Sub-Markov Random Walk for Image Segmentation

Xingping Dong, Jianbing Shen, *Senior Member, IEEE*, Ling Shao, *Senior Member, IEEE*, and Luc Van Gool

Abstract—A novel sub-Markov random walk (subRW) algorithm with label prior is proposed for seeded image segmentation, which can be interpreted as a traditional random walker on a graph with added auxiliary nodes. Under this explanation, we unify the proposed subRW and other popular random walk (RW) algorithms. This unifying view will make it possible for transferring intrinsic findings between different RW algorithms, and offer new ideas for designing novel RW algorithms by adding or changing auxiliary nodes. To verify the second benefit, we design a new subRW algorithm with label prior to solve the segmentation problem of objects with thin and elongated parts. The experimental results on both synthetic and natural images with twigs demonstrate that the proposed subRW method outperforms previous RW algorithms for seeded image segmentation.

Index Terms—Seeded image segmentation, subMarkov, random walk, optimization, label prior, complex texture.

I. INTRODUCTION

RANDOM walk (RW) has been widely used for many different tasks in computer vision and machine learning, such as segmentation [20], [23], [28], [35], clustering [19], [25], ranking [13], [31], classification [10], [41] and the other applications [1], [13], [18], [22], [26], [33]. Grady and Funka-Lea [15] first proposed the RW for medical image segmentation and extended it in [20] for general image segmentation. In their work, the user should give labels to a small number of pixels. Assuming a random walker starts at each unlabeled pixel, they calculate the probabilities that these random walkers first reach the pre-labeled pixels. These probabilities will determine the probability of each unlabeled pixel belonging to a label. By assigning each pixel to a label with the greatest probability, the interactive image segmentation

X. Dong and J. Shen are with the Beijing Laboratory of Intelligent Information Technology, School of Computer Science, Beijing Institute of Technology, Beijing 100081, China (e-mail: dongxingping@bit.edu.cn; shenjianbing@bit.edu.cn).

L. Shao is with the Department of Computer Science and Digital Technologies, Northumbria University, Newcastle upon Tyne NE1 8ST, U.K. (e-mail: ling.shao@ieee.org).

L. Van Gool is with the Computer Vision Laboratory, ETH Zürich, Zürich 8092, Switzerland (e-mail: vangool@vision.ee.ethz.ch).

result can be obtained. After [15], many related and important methods based on RW [17], [23], [34], [35]. In [17] have been proposed, the RW has been extended to segment out disconnected objects by using prior models without labeling each object. In other words, a user only needs to indicate labels on some objects and the other similar objects will be segmented out. Sinop and Grady [23] proposed a common framework to unify the RW, graph cuts, and shortest path algorithms for interactive segmentation. Furthermore, they added the popular watershed segmentation algorithm to this framework [35] and made the theoretical analysis for the connections between these algorithms. This unified framework brings some benefits, including opening new possibilities for using unary terms in traditional watershed algorithms to optimize more general models.

Recently, some researchers [27], [40] have focused on segmenting natural images with complex textures. They extend the RW algorithm and obtain better performance for these challenging images. In general, these algorithms are graph-based so we can use a graph to describe an image for introducing them. Kim *et al.* [27] proposed a random walker with a restarting probability (RWR) for segmentation. It means that this random walker will return to the starting node with a probability c at each step, and walk to other adjacent nodes with probability $1 - c$. Shen *et al.* [40] have developed the lazy random walk (LRW) for superpixel segmentation. A LRW will stay at the current node with a probability $1 - \alpha$ and walk out along the edges connected with the current node with probability α. Wu *et al.* [37] proposed another similar RW algorithm called partially absorbing random walk (PARW) for applications based on cluster, such as ranking and classification. In PARW, a random walker is absorbed at current node i with a probability α_i and follows a random edge out of it with probability $1 - \alpha_i$. And they analyze the relations between PARW and other popular ranking and classification models, such as PageRank [7], hitting and commute times [32], and semi-supervised learning [11], [16]. Comparing the above three RW-based algorithms, we can conclude that they all satisfy the subMarkov property [30], i.e., the sum of transition probabilities $q(i, j)$ that a random walker starts from a node to other adjacent nodes is less than or equal to 1. Is there a common framework to unify these algorithms? This is the problem we want to first solve in this work.

The other problem is how to segment objects with thin and elongated parts (twig problem) in natural images, which is difficult for most RW-based algorithms. For example, Figs. 1 (c)-(e) show the failed segmentation results of popular RW-based algorithms, including RW [20], RWR [27], and LRW [40]. One reason may be that the distance between
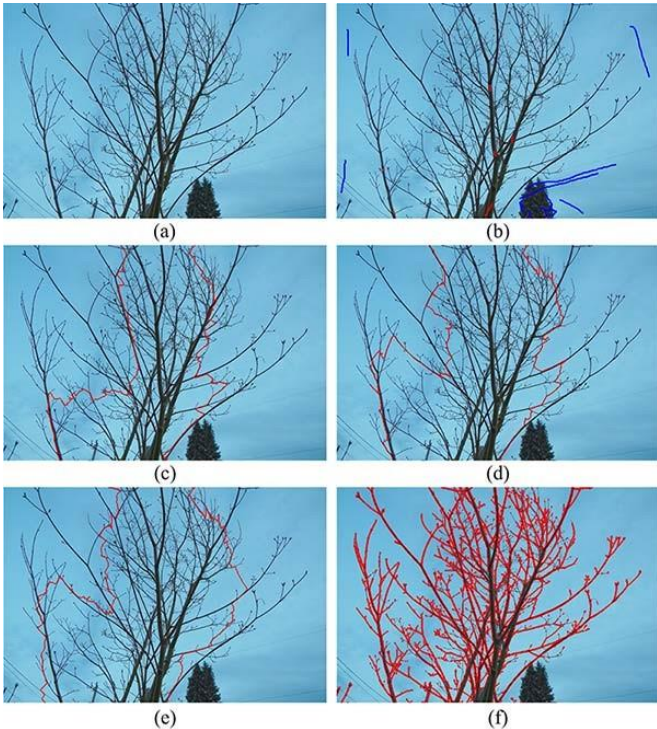
Fig. 1. Comparison results between our subRW method and three well-known RW-based algorithms. (a) Is an input image; (b) Is the scribbled image, where the red/blue scribbles specify the foreground/background regions respectively. (c)-(f) Are the corresponding results by RW [20], RWR [27], LRW [40], and our subRW algorithm with label prior. The red lines denote the boundaries of twigs segmentation.

the twig parts and the corresponding seeds is so far that the random walker with a small probability starts from these seeds to twig parts (or from twig to seeds).

In this paper, we propose a novel subMarkov random walk (subRW) framework to unify four RW-based algorithms: RW, RWR, LRW and PARW, and extend it by adding label prior to solve the twig problem. First, according to the subMarkov property, we build a subRW framework for image segmentation. In subRW, a random walker will leave a graph G from a node i with probability c_i and walk to the other adjacent nodes in G with probability $1 - c_i$. This random walker can be transformed to a random walker with Markov transition probability ($\sum q(i, j) = 1$) that walks in an expanded graph G_e. This graph is constructed by adding auxiliary staying nodes connected with seeds and auxiliary killing nodes connected with unseeded nodes into graph G. In order to further understand the subRW, we give a detailed optimization explanation. Then we unify the subRW and the aforementioned four RW-based algorithms in the expanded graph. After analyzing the connections between them, we design a new RW-based algorithm by changing edges or adding auxiliary nodes. According to this idea, we introduce a novel subRW with label prior to solve the twig problem. This label prior can be viewed as global 'seeds' connected with all nodes. Each global 'seed' corresponds to a label. So we can add some prior nodes connected with all nodes into graph G_e to build a new expanded graph G_p. Then we compute the probability that a random walker starting from each node reaches the staying nodes or the prior nodes in graph G_p, as

the likelihoods probability of corresponding labels. In other words, we want to compute the probability of reaching the user specified seeds plus the probability of reaching the global 'seeds'. These global 'seeds' will help to segment out the twig parts. As shown in Fig. 1 (f), our approach has successfully segmented the twigs of the tree using the proposed subRW algorithm with label prior information.

A preliminary version of this work was presented in [43], and the present is significantly different. The popular random walk with Markov transition probability is first added into our unified optimization framework, which makes this framework more complete. Considerable new theoretical analysis and proofs are added into the initial subRW algorithm, such as the uniqueness and a new optimization explanation for subRW with label prior, which make it applicable for more vision applications based on optimization. The optimization framework and explanation improves the initial algorithm to be more suitable for multi-label segmentation. We also extend the original experiments from 2-label segmentation to multi-label segmentation. Our source code will be available at.[1]

The main contributions are summarized as follows:

1) A novel random walk (subRW) with label prior is proposed for unifying well-known RW-based algorithms, such as RW [20], RWR [27], LRW [40], and PARW [37], which all satisfy the SubMarkov property, making it easier to convert the intrinsic findings between them.

2) The subRW is interpreted as a general optimization problem, which makes it easier to find the latent problem of the subRW for different vision applications. For example, from the optimization explanation of subRW with label prior, we find the consistence between label prior and reaching probability may be violated in multi-label segmentation, and we successfully solve this problem.

3) We further introduce a novel subRW algorithm by adding auxiliary nodes into the original graph. According to this idea, a novel subRW method with label prior is proposed to solve the twig segmentation problem with thin and elongated objects. The optimization explanation is also given.

II. A UNIFYING VIEW OF SUBRW

In this section, we propose a novel random walk algorithm with a subMarkov transition probability (subRW) for interactive multi-labeled image segmentation, provide the optimization explanation of subRW, and analyze the relations between this proposed algorithm and other popular RW algorithms, such as RW [20], RWR [27], LRW [40], and PARW [37].

We first give some important notations and their corresponding descriptions. An image is formulated as a weighted undirected graph $G = (V, E)$ with nodes $v \in V$, and edges $e \in E \subseteq V \times V$. Each node v_i represents an image pixel x_i. An edge e_{ij} connects two nodes v_i and v_j in 8-connected neighborhood system. A weight $w_{ij} \in \mathbf{W}$ of edge e_{ij} measures the likelihood that a random walker will cross this edge.

As many previous graph-based segmentation algorithms [20], [21], [27], [40], a weight w_{ij} is defined as the following weighting function:

$$w_{ij} = exp(-\frac{\|I_i - I_j\|^2}{\sigma}) + \epsilon, \tag{1}$$

where I_i and I_j are the pixel colors at two nodes v_i and v_j in Lab color space, σ is a controlling parameter which is set as $1/60$ in this paper, and ϵ is a small constant as 10^{-6}. The degree matrix $\mathbf{D}$ is a diagonal matrix where $\mathbf{D}_{ii} = d_i$, $d_i = \sum_{j \sim i} w_{ij}$ is the degree of a node v_i, $j \sim i$ represents a node v_j in the neighborhood (not includeing itself) of v_i. N is the number of nodes (pixels).

