Applied Soft Computing xxx (2016) xxx–xxx

Contents lists available at ScienceDirect

Applied Soft Computing

journal homepage: www.elsevier.com/locate/asoc

Image smoothing with generalized random walks: Algorithm and applications

Q1 Zhaobin Wang*, Hao Wang

Q2 *School of Information Science and Engineering, Lanzhou University, Lanzhou 730000, China*

ARTICLE INFO

Article history:
Received 28 May 2015
Received in revised form
18 December 2015
Accepted 5 January 2016
Available online xxx

Keywords:
Edge detection
Image decomposition
Image enhancement
Image smoothing
Random walks

ABSTRACT

A novel generalized random walks model based algorithm for image smoothing is presented. Unlike previous image smoothing methods, the proposed method performs image smoothing in a global weighted way based on graph notation, which can preserve important features and edges as much as possible. Based on the new random walks model, input image information and user defined smoothing scale information are projected to a graph, our method calculates the probability that a random walker starting at each pixel node position will first reach one of the pre-defined terminal node to achieve image smoothing, which goes to solving a system of linear equations, the system can be solved efficiently by lots of methods. Theoretical analysis and experimental results are reported to illustrate the usefulness and potential applicability of our algorithm on various computer vision fields, including image enhancement, edge detection, image decomposition, high dynamic range (HDR) image tone mapping and other applications.

1. Introduction

Q4 Image smoothing is one of the most fundamental and widely studied problem in computer vision. Image smoothing, also called image blurring. Traditional smoothing methods not only dissolve the noise, but also blur the important edges. To avoid this, edge preserving smoothing is proposed. Edge preserving smoothing method may be viewed as a compromise between smoothing and edge preserving. It wipes out tiny details and noise, and at the same time, preserves salient important edges. In the past few years, it has been successfully applied in the field of image processing such as edge detection, image restoration, image enhancement, and many other high-level image processing tasks. As an integral step of many computer vision problems, the results of image smoothing influence the performance of the whole vision system.

Our work is partially inspired by the well-known graph cut model [1], in this paper we propose a novel approach to edge preserving image smoothing with a similar graph structure. However, theoretical foundation of our work is based on random walks algorithm [2,3], the random walks algorithm is used for image segmentation by Grady, it solves this problem as following: given a random walker starting at each pixel position, what is the probability that it first reaches each of the pre-labeled nodes. A generalized random walks model for image smoothing is proposed. The smoothing problem is formulated on a carefully designed graph. First an image model is constructed, an input image is treated as a graph with a fixed number of vertices and edges. Each edge is assigned a positive value weight corresponding to image gradient. It indicates the likelihood that a random walker will cross that edge, which is the same as traditional random walks algorithm. Besides, two auxiliary nodes are added to the model and each pixel node is connected with the two nodes. Edges between the two nodes and pixel nodes are also assigned a weight to indicate that random walker will reach a terminal node from pixel node. By calculating the probability that the random walker first reaches one of the terminal nodes from each pixel node position, the smoothing result can be obtained.

The main contribution of the paper is to propose a new strategy that can project information of image structure and smoothing willing to a graph model. Our approach gives a new understanding of how edge preserving smoothing can be realized, which can fully explain the reason why image smoothing problem can be solved by random walks algorithm; Algorithmically, we propose a system of linear equations in an optimization framework for image smoothing and apply it to some interesting applications; The relationship between our method and generalized random walk with restart [4,5], anisotropic diffusion [6] and WLS [7] is also discussed.

The rest of the paper is organized as follows. Section 2 reviews previous methods. Differences between our algorithm and some existing image smoothing methods are also discussed. Section 3 provides implementation details of our method and theoretical connections with previous related work. Section 4 discusses experimental results and performance, along with comparisons with other smoothing methods. Finally, Section 5 gives the conclusions.

2. Prior work

Our work benefits from the rich body of work on image smoothing and random walk based applications. Some related work is reviewed in this section.

