Multiple Random Walkers and Their Application to Image Cosegmentation

Chulwoo Lee Won-Dong Jang
School of Electrical Engineering
Korea University
{chulwoo, wdjang}@mcl.korea.ac.kr

Jae-Young Sim
School of ECE
UNIST
jysim@unist.ac.kr

Chang-Su Kim
School of Electrical Engineering
Korea University
changsukim@korea.ac.kr

Abstract

A graph-based system to simulate the movements and interactions of multiple random walkers (MRW) is proposed in this work. In the MRW system, multiple agents traverse a single graph simultaneously. To achieve desired interactions among those agents, a restart rule can be designed, which determines the restart distribution of each agent according to the probability distributions of all agents. In particular, we develop the repulsive rule for data clustering. We illustrate that the MRW clustering can segment real images reliably. Furthermore, we propose a novel image cosegmentation algorithm based on the MRW clustering. Specifically, the proposed algorithm consists of two steps: inter-image concurrence computation and intra-image MRW clustering. Experimental results demonstrate that the proposed algorithm provides promising cosegmentation performance.

1. Introduction

A random walk, a process in which a walker moves randomly from one node to another in a graph, can be used to analyze the underlying data structure of the graph. Many properties of a random walk are quantifiable algebraically based on graph theory [6], and can be solved by optimization tools efficiently. Therefore, random walks have been used in various graph-based learning tasks, including data mining [4, 35] and interactive image segmentation [11, 15]. A study in [18, 19] showed that spectral clustering [26] is also related to the random walk theory.

Whereas the conventional random walk theory describes the movements of a single walker (or agent), we propose a system of multiple random walkers (MRW) to simulate multiple agents on a graph simultaneously. Those agents traverse the graph according to a transition matrix, but they also interact with one another to achieve a desired goal. Our MRW system can support a variety of interactions by employing different restart rules. In particular, we develop the repulsive restart rule for data clustering. With this restart rule, as the random process continues, multiple agents repel one another and form their own dominant regions. Eventually, the power balance among the agents is achieved, and their distributions converge. By comparing the stationary distributions, clustering can be achieved. We demonstrate that this MRW process can cluster point data and segment real images reliably.

Moreover, we apply the proposed MRW system to the problem of segmenting similar images jointly. Recently, attempts to extract common foreground objects from a set of similar images have been made. This approach, called *cosegmentation*, was first addressed by Rother *et al*. [23] and has been researched actively [20, 12, 14, 3, 5, 30, 7, 24, 31]. Compared with segmenting each image independently, it is advantageous to delineate similar objects from multiple images. However, since repeating image features do not always imply the most important and informative parts of a scene, cosegmentation is still a challenging vision problem.

For cosegmentation, we introduce the notion of concurrence distribution, which represents the similarity of each node in an image to foreground objects in the other images. Then, the MRW clustering is performed by incorporating the concurrence distribution into the repulsive restart rule. Experimental results show that the proposed MRW algorithm improves the segmentation accuracy significantly, as compared with recent state-of-the-art cosegmentation techniques [14, 30, 24, 31], on the iCoseg dataset [3].

The rest of this paper is organized as follows: Section 2 reviews related work. Section 3 proposes the MRW system for clustering. Section 4 discusses the characteristics of the MRW clustering with image examples. Sections 5 develops the MRW cosegmentation algorithm, and Section 6 presents experimental results. Section 7 concludes this work.

2. Related Work

Data Clustering: The task of partitioning data into disjoint subsets based on the underlying data structure is a primitive activity to perceive information. The volume of data that people produce and process has grown rapidly, and discovering meaningful information from big data has become an essential subject. Since robust clustering is difficult to

978-1-4673-6964-0/15/$31.00 ©2015 IEEE

achieve because of noise and outliers in real-world data, various approaches have been attempted [32, 25].

Many phenomena in social and biological systems, as well as engineering problems, can be modelled with graphs, where nodes represent system elements and edges connect the elements to represent their relationships. Graph-based clustering hence has drawn much attention, and numerous algorithms have been proposed [26, 21, 34, 16, 29].

Image Cosegmentation: Rother *et al*. [23] first formulated the cosegmentation problem by incorporating the inter-image consistency into a Markov random field (MRF) energy function. Instead of the l_1-norm, Mukherjee *et al*. [20] used the l_2-norm to measure the similarity between foreground histograms. Hochbaum and Singh [12] developed a reward model satisfying the submodular condition and thus solved it efficiently using the graph cuts. These early techniques have been proposed to extract almost identical objects from different backgrounds.

Joulin *et al*. [14] considered the cosegmentation of foreground objects with more appearance variations, by formulating the cosegmentation as a discriminative clustering problem. Batra *et al*. [3] developed an interactive cosegmentation algorithm, which intelligently suggests to the user where to scribble next. They also made a cosegmentation dataset with manually-annotated groundtruth, called iCoseg, publicly available.

Collins *et al*. [7] used the random walk algorithm in [11] as a core segmentation tool, and achieved the cosegmentation by enforcing the constraint that foreground histograms should match one another. The proposed algorithm is also based on a random walker process. However, while [11] considers whether a single walker reaches a foreground or background seed first, the proposed algorithm employs multiple random walkers, who interact one another, and derives segmentation results from their stationary distributions.

Recently, several cosegmentation algorithms have been proposed to achieve robust correspondence between foreground objects. Chang *et al*. [5] used saliency models to exclude regions that infrequently appear across images. Vicente *et al*. [30] generated object proposals from each image, and used a random forest classifier to score a pair of proposals from different images. Rubio *et al*. [24] developed a region matching method to establish the correspondences between common objects among images. Wang *et al*. [31] used a functional map to represent consistent appearance relations between a pair of images, and jointly optimized the segmentation maps of all images.

3. Multiple Random Walkers

This section introduces the notion of MRW. We first describe the conventional random walk, which is a Markov process of a single random walker (or agent). We then propose the MRW system to simulate movements and interactions of multiple agents on a graph.