In our approach, a user needs to indicate some scribbles on foreground objects and backgrounds as multi-labeled seeds. Here, we define these seeds as a set of labeled nodes $V_M = \{V^{l_1}, V^{l_2}, \cdots, V^{l_K}\}$. Then a set of labels $LS = \{l_1, l_2, \cdots, l_K\}$ is also defined, where K is the number of labels $V^{l_k} = \{v_1^{l_k}, v_2^{l_k}, \cdots, v_{M_k}^{l_k}\}$ and M_k is the number of seeds with label l_k.

A. The SubMarkov Random Walk

Given a weighted graph G, a set of labeled nodes V_M, and a set of unlabeled nodes V_U, where $V_U \cup V_M = V$, the multi-labeled image segmentation can be formulated as a labeling problem, where each node $v_i \in V$ should be assigned with a label from set LS. This problem can be solved by comparing the probability $r_i^{l_k}$ of each node belonging to a label l_k in our algorithm. Before computing this probability, we define the subMarkov transition probability q on V as follows:

Definition 1: q denotes a subMarkov transition probability if for each node v_i

$$\sum_{j \sim i} q(i, j) \leq 1. \tag{2}$$

According to [30], a subMarkov transition probability has the following property:

Property 1: Through adding an auxiliary node Δ, a sub-Markov transition probability q on G can be made into a (Markov) transition probability on $V \cup \{\Delta\}$ by setting $q(\Delta, \Delta) = 1$ and $q(i, \Delta) = 1 - \sum_{j \sim i} q(i, j)$. The probability $q(i, \Delta)$ can be viewed as a probability that a random walker leaves graph G.

According to the above property, we can design different subMarkov random walk algorithms by adding different auxiliary nodes. In fact, popular random walk algorithms, such as RW [20], RWR [27], LRW [40], and PARW [37], can be interpreted in this view (more details will be given in the next subsection). The other advantage of a subMarkov transition probability is that it will help to improve segmentation per-
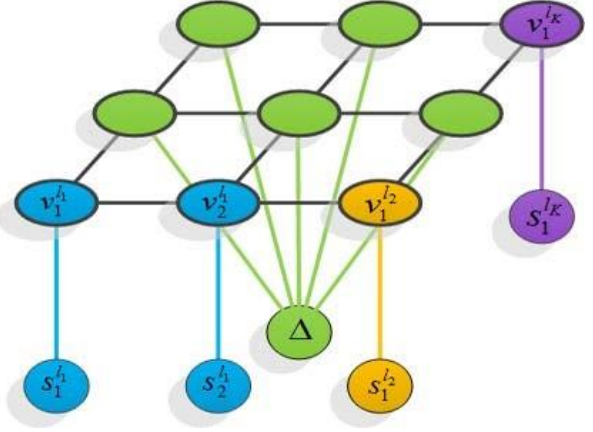

Fig. 2. The nodes graph of a subRW. The ellipse nodes denote the original nodes in V and the circle nodes are the newly added auxiliary nodes. The green ellipses are the unseeded nodes and the others are the seeded nodes.

node Δ connected with all unseeded nodes (e.g., the green circle node in Fig. 2). When a random walker from a node v_i reaches this node, it will be killed at this node and the corresponding probability will be omitted. In other words, an effective random walker will not reach this node. The other one is the staying node $s_m^{l_k}$ connected with the m-seeded node with label l_k (e.g., the blue, orange or purple circle nodes in Fig. 2). When a random walker reaches this node which can be viewed as a target node, it will stay at this node. We denote S_M as a set of staying nodes corresponding to seeds set V_M, and c_i as the leaving probability for each node in V. Then, the transition probability on $V \cup \{\Delta\} \cup S_M$ is formulated as follows:

$$q(i, j) = \begin{cases} c_i, & \text{if } i \in V, j \in \{\Delta\} \cup S_M \\ (1 - c_i)\dfrac{w_{ij}}{d_i}, & \text{if } j \sim i \in V \\ 1, & \text{if } i = j \in \{\Delta\} \cup S_M \\ 0, & \text{otherwise.} \end{cases} \tag{3}$$

Suppose a random walker starts from a node $v_i \in V$ and walks on $V \cup \{\Delta\} \cup S_M$ with the transition probability $q(i, j)$ in (3). By setting $r_{im}^{l_k}$ as the reaching probability that this random walker reaches the auxiliary staying node $s_m^{l_k}$, then we have:

$$r_{im}^{l_k} = \begin{cases} (1 - c_i)\displaystyle\sum_{j \sim i}\dfrac{w_{ij}}{d_i} r_{jm}^{l_k} + c_i \cdot 1 & \text{if } v_i = v_m^{l_k}, \\ (1 - c_i)\displaystyle\sum_{j \sim i}\dfrac{w_{ij}}{d_i} r_{jm}^{l_k} + c_i \cdot 0 & \text{otherwise.} \end{cases} \tag{4}$$

The vector notation $\mathbf{r}_m^{l_k} = [r_{im}^{l_k}]_{N \times 1}$, which is formulated as follows:

$$\mathbf{r}_m^{l_k}$$

formance in images with complex texture. More experiments and analysis for solving the complex texture problem given in Section IV-B.

We first consider a general subRW algorithm for interactive seeded image segmentation. Two kinds of auxiliary nodes are added into graph G to get an expanded graph G_e. As shown in Fig. 2, one kind of auxiliary node is a killing

$$\mathbf{r}_m^{l_k} = (\mathbf{I} - \mathbf{D_c})\mathbf{P}\mathbf{r}_m^{l_k} + \mathbf{D_c}\mathbf{b}_m^{l}, \tag{5}$$

where $\mathbf{b}_m^{l_k} = [b_{im}^{l_k}]_{N \times 1}$ is the N-dimensional indicating vector with $b_{im}^{l_k} = 1$ if $v_i = v^{l_k}$ and $b_{im}^{l_k} = 0$ otherwise. $\mathbf{D_c}$ is a diagonal matrix whose diagonal element is c_i i.e. $\mathbf{D_c} = diag(c_1, c_2, \cdots, c_N)$, and $\mathbf{I}$ is a $N \times N$ identity matrix. The transition matrix $\mathbf{P} = [p_{ij}]_{N \times N}$ is a row-normalized

matrix of the adjacency matrix $\mathbf{W}$ (defined in (1)):

$$p_{ij} = w_{ij}/d_i. \tag{6}$$

The number of a set of seeded nodes with a same label is often larger than one. A good RW approach should consider all seeded nodes. We use an average reaching probability $r_i^{l_k}$, that a random walker from a node v_i reaches a set of staying nodes with label l_k, as the likelihood of assigning this node to label l_k. (5) can be rewritten as follows:

$$\mathbf{r}_m^{l_k} = (\mathbf{I} - (\mathbf{I} - \mathbf{D_c})\mathbf{P})^{-1}\mathbf{D_c}\mathbf{b}_m^{l_k} = \mathbf{E}^{-1}\mathbf{D_c}\mathbf{b}_m^{l_k}, \tag{7}$$

where $\mathbf{E} = \mathbf{I} - (\mathbf{I} - \mathbf{D_c})\mathbf{P}$.

Thus, a vector formulation of this average steady-state probability $\mathbf{r}^{l_k}$ can be given as follows:

$$\mathbf{r}^{l_k} = \frac{1}{Z_k M_k}\sum_{m=1}^{M_k}\mathbf{r}_m^{l_k} = \frac{1}{Z_k M_k}\mathbf{E}^{-1}\mathbf{D_c}\mathbf{b}^{l_k}, \tag{8}$$

where $\mathbf{b}^{l_k} = [b_i^{l_k}]_{N \times 1}$ is a vector with $b_i^{l_k} = 1$ if $v_i \in \mathbf{V}^{l_k}$ and $b_i^{l_k} = 0$ otherwise, and Z_k is a normalized constant. The final labeling result (i.e., the segmentation result) for each node $v_i \in V$ is obtained as follows:

$$R_i = \arg\max_{l_k} r_i^{l_k}. \tag{9}$$

Note that, in (8), directly computing the inverse of matrix $\mathbf{E}$ may be very expensive, especially when the number of nodes is very large. Fortunately, (8) can be converted to solve the linear equations and the corresponding coefficient matrix $\mathbf{E}$ is a sparse matrix. We can use the existing sparse algorithms [3]–[5] to solve these linear equations. More importantly, if $0 < c_i \leq 1, \forall i \in V$, matrix $\mathbf{E}$ is non-singular (the proof is given in Appendix A). The solution of these linear equations is unique in theory, when $0 < c_i \leq 1, \forall i \in V$. Thus, the segmentation results are unique.

B. The Optimization Explanation of SubRW

Although the subRW algorithm is based on random walks, it can also be interpreted as a general energy optimization problem, which will make the subRW easier to be employed for general vision applications using optimization [29], [34]. Now we will give the optimization explanation in detail.

Firstly, suppose $\forall i, 0 < c_i < 1$, then let us consider the following objective function:

$$1 \quad N \quad N$$

By taking the partial derivative of $\mathbf{r}^{l_k}$, we have:

$$\frac{\partial O^{l_k}}{\partial \mathbf{r}_m} = (\mathbf{D} - \mathbf{W})\mathbf{r}_m^l + (\mathbf{I} - \mathbf{D_c})^{-1}\mathbf{D}\mathbf{D_c}(\mathbf{r}_m^l - \mathbf{b}_m^l)$$

$$= (\mathbf{I} - \mathbf{D_c})^{-1}\mathbf{D}[(\mathbf{I} - (\mathbf{I} - \mathbf{D_c})\mathbf{D}^{-1}\mathbf{W})\mathbf{r}_m^{l_k} - \mathbf{D_c}\mathbf{b}_m^{l_k}]$$

$$= (\mathbf{I} - \mathbf{D_c})^{-1}\mathbf{D}(\mathbf{E}\mathbf{r}_m^{l_k} - \mathbf{D_c}\mathbf{b}^{l_k}), \tag{12}$$

where $\mathbf{E} = \mathbf{I} - (\mathbf{I} - \mathbf{D_c})\mathbf{P} = \mathbf{I} - (\mathbf{I} - \mathbf{D_c})\mathbf{D}^{-1}\mathbf{W}$. By setting $\frac{\partial O^{l_k}}{\partial \mathbf{r}_m^{l_k}} = 0$, we can get (7). Thus, the optimal solution of minimizing the objective function O^{l_k} is equivalent to the solution of subRW.