2.1. Prior image smoothing methods

Anisotropic diffusion model introduced by Perona and Malik is widely used in practice and extensively studied in theory [6], it is modeled using partial differential equations and implemented as an iterative process, it uses an edge-stopping function of local

Q3 * Corresponding author. Tel.: +86 9318912786.
E-mail address: zhaobin_wang@foxmail.com (Z. Wang).

http://dx.doi.org/10.1016/j.asoc.2016.01.003
1568-4946/

gradient to make smoothing take place only in the interior of regions without crossing edges. It has been demonstrated to be able to achieve a good trade-off between noise removal and edge preservation. There has been a large number of works aimed at optimizing and extending the idea [8,9]. Rudin et al. proposed to regularize total variation, which utilizes the gradient sparsity enforced by an L1 penalty term to do edge preserving smoothing [10]. It has been successful in denoising problems. It is the inspiration source of much work. By now, the algorithm has also been used for other restoration tasks such as deblurring, blind deconvolution and inpainting [11]. The bilateral filter computes a smoothed output with a weighted average of neighboring pixels intensities, it takes into account spatial and color distances [12]. It is an extension of typical Gaussian smoothing. There exist a number of methods to boost the algorithm [12,13]. Because of its simplicity and effectiveness, it is one of the most popular smoothing methods.

In recent years, many exciting new techniques have been invented to solve this problem. All the methods have demonstrated satisfying results with good performance. Farbman et al. proposed to perform the edge preserving smoothing using the weighted least square (WLS) framework [7,14], edge preserving smoothing is viewed as a compromise between data term and regularization term. By minimizing the proposed energy functional, image smoothing result can be obtained by solving a large linear system. Xu et al. presented L0 gradient minimization for image smoothing, it minimizes a specific objective function [15], it can remove low-amplitude structures and globally preserve and enhance salient edges. Subr et al. considered edge preserving smoothing as interpolation between local signal extremes [16]. The method defines detail as oscillations between local minima and maxima. It smoothes high contrast texture while preserving salient edges. Guided image filter assumes there is a local linear model between the guidance image and the filtering output, and they seek a solution that minimizes the difference between filtering input and output while maintaining the linear model [17].

Among different smoothing schemes, our proposed graph theory based method has several good features in practical applications. It explicitly organizes the image elements into mathematical structures, requires no discretization and therefore incurs no discretization errors or ambiguities, and makes the formulation of the problem more flexible and the computation more efficient [3].

2.2. Random walks

The term random walk was first proposed by Karl Pearson in 1905 [19]. In his letter to nature, Pearson proposed the problem of random walk. At each step, a man started from a point and moved a fixed length, with a randomly chosen angle. He wanted to know the distribution of the man after many steps had been taken. In the same year, Albert Einstein published his paper on Brownian motion which he modeled as a random walk [20]. It had an enormous impact, and it gave strong evidence for discrete particles at a time when most scientists believed that matter was a continuum. The concept of random walk has been used in many fields, and it is nearly ubiquitous in science and engineering, including ecology, economics, physics, chemistry, and biology [20,22,23].

Various properties of random walks can be used to define the specified algorithm. The average first-passage time which is defined as the average number of steps a random walker starting in a state will take to enter another state for the first time; the average commute time which is defined as the average number of steps a random walker starting in a state will take before entering another state for the first time and go back to starting state; the escape probability which is defined as the probability that a random walk starting at a state will reach another state before returning to starting state, they are applied in many applications [23–27].

Random walks proposed by Grady are an improvement to traditional random walks on graph by highlighting the diversity and centrality simultaneously [3]. It obtains the affinities between each node and labeled nodes by measuring probability that a random walker will first reach a labeled node before other labeled nodes. The probability problem shares the same solution as the harmonic functions with given boundary conditions. The harmonic function defined on graph can be easily solved by minimizing a combinatorial formulation of the Dirichlet integral, and through certain mathematical transformations, the final question goes to solving a system of linear equations. Now the algorithm has been successfully applied in many fields of image processing, e.g. image segmentation, image fusion, image annotation and classification, 2D–3D conversion and other applications [2,29,30].

