i)} + B \mathbf{x}_M\right]$. Hence, we see that since $L_U^{\mathbf{x}^{(i)}}$ is positive definite, $(d\mathbf{x}^{(i)})^\top \frac{\partial E(\mathbf{x})}{\partial \mathbf{x}_U} = -\frac{\partial E(\mathbf{x})}{\partial \mathbf{x}_U}^\top \left(L_U^{\mathbf{x}^{(i)}}\right)^{-1} \frac{\partial E(\mathbf{x})}{\partial \mathbf{x}_U} < 0$, when $\frac{\partial E(\mathbf{x})}{\partial \mathbf{x}_U} \neq \mathbf{0}$. We therefore conclude that the update direction $d\mathbf{x}^{(i)}$ is a descent direction at $\mathbf{x}^{(i)}$. Since the energy function is bounded below by a unique minimum, our algorithm will converge to the correct solution. Notice that we choose a tolerance parameter δ , in order to regulate how close one wants to be to the true solution.

4.3. Automatic Selection of Asymmetric Penalty Schemes

So far, we have introduced our framework that extends the Random Walker algorithm to graphs with directed edges. We shall now discuss how the weights of these edges should be defined so that our proposed algorithm segments the object reliably. For this purpose, we devise a machinery that biases the direction of the object boundary gradients to

flow from regions which agree well with a learned intensity model of the object to regions that do not agree well.

More specifically, given the object/background seeds, which are interactively provided by a user, we build intensity models for the object as well as the background. A simple example of such models is a histogram of the seeds' intensities. These intensity models are then used to estimate the posterior probability P_i of each node $\mathbf{v}_i \in \mathcal{V}$ belonging to the object or not. We hence have an image whose intensity values are given by these posterior probabilities $\{P_i\}_{\mathbf{v}_i \in \mathcal{V}}$. Note that this is a representative intermediate segmentation of the image and *not* the final segmentation. The final segmentation is obtained by applying Algorithm 1 to this image of posterior probabilities.

We set the edge weights such that a decrease in the posterior probabilities across an edge is penalized more than an increase in the probabilities. For example, consider a pair of neighboring nodes \mathbf{v}_i and \mathbf{v}_j and assume without loss of generality that $P_i > P_j$. We define the weights of the directed edge as $w_{ij} = e^{-\beta_{hl}(P_i - P_j)^2}$ and $w_{ji} = e^{-\beta_{lh}(P_i - P_j)^2}$ for some $\beta_{hl} \geq \beta_{lh} \geq 0$. This scheme of choosing the edge weights helps in directing the boundary gradients to flow from pixels with a high probability of belonging to the object to pixels with a low probability.

Note that one can always apply Algorithm 1 to segment the original image rather than the image of probabilities, by manually deciding the asymmetric penalty scheme. The aforementioned scheme serves the purpose of increasing the generality of our interactive segmentation algorithm so that the intensity gradients to be favored do not need to be known in advance. Note that this penalty scheme is not specific to our algorithm and could easily be incorporated into other algorithms which make use of gradient directionality.

5. Results

In this section, we compare the segmentation results of our proposed framework with those of the Random Walker algorithm. We present qualitative as well as quantitative results on a database of 69 medical images. In particular, we try to show that introducing directional information improves the Random Walker algorithm's performance.

In the results, the seeds are displayed in red for the object and in green for the background. The segmentation boundary estimated by the algorithms is shown in blue. In all the experiments involving Algorithm 1, we use $\delta = 10^{-4}$ and set the value of ϵ to the machine precision. We use a 4 connected graph for our experiments and all the remaining edges are assigned zero weights.

Note that in the qualitative results we use Algorithm 1 to segment the original image rather than the image of posterior probabilities. Consequently, we manually decided whether we wanted to place a harder penalty on the increase

or decrease in intensity values, rather than adopting our proposed automatic scheme. The purpose of these examples is to analyze the advantage of introducing directional information in the classical Random Walker algorithm. In the quantitative results, we use the automatic scheme for setting the weights and operate on the image of posterior probabilities.

5.1. Qualitative Comparison on Synthetic Data

Figure 2 compares the results of segmenting synthetic images using the Random Walker algorithm with symmetric as well as asymmetric penalties. These images are designed such that the user is faced with multiple possibilities for the segmentation boundary.

The first image contains a grey disc ($I = 0.5$) against a black background ($I = 0$). In such an image, one would like to have the ability to place the object boundary at the inner or outer circle of the disc. Note that the classical Random Walker algorithm places the segmentation boundary at the inner circle. The second image of size 256×256 is similar in nature and contains a grey bar ($I = 0.5$) against a black background ($I = 0$). As motivated earlier, one would like to be able to place the object boundary at either edge of the bar. The classical Random Walker algorithm, however, always places the segmentation boundary at the center of the image due to symmetry of the seeds with respect to the image. In both cases, we see that the introduction of directional information in the Random Walker algorithm can be used to obtain the desired segmentations.