Let us analyze the objective function (10). This objective function consists of two components. The first component is a common smoothing term. Minimizing this term will keep the consistence of the local probabilities, i.e., when the nodes are more similar, their probabilities are more likely to be the same. The second component can be viewed as a unary term corresponding to seeds. By minimizing this term, probability $r_{im}^{l_k}$ keeps close to the predefined value $b_{im}^{l_k}$ which represents the information of seeds. c_i is the regularization parameter, and it decides which component is more important.

C. Relations With Other Well-Known RW Algorithms

We will further analyze the relations between the proposed subMarkov random walk and other popular algorithms: RW [20], RWR [27], LRW [40], and PARW [37]. Then we find these algorithms can be unified and related with the subRW.

1) Relations With RW: In [20], Grady proposed a random walk algorithm with Markov transition probability for segmentation, which can be viewed as a special case of subRW. This algorithm places a random walker at each unlabeled node and then computes which labeled nodes they first arrive at. This computation would be completely impractical. Fortunately, the combinatorial Dirichlet algorithm provides a simple and convenient method for computing this problem [20]. Here, we directly give formulation of this solution. More details can be found in [20]. Denote $\mathbf{x}^{l_k}$ as the probabilities of random walkers first arriving at the labeled nodes with label l_k and $\mathbf{L}$ as the Laplacian matrix, where $\mathbf{L} = \mathbf{D} - \mathbf{W}$ [6]. Without loss of generality, we may assume that the nodes in $\mathbf{L}$ and $\mathbf{x}^{l_k}$ are ordered such that the labeled nodes are in first and the unlabeled nodes are in second. Then we have the block matrix of $\mathbf{L}$ and $\mathbf{x}^{l_k}$:

$$\mathbf{L} = \begin{bmatrix} \mathbf{L}_M & \mathbf{B} \\ \mathbf{B}^T & \mathbf{L}_U \end{bmatrix}, \quad \mathbf{x}^{l_k} = \begin{bmatrix} \mathbf{x}_M^{l_k} \\ \mathbf{x}_U^{l_k} \end{bmatrix}. \tag{13}$$

$$O^{l_k} = \frac{\overline{}}{2} \sum_{i=1}^{} \sum_{j=1}^{} w_{ij} (r_{im}^{l_k} - r_{jm}^{l_k})^2$$

$$+ \frac{1}{2} \sum_{i=1}^{N} \frac{d_i c_i}{1 - c_i} (r_{im}^{l_k} - b_{im}^{l_k})^2. \tag{10}$$

The vector formulation of above equations is defined as:

$$O^{l_k} = \frac{1}{2} \mathbf{r}_m^{l_k \, T} (\mathbf{D} - \mathbf{W}) \mathbf{r}_m^{l_k}$$

$$+ \frac{1}{2} (\mathbf{r}_m^{l_k} - \mathbf{b}_m^{l_k})^T (\mathbf{I} - \mathbf{D_c})^{-1} \mathbf{D} \mathbf{D_c} (\mathbf{r}_m^{l_k} - \mathbf{b}_m^{l_k}). \tag{11}$$

Different from the subRW, this method only needs to calculate the probabilities of unlabeled nodes and the final solution is as follows:

$$\mathbf{L}_U \mathbf{x}_U^{l_k} = -\mathbf{B}^T \mathbf{x}_M^{l_k}, \tag{14}$$

where $\mathbf{x}_M^{l_k}$ is given by the labeled nodes.

By setting the indicating vector $\mathbf{b}^{l_k} = [\mathbf{b}_M^{l_k \, T}, \mathbf{b}_U^{l_k \, T}]^T$, then we obtain:

$$\mathbf{x}_M^{l_k} = \mathbf{b}_M^{l_k}. \tag{15}$$

For the comparison between RW and subRW, we should rewrite (14). Denote the block matrix of $\mathbf{W}$, $\mathbf{D}$ as:

$$\mathbf{W} = \begin{matrix} \mathbf{W}_M & \mathbf{W}_B \\ \mathbf{W}_B^T & \mathbf{W}_U \end{matrix}, \quad \mathbf{D} = \begin{matrix} \mathbf{D}_M & 0 \\ 0 & \mathbf{D}_U \end{matrix}. \quad (16)$$

Note that $\mathbf{L} = \mathbf{D} - \mathbf{W}$, so $\mathbf{L}_U = \mathbf{D}_U - \mathbf{W}_U$ and $\mathbf{B} = -\mathbf{W}_B$. Substitute these above equations into (14), we then have:

$$(\mathbf{D}_U - \mathbf{W}_U)\mathbf{x}_U^{l_k} = \mathbf{W}_B^T \mathbf{x}_M^{l_k} \quad (17)$$

$$\mathbf{D}_U \mathbf{x}_U^{l_k} = \mathbf{W}_U \mathbf{x}_U^{l_k} + \mathbf{W}_B^T \mathbf{x}_M^{l_k}. \quad (18)$$

Rewrite (18) as follows:

$$d_i x_i^{l_k} = \sum_{j \sim i, j \in V_U} w_{ij} x_i^{l_k} + \sum_{j \sim i, j \in V_M} w_{ij} x_j^{l_k}, \quad i \in V_U \quad (19)$$

$$x_i^{l_k} = \sum_{j \sim i} \frac{w_{ij}}{d_i} x_j^{l_k}, \quad i \in V_U, \quad (20)$$

where $x_j^{l_k} = b_i^{l_k}$, for $j \in V_M$.

Recall (8) for computing the desired probabilities of subRW, we can rewrite it as follows:

$$r_i^{l_k} = (1 - c_i) \sum_{j \sim i} \frac{w_{ij}}{d_i} r_j^{l_k} + \frac{c_i}{Z_k M_k} \cdot b_i^{l_k}. \quad (21)$$

Comparing (15), (20) and (21), it indicates that if $Z_k = 1/M_k$, $c_i = \begin{cases} 1, i \in V_M \\ 0, i \in V_U, \end{cases}$ then the subRW is equivalent to the RW.

2) Relations With RWR: Kim *et al.* [27] suppose a random walker starts from a m-th seed node $v_m^{l_k}$ of label l_k in a graph G. Different from the traditional random walker, it has a restarting probability c to return to seed $v_m^{l_k}$ at each step. Then each node is assigned a steady-state probability $fr_{im}^{l_k}$ that this random walker will finally stay at this node, which is formulated as:

$$fr_i^{l_k} = (1 - c) \sum w_{ij} fr^{l_k} + c \cdot b^{l_k}. \quad (22)$$

a node v_i has the same label with seeded node v^{l_k}, can be formulated as:

$$\mathbf{fl}_m^{l_k} = (\mathbf{I} - \alpha\mathbf{S})^{-1} \mathbf{b}_m^{l_k}, \quad (24)$$

where $\mathbf{S} = \mathbf{D}^{-1/2}\mathbf{W}\mathbf{D}^{-1/2}$, $\mathbf{D}$ is a diagonal matrix and $\mathbf{D}_{ii} = d_i$.

We rewrite (24) as follows:

$$\mathbf{fl}_m^{l_k} = \alpha\mathbf{D}^{-1/2}\mathbf{W}\mathbf{D}^{-1/2}\mathbf{fl}_m^{l_k} + \mathbf{b}_m^{l_k}, \quad (25)$$

In fact, the labeling result of LRW (or other algorithms based on RW) only depends on the probabilities of each node with different labels, i.e., $Rl_i = \arg\max_{l_k} fl_{im}^{l_k}$. So we can scale these probabilities to interpret the LRW from the subRW point of view, which will not change this labeling result. By setting $\beta = (1 - \alpha)d_{min}^{1/2}$, $d_{min} = \min_{i=1:N} d_i$, and $\mathbf{rl}_m^{l_k} = \beta\mathbf{D}^{-1/2}\mathbf{fl}_m^{l_k}$, we then multiply (25) by $\beta\mathbf{D}^{-1/2}$ to obtain:

$$\mathbf{rl}_m^{l_k} = \alpha\mathbf{D}^{-1}\mathbf{W}\mathbf{rl}_m^{l_k} + \beta\mathbf{D}^{-1/2}\mathbf{b}_m^{l_k}. \quad (26)$$

The above equation can be rewritten as follows:

$$rl_{im}^{l_k} = \alpha \sum_{j \sim i} \frac{w_{ij}}{d_i} rl_{jm}^{l_k} + (1 - \alpha)\gamma_i b_{im}^{l_k} + (1 - \alpha)(1 - \gamma_i) \cdot 0, \quad (27)$$

where $\gamma_i = (\frac{d_{min}}{d_i})^{1/2}$.

According to this equation, we find that the LRW can also be viewed as a subMarkov random walk by adding an edge between each seeded node and the killing node . Then the corresponding transition probability on $V \cup \{ \} \cup S_M$ can be formulated as:

$$q(i, j) = \begin{cases} 1 - \alpha, & \text{if } i \in V_U, j = \\ (1 - \alpha)\gamma_i, & \text{if } i \in V_M, j \in S_M \\ (1 - \alpha)(1 - \gamma_i), & \text{if } i \in V_i, j \in S_M \\ \alpha\frac{w_{ij}}{d_i}, & \text{if } j \sim i \in V \end{cases} \quad (28)$$

$$\begin{cases} 1, & \text{if } i = j \in \{\ \} \cup S_M \\ 0, & \text{otherwise.} \end{cases}$$

(4) of subRW is rewritten as follows:

$$r^{l_k}_{im} = (1 - c_i) \sum_{j \sim i} \frac{w_{ij}}{d_i} r^{l_k}_{jm} + c_i \cdot b^{l_k}_{im}. \qquad (23)$$

Combining (22) and (23), we find that RWR is a special case of subRW with leaving probability $c_i = c$, $i = 1, 2, \cdots, N$. In other words, the subRW can be viewed as a set of RWR with a different restarting probability at each node.

3) Relations With LRW: In [40], Shen *et al.* proposed a lazy random walker for superpixel segmentation, which is viewed as a multi-labeled segmentation problem. Under their framework, a random walker will stay at the current position with probability $(1 - \alpha)$ and walk out along an arbitrary edge with probability α. They use the commute time CT_{ij} [14] to measure the probability that these two nodes belong to the same label, where CT_{ij} is the expected number of steps for a lazy random walker starting at v_i to v_j and then returning. After normalizing the commute time, probability $fl^{l_k}_m$, that

4) Relations With PARW: In [37], partially absorbing random walks (PARWs) model were proposed for ranking, clustering, and classification. Wu *et al.* [37] suppose that a PARW is absorbed at current node i with probability λ_i, and walks out of it following a random edge with probability $1 - \lambda_i$. They set a_{ij} as the absorption probability, where a PARW starting from node i will be absorbed at node j with probability a_{ij}. Then matrix $\mathbf{A} = [a_{ij}]_{N \times N}$ of absorption probabilities is formulated as:

$$\mathbf{A} = (\ + \mathbf{D} - \mathbf{W})^{-1}, \qquad (29)$$

where $= diag(\lambda_1, \lambda_2, \cdots, \lambda_N)$, $\lambda_i \quad 0, i = 1, \cdots, N$.