3. Proposed smoothing method

3.1. Problem formulation

Unlike most previous image smoothing methods, we consider image smoothing as a problem: given a random walker starting at each pixel node position, what is the probability that he first reaches one of the two terminal nodes? As illustrated in Fig. 1, a walker starting from an image pixel node position, he may go directly to the terminal nodes or he may wander around his starting position before he reaches terminal nodes. At last, he will reach one of the terminal nodes before another. Our method calculates the probability that he will end up with reaching each terminal nodes, and takes each probability starting from each pixel positions as smoothed pixels intensities. The process is explained in detail below.

Suppose an input image I with n pixels is scaled to [01], image I can be written in a vector form $[I_1, I_2, \ldots, I_n]^T$. Image pixel lattices are represented in the form of a weighted undirected graph $G = (V, E, W)$, where V is a set of vertices $\{1, 2, \ldots, n\}$, represents image pixels with intensities $\{I_1, \ldots, I_n\}$, E is a set of edges $\{e_{ij}\}$ connecting two neighboring pixels I_i and I_j, W is a weighted adjacency matrix of input image, whose elements $\{w_{ij}\}$ are assigned to each edge e_{ij}. In this work, we use the typical Gaussian weighting function to calculate the weight given by

$$w_{ij} = e^{-\beta(I_i - I_j)^2}, \quad (1)$$

where I_i and I_j indicate the image intensity at pixel i and j, and β is a free parameter. A big weight value means a close relationship.

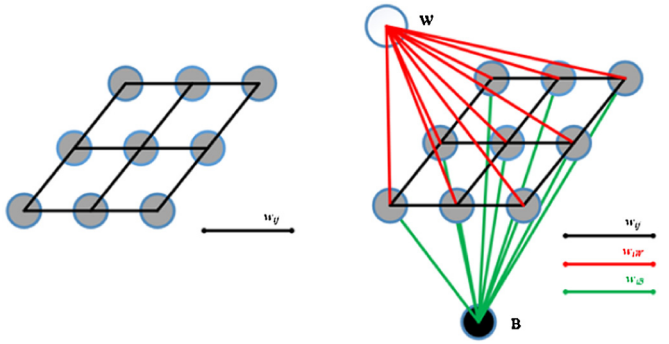

Fig. 1. Traditional random walk model (left) and the proposed model (right) for image smoothing (3 × 3 image). On the traditional graph model, each pixel is connected with its four neighbors (if it has). We added some other vertex and edges in our proposed graph model for image smoothing. Terminal nodes **W** and **B** are added and further connected with each pixel node. Random walkers starting from each pixel node position can walk to its surrounding pixel node position through the edge, in the long term walking process, he will wander among the pixels nodes before reaching terminal nodes. By calculating the probability walkers will first arrive terminal node **W** before **B**, we get our image smoothing result. Section 3 gives detailed instructions.

Please cite this article in press as: Z. Wang, H. Wang, Image smoothing with generalized random walks: Algorithm and applications, Appl. Soft Comput. J. (2016), http://dx.doi.org/10.1016/j.asoc.2016.01.003

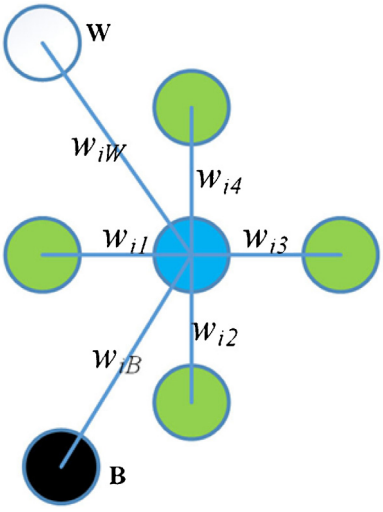

Fig. 2. Weight assignment. Take central blue pixel as an example. First, it has four edges connecting itself with its four neighboring pixels, the weight is calculated using Eq. (1); second, it has two edges connecting itself with terminal node **W** and **B**, the weight is defined as Eq. (3). Note that the main difference between our method with traditional random walks algorithm is the existence of edges between terminal nodes and pixel nodes, which adapts traditional random walks to more applications.