3.1. Single Random Walker

Let $G = (V, E)$ be an undirected, weighted graph. The node set V consists of data points $\mathbf{x}_i$, $i = 1, \ldots, N$. Edge e_{ij} in the edge set E connects $\mathbf{x}_i$ and $\mathbf{x}_j$. Also, let $\mathbf{W} \in \mathbb{R}^{N \times N}$ be a symmetric matrix, in which the (i, j)th element w_{ij} is the weight of e_{ij}, representing the affinity between $\mathbf{x}_i$ and $\mathbf{x}_j$. Typically, w_{ij} is defined as

$$w_{ij} = \begin{cases} \exp\left(-\frac{d^2(\mathbf{x}_i, \mathbf{x}_j)}{\sigma^2}\right) & \text{if } e_{ij} \in E, \\ 0 & \text{otherwise,} \end{cases} \tag{1}$$

where d is a dissimilarity function and σ^2 is a scale parameter.

A random walker travels on the graph G. The transition probability a_{ij} that the walker moves from node j to node i is obtained by dividing w_{ij} by the degree of node j, *i.e.*, $a_{ij} = w_{ij} / \sum_k w_{kj}$. In other words, the transition matrix $\mathbf{A} = [a_{ij}]$ is computed by normalizing each column of the affinity matrix $\mathbf{W}$. Let $\mathbf{p}^{(t)} = [p_1^{(t)}, \cdots, p_N^{(t)}]^T$ be a column vector, in which $p_i^{(t)}$ denotes the probability that the walker is found at node i at time instance t. We then have the temporal recursion

$$\mathbf{p}^{(t+1)} = \mathbf{A}\mathbf{p}^{(t)}. \tag{2}$$

If the graph has a finite number of nodes and is fully connected, $\mathbf{A}$ is irreducible and primitive [13]. Then, the walker has a unique stationary distribution $\boldsymbol{\pi}$, satisfying $\boldsymbol{\pi} = \mathbf{A}\boldsymbol{\pi}$, and $\boldsymbol{\pi} = \lim_{t \to \infty} \mathbf{p}^{(t)}$ regardless of an initial condition $\mathbf{p}^{(0)}$. The stationary distribution $\boldsymbol{\pi}$ conveys useful information about the underlying data structure of the graph, and it is thus exploited in vision applications [11, 10].

RWR [22] is a generalization of the random walk, in which the walker is compelled to return to specified nodes with a restart probability ϵ. The RWR recursion is

$$\mathbf{p}^{(t+1)} = (1 - \epsilon)\mathbf{A}\mathbf{p}^{(t)} + \epsilon\mathbf{r}, \tag{3}$$

where $\mathbf{r} = [r_1, r_2, \cdots, r_N]^T$ is the restart distribution with $\sum_i r_i = 1$ and $r_i \geq 0$. With probability $1 - \epsilon$, the walker moves ordinarily as in (2). On the other hand, with probability ϵ, the walker is forced to restart with the distribution $\mathbf{r}$. When $r_i = 1$ and $r_j = 0$ for all $j \neq i$, the stationary distribution of the RWR represents the affinity of each node to the specific node i. This property has been exploited in image segmentation [15] and data mining [22].

We note that RWR can be interpreted as the ordinary random walk as well. The RWR recursion in (3) can be rewritten as

$$\mathbf{p}^{(t+1)} = (1 - \epsilon)\mathbf{A}\mathbf{p}^{(t)} + \epsilon\mathbf{B}\mathbf{p}^{(t)} = \tilde{\mathbf{A}}\mathbf{p}^{(t)}, \tag{4}$$

where $\mathbf{B}$ is a square matrix, all columns of which are equal to $\mathbf{r}$, and $\tilde{\mathbf{A}} = (1-\epsilon)\mathbf{A} + \epsilon\mathbf{B}$ is the equivalent transition matrix. There exists i such that $r_i > 0$. Thus, $b_{ii} > 0$ and $\tilde{a}_{ii} > 0$. It is known in matrix analysis [13] that an irreducible matrix with at least one positive diagonal elements is primitive. Therefore, the RWR recursion yields a unique stationary distribution regardless of an initial condition, as long as the graph is connected (it does not need to be fully connected).

3.2. MRW

The conventional random walk in (2) or (3) describes the movements of a single agent. In contrast, we consider multiple agents who share the same graph and affect the movements of one another.

Suppose there are K agents on a graph. Let $\mathbf{p}_k^{(t)}$ denote the probability distribution of agent k at time t. Similar to (3), random movements of agent k are governed by

$$\mathbf{p}_k^{(t+1)} = (1-\epsilon)\mathbf{A}\mathbf{p}_k^{(t)} + \epsilon\mathbf{r}_k^{(t)}, \quad k = 1, \ldots, K. \tag{5}$$

Thus, with probability $1-\epsilon$, agent k travels on the graph according to the transition matrix $\mathbf{A}$, independently of the other agents. However, with probability ϵ, agent k visits the nodes according to the time-varying restart distribution $\mathbf{r}_k^{(t)} = [r_{k,1}^{(t)}, \cdots, r_{k,N}^{(t)}]^T$.

We can make the agents interact with one another, by determining the restart distribution as

$$\mathbf{r}_k^{(t)} = (1-\delta^t)\mathbf{r}_k^{(t-1)} + \delta^t \boldsymbol{\phi}_k(\mathcal{P}^{(t)}) \tag{6}$$

where the function $\boldsymbol{\phi}_k$ is referred to as the restart rule. It determines a probability distribution $\boldsymbol{\phi}_k$ from $\mathcal{P}^{(t)} = \{\mathbf{p}_k^{(t)}\}_{k=1}^K$, which is the set of the probability distributions of all agents at time t.