To solve the segmentation or labeling problem, we can use the absorption probability $rp^{l_k}_{im}$ that a PARW starting from a seeded node $v^{l_k}_m$ is absorbed at node v_i, as the probabilities belonging to label l_k. The vector notation $\mathbf{rp}^{l_k}_m$ is:

$$\mathbf{rp}^{l_k}_m = \mathbf{A}\mathbf{b}^{l_k}_m. \qquad (30)$$

By combining (29), (30) can be rewritten as:

$$\mathbf{rp}_m^{l_k} = (\mathbf{\Lambda} + \mathbf{D})^{-1}(\mathbf{W}\mathbf{rp}_m^{l_k} + \mathbf{\Lambda}\mathbf{b}_m^{l_k}). \tag{31}$$

The above equation is equivalent to:

$$rp_{im} = \sum_{j \sim i} \frac{w_{ij}}{d_i + \lambda_i} rp_{im}^{l_k} + \frac{\lambda_i}{d_i + \lambda_i} b_{im}^{l_k}. \tag{32}$$

By comparing the above equation and (23), we find that (32) is equivalent to (23) when $c_i = \frac{\lambda_i}{d_i + \lambda_i}$, i.e. the PARW is equivalent to the sub$\overline{\text{RW}}$. In fact, after substituting $c_i = \frac{\lambda_i}{d_i + \lambda_i}$ into $\frac{1-c_i}{d_i}$, we have $\frac{1-c_i}{d_i} = \frac{1}{d_i + \lambda_i}$.

5) Merits of a Unifying View: We have shown that subRW can unify or relate the popular models based on RW. There are at least two merits of the unifying view. First, it builds the connections between different RW-based algorithms, so that it is easier to transfer findings between them. For example, in PARW [37], the authors have proved that the choice of $\lambda_i = \alpha$ ($\forall i$) can fulfil the cluster assumption. In other words, by fixing all λ_i as a proper parameter, a desirable clustering performance with PARWs will be achieved. If we want to apply subRW to solve a clustering problem, the leaving probability c_i can be fixed as $\frac{\alpha}{d_i + \alpha}$ to fulfill the cluster assumption - when $c_i = \frac{\lambda_i}{d_i + \lambda_i}$, the subRW is equivalent to the PARW. Another example is for segmentation. In RWR [27], the authors have discussed the influence of parameter c. In this paper, we have chosen parameter c_i according to their suggestion since the subRW is the generalized version of RWR. Second, a unifying view offers a new way to design novel RW-based algorithms by adding some new auxiliary nodes or changing the edges between auxiliary nodes and original nodes in V. For example, the LRW can be viewed as an expansion of the subRW. Inspired by the second merit, we design a new subRW with label prior to segment out objects with twigs.

III. SubRW With Label Prior for Seeded Image Segmentation

An object with twigs can be separated into two parts: main branch object and twig part. Usually, the twig part is similar to the main object, so appropriate user-specified scribbles on the main object have included enough information for segmenting out the twig part. But most RW-based algorithms do not make full use of this information and often omit the twig part. In this section, we want to add a label prior constructed by these scribbles into the subRW to help segment out the twig part.

A. Adding Label Prior

In general, user-specified scribbles are considered as exact label prior. Unfortunately, this prior only works at the seeded nodes and all unseeded nodes do not have this prior. Therefore, we want to give all nodes in V a new label prior,
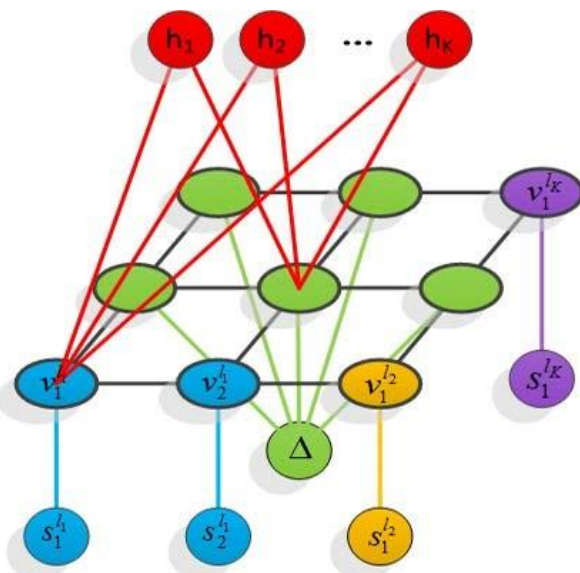

Fig. 3. The nodes graph with prior nodes of a subRW. The red circle nodes represent the prior nodes, which should be connected with all the original ellipse nodes. We only show two edges for each prior node for simplification. One edge connects a seeded node, and the other connects an unseeded node. Except for the red circle nodes, the other nodes are defined as the same as the nodes in Fig. 2.

where u_k^i denotes the probability density belonging to H_k at node v_i. This distribution can be learned via many techniques, such as kernel estimation and Gaussian Mixture Model (GMM). Here, the GMM is used as the prior model. Each prior distribution H_k is viewed as a GMM learned by the seeded nodes with label l_k (more details about GMM learning can be seen in [24], [36], and [38]). Given these prior distributions, we can add a set of prior auxiliary nodes $H_M = \{h_1, h_2, \cdots, h_K\}$ into expanded graph G_e and get a graph with prior $\tilde{G}$. An example is shown in Fig. 3. Each prior node is connected with all nodes in V and weight w_{ih_k} of an edge between a prior node h_k and a node $v_i \in V$ is proportional to probability density u_i^k, i.e., $w_{ih_k} \propto u_i^k$. In this paper, we set weight w_{ih_k} as:

$$w_{ih_k} = (1 - c_i)\lambda u_i^k, \tag{33}$$

where λ is a regularization parameter, which measures the importance of the prior distribution.

Then the transition probability on $V \cup \{\Delta\} \cup S_M \cup H_M$ is formulated as follows:

$$\bar{q}(i, j) = \begin{cases} c_i, & \text{if } i \in V, j \in \{\Delta\} \cup S_M \\ (1 - c_i)\dfrac{\lambda u_i^k}{d_i + \lambda g_i}, & \text{if } i \in V, j = h_k \\ (1 - c_i)\dfrac{w_{ij}}{d_i + \lambda g_i}, & \text{if } j \sim i \in V \\ 1, & \text{if } i = j \in \{\Delta\} \cup S_M \end{cases}$$

which maybe less exact than user scribbles, but can be

$0,$ otherwise, $\cup H_M$

$$(34)$$

used for unseeded nodes. This label prior is constructed by the user scribbles, i.e., the seeded nodes. We can use probability distributions to build the prior model. Assume a label l_k has an intensity distribution H_k for each node,

where $g_i = \sum_{k=1}^{K} u_i^k$.

Given a transition probability $\bar{q}$ on a graph with prior $\bar{G}$, the probability $\bar{r}_{im}^{l_k}$, that a random walker from a node $v_i \in V$ reaches the m-th staying node $s^{l_k^m}$ with label l_k or prior node h_k,

is formulated as follows:

$$\bar{r}_{im}^{\ l_k} = (1 - c_i) \sum_{j \sim i \in V} \frac{w_{ij}\bar{r}_{jm}^{\ l_k}}{d_i + \lambda g_i} + (1 - c_i)\frac{\lambda u_i^k}{d_i + \lambda g_i} + c_i b_{im}^{l_k},$$

(35)

where $b_{im}^{l_k} = 1$ if $v_i = v_m^{l_k}$ and $b_{im}^{l_k} = 0$ otherwise.

In fact, this prior node h_k can be viewed as a new staying node with label l_k, so this reaching probability of h_k should be considered. By setting a vector $\bar{\mathbf{r}}_m^{l_k} = [\bar{r}_{im}^{l_k}]_{N \times 1}$, we can get the vector formulation of (35):

$$\begin{aligned}
\bar{\mathbf{r}}_m^{l_k} &= (\mathbf{I} - \mathbf{D_c})\bar{\mathbf{P}}\bar{\mathbf{r}}_m^{l_k} + (\mathbf{I} - \mathbf{D_c})\bar{\mathbf{u}}^k + \mathbf{D_c}\mathbf{b}_m^{l_k} \\
&= (\mathbf{I} - (\mathbf{I} - \mathbf{D_c})\bar{\mathbf{P}})^{-1}((\mathbf{I} - \mathbf{D_c})\bar{\mathbf{u}}^k + \mathbf{D_c}\mathbf{b}_m^{l_k}) \\
&= \bar{\mathbf{E}}^{-1}((\mathbf{I} - \mathbf{D_c})\bar{\mathbf{u}}^k + \mathbf{D_c}\mathbf{b}_m^{l_k}),
\end{aligned}$$

(36)

where $\bar{\mathbf{E}} = \mathbf{I} - (\mathbf{I} - \mathbf{D_c})\bar{\mathbf{P}}$, the transition probability matrix $\bar{\mathbf{P}} = [\bar{p}_{ij}]_{N \times N}$ in V is defined as:

$$\bar{p}_{ij} = \frac{w_{ij}}{d_i + \lambda g_i},$$

(37)

$\bar{\mathbf{u}}^k = [\bar{u}_i^k]_{N \times 1}$ is a vector with

$$\bar{u}_i^k = \frac{\lambda u_i^k}{d_i + \lambda g_i}.$$

(38)

As mentioned before, we use the average reaching probability $\bar{r}_i^{l_k}$ for each node $v_i \in V$ as the probabilities belonging to label l_k. The vector notation $\bar{\mathbf{r}}^{l_k}$ is formulated as:

$$\bar{\mathbf{r}}^{l_k} = \frac{1}{Z_k}\bar{\mathbf{E}}^{-1}((\mathbf{I} - \mathbf{D_c})\bar{\mathbf{u}}^k + \frac{1}{M_k}\mathbf{D_c}\mathbf{b}^{l_k}),$$

(39)

The final labeling (segmentation) result with a label prior is obtained as follows:

$$\bar{R}_i = \arg\max_{l_k} \bar{r}_i^{l_k},$$

(40)

where $\bar{R}_i$ represents the final label for each node, i.e., each pixel in an image.

