Multilabel Random Walker Image Segmentation Using Prior Models [*]

Leo Grady
Imaging and Visualization Department
Siemens Corporate Research
Princeton, NJ 08540
leo.grady@siemens.com

Abstract

The recently introduced random walker segmentation algorithm of [14] has been shown to have desirable theoretical properties and to perform well on a wide variety of images in practice. However, this algorithm requires user-specified labels and produces a segmentation where each segment is connected to a labeled pixel. We show that incorporation of a nonparametric probability density model allows for an extended random walkers algorithm that can locate disconnected objects and does not require user-specified labels. Finally, we show that this formulation leads to a deep connection with the popular graph cuts method of [8, 24].

1 Introduction

Traditional machine learning and statistical pattern recognition systems [15, 3] typically make decisions about test points without making use of their relationship to each other. Due to the spatial nature of an image, such approaches are often inappropriate and their application results in noisy segmentations with many small, fragmented, pieces of an object scattered through the image. This is why the computer vision community has trended toward spatial algorithms, such as normalized cuts [22], graph cuts [8], watersheds [20], active contours [16], level sets [21] and the random walker algorithm [14]. However, in some segmentation scenarios, the objects of interest may be reasonably characterized by an intensity (feature) distribution. For such a situation, it is important to be able to integrate intensity information into a spatial algorithm.

The recently introduced random walker segmentation algorithm of [14] has been shown to have desirable theoretical properties and perform well on a wide variety of images in practice. The algorithm was designed to be a general-purpose interactive segmentation tool, such that a user could mark a few pixels with an arbitrary number of labels and expect a quality result, regardless of the data set or the segmentation goal. Using this technique, a segmentation is obtained for a pixel by computing, for each label, the probability that a random walker starting its walk at that pixel first reaches a seed with that label. The pixel is then assigned the label with the greatest probability. A user-specified **seed** is a pixel that has been given a labeling by the user. It was shown in [14] that the probabilities may be computed analytically by solving a sparse, symmetric, positive-definite, system of linear equations instead of performing a random walk simulation.

Specifically, it was shown in [14] that the random walker algorithm has the following properties:

1. The solution for the probabilities is unique.
2. The expected value of the probabilities for an image of pure noise, given by identically distributed (not necessarily independent) random variables, is equal to those obtained on a uniform image.
3. The expected value of the probabilities in the presence of random, uncorrelated weights is equal to the probabilities obtained by using weights equal to the mean of each random variable.

However, this algorithm has three properties that could be problematic for certain segmentation tasks: each segment must be connected to a seed, only intensity *gradients* were used instead of employing absolute intensity information, and the algorithm requires user-specified seeds. For many segmentation tasks, these properties are desirable. Ignoring absolute intensity information increases robustness to quantization, shifted or inverted intensities and requiring connectedness of each segment to a seed prevents a noisy, scattered segmentation of small regions. However, segmentation of an image containing objects of interest that have a great many disconnected pieces is tedious for the user,

[*]Published in: Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Vol. 1, p. 763–770, June, San Diego

since a seed must be placed inside each disconnected piece. If a consistent intensity profile characterizes an object of interest, then this information should be incorporated into the segmentation. Finally, we may have an object intensity model, but user input is unavailable. For these cases, the present work describes an *extended random walker* algorithm that uses an intensity model obtained either *a priori* or via a density estimation from user-input seeds. For clarity and simplicity of exposition, we develop the present extension in the context of images with a single channel (hereafter referred to as *intensity*) and user-specified seeds. However, we stress that the algorithm applies equally to multichannel image and an absence of user interaction (assuming a model is available).

Figure 1 illustrates the goal of this work — joining an aspatial algorithm (represented by the density estimation) with the spatial random walkers algorithm. The user has supplied two groups of seeds, representing the "blood cell" label and the "background" label. Applying the random walkers algorithm yields a correct segmentation of the cell within which the seeds were placed, but incorrectly identifies the other cells. In contrast, using a simple density estimation of the two groups finds pieces of the cells and background, but ultimately yields a fractured segmentation that lacks spatial cohesion. Our goal is to combine the intensity profiling and long-range aspects of the density estimation approach with the spatial cohesion of the random walker algorithm in a principled way that produces the correct result, despite variability of the intensity values present in the image.