In (6), δ is a constant within $[0, 1]$, called the cooling factor. In an extreme case of $\delta = 0$, the restart distribution $\mathbf{r}_k^{(t)}$ becomes time-invariant, and the MRW recursion of each agent in (5) is identical with the RWR recursion in (3). In the other extreme case of $\delta = 1$, $\mathbf{r}_k^{(t)} = \boldsymbol{\phi}_k(\mathcal{P}^{(t)})$ does not directly depend on the previous distribution $\mathbf{r}_k^{(t-1)}$. Suppose that $0 < \delta < 1$. We have

$$\|\mathbf{r}_k^{(t)} - \mathbf{r}_k^{(t-1)}\|_\infty = \delta^t \|\boldsymbol{\phi}_k(\mathcal{P}^{(t)}) - \mathbf{r}_k^{(t-1)}\|_\infty \le \delta^t. \tag{7}$$

Thus, if $s \ge t \ge T$, $\|\mathbf{r}_k^{(s)} - \mathbf{r}_k^{(t)}\|_\infty \le \delta^T/(1-\delta)$. So each element in the restart distribution $\mathbf{r}_k^{(t)}$ is a Cauchy sequence in terms of time t. Since a Cauchy sequence in $\mathbb{R}$ is convergent, the restart distribution $\mathbf{r}_k^{(t)}$ converges to a fixed distribution $\mathbf{r}_k^{(\infty)}$. Therefore, as t approaches infinity, the MRW recursion in (5) becomes the RWR recursion, and agent k has a stationary distribution eventually. To summarize we have the following convergence theorem.

Theorem 1 *If the graph is connected and* $0 \le \delta < 1$*, each agent in the MRW system in* (5) *and* (6) *has a stationary distribution, given by*

$$\boldsymbol{\pi}_k = \lim_{t\to\infty} \mathbf{p}_k^{(t)}. \tag{8}$$

The term 'multiple random walks' was used in [2, 8], but their graph simulations are different from the proposed MRW system. They assume that multiple particles are independent of one another, or annihilate one another, or coalesce into one at a meeting node. Then, the expected time for covering all nodes, or annihilating all particles, or coalescing into a single particle is computed. On the contrary, in our system, multiple walkers adapt their movements based on other walkers' distributions. Then, we extract useful information from the stationary distributions of the multiple walkers.

3.3. Repulsive Restart Rule for Clustering

By designing the restart rule $\boldsymbol{\phi}_k$ in (6), we can simulate a variety of agent interactions to achieve a desired goal. As a particular example, we propose the repulsive restart rule for clustering data. For notational simplicity, let us omit the superscripts for time instances.

In the MRW system,

$$\mathbf{p}_k = [p(\mathbf{x}_1|\omega_k), \cdots, p(\mathbf{x}_N|\omega_k)]^T \tag{9}$$

where $p(\mathbf{x}_i|\omega_k)$ is the probability that agent k is found at node i. According to the Bayes' rule, the posterior probability is given by

$$p(\omega_k|\mathbf{x}_i) = \frac{p(\mathbf{x}_i|\omega_k)p(\omega_k)}{\sum_l p(\mathbf{x}_i|\omega_l)p(\omega_l)}, \tag{10}$$

which represents the probability that node i is occupied by agent k. The repulsive restart rule sets the ith element of $\boldsymbol{\phi}_k(\mathcal{P})$ as

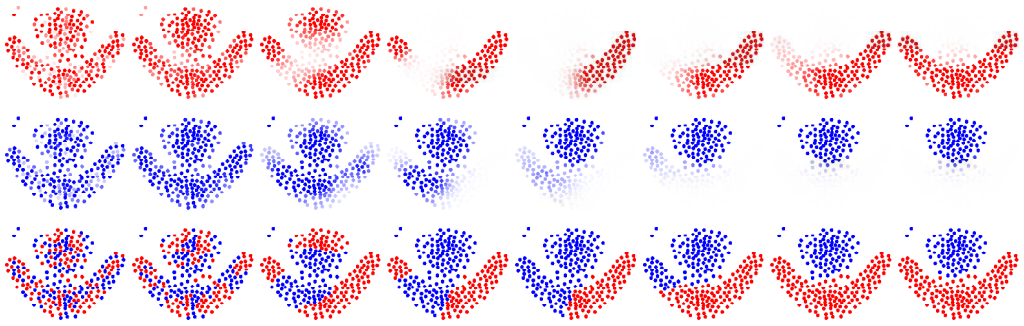
$$\phi_{k,i} = \alpha \cdot p(\omega_k|\mathbf{x}_i) \cdot p(\mathbf{x}_i|\omega_k) \tag{11}$$

where α is a normalizing factor to make $\boldsymbol{\phi}_k(\mathcal{P})$ a probability distribution. Suppose that agent k is dominant at node i, *i.e.*, it has a high posterior probability $p(\omega_k|\mathbf{x}_i)$ and a high likelihood $p(\mathbf{x}_i|\omega_k)$. Then, it restarts at that node with a high probability, and tends to become more dominant. This has the effect that a dominant agent at a node repels the other agents. The repulsive restart rule in (11) can be rewritten as

$$\boldsymbol{\phi}_k(\mathcal{P}) = \alpha\mathbf{Q}_k\mathbf{p}_k \tag{12}$$

where $\mathbf{Q}_k$ is a diagonal matrix whose (i,i)th element is the posterior probability $p(\omega_k|\mathbf{x}_i)$.

For clustering, we perform the MRW simulation in (5) and (6), by employing the restart rule in (12), to obtain the



(a) $t = 0$ (b) $t = 1$ (c) $t = 5$ (d) $t = 15$ (e) $t = 30$ (f) $t = 70$ (g) $t = 100$ (h) $t = 200$

Figure 1. Double random walkers with the repulsive restart rule. Red and blue walkers move interactively to divide a point set into two clusters. The top two rows depict the probability distributions of the red and blue walkers, and the bottom row shows the clustering results. The clustering decides that a point belongs to the red cluster, if the red walker has a higher probability at the point than the blue walker.

stationary distribution $\boldsymbol{\pi}_k$ of each agent k. Then, node i is assigned a cluster label l_i based on the maximum a posteriori (MAP) criterion,

$$l_i = \arg\max_k p(\omega_k|\mathbf{x}_i). \tag{13}$$

Figure 1 illustrates an MRW process of double random walkers with the repulsive restart rule. In Figure 1(a), the top two rows show initial probability distributions of the red and blue walkers, which are randomly generated. The bottom row is the result of the clustering. The clustering result at $t = 0$ is meaningless. However, as the iteration goes on, each walker repels the other walker, while forming a dominant cluster region. Consequently, the probability distributions of the two walkers converge, respectively, and the power balance between the walkers is achieved. By comparing those probabilities at each node, we obtain the clustering result in Figure 1(h), which coincides with the intuitive clustering of the human visual system.