In fact, solving (39) can be converted to solving the equivalent sparse linear equations. We will scale the probability density to better utilize the GMMs, since the GMMs need the log operation to reduce the cost of computation. These algorithm steps are shown in **Algorithm 1**. And the solution of these equations is also unique, if $0 < c_i \leq 1, \forall i \in V$ (the proof is given in Appendix A).

B. The Optimization Explanation

Similar to the subRW, we can also give the optimization

Algorithm 1 Segmentation by SubRW With Label Prior

Input: An image $V(v_i)$ and K kinds of user scribbles $V_M = \{V^{l_1}, V^{l_2}, \cdots, V^{l_K}\}$, the parameters $\mathbf{D_c} = diag(c_1, c_2, \cdots, c_N)$, λ, γ;
Output: The segmentation result $\bar{R}_i$ for each pixel;
1: Obtain the indicating vectors $\mathbf{b}^{l_k}, k = 1 : K$ of scribbles;
2: Define an adjacency matrix $\mathbf{W} = [w_{ij}]_{N \times N}$ with 8-neighbors graph structure by (1);
3: Generate the GMMs with five components from all scribbles and then get the probability density u_i^k;
4: Scale probability: $u_i^k \leftarrow max(100 + log(u_i^k), 10^{-10})$;
   if $\gamma = 1$ **then**
5:     Obtain coarse segmentation result $CR_i = \arg\max_k u_i^k$;
6:     Get the candidate vector $\mathbf{cr}^k$ from CR_i;
7:     Reset the prior vector $\mathbf{u}^k \leftarrow \mathbf{u}^k \odot \mathbf{cr}^k$;
   end if
8: Set $g_i \leftarrow g_i^k$, where $g_i^k = u_i^k + \max_{t \neq k} u_i^t$;
9: Compute the transition probability matrix $\bar{\mathbf{P}}$ and the vector $\bar{\mathbf{u}}^k$ by (37) and (38);
12: Solve linear equations: $\bar{\mathbf{E}}\bar{\mathbf{r}}^{l_k} = (\mathbf{I} - \mathbf{D_c})\bar{\mathbf{u}}^k + \frac{1}{M_k}\mathbf{D_c}\mathbf{b}^{l_k}$;
13: Normalize the reaching probabilities: $\bar{\mathbf{r}}^{l_k} \leftarrow \frac{1}{Z_k}\bar{\mathbf{r}}^{l_k}$;
14: Obtain segmentation result $\bar{R}_i = \arg\max_{l_k} \bar{r}_i^{l_k}$;

The vector formulation is as follows:

$$\begin{aligned}
\bar{O}^{l_k} &= \frac{1}{2}\mathbf{r}_m^{l_k T}(\mathbf{D} - \mathbf{W})\bar{\mathbf{r}}^{l_k} \\
&+ \frac{1}{2}(\bar{\mathbf{r}}_m^{l_k} - \mathbf{b}_m^{l_k})^T (\mathbf{I} - \mathbf{D_c})^{-1}(\mathbf{D} + \lambda\mathbf{D_g})\mathbf{D_c}(\bar{\mathbf{r}}_m^{l_k} - \mathbf{b}_m^{l_k}) \\
&+ \frac{\lambda}{2}(\bar{\mathbf{r}}_m^{l_k} - \mathbf{e})^T \mathbf{D_u^k}(\bar{\mathbf{r}}_m^{l_k} - \mathbf{e}) + \frac{\lambda}{2}\sum_{t=1,t=k}^{K} \bar{\mathbf{r}}_m^{l_k T} \mathbf{D_u^t}\bar{\mathbf{r}}_m^{l_k},
\end{aligned}$$

(42)

where $\mathbf{D_g} = diag(g_1, g_2, \cdots, g_N)$ is a diagonal matrix, $\mathbf{D_u^k} = diag(u_1^k, u_2^k, \cdots, u_N^k)$ is also a diagonal matrix, and $\mathbf{e} = [1, 1, \cdots, 1]^T$ is a $N \times 1$ vector. We set $\mathbf{D_a} = \mathbf{D} + \lambda\mathbf{D_g}$ and $\mathbf{D_\eta} = \mathbf{I} - \mathbf{D_c}$.

By taking the partial derivative of $\bar{\mathbf{r}}_m^{l_k}$, we have:

$$\begin{aligned}
\frac{\partial \bar{O}^{l_k}}{\partial \bar{\mathbf{r}}_m^{l_k}} &= (\mathbf{D} - \mathbf{W})\bar{\mathbf{r}}^{l_k} + \mathbf{D_\eta}^{-1}\mathbf{D_a}\mathbf{D_c}(\bar{\mathbf{r}}^{l_k} - \mathbf{b}^{l_k}) \\
&+ \lambda(\mathbf{D_u^k}\bar{\mathbf{r}}_m^{l_k} + \sum_{t=1,t=k}^{K} \mathbf{D_u^t}\bar{\mathbf{r}}_m^{l_k} - \mathbf{u}^k)
\end{aligned}$$

$$= (\mathbf{D_a} - \mathbf{W})\bar{\mathbf{r}}^{l_k}$$

explanation for the subRW with label prior. Suppose $\forall i,\ 0 < c_i < 1$, then the objective function is as follows:

$$\bar{O}^{l_k} = \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} w_{ij} (\bar{r}^{l_k}_{im} - \bar{r}^{l}_{jm})^2$$

$$+ \frac{1}{2} \sum_{i=1}^{N} \frac{(d_i + \lambda g_i)c_i}{1 - c_i} (\bar{r}^{l_k}_{im} - b^{l_k}_{im})^2$$

$$+ \frac{1}{2} \sum_{i=1}^{N} \lambda u_i^k (\bar{r}^{l_k}_{im} - 1)^2 + \frac{1}{2} \sum_{t=1, t \ne k}^{K} \sum_{i=1}^{N} \lambda u_i^t \bar{r}^{l_k}_{im}{}^2. \tag{41}$$

$$\eta\ \mathbf{D_a D_c}(\bar{\mathbf{r}}^{\ k}_k$$

$$_m + \mathbf{D^{-1}}{}^l - \mathbf{b}_m) - \lambda \mathbf{u^k}$$

$$= \mathbf{D}_\eta^{-1}\mathbf{D_a}[(\mathbf{D}_\eta + \mathbf{D_c} - \mathbf{D_a^{-1}}\mathbf{D}_\eta \mathbf{W})\bar{\mathbf{r}}^{l_k}_m$$

$$- \mathbf{D_c}\, \mathbf{b}^{l_k}_m - \mathbf{D^{-1}_a}\mathbf{D}_\eta\, \lambda \mathbf{u}^k_\eta]$$

$$= \mathbf{D}_\eta^{-1}\mathbf{D_a}[\bar{\mathbf{E}}\bar{\mathbf{r}}^l_k - (\mathbf{D}_\eta\, \bar{\mathbf{u}}^k + \mathbf{D_c}\, \mathbf{b}^l_m)], \tag{43}$$

where $\mathbf{D}_\eta \bar{\mathbf{u}}^k = \mathbf{D}_\eta \mathbf{D}_1 \lambda \mathbf{u}^k = \mathbf{D}_a^{-1} \mathbf{D}_\eta \lambda \mathbf{u}^k$, $\bar{\mathbf{E}} = \mathbf{I} - (\mathbf{I} - \overline{\mathbf{D_c}})\bar{\mathbf{P}} = \mathbf{I} - \mathbf{D}_\eta \mathbf{D_a^{-1}}\mathbf{W} = \mathbf{D}_\eta + \mathbf{D_c} - \mathbf{D_a^{-1}}\mathbf{D}_\eta \mathbf{W}$. By setting $\frac{\partial \bar{O}^{l_k}}{\partial \bar{\mathbf{r}}^{l_k}_m} = 0$, we can get (36).

Thus, the optimal solution of minimizing objective function $\bar{O}^{l_k}$ is equivalent to the solution of subRW with label prior.

From (41), we can find that it consists of three components. The first two components are the smooth term and the unary term, which are similar to the components in (10). The last component corresponds to the label prior. By minimizing this component, the reaching probability $\bar{r}_{im}^{l_k}$ will be consistent with the label prior. Note that, if we only consider the label prior, the reaching probability $\bar{r}_{im}^{l_k}$ should satisfy the following proposition.

Proposition 1: If the probability density u_i^k of label l_k is larger than the probability densities of any other labels, i.e., $u_i^k > \max_{t \neq k} u_i^t$, then the corresponding reaching probability $\bar{r}_{im t}^{l_k}$ should be close to 1.

However, from the last component of (41), we can find that only when $u_i^k > \sum_{t=1, t \neq k}^{K} u_i^t$, $\bar{r}_{imt}^{l_k}$ will be close to 1. When $K > 2$, this situation is not expected. Thus, a proper adjustment is made to correct this problem. Denote $g_i^k = u_i^k + \max_{t \neq k} u_i^t$ and set $g_i = g_i^k$ for each label, then the last component of (41) will become:

$$\frac{1}{2} \sum_{i=1}^{N} \lambda u_i^k (\bar{r}_{im}^{l_k} - 1)^2 + \frac{1}{2} \sum_{i=1}^{N} \lambda (\max_{t \neq k} u_i^t) \bar{r}_{im}^{l_k \, 2}. \qquad (44)$$

By minimizing this component, $\bar{r}_{im}^{l_k}$ will satisfy **Proposition 1**. In practice, the above adjustment is made by setting $g_i = g_i^k$ in (39). More details are shown in **Algorithm 1**.

C. Noise Reduction

Adding the label prior may produce some noise. One solution is decreasing parameter λ. However, when λ is too small, the twig parts may be lost. Then we need to use the other strategy to reduce noise. Combining the label prior value for each node, we can get a coarse segmentation result:

$$CR_i = \arg\max_k u_i^k. \qquad (45)$$

Although there is much noise in this coarse segmentation, it mostly does not connect with the main part. So we can choose the connected regions with seeds from the coarse segmentation as the candidate regions. Then we only add the label prior into these candidate regions, which will keep the prior information of twigs and decrease the noise. Furthermore, the candidate regions are dilated to add more prior information into the boundary regions. Since the contrast of the prior information near boundaries is high, it will help to find correct boundaries. We denote the indicating vector of the dilated candidate regions as $\mathbf{cr}^k = [cr_i^k]_{N \times 1}$, i.e., $cr_i^k = 1$, if node i belongs to these regions; $cr_i^k = 0$, otherwise. Then the prior vector

IV. Experimental Results

In this section, we evaluate the performance of the proposed subRW on both synthetic and natural images. We compare our algorithm with state-of-the-art methods including RW [20], RWR [27], and LRW [40] in qualitative and quantitative evaluation. The implementation codes of these algorithms are offered by the original authors, and the suggested parameters in their papers are used to run the experiments. Our algorithm includes two main parameters: the leaving probabilities $c_1, c_2, \cdots, c_N$ and the label prior parameter λ. The leaving probabilities control the probability that a random walker reaches the staying nodes (seeds), which principally influence the regions without twigs. As mentioned before, when all of c_i are set as the same constant, the subRW is equivalent to the RWR. And a RWR with a proper restarting probability performs well in most images without twigs. Then we empirically set the leaving probabilities to a constant $c_i = 4e - 4$.