Degree d_i of vertex i is defined as the sum of the weights of the edges, that connect i with its neighboring pixel node,

$$d_i = \sum_{e_{ij} \in E} w_{ij}. \tag{2}$$

By inspecting edge weights of traditional graph model, it can be figured out that the model only contains intensity gradient information, while intensity information is ignored. In our proposed model, two auxiliary nodes are added and connected to pixel nodes, by properly setting the weights of the edges connecting terminal nodes to pixel nodes, we can map the intensity information to the graph model.

Fig. 1 shows the traditional graph model and the proposed model, terminal nodes **W** and **B** are added. Each pixel nodes is connected with terminal nodes, with weights

$$w_{iW} = \mu I_i d_i, \quad w_{iB} = \mu(1 - I_i)d_i \tag{3}$$

assigning to edges connecting image pixel node i to **W** and **B**, respectively. Fig. 2 shows weight assignment of the blue node, it has four edges connected to its four neighbors, which is assigned with w_{i1}, w_{i2}, w_{i3} and w_{i4}, the weight is calculated using above mentioned Gaussian weighting function equation (1), it is proportional to the probability that a random walker may cross the edge. Besides, it has two edges connected to terminal node, which is assigned with weights w_{iW} and w_{iB}.

The probabilities of each edges that will be crossed by a random walker are directly proportional to the weights assigned to them. So at each step, random walker has a probability p_n to go to his neighbouring node, and p_t to go to terminal nodes, where $p_t + p_n = 1$.

$$\begin{aligned} p_n &= \frac{\sum_{i=1}^{4} w_{ni}}{\sum_{i=1}^{4} w_{ni} + \mu I d_n + \mu(1 - I)d_n} \\ &= \frac{d_n}{(1+\mu)d_n} \\ &= \frac{1}{1+\mu} \end{aligned} \tag{4}$$

$$p_t = 1 - p_n = \frac{\mu}{1+\mu} \tag{5}$$

So we can imagine that if μ is infinite, then P_n will be zero, random walker starting from each pixel position has two ways to go, he can either go the way to **W** or **B**. As weight of one edge connected to terminal node **W** is $\mu I_i d_i$, and the other edges connected to terminal node **B** have a weight $\mu(1 - I_i)d_i$, so the probability that a random walker first reach terminal node **W** will be I_i, which is the same as its starting pixel intensity. This leads to an unsmoothed result.

However, if μ is not infinite, then the random walker starting from each pixel position has six paths to go, he can go the direction of terminal node **W** and **B** with probability P_t, or he can go to his neighboring pixels with probability P_n. It means in this situation, a random walker has a big probability to take more steps at wandering around his starting points before he reaches either terminal nodes. When goes to his neighbors, he has a different probability to first reach terminal node **W**. This means the probability that he first reaches terminal node **W** can be affected by his neighboring pixels. So in this situation we can obtain a smoothed result. As random walker always tends to go to his neighboring node which has a similar pixel intensity to his starting pixel, so our method has an edge preserving property.

Next, we will give a mathematical solution to the probability and a more detail theoretical analysis of the process.