4. MRW Clustering – Image Examples

This section illustrates how the MRW clustering algorithm with the repulsive restart rule can segment real images. With those image examples, we discuss the characteristics of the MRW algorithm.

4.1. Methodology

Given an input image, we first construct a graph $G = (V, E)$. The image is initially over-segmented into SLIC super-pixels [1], each of which becomes a node in V. For the edge set E, we use the edge connection scheme in [33]. Specifically, each node is connected not only to its spatial neighbors but also to the neighbors of the neighbors, and all boundary nodes at the image border are connected to one another.

For each edge e_{ij}, we determine the affinity weight w_{ij} in (1), by employing the dissimilarity function

$$d(\mathbf{x}_i, \mathbf{x}_j) = \sum_l \lambda_l d_l(\mathbf{x}_i, \mathbf{x}_j). \tag{14}$$

We use five dissimilarities d_l of node features, including RGB and LAB super-pixel means, boundary cues, bag-of-visual-words histograms of RGB and LAB colors [27]. We average those dissimilarities using empirically determined weights λ_l. Please refer to the supplemental materials for details. By normalizing each column of the affinity matrix $\mathbf{W} = [w_{ij}]$, we obtain the transition matrix $\mathbf{A}$.

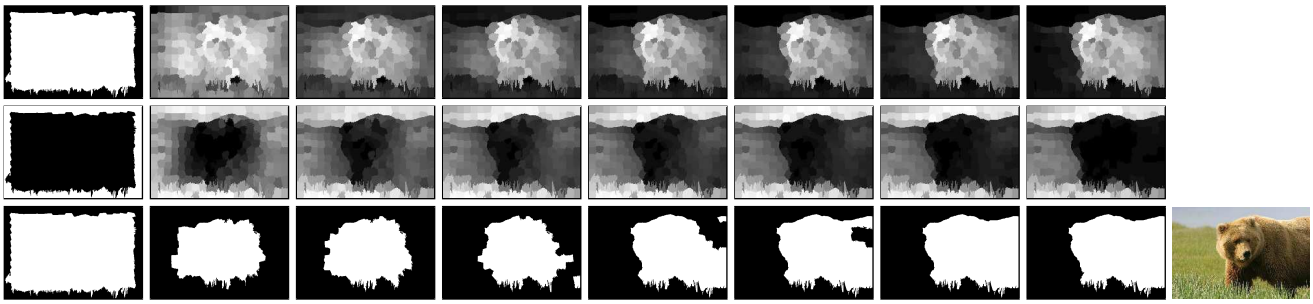
To achieve bilayer segmentation, we employ double random walkers, called foreground walker and background walker, whose probability distributions are denoted by $\mathbf{p}_\mathrm{f}$ and $\mathbf{p}_\mathrm{b}$, respectively. These two walkers make interactions according to

$$\begin{aligned} \mathbf{p}_\mathrm{f}^{(t+1)} &= (1-\epsilon)\mathbf{A}\mathbf{p}_\mathrm{f}^{(t)} + \epsilon\mathbf{r}_\mathrm{f}^{(t)} \\ \mathbf{p}_\mathrm{b}^{(t+1)} &= (1-\epsilon)\mathbf{A}\mathbf{p}_\mathrm{b}^{(t)} + \epsilon\mathbf{r}_\mathrm{b}^{(t)} \end{aligned} \tag{15}$$

with the repulsive restart rule. By **Theorem 1**, it is guaranteed that the probability distributions converge to stationary distributions $\boldsymbol{\pi}_\mathrm{f}$ and $\boldsymbol{\pi}_\mathrm{b}$, respectively. However, $\boldsymbol{\pi}_\mathrm{f}$ and $\boldsymbol{\pi}_\mathrm{b}$ depend on the initial distributions $\mathbf{p}_\mathrm{f}^{(0)}$ and $\mathbf{p}_\mathrm{b}^{(0)}$. Assuming the center prior that foreground objects tend to appear near the image center, we initialize $\mathbf{p}_\mathrm{f}^{(0)}$ and $\mathbf{p}_\mathrm{b}^{(0)}$ to be uniformly distributed at the interior nodes and the boundary nodes, respectively, as shown in Figure 2(a).

4.2. Examples

Figure 2 illustrates the repulsive MRW process in an image, which is shown in the lower right corner. In Figure 2(a), the top two rows show the initial probability distributions of the foreground and background walkers, based on the center prior, respectively. The bottom row is the segmentation result, based on the MAP decision in (13). At early stages, the foreground region is identified around



(a) $t = 0$ (b) $t = 3$ (c) $t = 10$ (d) $t = 15$ (e) $t = 20$ (f) $t = 25$ (g) $t = 30$ (h) $t = 50$

Figure 2. A repulsive MRW process of foreground and background walkers in an image. The input image is shown in the lower right corner. At each time instance t, from top to bottom, the probability distributions of the foreground and background walkers, $\mathbf{p}_{\mathrm{f}}$ and $\mathbf{p}_{\mathrm{b}}$, and the segmentation result are shown.

the image center due to the initial distributions. However, as the iteration continues, the foreground walker explores nearby similar nodes, while competing with the background walker. The repulsive restart rule facilitates discriminative clustering. Finally, in Figure 2(h), the probability distributions converge, and we obtain a satisfactory segmentation result that extracts the bear faithfully.

Figure 3 shows more examples. For comparison, we also provide the segmentation results of the normalized cuts (Ncut) algorithm [26] and the spectral k-means (SKM) algorithm [21]. Both Ncut and SKM are spectral clustering algorithms, which can be interpreted using the framework of the conventional random walks [18, 19]. Thus, the proposed MRW is related to Ncut and SKM. However, the proposed algorithm is different from the spectral clustering. Note that, using the definition of the transition matrix in [18, 19], the spectral clustering processes 'right' eigenvectors of the transition matrix for the clustering. On the contrary, the proposed algorithm uses the stationary distribution, which is a 'left' eigenvector of the transition matrix. Furthermore, whereas the conventional random walks consider only a single agent, the proposed algorithm employs multiple agents and exploits their interactions.