The mixing of statistical information into a spatial approach in not new in the computer vision literature. For energy methods of segmentation, this effect is often achieved naturally by adding energy terms and performing minimization on the total energy (e.g., [18]). However, some spatial algorithms, such as the watershed transform [20], do not easily admit the incorporation of image priors. The novelty in this work is to extend the success of the random walker approach by employing image priors to find disconnected pieces of an object and to remove the necessity of user interaction.

The most closely related approach to the present work is the graph cuts method of [9, 8] with a "data term" representing the priors. The graph cuts algorithm may also be used to find a minimum cut between user-specified seed groups [7]. As will be discussed in further detail below, the random walker and graph cuts algorithms obtain a segmentation through minimization of the same functional, with the difference that the random walker algorithm minimizes the functional over the space of real numbers and the graph cuts algorithm performs the minimization over the set of integers. Although this difference might appear slight at first, the segmentations obtained from the two algorithms have different properties and may behave differently on the same image. Specifically, the random walkers algorithm has provable robustness to noise, extends easily (and exactly) to an arbitrary number of labels and offers a confidence value that a given node belongs to a particular segment (as represented by the probability). Furthermore, as noted by Shi and Malik [22], minimum cuts has a tendency to find a small cut. The reason for this is that graph cuts will find the smallest cut between the seeds (terminals), resulting in a tendency to find the cut that barely encloses the seeds in situations where the desired boundary is weakly defined or few seeds are placed. Since the random walker algorithm is not seeking the smallest boundary, it does not suffer from this "small cut" problem. However, the graph cut algorithm is guaranteed to give the minimum cut between two groups of labeled nodes.

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). Formulation of the algorithm on a graph allows the application of the algorithm to surface meshes or space-variant images [23, 13]. 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.

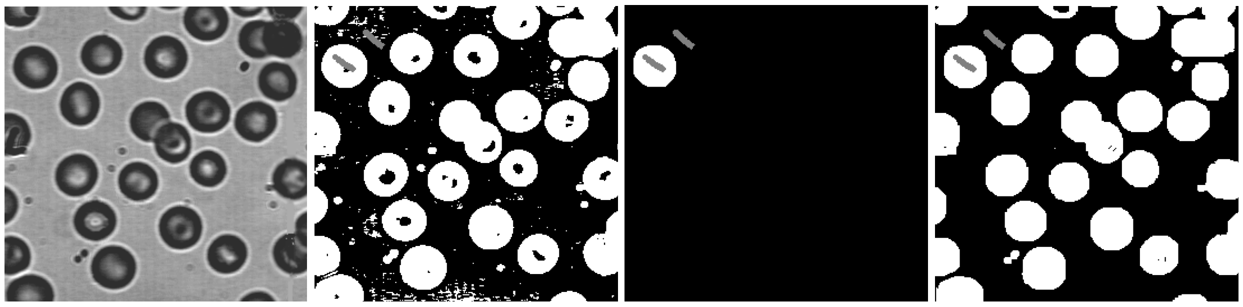
The random walker formulation will first be reviewed and then extended to incorporate intensity priors. We then discuss connections to the graph cuts algorithm and provide implementation details of the algorithm. Results are displayed for several images and we conclude with a discussion.

2 Development

The segmentation is formulated on a weighted graph, where each node represents a pixel or voxel. A **graph** is a pair $G = (V, E)$ with vertices $v \in V$ 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}. Let $n = |V|$ and $m = |E|$ where $|\cdot|$ denotes cardinality. A **weighted graph** has a value (typically nonnegative and real) assigned to each edge called a **weight**. The weight of 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.

2.1 Review of random walker formulation

Given a weighted graph, a set of marked (labeled) nodes, V_M, and a set of unmarked nodes, V_U, such that $V_M \cup V_U = V$ and $V_M \cap V_U = \emptyset$, we would like to label each node



(a) Original image (b) Density estimation only (c) Random walkers only (d) RW with density est.

Figure 1. Gray markers indicate user-specified seeds specifying cells and background. In the output, white regions correspond to pixels determined to be "cell" and black regions to pixels determined to be "background". The random walkers algorithm by itself is capable of finding a spatially coherent object, but unable to locate similar, disconnected, pieces. A simple density estimation classifies disconnected pixels, but yields a fragmented, spatially insensitive, segmentation. Use of density estimation as a set of priors to the random walker algorithm results in a segmentation that overcomes the limitation of both individual approaches. This image was processed using the parameters: $\boldsymbol{\beta = 500}$, $\boldsymbol{\gamma = 1e^{-2}}$, $\boldsymbol{\sigma = 100}$.