3.2. Implementation

In a similar logic as traditional random walks algorithm, we consider **W** and **B** as two marked pixels, it is desired to find the probability that random walk starting from each pixel position reaches terminal node **W** before **B**. One possible solution is that we can start many random walkers at each pixel position and find the average number. This method is known as a Monte Carlo method [31]. It is a colorful way to solve the problem, but quite inefficient [32]. It is difficult to directly implement the idea, as the computation will be too expensive to afford. Luckily the combinatorial Dirichlet problem has the same solution as the desired random walker probability [33], so the probability can be efficiently solved by minimizing the combinatorial Dirichlet integral.

In this section, a combinatorial formulation of the Dirichlet integral is given, and steps of minimize the integral are reviewed and our method is deduced.

First, some used matrices are defined. The weighted adjacency matrix of the graph model is defined by:

$$W_{smooth} = \left[\begin{array}{cc:ccc} \multicolumn{2}{c:}{\multirow{2}{*}{0}} & w_{1W} & \cdots & w_{nW} \\ & & w_{1B} & \cdots & w_{nB} \\ \hdashline w_{1W} & w_{1B} & w_{11} & \cdots & w_{1n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ w_{nW} & w_{nB} & w_{n1} & \cdots & w_{nn} \end{array}\right], \tag{6}$$

where $w_{iW} = \mu I_i d_i$, $w_{iB} = \mu(1 - I_i)d_i$. Define D_{smooth} as diagonal matrix with degree of each node along the diagonal.

$$D_{smooth} = \left[\begin{array}{cc:ccc} d_W & & & & \\ & d_B & & & \\ \hdashline & & (1+\mu)d_1 & & \\ & & & \ddots & \\ & & & & (1+\mu)d_n \end{array}\right], \tag{7}$$

Please cite this article in press as: Z. Wang, H. Wang, Image smoothing with generalized random walks: Algorithm and applications, Appl. Soft Comput. J. (2016), http://dx.doi.org/10.1016/j.asoc.2016.01.003

So the combinatorial Laplacian matrix [34,35] of the proposed graph model can be described as

$$\begin{aligned} L_{smooth} &= D_{smooth} - W_{smooth} \\ &= \left[\begin{array}{c|ccc} d_W & & & \\ & d_B & & \\ \hline & & (1+\mu)d_1 & \\ & & & \ddots \\ & & & (1+\mu)d_n \end{array}\right] - \left[\begin{array}{c|ccc} 0 & w_{1W} & \cdots & w_{nW} \\ & w_{1B} & \cdots & w_{nB} \\ \hline w_{1W} & w_{1B} & w_{11} & \cdots & w_{1n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ w_{nW} & w_{nB} & w_{n1} & \cdots & w_{nn} \end{array}\right], \\ &= \left[\begin{array}{cc|c} d_W & & B \\ & d_B & \\ \hline & B^T & (1+\mu)D - W \end{array}\right] \end{aligned} \tag{8}$$

where D is a diagonal matrix with pixel node degree $\{d_i, \ldots, d_n\}$along the diagonal, and W is a weighted adjacency matrix of input image. The element w_{ij} of W is edge weight of pixel grid, it is calculated using Gaussian function as defined above, and

$$B = \begin{bmatrix} -w_{1W} & \cdots & -w_{nW} \\ -w_{1B} & \cdots & -w_{nB} \end{bmatrix}. \tag{9}$$

Then, a combinatorial formulation of Dirichlet integral can be formulated as:

$$E_{RW} = \frac{1}{2} g^T L_{smooth} g. \tag{10}$$

Last, above energy functional is minimized, Eq. (10) can be decomposed into block form for analysis:

$$\begin{aligned} E_{RW} &= \frac{1}{2}\begin{bmatrix} f_{ter}^T & f^T \end{bmatrix}\left[\begin{array}{cc|c} d_W & & B \\ & d_B & \\ \hline B^T & & (1+\mu)D - W \end{array}\right]\begin{bmatrix} f_{ter} \\ f \end{bmatrix} \\ &= \frac{1}{2}(f_{ter}^T \begin{bmatrix} d_W & \\ & d_B \end{bmatrix} f_{ter} + 2 f^T B^T f_{ter} + f^T((1+\mu)D - W)f) \\ &= f^T B^T f_{ter} + \frac{1}{2} f^T((1+\mu)D - W)f^T + const \end{aligned} \tag{11}$$