In Figure 3, we see that Ncut and SKM tend to divide images along the strongest edges. In contrast, the proposed algorithm extracts more meaningful foreground regions. We note that this is not a fair comparison, since the proposed algorithm assumes the center prior in the initialization of the foreground and background distributions. The objective of this comparison is to demonstrate different characteristics of the proposed algorithm from the spectral clustering, not to claim superior performance of the proposed algorithm.

The proposed algorithm is fully automatic, but it has common components with the interactive segmentation techniques [11, 15]. Especially, in [15], for each cluster label, the RWR recursion is performed by employing user scribbles as the restart distribution. The segmentation is achieved by comparing the stationary distributions for the

(a) Input (b) Ncut (c) SKM (d) RWR (e) MRW

Figure 3. Bilayer segmentation results of the proposed MRW clustering in comparison with the conventional approaches: Ncut [26], SKM [21], and RWR. In the RWR results, the initial distributions based on the center prior are used as the fixed restart distributions.

different labels. Thus, the initial distributions in the proposed algorithm can be regarded as those scribbles. But, whereas [15] assumes that user scribbles are completely trustable, the proposed algorithm automatically begins with rough initial guesses. As shown in Figure 2, the rough guesses at $t = 0$ are refined according to the interactions of the walkers, as t increases. In other words, while [15] uses fixed restart distributions, the proposed algorithm evolves restart distributions adaptively to image contents in a time-varying manner. Figure 3(d) shows the RWR segmentation results when the rough guesses based on the center prior are

used as the fixed restart distributions. RWR yields inferior performance to the proposed algorithm, since the walkers do not interact and the restart distributions are fixed.

5. Image Cosegmentation

We propose a novel algorithm for image cosegmentation based on the MRW system. The input is a set of images $\mathcal{I} = \{I_1, \cdots, I_Z\}$, which contain similar objects. The objective is to segment those objects jointly from all input images.

5.1. Initialization

For each image I_u in $\mathcal{I}$, we construct a graph independently of the other images, and adopt a foreground walker and a background walker. We use the graph construction scheme in Section 4.1. Let $\mathbf{p}_{\mathrm{f}(u)}$ and $\mathbf{p}_{\mathrm{b}(u)}$ denote the probability distributions of the foreground walker and the background walker in I_u, respectively. They are also initialized based on the center prior, as described in Section 4.1.

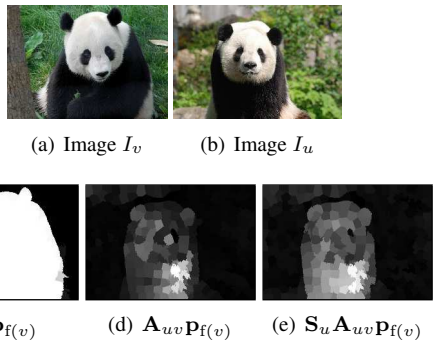
5.2. Inter-Image Concurrence Computation

To exploit the correlation among images, we compute the concurrence distribution $\mathbf{c}_{\mathrm{f}(u)}$ of the foreground walker, which indicates the similarity of each node in image I_u to foreground objects in the other images.

Let $\mathbf{W}_{uv} \in \mathbb{R}^{M \times N}$ be a matrix, in which the (i,j)th element w_{ij} represents the affinity from node j in image I_v to node i in image I_u. Here, M and N are the numbers of nodes in I_u and I_v, respectively. These inter-image affinities are computed between all nodes in I_v and all nodes in I_u. We compute each w_{ij} using a dissimilarity function, similar to (14). However, we use features, *e.g.* SIFT [17] and texton [9], which are widely used in object detection and classification. For more details, please refer to the supplemental materials. Then, we obtain the transfer matrix $\mathbf{A}_{uv}$ from I_v to I_u by normalizing each column in $\mathbf{W}_{uv}$.

The transfer matrix $\mathbf{A}_{uv}$ represents the correspondences from nodes in I_v to nodes in I_u probabilistically. It hence can transfer the foreground distribution $\mathbf{p}_{\mathrm{f}(v)}$ in I_v to I_u. The transferred distribution, $\mathbf{A}_{uv}\mathbf{p}_{\mathrm{f}(v)}$, can be used as an inter-image estimate of the object location in I_u. However, since the correspondences may be inaccurate especially on homogeneous regions, $\mathbf{A}_{uv}\mathbf{p}_{\mathrm{f}(v)}$ may be concentrated around a few nodes. For example, the foreground distribution $\mathbf{p}_{\mathrm{f}(v)}$ in Figure 4(c) is transferred to $\mathbf{A}_{uv}\mathbf{p}_{\mathrm{f}(v)}$ in Figure 4(d), but only the lower right part of the object is overly highlighted. To overcome this limitation, in image I_u, we perform the RWR recursion in (3) by employing $\mathbf{A}_{uv}\mathbf{p}_{\mathrm{f}(v)}$ as the restart distribution $\mathbf{r}$. Notice that the corresponding stationary distribution can be computed as $\mathbf{S}_u\mathbf{r} = \mathbf{S}_u\mathbf{A}_{uv}\mathbf{p}_{\mathrm{f}(v)}$, where

$$\mathbf{S}_u = \epsilon \left(\mathbf{I} - (1-\epsilon)\mathbf{A}_u\right)^{-1}. \tag{16}$$



(a) Image I_v (b) Image I_u

(c) $\mathbf{p}_{\mathrm{f}(v)}$ (d) $\mathbf{A}_{uv}\mathbf{p}_{\mathrm{f}(v)}$ (e) $\mathbf{S}_u\mathbf{A}_{uv}\mathbf{p}_{\mathrm{f}(v)}$

Figure 4. Transferring the foreground distribution $\mathbf{p}_{\mathrm{f}(v)}$ in image I_v to image I_u. For the purpose of illustration, the manually obtained ground-truth $\mathbf{p}_{\mathrm{f}(v)}$ is used in this example.

In Figure 4(e), we see that this new estimate $\mathbf{S}_u\mathbf{A}_{uv}\mathbf{p}_{\mathrm{f}(v)}$ approximates the object location in I_u more faithfully.