$v_i \in V_U$ with a label from the set $G = \{g^1, g^2, \dots, g^k\}$ having cardinality $k = |G|$. We term a node, $v_i \in V_U$, as **free** because its label is not initially known. Assume that each node $v_i \in V_M$ has also been assigned a label, $y_i \in G$. The random walker approach to this problem given in [14] is to assign to each node, $v_i \in V_U$, the probability, x_i^s, that a random walker starting from that node first reaches a marked node, $v_j \in V_M$, assigned to label g^s. The segmentation is then completed by assigning each free node to the label for which it has the highest probability, i.e., $y_i = \max_s v_i^s$. Note that the values for y_i, if $v_i \in V_M$, are given by user-interaction.

It is known [14, 11] that the minimization of

$$E_{\text{spatial}} = x^{sT} L x^s, \tag{1}$$

for an $n \times 1$, real-valued, vector, x^s, defined over the set of nodes (i.e., a cochain) yields the probability, x_i^s, that a random walker starting from node, v_i, first reaches a node $v_j \in V_M$ with label g^s (set to $x_j^s = 1$), as opposed to first reaching a node, $v_j \in V_M$, with label $g^{q \neq s}$ (set to $x_j = 0$), where L represents the combinatorial Laplacian matrix [17] defined as

$$L_{v_i v_j} = \begin{cases} d_{v_i} & \text{if } i = j, \\ -w_{ij} & \text{if } v_i \text{ and } v_j \text{ are adjacent nodes}, \\ 0 & \text{otherwise}. \end{cases} \tag{2}$$

The notation $L_{v_i v_j}$ is used to indicate that the matrix L is indexed by vertices v_i and v_j.

By partitioning the Laplacian matrix into marked (i.e., pre-labeled) and unmarked (i.e., free) blocks

$$L = \begin{bmatrix} L_M & B \\ B^T & L_U \end{bmatrix}, \tag{3}$$

and denoting an $|V_M| \times 1$ indicator vector, f^s, as

$$f_j^s = \begin{cases} 1 & \text{if } y_j = g^s, \\ 0 & \text{if } y_j \neq g^s, \end{cases} \tag{4}$$

the minimization of (1) with respect to x_U^s is given by the system

$$L_U x_U^s = -B f^s, \tag{5}$$

which is a sparse, symmetric, positive-definite, system of linear equations. By virtue of x_i being a probability

$$\sum_s x_i^s = 1 \ \ \forall i, \tag{6}$$

only $k - 1$ linear systems must be solved, since the final system may calculated for simply via (6). Note that (6) may also be derived directly or by recourse to superposition in a circuit analogy [14]. Although we are using random walkers for our segmentation, the deep connection with potential theory allows us to simply and, more importantly, deterministically, solve a system of linear equations to find the probabilities. The mathematics above reveal one property of the random walker algorithm: In the absence of labeled points (i.e., $V_M = \emptyset$), the probabilities are undefined. Therefore, in the original work, this algorithm was presented as a strictly semi-automated segmentation algorithm. We will now present how the incorporation of priors into the above framework yields a segmentation algorithm that need not have any user interaction.

2.2 Label priors

Assume we have a set of real-valued, nodewise priors, λ_i^s, that represent the probability density that the intensity at node, v_i, belongs to the intensity distribution of label g^s. Assuming that each label is equally likely, Bayes' theorem gives the probability that a node, v_i, belongs to label g^s as

$$x_i^s = \frac{\lambda_i^s}{\sum_{q=1}^k \lambda_i^q}. \tag{7}$$

We may write (7) in vector notation as

$$\left(\sum_{q=1}^k \Lambda^q\right) x^s = \lambda^s, \tag{8}$$

where Λ^s is understood to be a diagonal matrix with the values of λ^s on the diagonal. In MATLAB notation, $\Lambda = \mathrm{diag}\,(\lambda^s)$.