A. Complex Texture Problem

In this experiment, we compared our algorithm with the other algorithms on synthetic textured images. Each textured image includes four or five types of texture. The goal of segmentation is to extract the texture with twig parts from these images. The segmentation results in Fig. 4 (e) show that all the twigs are completely segmented out and the boundaries of the main part are also correctly detected by our algorithm. As shown in Fig. 4 (b)-(d), the other RW algorithms [20], [27], [40] do not perform well for these textured images. The RW method almost does not find the right boundaries since there are too many short noisy edges in textured images. These edges can be viewed as barriers which prevent a random walker to walk across them. When there are many barriers between an object node and the corresponding seeds, it is difficult for a random walker from this node to reach the right seed. Thus RW fails to segment many nodes of the object. Both RWR and LRW reduce the probability that a random walker walks on the original image graph, which thus reduces the influence of these barriers. Then the random walker in RWR or LRW will be more likely to find the correct boundaries. Fig. 4 (c) and (d) show the similar results by RWR and LRW, since they can be viewed as similar subRWs without label prior. In the top row, they perform well and only lose some twigs and produce some error boundaries. However, in the bottom row, they only detect very few right boundaries. This may be due to that the twigs are too long and complicated. Our algorithm still performs very well for this complicated situation. Our algorithm also reduces the influence of barriers by decreasing the walking probability on

$\mathbf{u}^k = [u_i^k]_{N \times 1}$ is reset as: $\mathbf{u}^k \leftarrow \mathbf{u}^k \quad \mathbf{cr}^k$, where $\quad$ means the corresponding elements of two vectors are multiplied respectively and a vector is returned with the same scale. For some images with a high contrast between the object and the background, it is not necessary to make this noise removal process. So we offer a selecting parameter γ for users to choose whether to perform this process.

Finally, we summarize the main steps of the proposed subMarkov random walk with label prior in **Algorithm 1**.

the original image graph. The label prior will help the random walker from a node of twigs to reach the right global 'seed' (i.e, prior node).

B. Qualitative and Quantitative Comparisons

We have also compared our algorithm with the other three well-known RW algorithms for natural images shown in Fig. 5 and Fig. 6. These images are taken from three datasets: the Berkeley segmentation dataset (BSD) [9], the
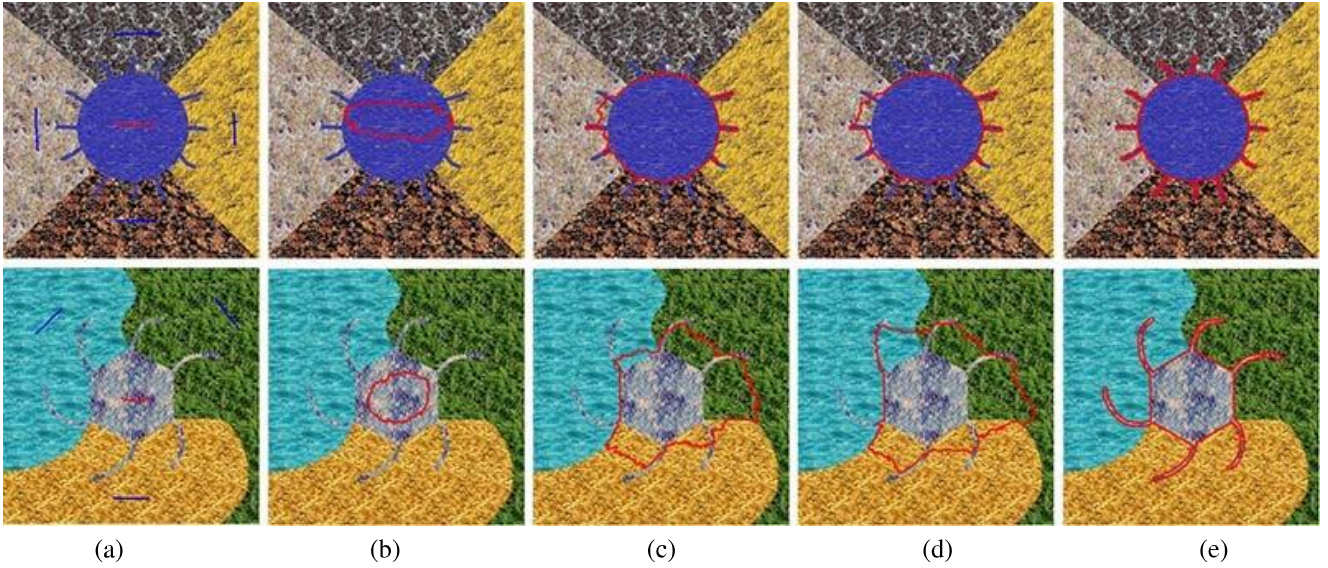
Fig. 4. Comparison between our algorithm and other RW algorithms on synthetic texture images. (a) The input scribbled images. (b), (c), (d) and (e) are the segmentation results by RW [20], RWR [27] with $c = 1e - 6$, LRW [40] with $1 - \alpha = 1e - 6$, and our subRW with $\gamma = 1$, $c_i = 4e - 4$, $\lambda = 4e - 11$.
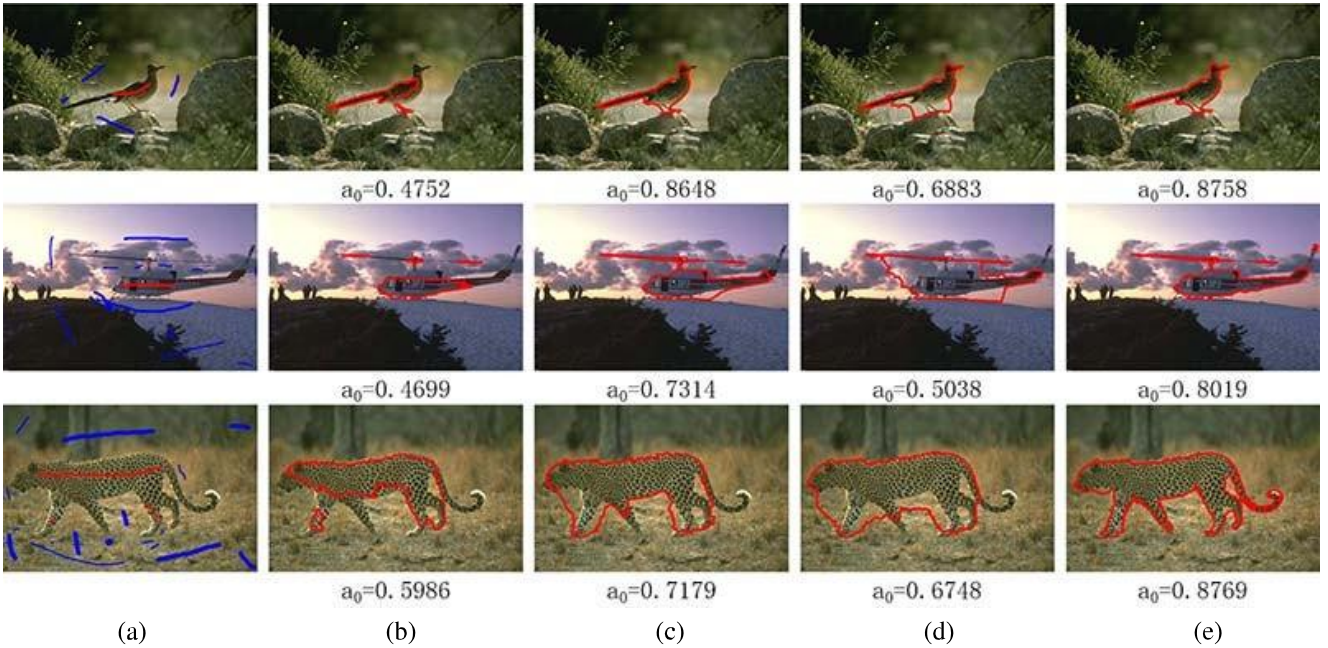

Fig. 5. Comparison between our algorithm and state-of-the-art algorithms. (a) Input scribbled images. (b), (c), (d) and (e) are segmentation results of RW [20], RWR [27] with $c = 4e - 4$, LRW [40] with $1 - \alpha = 1e - 4$, and our method with $c_i = 4e - 4$, $\lambda = 2e - 10$. The value of γ in our approach is empirically chosen, where $\gamma = 1$ for the top one image and $\gamma = 0$ for the bottom two images. If $\gamma = 1$, many twigs of the bottom three images will be lost.

Microsoft Research Cambridge object recognition image database (MSRC), and Visual Object Classes Challenge 2012 (VOC2012). The manual labeled ground-truth masks are provided in these three datasets, and the VOC2012 dataset only offers ground-truth masks of a part of images. However, the masks in MSRC and VOC2012 are not accurate enough or complete for our experiments, and the VOC2012 does not offer the masks of background. Thus, we only used the ground truth masks from BSD and labeled the new ground truth masks for a part of images from MSRC and VOC2012 ourselves.

Two metrics are used for the quantitative comparisons. The first one is a normalized overlap score a_0 [23] which measures the accuracy of the object segmentation results. The overlap region is formulated as: $a_0 = \frac{\tilde{R} \cap \tilde{G}}{\tilde{R} \cup \tilde{G}}$, where $\tilde{R}$ is the set of pixels assigned as the object from the segmentation result and $\tilde{G}$ is that from the ground-truth.