By integrating the inter-image estimates from all images, we obtain the concurrence distribution $\mathbf{c}_{\mathrm{f}(u)}$ of the foreground walker in I_u as

$$\mathbf{c}_{\mathrm{f}(u)} = \frac{1}{Z}\mathbf{S}_u \sum_v \mathbf{A}_{uv}\mathbf{p}_{\mathrm{f}(v)}, \tag{17}$$

where Z is the number of input images. In other words, $\mathbf{c}_{\mathrm{f}(u)}$ is the overall inter-image estimate of the object location in I_u, obtained from all images. We compute the concurrence distribution $\mathbf{c}_{\mathrm{b}(u)}$ of the background walker in a similar manner.

5.3. Intra-Image MRW Clustering

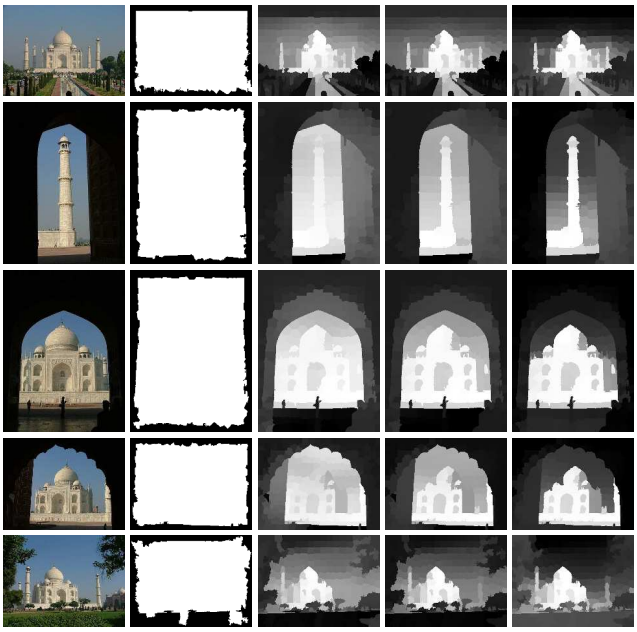
Next, we perform an MRW clustering process to refine the foreground and background distributions, $\mathbf{p}_{\mathrm{f}(u)}$ and $\mathbf{p}_{\mathrm{b}(u)}$, by employing the concurrence distributions $\mathbf{c}_{\mathrm{f}(u)}$ and $\mathbf{c}_{\mathrm{b}(u)}$.

The repulsive restart rule in (12) is effective for clustering nodes according to their intra-image feature vectors. On the other hand, the concurrence distributions $\mathbf{c}_{\mathrm{f}(u)}$ and $\mathbf{c}_{\mathrm{b}(u)}$ provide the inter-image estimates of object and background locations, respectively. Thus, to exploit both the intra and inter information, we define the hybrid restart rule for the foreground walker at image I_u as

$$\boldsymbol{\phi}_{\mathrm{f}(u)}\left(\{\mathbf{p}_{\mathrm{f}(u)}, \mathbf{p}_{\mathrm{b}(u)}\}\right) = \gamma\,\alpha\,\mathbf{Q}_{\mathrm{f}(u)}\mathbf{p}_{\mathrm{f}(u)} + (1-\gamma)\mathbf{c}_{\mathrm{f}(u)} \tag{18}$$

where $\mathbf{Q}_{\mathrm{f}(u)}$ is a diagonal matrix whose elements are the posterior probabilities of the foreground walker and α is a normalizing factor, as in (12). Also, γ is a parameter to controls the balance between the repulsive interaction and the concurrence preservation. In this work, γ is fixed to 0.3. We also define the hybrid restart rule $\boldsymbol{\phi}_{\mathrm{b}(u)}$ for the background walker in a similar manner.

Using these hybrid restart rules, we perform the iterative MRW process to obtain the stationary distributions $\boldsymbol{\pi}_{\mathrm{f}(u)}$



(a) Input (b) Initial (c) 1st pass (d) 2nd pass (e) 6th pass

Figure 5. The evolution of foreground distributions in the multi-pass clustering. As the passes go on, the Taj Mahal is more clearly highlighted.

and $\boldsymbol{\pi}_{\mathrm{b}(u)}$ of the foreground and background walkers. Then, from the stationary distributions, we obtain the posterior probabilities of the foreground and background walkers at each node. Finally, we can achieve the bilayer segmentation of image I_u based on the MAP criterion in (13).

5.4. Multi-Pass Refinement

A single execution of the two steps, *i.e.* (Step 1) inter-image concurrence computation and (Step 2) intra-image MRW clustering, provides satisfactory cosegmentation results in most cases. However, multiple executions of the steps are beneficial in some cases. In such multi-pass refinement, the resultant stationary distributions at Step 2 are used as the input for computing the concurrent distributions at Step 1 in the next execution.

Figure 5 exemplifies the multi-pass refinement. Figure 5(b) shows the initial foreground distributions. In the first pass, they are used to compute the stationary distributions of the foreground walkers in Figure 5(c). Those stationary distributions highlight not only the Taj Mahal but also less important regions, such as the sky. By feeding these distributions as new initial distributions in the second pass, we obtain the stationary distributions in Figure 5(d), and so on. Note that, as the passes go on, the Taj Mahal is more clearly highlighted.

We need a stopping criterion for the multi-pass refinement. The fundamental assumption of cosegmentation is that foreground objects in images should be similar to one another. In other words, every foreground node in image I_u should have a similar node in another image I_v. To quantify this property, let C_u denote the set of the nodes in I_u that are classified as the foreground. Then, the foreground distance $d_{\mathrm{f}}(C_u, C_v)$ from I_u to I_v is defined as

$$d_{\mathrm{f}}(C_u, C_v) = \sum_{i \in C_u} \min_{j \in C_v} d(\mathbf{x}_i, \mathbf{x}_j) \tag{19}$$

where d is the dissimilarity function used to compute the inter-image affinity $\mathbf{W}_{uv}$. Then, the overall foreground distance among all images is computed as $\sum_{u,v} d_{\mathrm{f}}(C_u, C_v)$. The multi-pass refinement is terminated when the overall distance stops decreasing.