It is clear that (8) is the minimum energy distribution for the **aspatial functional**[1]

$$E^s_{\mathrm{aspatial}}(x^s) = \sum_{q=1,q\neq s}^k x^{qT}\Lambda^q x^q + (x^s-1)^T \Lambda^s (x^s-1)\,. \tag{9}$$

These energies may be combined into a single functional with the introduction of a free parameter γ as

$$E^s_{\mathrm{Total}} = E^s_{\mathrm{spatial}} + \gamma E^s_{\mathrm{aspatial}}, \tag{10}$$

and minimized with respect to the free (i.e., not pre-labeled) nodal probabilities. Without loss of generality, assume for the moment that there are no pre-labeled nodes (i.e., all x_i are free). The minimum energy of (10) is obtained when x^s satisfies the solution to[2]

$$\left(L + \gamma\sum_{r=1}^k \Lambda^r\right) x^s = \lambda^s. \tag{11}$$

Note that, despite the singularity of L [2], the combined matrix in (11) is guaranteed to be positive definite (and therefore nonsingular), since L is positive semi-definite and the diagonal matrices are strictly positive definite. For this reason, the incorporation of priors into the formulation circumvents the problems associated with equation (5) that demanded user-specified labels in the original work. However, if desired, user-specified seeds may also be incorporated by solving the system[3]

$$\left(L_U + \gamma\sum_{r=1}^k \Lambda^r_U\right) x^s_U = \lambda^s_U + Bf^s, \tag{12}$$

for the unlabeled nodes V_U.

Compare (11) for a lattice with known priors and (5) for a lattice modified to include an extra (labeled) node for each label, g^s, that is connected to each node in the lattice, v_i, with weight equal to $\gamma\lambda_i^s$, as depicted in Figure 2. For these cases, (11) and (5) are the same with $Bf^s = \lambda^s$ and L_U is simply the Laplacian for the lattice. The additions to the diagonal of L representing the lattice in (11) is residue from the deletion of the marked (floating) nodes. Therefore, the incorporation of priors (in (11)) yields the same solution as would be obtained for the random walker probabilities on an augmented graph. It is more convenient to consider the augmented graph, since the inclusion of priors may now be treated in the original, random walker, framework. Specifically, the proofs given in [14] concerning the robustness and behavior of the random walker algorithm also apply when priors are included, since the inclusion of priors is equivalent to the original random walker problem solved on the augmented graph of Figure 2. Note that use of pre-labeled nodes with priors also leads to solving (5) on the augmented graph. To be clear, the solution obtained through incorporation of priors into the total energy of (10) is equivalent to the solution to just the original random walkers algorithm on the augmented graph in Figure 2.

We note that, because of the identification of the minimization of (10) with the general random walker problem in [14], we are guaranteed that the solutions, x_i^s, found for each label, g^s, sum to unity as required by a probability, i.e., $\sum_s x_i^s = 1$. The unity constraint for each node indicates that only $k-1$ solutions to (11) are required.

The construction of a graph with additional nodes, as depicted in Figure 2, as well as the development in this section, bears a close resemblance to the construction of the graph cuts problem with the inclusion of nodewise priors as detailed in [1, 9, 8]. In the terminology of [8], our weights w_{ij} between nodes v_i and v_j are attached to *N-links* and the $\gamma\lambda_i^s$ are attached to *T-links*. There are, however, three notable differences between these two formalisms. First, the algorithm described in the present work minimizes the energy in (10) over the field of real-valued probabilities instead of binary values. Although this may appear to be an inconsequential difference, a unique solution is guaranteed for a real-valued solution (since the matrix in (11) is nonsingular), while it is not guaranteed for a binary so-

[1] 10/19/09 LJG: This equation is reproduced as printed, but should instead read $E^s_{\mathrm{aspatial}}(x^s) = \sum_{q=1,q\neq s}^k x^{sT}\Lambda^q x^s + (x^s-1)^T \Lambda^s (x^s-1)$. Thanks to Rui Shen for calling this to my attention.

[2] 11/13/07 LJG: This equation is reproduced as printed, but should instead read $\left(L + \gamma\sum_{r=1}^k \Lambda^r\right) x^s = \gamma\lambda^s$. Thanks to Sebastian Nowozin for calling this to my attention.

[3] 11/13/07 LJG: This equation is reproduced as printed, but should instead read $\left(L_U + \gamma\sum_{r=1}^k \Lambda^r_U\right) x^s_U = \gamma\lambda^s_U - Bf^s$. Thanks to Sebastian Nowozin for calling this to my attention.