A higher score indicates better segmentation performance. This metric is effective for objective evaluation. However, it cannot fully reflect the evaluation mechanism of human visual system. It may ignore the connectivity or the coherence along the segmented boundaries. This limitation has been shown
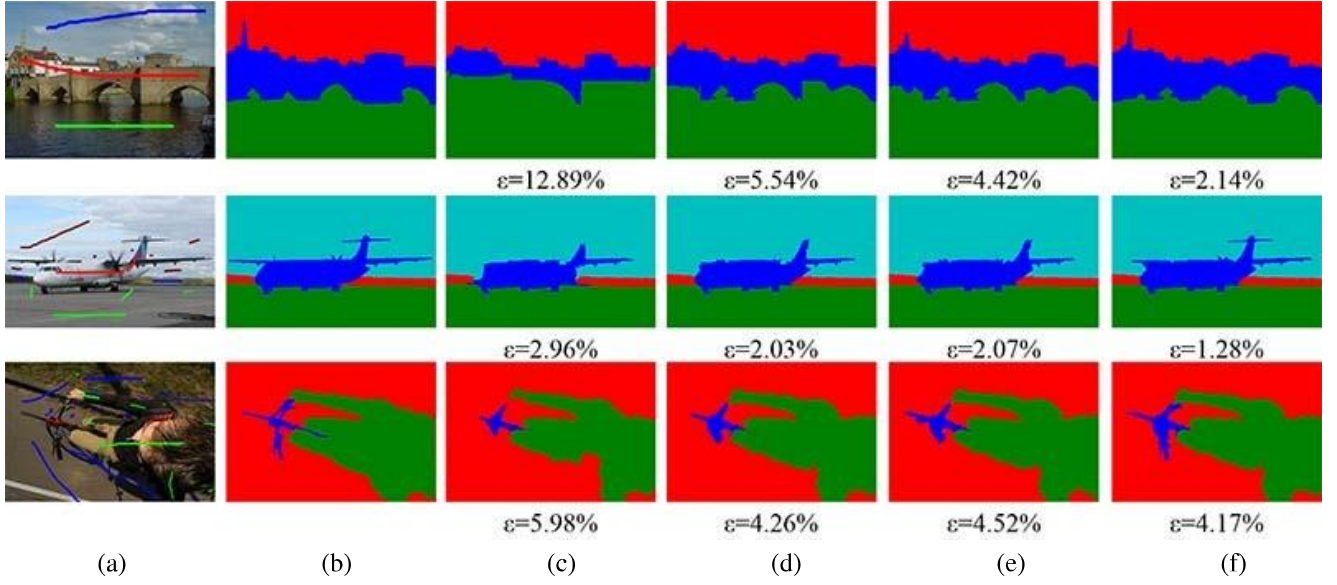
Fig. 6. Comparison results with multiple labels. (a) Input scribbled images. Top two images are from MSRC and others are from VOC2012. (b) Ground truth. (c)-(f) are segmentation results of RW [20], RWR [27] with $c = 4e - 4$, LRW [40] with $1 - \alpha = 1e - 4$, and our method with $c_i = 4e - 4$, $\lambda = 2e - 10$, $\gamma = 1$. The error ε denotes the percentage of incorrectly labeled pixels.

in [39]. To further verify the performance of our method, we adopt the second metric: the percentage of incorrectly labeled pixels ε, which is also called error rate. Let us denote

$L_A = [l_A^1, \cdots, l_A^N]^T$ and $L_B = [l_B^1, \cdots, l_B^N]^T$ as the sets of

labels from the segmentation results and from the ground-truth. Then, the ε is defined as follows: $\varepsilon = \frac{\sum_{i=1}^{N} l_A^i \oplus l_B^i}{N} \times 100$, where

$l_A^i \oplus l_B^i = 1$, if $l_A^i = l_B^i$; $l_A^i \oplus l_B^i = 0$, otherwise.

We evaluated our algorithm and other three algorithms on two segmentation problems. The first one is fore-ground/background segmentation, i.e., 2-label segmentation, which is the most common segmentation problem. And the second is multi-label segmentation. Both metrics are applied for measuring the quantitative performance. But the main met-ric is chosen by different segmentation problems. The overlap score a_0 measures the accuracy of the object, which is more suitable for the 2-label segmentation. In 2-label segmentation, more attention may be attracted by the foreground, i.e., the object in an image. The error rate ε indicates the segmentation performance of all labels (not only considering the object), which is more suitable for the multi-label segmentation. Thus, the overlap score a_0, as the main metric of 2-label segmen-tation, is directly shown in the qualitative results in Fig. 5. Similarly, the error rate ε is also directly shown in Fig. 6. Next, we will describe these experiments in detail.

In 2-label segmentation, we choose the natural images that only include one object with twigs. Some of them have complex textures like the cheetah, and others possess very thin twigs such as the helicopter. Fig. 5 shows that our algorithm outperforms the other algorithms no matter in quality or in quantity. In the qualitative comparisons, it is obvious that our method not only successfully segments out the most twigs of objects, but also adheres the edges to the boundaries better than the others. This is due to the added label prior which also has an impact on the main part of the object. In the

TABLE I
THE PERCENTAGE OF INCORRECTLY LABELED PIXELS
FOR 2-LABEL SEGMENTATION

	RW [20]	RWR [27]	LRW [40]	ours
1 1 1	6.68%	5.97%	5.94%	**0.90%**
airplane2	2.66%	2.66%	3.71%	**1.09%**
bird	1.65%	0.44%	1.33%	**0.40%**
elk	2.02%	1.64%	2.87%	**1.19%**
h l l t	3.33%	1.81%	5.40%	**1.26%**
h t h	7.90%	6.36%	8.23%	**2.71%**
boat	2.82%	4.19%	6.52%	**1.17%**
Average time (s)	**0.69**	2.12	2.11	2.41

quantitative comparisons, the advantage of our method is also significant such as the airplane and the cheetah in Fig. 5, where the twig parts are very small and our improvements over other algorithms are still very evident for these complicated images. The comparison of error rate is shown in Table I, which also verifies our improvements.

In multi-label segmentation, we also choose the natural images with twigs shown in Fig. 6. Some of them have multi-background, such as the backgrounds of the bridge include the water and the sky. Some of them contain multi-objects, like the bike image where the boy is also an object. The qualitative and quantitative results are shown in Fig. 6. Similar to 2-label segmentation, our algorithm still yields better performance than the other algorithms in multi-label segmentation. Most twigs are segmented out and the adherence of edges in the objects are better than others as well. For example, the tail and the wings of the airplane have been successfully segmented out by our algorithm, but the other methods lost these twigs. And for the legs of the cow, our algorithm has better edges' adherence. In quantitative comparisons, the error

rate in Fig. 6 and the average overlap score in Table II also show the significant improvement of our algorithm. Note that,

		RWR [27]	LRW [40]	ours
1 1 1		0.7317	5.94%	**0.7973**
b 1 l		0.7855	3.71%	**0.9184**
cow		0.7562	1.33%	**0 5**
airplane2		0.8642	2.87%	**0.9232**
plant		0.9169	5.40%	**0 93 (**
hike		0.6970	8.23%	**0.7424**
chair		0.5064	6.52%	**0.7125**
Average time (s)			1.51	2.57

the average overlap score is the average of all objects' overlap scores in one image. It is worth mentioning that the label prior of GMM may produce more noise than 2-label segmentation. If we do not reduce the noise (i.e., setting γ = 0), the segmentation results will be bad or even completely wrong. Thus, we reduce the noise by setting γ = 1 for most images. As mentioned before, this process may lose some twigs such as the airplane2 in Fig. 6. However, our algorithm still has better performance.

We also compare the run time among these algorithms. All the average run time in Tables I and II are measured in seconds with our unoptimized Matlab implementation on

a PC with Intel Core Processor i7-2600 CPU @3.40GHz with 6GB RAM. Our approach is slightly slower than RWR and LRW, since our method additionally builds label prior

for achieving better performance, which needs extra time to

construct GMMs. However, as shown in Fig. 6, our subMarkov RW algorithm significantly improve the seeded image seg-

mentation performance. The additional computational cost is worthy to obtain better segmentation performance using our subMarkov RW algorithm with Label Prior.

V. CONCLUSIONS

We have presented a novel framework based on the sub-Markov random walk for interactive seeded image segmen-

tation in this work. This framework can be explained as a

traditional random walker that walks on the graph by adding some new auxiliary nodes, which makes our framework easily interpreted and more flexible. Under this framework, we unify the well-known RW-based algorithms, which satisfy the sub-Markov property and build bridges to make it easy to transform the findings between them. Furthermore, we have designed a

novel subRW with label prior to solve the twigs segmentation

problems by adding prior nodes into our framework. The

experimental results have shown that our algorithm outper-forms the state-of-the-art RW-based algorithms. This also proves that it is practicable to design a new subRW algorithm by adding new auxiliary nodes into our framework. In the future, we will extend our algorithm to more applications,

such as centerline detection at 3D medical images [42] and

A. The Proof of the Matrix $\mathbf{E}$ in (8) Is Nonsingular

Before the proof, we give some definitions and an important theorem from http://en.wikipedia.org/wiki/Diagonally_dominant_matrix.

Definition 2: The matrix $\mathbf{A}$ is diagonally dominant if

$$|a_{ii}| \geq \sum_{j \neq i} |a_{ij}| \quad \text{for all } i, \tag{46}$$

where a_{ij} denotes the entry in the ith row and jth column.

Definition 3: If a strict inequality ($>$) is used in (46), then this matrix is said to be strictly diagonal dominant.

Theorem 1: A strictly diagonally dominant matrix (or an irreducibly diagonally dominant matrix) is non-singular, which is known as the Levy-Desplanques theorem [2].

Proof: According to **Theorem 1**, we only need to prove the matrix $\mathbf{E} = \mathbf{I} - (\mathbf{I} - \mathbf{D_c})\mathbf{P}$ is strictly diagonal dominant. We first give the formulation of each element in $\mathbf{E}$ as:

$$e_{ij} = \begin{cases} 1, & i = j, \\ -\dfrac{(1 - c_i)w_{ij}}{d_i}, & j \sim i, \\ 0, & otherwise. \end{cases} \tag{47}$$

Thus, if $0 < c_i \leq 1$, then $\forall i$,

$$\sum_{j \neq i} |e_{ij}| = \sum_{j \sim i} |e_{ij}| = \frac{1 - c_i}{d_i} \sum_{j \sim i} w_{ij}$$

$$= 1 - c_i < 1 = |e_{ii}| \tag{48}$$

So according to **Definition 3**, the matrix $\mathbf{E}$ is strictly diagonal dominant.