Since each image is over-segmented into super-pixels to reduce the number of graph nodes, the resultant foreground and background distributions can be further refined at the pixel level using a bilateral filter [28]. Based on the spatial distances and the LAB color differences in an input image, we filter the foreground and background distributions. The segmentation accuracy is improved by about 1% with this pixel-level refinement. **Algorithm 1** summarizes the proposed cosegmentation algorithm.

Algorithm 1 Image Cosegmentation

Input: Graphs $\mathcal{G} = \{G_1, \cdots, G_Z\}$ for a set of input images $\mathcal{I}$

1: Initialize $\mathcal{P}_{(u)} = \{\mathbf{p}_{\mathrm{f}(u)}, \mathbf{p}_{\mathrm{b}(u)}\}$ for each I_u ▷ Section 4.1
2: **repeat** for each image I_u
3: Inter-image concurrence computation ▷ Section 5.2
4: Intra-image MRW clustering ▷ Section 5.3
5: Foreground extraction $\mathcal{C} = \{C_1, \cdots, C_Z\}$
6: Computation of the foreground distance $\sum_{u,v} d_{\mathrm{f}}(C_u, C_v)$
7: **until** the foreground distance stops decreasing
8: Pixel-level refinement

Output: Segmentation maps $\mathcal{C} = \{C_1, \cdots, C_Z\}$

6. Experimental Results

We test the performance of the proposed MRW cosegmentation (MRW-CS) algorithm on the iCoseg dataset [3], which is composed of 643 images of 38 object classes. For each object class, similar images were searched with the same theme, and sometimes were selected from the same user's photo album. This is a realistic scenario for image cosegmentation, which attempts to improve the segmentation accuracy of an input image by employing similar images. A segmentation accuracy is defined as the percentage of correctly labeled pixels.

MRW-CS performs the inter-image concurrence computation to exploit the correlation across images. If this step is skipped, MRW-CS is reduced to the MRW clustering in Section 4, which segments each image independently. Thus, we refer to the MRW clustering in Section 4 as the

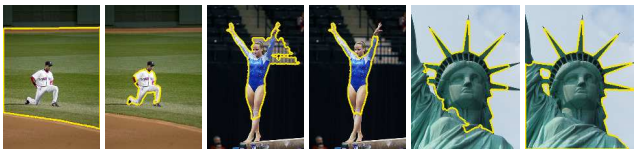


Figure 6. Pairs of segmentation results, obtained by MRW-IS (left) and MRW-CS (right).

Table 1. Cosegmentation performance on iCoseg data set.

	Class	# of images (used/total)	[14]	[30]	[24]	[31]	MRW IS	MRW CS
1	Alaskan bear	9/19	74.8	90.0	86.4	**90.4**	78.0	87.3
2	Balloon	8/24	85.2	90.1	89.0	90.4	64.1	**97.7**
3	Baseball	8/25	73.0	90.1	90.5	94.2	52.8	**97.1**
4	Bear	5/5	74.0	**95.3**	80.4	88.1	75.1	93.7
5	Elephant	7/15	70.1	43.1	75.0	86.7	68.1	**93.1**
6	Ferrari	11/11	85.0	89.9	84.3	**95.6**	83.7	91.9
7	Gymnastics	6/6	90.9	91.7	87.1	90.4	90.2	**96.1**
8	Kite	8/18	87.0	90.3	89.8	93.9	70.6	**95.7**
9	Kite panda	7/7	73.2	90.2	78.3	93.1	77.1	**96.0**
10	Liverpool	9/33	76.4	87.5	82.6	**89.4**	71.1	88.5
11	Panda	8/25	84.0	**92.7**	60.0	88.6	85.8	84.8
12	Skating	7/11	82.1	77.5	76.8	78.7	76.8	**91.6**
13	Statue	10/41	90.6	93.8	91.6	**96.8**	78.6	94.5
14	Stonehenge	5/5	56.6	63.3	87.3	92.5	77.3	**95.9**
15	Stonehenge 2	9/18	86.0	88.8	88.4	87.2	82.6	**90.7**
16	Taj Mahal	5/5	73.7	91.1	88.7	92.6	70.7	**95.2**
	Average		78.9	85.4	83.5	90.5	75.2	**93.1**

independent segmentation (MRW-IS). For both CS and IS, we fix the restart probability ϵ in (5) to 0.2, and the cooling factor δ in (6) to 0.95. Also, for computing posterior probabilities in (10), the prior probabilities for foreground and background walkers are set to $p(\omega_{\mathrm{f}}) = p(\omega_{\mathrm{b}}) = \frac{1}{2}$.

Figure 6 compares MRW-CS with MRW-IS. IS attempts to separate an object from its background, but the definition of 'object' is ambiguous. For instance, given the single image of 'Statue,' it is not clear whether the object should consist of the head only or both the head and body. Thus, IS may fail to delineate desired objects, especially in the cases of weak boundaries or background clutter. In contrast, CS overcomes the ambiguity, by extracting regions that occur repeatedly across images. In Figure 6, CS segments out the baseball player, the gymnast, and the statue faithfully.

Table 1 compares the accuracies of the proposed MRW-IS and MRW-CS with those of conventional algorithms. For each class, a number of images are selected for the cosegmentation. Since [30], the number has been fixed, but the combination of the images is unknown. Hence, for each class, the cosegmentation is performed with randomly selected images. Then, the accuracy is averaged over 20 random selections. Note that CS outperforms the conventional algorithms in most cases. On average, as compared with the state-of-the-art in [31], CS improves the accuracy by about 3%. Figure 7 demonstrates that the proposed MRW-CS yields fine results, even though individual images are challenging. Due to the page limitation, we provide more examples in the supplemental materials.

(a) Balloon

(b) Baseball

(c) Kite Panda

(d) Skating

(e) Statue

(f) Stonehenge

Figure 7. Examples of cosegmentation results. Best viewed in color.

7. Conclusions

We proposed the MRW system to simulate the interactions of multiple agents on a graph. To achieve desired interactions, a restart rule can be designed to determine the restart distribution of an agent according to the probability distributions of all agents. As a particular example, we developed the repulsive rule for data clustering. We discussed the characteristics of the MRW clustering with image examples. Moreover, we proposed an efficient image cosegmentation algorithm, composed of two main steps: inter-image concurrence computation and intra-image MRW clustering. Experimental results demonstrated that the proposed algorithm outperforms the state-of-the art techniques significantly on the iCoseg dataset. Future research issues include the exploration of more applications of the MRW system and the development of other restarting rules.