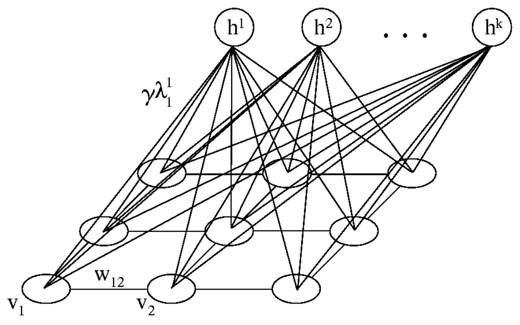


Figure 2. Mathematically, the use of intensity priors is equivalent to using k labeled, "floating" nodes that correspond to each label and are connected to each node. Note the similarity between this construction and that of the graph cuts algorithm in [9, 8]. See text for a comparison.

lution. Another difference is that, as shown in [8], using graph cuts to minimize the energy with more than two labels does not guarantee an exact minimum, only a solution within a bound of the true minimum. In contrast, multiple labels are handled naturally in the present formulation, yielding the desired probabilities exactly regardless of the number of labels that are used. Finally, with graph cuts, the T-link weights are typically taken as $\log(\lambda)$ due to the original MAP formulation of the problem [1, 9, 6] as opposed to the sum of energy terms derivation that is employed in the present work.

3 Algorithmic details

The algorithm may be described in four steps which will be detailed in this section:

1. With a prior model of label intensities (obtained possibly through estimation from user interaction), generate the probability, λ_i^s, that each node, v_i, belongs to each label, g^s, based on its intensity, I_i.

2. Generate edge weights, w_{ij}, between connected nodes v_i and v_j.

3. Solve the system of equations defined in (12) for the probabilities, x_i^s, corresponding to each label, g^s. This need only be done $k-1$ times, since the final set of probabilities may be calculated via the unity sum condition, i.e., $x_i^k = 1 - \sum_{s<k} x_i^s$.

4. Assign each node, v_i, to the label, g^s, with highest probability, x_i^s, i.e., $y_i = \max_s(x_i^s)$.

3.1 Prior model

Segmentation tasks are often specified in the context of a particular problem domain. In such a situation, the number of desired labels is known *a priori* and a probability density estimation may be generated from training (i.e., prelabeled) images via a wide array of techniques [3].

Although many advanced techniques are available for density estimation, we used a simple kernel estimation to produce the probability densities. Assume that we have a set of training nodes, with intensities denoted $T = \{t_1, t_2, \ldots, t_c\}$, and corresponding labels, denoted $R = \{r_1, r_2, \ldots, r_c\}$, where $c = |T| = |R|$ is the number of training points and $r_i \in G$. Note that the training points may be defined with prior training images, or may be given interactively by a user. Specifically, we quantized each image intensity to 256 levels (to preserve reuse of parameters across image modalities) and used a Gaussian kernel to produce the densities corresponding to each of the k labels.

The probability, λ_i^s, that node v_i is generated from the distribution corresponding to label g^s is generated through

$$\lambda_i^s = \frac{1}{Z^s} \sum_{q, r_q = g^j} \mathrm{e}^{\frac{(I_i - t_q)^2}{\sigma}}, \tag{13}$$

where σ is a free parameter and Z^s is a normalizing constant for label g^s equal to

$$Z^s = \sum_{p=0}^{255} \sum_{q, r_q = g^s} \mathrm{e}^{\frac{(p - t_q)^2}{\sigma}}. \tag{14}$$

In practice, a normalized histogram is generated from the Gaussian kernels for each label and the probabilities are simply read off for each test intensity. Estimating densities for multidimensional images and a large number of training points would require more advanced methods of density estimation in order to be practical.

Of course, an intensity profile may not be the most appropriate descriptor for some segmentation tasks. We focus here on an intensity description purely for the simplicity of algorithm exposition — generating distributions based on a texture analysis or filtering may be better suited for some segmentation problems.

3.2 Choosing weights

Although several functions exist for mapping nodal intensities to connecting weights (e.g., see [4] for an excellent treatment), we chose the ubiquitous function [19]

$$w_{ij} = \mathrm{e}^{-\beta(I_i - I_j)^2}. \tag{15}$$

In practice, we employ

$$w_{ij} = \mathrm{e}^{\frac{\beta}{\rho}(I_i - I_j)^2} + \epsilon, \tag{16}$$

where ϵ is a small constant (we take $\epsilon = 10^{-6}$) and ρ is a normalizing constant $\rho = \max(I_i - I_j), \forall i, j$. The purpose of (16) is to keep the choice of β relevant to images of different quantization and contrast, as well as make sure that none of the weights go identically to zero.