B. Unique Solution of (39)

Proof: If the coefficient matrix $\bar{\mathbf{E}} = \mathbf{I} - (\mathbf{I} - \mathbf{D_c})\bar{\mathbf{P}}$ is

non-singular, then the equivalent linear equations of (39) have unique solution, i.e., the solution of (39) is unique. So we only need to prove the matrix $\bar{\mathbf{E}}$ is non-singular. Similar to the above proof, we should prove $\bar{\mathbf{E}}$ is strictly diagonal dominant. We rewrite this matrix as follows:

$$\bar{e}_{ij} = \begin{cases} 1, & i = j, \\ -\dfrac{(1 - c_i)w_{ij}}{d_i + \lambda g_i}, & j \sim i, \\ 0, & otherwise. \end{cases} \tag{49}$$

Thus, if $0 < c_i \leq 1$, then $\forall i$,

$$\sum_{j \neq i} |\bar{e}_{ij}| \sum_{j \sim i} |\bar{e}_{ij}| = \frac{1 - c_i}{\ } \sum w_{ij}$$

classification [41].

$$\left| \sum_{j=i} \right| = \sum_{j \sim i} \frac{d_i + \lambda g_i}{d_i} \sum_{j \sim i}$$

$$= (1 - c_i) \frac{d_i}{d_i + \lambda g_i} < 1 = |\bar{e}_{ii}| \qquad (50)$$

Appendix A
Implementation Details

In this section, we describe our exact formulations with more detailed explanations for completeness.

According to **Definition 3**, the matrix $\bar{\mathbf{E}}$ is also strictly diagonal dominant.

REFERENCES

[1] H. Wechsler and M. Kidode, "A random walk procedure for texture discrimination," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 1, no. 3, pp. 272–280, Jul. 1979.

[2] R. A. Horn and C. R. Johnson, *Matrix Analysis*. Cambridge, U.K.: Cambridge Univ. Press, 1985.

[3] J. J. Dongarra, I. S. Duff, D. C. Sorensen, and H. A. Van Der Vorst, *Solving Linear Systems on Vector and Shared Memory Computers*. Philadelphia, PA, USA: SIAM, 1991.

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

[5] G. H. Golub and C. F. van Loan, *Matrix Computations*, 3rd ed. Baltimore, MD, USA: The Johns Hopkins Univ. Press, 1996.

[6] F. R. Chung, *Spectral Graph Theory*, vol. 92. Providence, RI, USA: AMS, 1997.

[7] L. Page, S. Brin, R. Motwani, and T. Winograd, "The PageRank citation ranking: Bringing order to the Web," Tech. Rep., 1999.

[8] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 8, pp. 888–905, Aug. 2000.

[9] D. Martin, C. Fowlkes, D. Tal, and J. Malik, "A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics," in *Proc. IEEE ICCV*, Jul. 2001, pp. 416–423.

[10] T. Jaakkola and M. Szummer, "Partially labeled classification with Markov random walks," in *Proc. NIPS*, vol. 14. 2002, pp. 945–952.

[11] X. Zhu, Z. Ghahramani, and J. Lafferty, "Semi-supervised learning using Gaussian fields and harmonic functions," in *Proc. 12th ICML*, vol. 3. 2003, pp. 912–919.

[12] D. Zhou and B. Schölkopf, "Learning from labeled and unlabeled data using random walks," in *Proc. 26th DAGM Symp.*, 2004, pp. 237–244.

[13] J.-Y. Pan, H.-J. Yang, C. Faloutsos, and P. Duygulu, "Automatic multimedia cross-modal correlation discovery," in *Proc. ACM SIGKDD*, 2004, pp. 653–658.

[14] J. Ham, D. D. Lee, S. Mika, and B. Schölkopf, "A kernel view of the dimensionality reduction of manifolds," in *Proc. IEEE ICML*, 2004, pp. 1–9.

[15] L. Grady and G. Funka-Lea, "Multi-label image segmentation for medical applications based on graph-theoretic electrical potentials," in *Proc. ECCV Workshops CVAMIA MMBIA*, 2004, pp. 230–245.

[16] D. Zhou, O. Bousquet, T. N. Lal, J. Weston, and B. Schölkopf, "Learning with local and global consistency," in *Proc. NIPS*, 2004, pp. 321–328.

[17] L. Grady, "Multilabel random walker image segmentation using prior models," in *Proc. IEEE CVPR*, Jun. 2005, pp. 763–770.

[18] M. Gori, M. Maggini, and L. Sarti, "Exact and approximate graph matching using random walks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 7, pp. 1100–1111, Jun. 2005.

[19] L. Yen, D. Vanvyve, F. Wouters, F. Fouss, M. Verleysen, and M. Saerens, "Clustering using a random walk based distance measure," in *Proc. 13th Eur. Symp. Artif. Neural Netw.*, 2005, pp. 317–324.

[20] L. Grady, "Random walks for image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 11, pp. 1768–1783, Nov. 2006.

[21] L. Grady and E. L. Schwartz, "Isoperimetric graph partitioning for image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 3, pp. 469–475, Mar. 2006.

[22] H. Tong, C. Faloutsos, and J.-Y. Pan, "Fast random walk with restart and its applications," in *Proc. ICDM*, 2006, pp. 613–622.

[23] A. K. Sinop and L. Grady, "A seeded image segmentation framework unifying graph cuts and random walker which yields a new algorithm," in *Proc. IEEE ICCV*, Oct. 2007, pp. 1–8.

[24] S. Calinon, F. Guenter, and A. Billard, "On learning, representing, and generalizing a task in a humanoid robot," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 37, no. 2, pp. 286–298, Apr. 2007.

[25] H. Qiu and E. R. Hancock, "Clustering and embedding using commute times," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 11, pp. 1873–1890, Nov. 2007.

[26] F. Fouss, A. Pirotte, J.-M. Renders, and M. Saerens, "Random-walk computation of similarities between nodes of a graph with application to collaborative recommendation," *IEEE Trans. Knowl. Data Eng.*, vol. 19, no. 3, pp. 355–369, Mar. 2007.

[27] T. H. Kim, K. M. Lee, and S. U. Lee, "Generative image segmentation using random walks with restart," in *Proc. ECCV*, 2008, pp. 264–275.

[28] L. Grady and A. K. Sinop, "Fast approximate random walker segmentation using eigenvector precomputation," in *Proc. IEEE CVPR*, Jun. 2008, pp. 1–8.

[29] J. Wang, F. Wang, C. Zhang, H. C. Shen, and L. Quan, "Linear neighborhood propagation and its applications," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 9, pp. 1600–1615, Sep. 2009.

[30] G. F. Lawler and V. Limic, *Random Walk: A Modern Introduction*. Cambridge, U.K.: Cambridge Univ. Press, 2010.

[31] Q. Mei, J. Guo, and D. Radev, "DivRank: The interplay of prestige and diversity in information networks," in *Proc. ACM KDD*, 2010, pp. 1009–1018.

[32] U. von Luxburg, A. Radl, and M. Hein. (2010). "Hitting and commute times in large graphs are often misleading." [Online]. Available: http://arxiv.org/abs/1003.1266

[33] V. Gopalakrishnan, Y. Hu, and D. Rajan, "Random walks on graphs for salient object detection in images," *IEEE Trans. Image Process.*, vol. 19, no. 12, pp. 3232–3242, Dec. 2010.

[34] L. J. Grady and J. Polimeni, *Discrete Calculus: Applied Analysis on Graphs for Computational Science*. Springer, 2010.

[35] C. Couprie, L. Grady, L. Najman, and H. Talbot, "Power watershed: A unifying graph-based optimization framework," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 7, pp. 1384–1399, Jul. 2011.

[36] S. Jegelka and J. Bilmes, "Submodularity beyond submodular energies: Coupling edges in graph cuts," in *Proc. IEEE CVPR*, Jun. 2011, pp. 1897–1904.

[37] X.-M. Wu, Z. Li, A. M. So, J. Wright, and S.-F. Chang, "Learning with partially absorbing random walks," in *Proc. NIPS*, 2012, pp. 3077–3085.

[38] P. Kohli, A. Osokin, and S. Jegelka, "A principled deep random field model for image segmentation," in *Proc. IEEE CVPR*, Jun. 2013, pp. 1971–1978.

[39] B. Ham, D. Min, and K. Sohn, "A generalized random walk with restart and its application in depth up-sampling and interactive segmentation," *IEEE Trans. Image Process.*, vol. 22, no. 7, pp. 2574–2588, Jul. 2013.

[40] J. Shen, Y. Du, W. Wang, and X. Li, "Lazy random walks for superpixel segmentation," *IEEE Trans. Image Process.*, vol. 23, no. 4, pp. 1451–1462, Apr. 2014.

[41] X. Zhu, W. Nejdl, and M. Georgescu, "An adaptive teleportation random walk model for learning social tag relevance," in *Proc. ACM SIGIR*, 2014, pp. 223–232.

[42] A. Sironi, V. Lepetit, and P. Fua, "Multiscale centerline detection by learning a scale-space distance transform," in *Proc. IEEE CVPR*, Jun. 2014, pp. 2697–2704.

[43] X. Dong, J. Shen, and L. Van Gool, "Segmentation using subMarkov random walk," in *Energy Minimization Methods in Computer Vision and Pattern Recognition*. Springer, 2015, pp. 237–248.

Xingping Dong is currently pursuing the Ph.D. degree with the School of Computer Science, Beijing Institute of Technology, Beijing, China. His current research interests include random walks and image segmentation.

Jianbing Shen (M'11–SM'12) is currently a Full Professor with the School of Computer Science, Beijing Institute of Technology, Beijing, China. His research interests include computer vision and multimedia processing. He has also obtained many flagship honors, including the Fok Ying Tung Education Foundation from the Ministry of Education, the Program for Beijing Excellent Youth Talents from Beijing Municipal Education Commission, and the Program for New Century Excellent Talents from the Ministry of Education. He serves as an Associate Editor of *Neurocomputing*.

Ling Shao (M'09–SM'10) is currently a Full Professor and the Head of the Computer Vision and Artificial Intelligence Group with the Department of Computer Science and Digital Technologies, Northumbria University, Newcastle upon Tyne, and an Advanced Visiting Fellow with the Department of Electronic and Electrical Engineering, University of Sheffield. His research interests include computer vision, image processing, pattern recognition, and machine learning. He is a fellow of the British Computer Society and IET, and a Life Member of ACM. He is an Associate Editor of the IEEE TRANSACTIONS ON IMAGE PROCESSING, and other journals.

Luc Van Gool received the degree in electromechanical engineering from the Katholieke Universiteit Leuven, in 1981. He became a Full Professor in 1998. He also became a Full Professor with ETH Zurich, Zurich, Switzerland. He leads computer vision research at both sites. He has authored over 200 papers in this field. He was the Program Chair of ICCV 2005 and the General Chair of ICCV 2011 and ECCV 2014. His main interests include 3D reconstruction and modeling, tracking and gesture analysis, and object recognition. He has received several best papers (David Marr Prize in 1998, CVPR Best Paper in 2007, ACCV in 2007, and ICRA in 2009). He is the Co-Founder of the Eyetronics, GeoAutomation, Kooaba, eSaturnus, and Procedural companies.