Acknowledgements

This work was supported partly by the Global Frontier R&D Program on Human-centered Interaction for Coexistence funded by the National Research Foundation of Korea (NRF) grant funded by the Korean Government (MSIP) (2011-0031648), and partly by the NRF grant funded by the Korea government (MSIP) (No. 2012-011031).

References

[1] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Susstrunk. SLIC superpixels compared to state-of-the-art superpixel methods. *IEEE TPAMI*, 34(11):2274–2282, Nov. 2012.

[2] N. Alon, C. Avin, M. Koucký, G. Kozma, Z. Lotker, and M. R. Tuttle. Many random walks are faster than one. In *Proc. ACM SPAA*, pages 110–128, 2008.

[3] D. Batra, A. Kowdle, D. Parikh, J. Luo, and T. Chen. iCoseg: Interactive co-segmentation with intelligent scribble guidance. In *Proc. IEEE CVPR*, pages 3169–3176, 2010.

[4] S. Brin and L. Page. The anatomy of a large-scale hypertextual web search engine. *Computer Networks and ISDN Systems*, 30(1):107–117, 1998.

[5] K.-Y. Chang, T.-L. Liu, and S.-H. Lai. From co-saliency to co-segmentation: An efficient and fully unsupervised energy minimization model. In *Proc. IEEE CVPR*, pages 2129–2136, 2011.

[6] F. R. K. Chung. *Spectral Graph Theory*. American Mathematical Society, 1997.

[7] M. Collins, J. Xu, L. Grady, and V. Singh. Random walks based multi-image segmentation: Quasiconvexity results and GPU-based solutions. In *Proc. IEEE CVPR*, pages 1656–1663, 2012.

[8] C. Cooper, A. Frieze, and T. Radzik. Multiple random walks in random regular graphs. *SIAM J. Discrete Mathematics*, 23(4):1738–1761, 2010.

[9] T. Deselaers and V. Ferrari. Global and efficient self-similarity for object classification and detection. In *Proc. IEEE CVPR*, pages 1633–1640, 2010.

[10] V. Gopalakrishnan, Y. Hu, and D. Rajan. Random walks on graphs for salient object detection in images. *IEEE Trans. Image Process.*, 19(12):3232–3242, 2010.

[11] L. Grady. Random walks for image segmentation. *IEEE TPAMI*, 28(11):1768–1783, 2006.

[12] D. Hochbaum and V. Singh. An efficient algorithm for co-segmentation. In *Proc. IEEE ICCV*, pages 269–276, 2009.

[13] R. A. Horn and C. R. Johnson. *Matrix Analysis*. Cambridge University Press, 1985.

[14] A. Joulin, F. Bach, and J. Ponce. Discriminative clustering for image co-segmentation. In *Proc. IEEE CVPR*, pages 1943–1950, 2010.

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

[16] Z. Li, J. Liu, S. Chen, and X. Tang. Noise robust spectral clustering. In *Proc. IEEE ICCV*, pages 1–8, 2007.

[17] C. Liu, J. Yuen, and A. Torralba. SIFT flow: Dense correspondence across scenes and its applications. *IEEE TPAMI*, 33(5):978–994, May 2011.

[18] M. Meila and J. Shi. Learning segmentation by random walks. In *Adv. Neural Inf. Process. Syst.*, pages 873–879, 2001.

[19] M. Meila and J. Shi. A random walks view of spectral segmentation. In *Proc. AISTATS*, 2001.

[20] L. Mukherjee, V. Singh, and C. Dyer. Half-integrality based algorithms for cosegmentation of images. In *Proc. IEEE CVPR*, pages 2028–2035, 2009.

[21] A. Y. Ng, M. I. Jordan, and Y. Weiss. On spectral clustering: Analysis and an algorithm. In *Adv. Neural Inf. Process. Syst.*, volume 2, pages 849–856, 2002.

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

[23] C. Rother, T. Minka, A. Blake, and V. Kolmogorov. Cosegmentation of image pairs by histogram matching - incorporating a global constraint into MRFs. In *Proc. IEEE CVPR*, pages 993–1000, 2006.

[24] J. Rubio, J. Serrat, A. Lopez, and N. Paragios. Unsupervised co-segmentation through region matching. In *Proc. IEEE CVPR*, pages 749–756, 2012.

[25] S. E. Schaeffer. Graph clustering. *Computer Science Review*, 1(1):27–64, 2007.

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

[27] J. Sivic and A. Zisserman. Video google: A text retrieval approach to object matching in videos. In *Proc. IEEE CVPR*, pages 1470–1477, 2003.

[28] C. Tomasi and R. Manduchi. Bilateral filtering for gray and color images. In *Proc. IEEE ICCV*, pages 839–846, 1998.

[29] S. M. van Dongen. *Graph Clustering by Flow Simulation*. Ph.D. dissertation, Universiteit Utrecht, 2001.

[30] S. Vicente, C. Rother, and V. Kolmogorov. Object cosegmentation. In *Proc. IEEE CVPR*, pages 2217–2224, 2011.

[31] F. Wang, Q. Huang, and L. Guibas. Image co-segmentation via consistent functional maps. In *Proc. IEEE ICCV*, pages 849–856, 2013.

[32] R. Xu and D. Wunsch II. Survey of clustering algorithms. *IEEE Trans. Neural Netw.*, 16(3):645–678, 2005.

[33] C. Yang, L. Zhang, H. Lu, X. Ruan, and M.-H. Yang. Saliency detection via graph-based manifold ranking. In *Proc. IEEE CVPR*, pages 3166–3173, 2013.

[34] L. Zelnik-Manor and P. Perona. Self-tuning spectral clustering. In *Adv. Neural Inf. Process. Syst.*, pages 1601–1608, 2004.

[35] D. Zhou, J. Weston, A. Gretton, O. Bousquet, and B. Schölkopf. Ranking on data manifolds. In *Adv. Neural Inf. Process. Syst.*, volume 16, pages 169–176. 2004.