3.3 Numerical solution

As with the original random walker algorithm [14], the main computational hurdle in the extended version described in this work is the solution to the large, sparse, symmetric, positive-definite, system of linear equations in (12). Many methods exist to solve a system of equations [12], although the high memory consumption of most direct methods (e.g., LU decomposition) precludes practical use in this situation, except in the case of small images. Instead, iterative methods such as preconditioned conjugate gradient are more appropriate, due to an acceptable memory consumption and easy parallelization [10, 5].

If the prior distributions were uniform, then $\gamma \sum_{s=1}^{k} \Lambda^s = \frac{\gamma k}{n} I$ and the addition of this matrix to L (or L_M, if user-defined seeds are used) would be guaranteed to reduce the (Euclidean) condition number that is known to affect the convergence of conjugate gradients [12]. Although the densities will not, in general, be uniform, we have noticed an empirical improvement in the convergence of the conjugate gradient method when applied to (12) over (5).

3.4 Computational complexity

Given a fixed quantization, and a number of training points and labels that are independent of the number of pixels, the density estimation is a constant time operation and the subsequent assignment of a probability to each node is $\mathcal{O}(n)$.

Solving the system of linear equations defined in (12) is the main computational hurdle of the algorithm and, in practice, requires the most time. However, if we employ a graph of bounded degree, d, on the unlabeled nodes (e.g., $d = 4$ for a 4-connected lattice), then the sparse matrix multiply employed in each iteration of conjugate gradients requires no greater than dn operations. Consequently, if we assume that a fixed number of iterations are employed, then the solution to the system of equations is performed in $\mathcal{O}(n)$ operations. Because each phase of the algorithm, including the initial weight assignment and final pixelwise maximum-likelihood segmentation, are linear time operations, the entire algorithm requires $\mathcal{O}(n)$ operations.

In practice, the entire algorithm (from initial prior and weight/matrix generation to final label assignment) requires approximately 3 seconds on a 256×256 image for an Intel Xeon 2.40GHz with 3GB of RAM.

4 Results

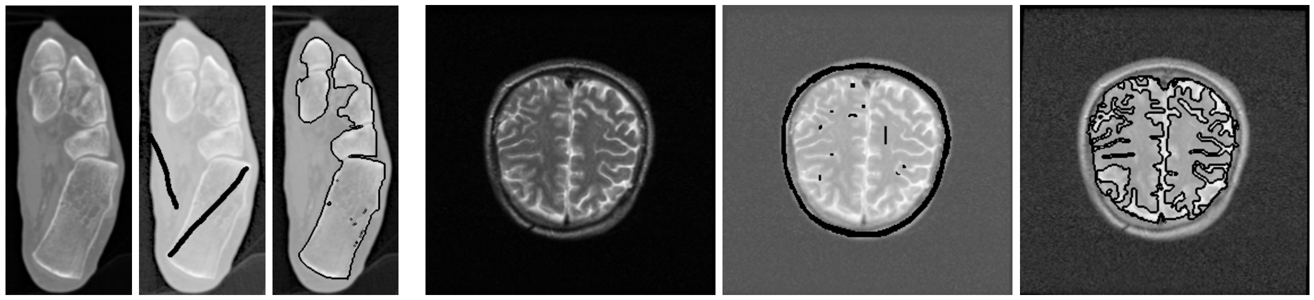
This extension of the work in [14] is best suited to problems with disconnected objects, where the intensity profile largely characterizes the desired objects, but the presence of noise or irregularities results in a need to respect spatial cohesion. A histological example of this type of segmentation task was given above in Figure 1 where the user would prefer to label only one cell and the background. Although the irregularities in the background and within the cells would cause a priors-only solution to yield a noisy, fragmented, segmentation with many errors, use of the spatial energy term from [14] yields a quality segmentation. Figure 3 shows four examples of medical images that have been segmentated with our extended random walker algorithm. All of the segmentations in this paper are done using the same values for the parameters. In each of the images in Figure 3, the labeled object is disconnected, preventing straightforward application of [14]. However, a density estimation of the intensity is also insufficient to characterize the objects without introducing significant noise into the final segmentation. Although Figure 3 displays results on medical images, there is nothing inherent about this algorithm that pertains specifically to medical images. In other words, this algorithm, both the original random walkers and this extension, is a general purpose segmentation tool that makes no assumptions about the image type or the segmentation task.