5.2. Qualitative Comparison on Real Data

Figure 3 presents a qualitative comparison of the performance of the classical Random Walker algorithm on real

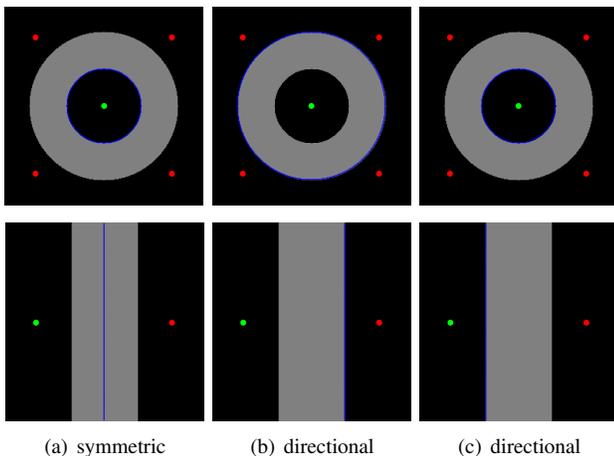


Figure 2. Results on synthetic images: Column(a) shows the results obtained using symmetric penalties for increases and decreases in intensities. Column (b) shows the results obtained when we place higher penalties for decrease in intensities and column (c) indicates results with higher penalties for increase in intensities.

medical images with that of our proposed framework. We use $\beta = 90$ to set the symmetric penalties for the Random Walker algorithm, and $\beta_{hl} = 90$ and $\beta_{lh} = 10$ to define asymmetric penalties. For the first two images, we place harsher penalties on increases in intensity. For the last two images, we place harsher penalties on decreases in intensity.

In most images, the image contrast at the object boundary is not very high, thereby causing the classical Random Walker algorithm to place erroneous boundaries at the sharpest edges close to the true boundary. In addition, there are cases when the effect of the seeds is restricted by the Random Walker algorithm to a neighborhood that is smaller than we would like. This behavior is due to the presence of sharp edges close to the seeds. We see that our proposed framework is able to get better segmentations and obtain a more accurate boundary, as is reflected in the images.

5.3. Quantitative Comparison on Real Data

We evaluated our algorithm on 69 3-D medical datasets containing a single target that was manually segmented by a clinical practitioner. Each volume also contains manually-placed foreground and background seeds by the same clinical

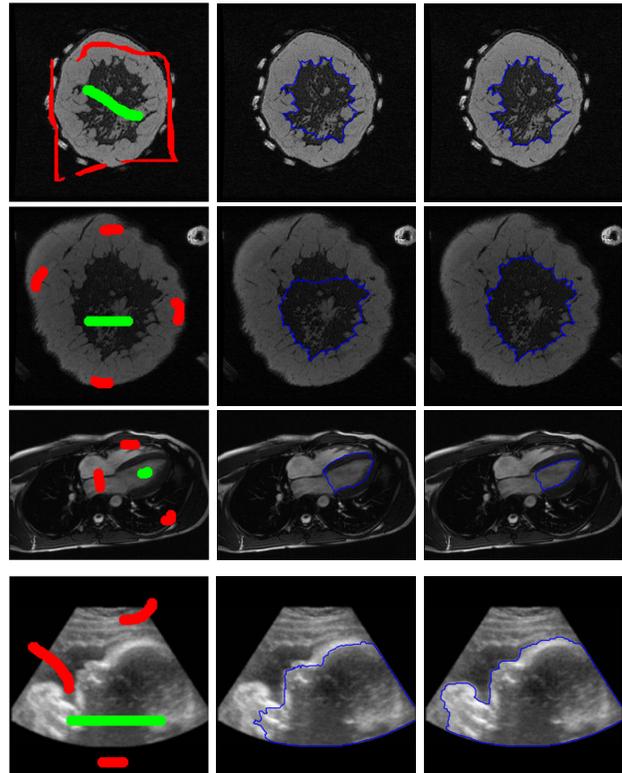


Figure 3. Our extension of the Random Walker algorithm that introduces directional nature can be used to improve the segmentation results given by the classical Random Walker algorithm.

cal practitioner that provided the manual segmentation. The data was acquired from a Siemens computed tomography (CT) scanner and contained a range of segmentation targets including tumors, lymph nodes, cysts and other lesions. The data was acquired using different scanners, with different reconstruction kernels and the clinical input (ground truth and seeds) was given by different clinical partners. Therefore, our results should not be biased by the details of a particular acquisition protocol or clinical individual. The datasets we used for segmentation were typically cropped from larger data acquisitions and ranged in size from roughly $40 \times 40 \times 40$ to $128 \times 128 \times 128$. We use the seeds contained in the datasets in order to find the segmentation using the Random Walker algorithm and our proposed method. For the classical Random Walker algorithm, we set the symmetric penalties using $\beta = 50$ across all the trials. We use the parameters $\beta_{hl} = 50$ and $\beta_{lh} = 12.25$ to define the asymmetric penalties in our proposed algorithm.