5 Conclusion

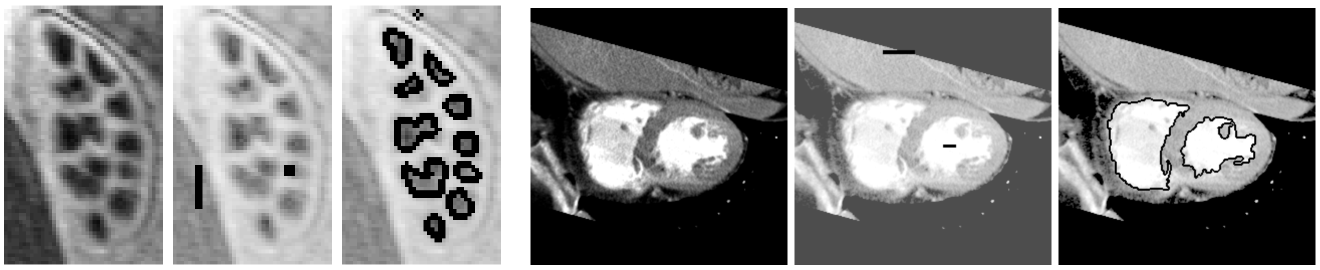
Although producing high-quality segmentations on a wide variety of images, the random walker segmentation algorithm first presented in [14] has three properties that could be problematic for certain segmentation tasks: each segment must be connected to a seed, only intensity *gradients* were used instead of employing absolute intensity information, and the algorithm requires user-specified seeds. We have shown in this work that the incorporation of a prior model into the energy minimization yields an extended algorithm that overcomes these problems. Specifically, the incorporation of priors is most beneficial when the image consists of many objects bearing a single label that have similar intensity profiles.

A pixelwise prior model alone is frequently incapable of producing a quality segmentation, leading instead to a noisy, fragmented, solution. Without use of spatial cohesion, such an approach results in many tiny, often one or two pixel, pieces of a segment. Therefore, addition of the spatial energy term also lends strength to a pointwise, Bayesian, generative approach to segmentation.

Combination of the aspatial (priors) energy term and the spatial (random walkers) energy term was shown to yield



(a) Ankle (b) Seeds (c) Seg. (d) MR Grey/White matter (e) Seeds (f) Segments



(g) Kidney (h) Seeds (i) Segments (j) CT Cardiac blood pool (k) Seeds (l) Segments

Figure 3. The user-specified labels are indicated by black lines in the middle figure labeled “seeds”. The black lines in the “segments” figure indicate segment boundaries. Each image had two seed groups, except for the gray/white matter segmentation, which has three groups (gray matter, white matter, background). All of the seeds inside the brain are gray matter seeds except for the vertical line in the center of the white matter. Each image was processed using the same parameters: $\beta = 500$, $\gamma = 1e^{-2}$, $\sigma = 100$

a functional that may be interpreted in the framework of the random walkers algorithm alone. Namely, each label is viewed as a "floating" seed node that is connected to every other node with weight corresponding to the prior density of the pixel intensity to that label, as illustrated in Figure 2. Therefore, the extended algorithm presented here may be interpreted in the original random walker formalism with a simple modification of the graph itself. This realization allows the proofs in [14] on noise robustness, to be applied without modification to the priors extension. Furthermore, the graph construction in Figure 2 exactly mirrors the graph construction employed in the graph cuts algorithm [9, 8] when a "data term" is employed. This close connection is not surprising, since the functional minimized to find the random walker probabilities is identical to the functional minimized to find the smallest graph cut, with the difference that the minimization in the case of the random walkers algorithm is performed over the field of real numbers instead of binary values. This difference between the two algorithms, optimization over the reals instead of (binary) integers, may seem subtle and inconsequential, but it is shown in [14] that the two algorithms exhibit different properties. Specifically, the random walkers algorithm does not suffer from the "small cut" problem of graph cuts, has provable robustness to noise, extends easily (and exactly) to an arbitrary number of labels and yields a probability that a given node belongs to a segment. In contrast, the graph cut algorithm is guaranteed to give the minimum cut between the labeled nodes.