Let $G(v_i)$ and $S(v_i)$ denote the ground truth segmentation and the estimated segmentation for a node $v_i \in \mathcal{V}$. We used the metrics *normalized volume difference* (ν) and *normalized object overlap* (ρ) as defined in (12) and (13) to evaluate the quality of segmentation. In general, a good segmentation result is marked by a low value for ν and a high value for ρ . Table 2 shows the statistics for these metrics obtained after using both the algorithms to segment the volumes in the database.

Normalized volume difference :

$$\nu = \frac{|\sum_{v_i \in \mathcal{V}} G(v_i) - \sum_{v_i \in \mathcal{V}} S(v_i)|}{\sum_{v_i \in \mathcal{V}} G(v_i)}. \quad (12)$$

Normalized object overlap :

$$\rho = \frac{\sum_{v_i \in \mathcal{V}} G(v_i) \cdot S(v_i)}{\sum_{v_i \in \mathcal{V}} G(v_i) + S(v_i) - G(v_i) \cdot S(v_i)}. \quad (13)$$

We see that our proposed framework gives improvements on the order of 0.1% – 0.2% in these metrics. While these improvements are not necessarily significant, we argue that the seeds marked by the medical practitioners are good enough to provide accurate segmentations using the Random Walker algorithm.

Method	ν	ρ
Random Walker	27.34% (mean)	63.92% (mean)
(symmetric)	14.74% (median)	68.76% (median)
Random Walker	26.98% (mean)	64.06% (mean)
(directional)	14.29% (median)	68.83% (median)

Table 2. Quantitative comparison of the classical Random Walker algorithm with our proposed framework that introduces directional nature in the Random Walker algorithm.

6. Conclusions

We have presented an interactive image segmentation algorithm that extends the Random Walker algorithm to operate on combinatorial graphs with directed edges. The segmentation problem is posed as the minimization of a quadratic energy function defined on this graph. The proposed energy has a unique minimizer, which can be estimated using an iterative scheme proposed by us. Our scheme typically requires just 4–5 iterations to converge to the solution, with a Random Walker like optimization being solved at each iteration. Comprehensive testing shows that the introduction of directional nature in the Random Walker algorithm helps improve its segmentation results.

7. Acknowledgements

Dheeraj Singaraju would like to thank Dr. Gareth Funka-Lea for giving him the opportunity to intern at Siemens Corporate Research, and making this work possible.

References

- [1] Y. Boykov and G. Funka-Lea. Graph Cuts and Efficient N-D Image Segmentation. *IJCV*, 70(2):109–131, 2006.
- [2] V. Caselles, R. Kimmel, and G. Sapiro. Geodesic Active Contours. *IJCV*, 22(1):61–79, 1997.
- [3] P. Doyle and L. Snell. *Random Walks and Electric Networks*. Number 22 in The Carus Mathematical Monographs. The Mathematical Society of America, 1984.
- [4] L. Grady. Random Walks for Image Segmentation. *IEEE Trans. on PAMI*, 28(11):1768–1783, November 2006.
- [5] M. Holtzman-Gazit, R. Kimmel, N. Peled, and D. Goldsher. Segmentation of Thin Structures in Volumetric Medical Images. *IEEE Trans. on Image Processing*, 15(2):354–363, 2006.
- [6] V. Kolmogorov and Y. Boykov. What Metrics Can Be Approximated by Geo-Cuts, or Global Optimization of Length/Area and Flux. In *ICCV*, volume 1, pages 564–571, 2005.
- [7] V. Kolmogorov and R. Zabih. What Energy Functions Can Be Minimized Via Graph Cuts? *IEEE Trans. on PAMI*, 26(2):147–159, 2004.
- [8] V. Lempitsky, Y. Boykov, and D. Ivanov. Oriented Visibility for Multiview Reconstruction. In *ECCV*, volume 3 of *LNCS* 3953, pages 226–238, May 2006.
- [9] R. Malladi, J. Sethian, and B. C. Vemuri. Shape Modeling with Front Propagation: A Level Set Approach. *IEEE Trans. on PAMI*, 17(2):158–175, Feb. 1995.
- [10] A. K. Sinop and L. Grady. A Seeded Image Segmentation Framework Unifying Graph Cuts and Random Walker Which Yields A New Algorithm. In *ICCV*, 2007.
- [11] A. Vasilevskiy and K. Siddiqi. Flux Maximizing Geometric Flows. *IEEE Trans. on PAMI*, 24(12):1565–1578, 2002.