Future work includes progression beyond a simple intensity-based prior and more advanced density estimation. Given the similarity in construction between this extended random walker algorithm and the graph cuts algorithm with a data term of [9, 8], a natural course would be to apply this algorithm to the problems tackled with that algorithm, namely image restoration and stereo segmentation.

References

[1] J. Besag. On the statistical analysis of dirty pictures. *Journal of the Royal Statistical Society, Series B*, 48(3):259–302, 1986.

[2] N. Biggs. Algebraic potential theory on graphs. *Bulletin of London Mathematics Society*, 29:641–682, 1997.

[3] C. M. Bishop. *Neural Networks for Pattern Recognition*. Oxford University Press, Oxford, England, 1995.

[4] M. J. Black, G. Sapiro, D. H. Marimont, and D. Heeger. Robust anisotropic diffusion. *IEEE Transactions on Image Processing*, 7(3):421–432, March 1998.

[5] J. Bolz, I. Farmer, E. Grinspun, and P. Schröder. Sparse matrix solvers on the GPU: Conjugate gradients and multigrid. In *ACM Transactions on Graphics*, volume 22 of *SIGGRAPH*, pages 917–924, July 2003.

[6] Y. Boykov and M.-P. Jolly. Interactive organ segmentation using graph cuts. In *Medical Image Computing and Computer-Assisted Intervention*, pages 276–286, Pittsburgh, PA, October 2000.

[7] Y. Boykov and M.-P. Jolly. *Interactive graph cuts* for optimal boundary & region segmentation of objects in N-D images. In *Proc. of ICCV 2001*, pages 105–112, 2001.

[8] Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. *IEEE PAMI*, 23(11):1222–1239, November 2001.

[9] B. P. D. Greig and A. Seheult. Exact maximum *a posteriori* estimation for binary images. *Journal of the Royal Statistical Society, Series B*, 51(2):271–279, 1989.

[10] J. J. Dongarra, I. S. Duff, D. C. Sorenson, and H. A. van der Vorst. *Solving Linear Systems on Vector and Shared Memory Computers*. Society for Industrial and Applied Mathematics, Philadelphia, 1991.

[11] P. Doyle and L. Snell. *Random walks and electric networks*. Number 22 in Carus mathematical monographs. Mathematical Association of America, Washington, D.C., 1984.

[12] G. Golub and C. Van Loan. *Matrix Computations*. The Johns Hopkins University Press, 3rd edition, 1996.

[13] L. Grady. *Space-Variant Computer Vision: A Graph-Theoretic Approach*. PhD thesis, Boston University, Boston, MA, 2004.

[14] L. Grady and G. Funka-Lea. Multi-label image segmentation for medical applications based on graph-theoretic electrical potentials. In *Computer Vision and Mathematical Methods in Medical and Biomedical Image Analysis, ECCV 2004*, number LNCS3117, pages 230–245, Prague, Czech Republic, May 2004. Springer.

[15] A. K. Jain, R. P. Duin, and J. Mao. Statistical pattern recognition: A review. *IEEE Pattern Analysis and Machine Intelligence*, 22(1):4–37, January 2000.

[16] M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active contour models. *International Journal of Computer Vision*, 1(4):321–331, 1987.

[17] R. Merris. Laplacian matrices of graphs: A survey. *Linear Algebra and its Applications*, 197, 198:143–176, 1994.

[18] J.-M. Morel and S. Solimini. *Variational Methods in Image Segmentation*. Birkhauser, BostonBasel -Berlin, 1994.

[19] P. Perona and J. Malik. Scale-space and edge detection using anisotropic diffusion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(7):629–639, July 1990.

[20] J. Roerdink and A. Meijster. The watershed transform: definitions, algorithms, and parallellization strategies. *Fund. Informaticae*, 41:187–228, 2000.

[21] J. A. Sethian. *Level Set Methods and Fast Marching Methods*. Cambridge University Press, 1999.

[22] J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE PAMI*, 22(8):888–905, Aug. 2000.

[23] R. Wallace, P.-W. Ong, and E. Schwartz. Space variant image processing. *International Journal of Computer Vision*, 13(1):71–90, Sept. 1994.

[24] Z. Wu and R. Leahy. An optimal graph theoretic approach to data clustering: Theory and its application to image segmentation. *IEEE Pattern Analysis and Machine Intelligence*, 11:1101–1113, 1993.