where *const* means a constant number which is independent of f, f_{ter} and f represent intensities of two terminal nodes and n image pixels. f_{ter} is the intensities of two terminal nodes, which is 1 and 0 respectively. It is the initial state. So E_{RW} can be further simplified as

$$\begin{aligned} E_{RW} &= f^T \begin{bmatrix} -w_{1W} & -w_{1B} \\ \vdots & \vdots \\ -w_{nW} & -w_{nB} \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} + \frac{1}{2} f^T((1+\mu)D - W)f + const \\ &= f^T \begin{bmatrix} -w_{1W} \\ \vdots \\ -w_{nW} \end{bmatrix} + \frac{1}{2} f^T((1+\mu)D - W)f + const \\ &= -\mu f^T D I + \frac{1}{2} f^T((1+\mu)D - W)f + const \end{aligned} \tag{12}$$

where f is the smoothed pixel intensity. In fact, f is the only critical point that minimizes the E_{RW}, so it can be obtained by differentiating E_{RW} to f and the final question goes to solve the equation:

$$-\mu D I + ((1+\mu)D - W)f = 0. \tag{13}$$

3.3. Computation complexity analysis

Eq. (14) can be solved using matrix inversion in principle by solving following equation,

$$f = \mu[(1+\mu)E - D^{-1}W]^{-1} I \tag{14}$$

However, it costs $O(n^3)$ operations, where n is the number of pixels. And applying the inverse matrix requires $O(n^2)$ operations [7]. Solving the sparse linear equation system equation (15) is more computational efficient. Above equations is a system of linear equations with n unknowns.

$$\frac{1}{\mu}[(1+\mu)E - D^{-1}W]f = I \tag{15}$$

The resulting linear system is a form of $Ax = b$, and the above equation is nonsingular [36]. It is easy to be solved using modern numerical linear algebra. The basic steps of our algorithm are summarized in Algorithm 1.

Algorithm 1. Basic steps of the proposed smoothing algorithm
1: Input I, β, μ.
2: Calculate weights w_{ij} of each edges according to Eq. (1).
3: Construct the weighted adjacency matrix W of input image I.
4: Construct the diagonal matrix D with degree of each node along the diagonal.
5: Solve Eq. (13).
6: Output smoothed image f.

Besides, to some extents, we have to say that the solution of above equations for image smoothing sometimes is predictable. As the smoothed image is always near the initial image, especially when the smoothing scale is small. So this equation is suggested to be solved by iterative solvers. These solvers have the advantage of a small memory requirement and the ability to represent the matrix vector multiplication as a function. When we set the image to be smoothed as the initial value, the computation speed can be further improved. Many good methods exist on the solution to large, sparse, symmetric, linear systems of equations. An appropriate preconditioner or a multi-grid solver requires only $O(n)$ operations.

Random walk problem has a close connection with electrical circuits [32,37]. Specifically, in our proposed model. If we replace edge weight of the proposed model with the same value conductor, and add a voltage source between terminal nodes, the voltage of each pixel position will be the same as our smoothing pixel intensity. Fig. 3 shows the corresponding relationship. The direct correspondence between our proposed method and analog electric circuits make a hardware (e.g., VLSI) implementation possible [3].

3.4. Parameter setting

In this part, we briefly discuss parameter setting in our model. As pointed out in Eq. (15), our smoothing result can be expressed as applying the operator $\mu[(1+\mu)E - D^{-1}W]^{-1}$ to the input image vector. Each row of the operator matrix may be thought as a kernel that determines the weight value of the corresponding pixel, output smoothing result can be thought as a weighted combination of other pixels in the input image. Fig. 6 gives weight distribution of our smoothing method. Reasonable parameter setting is vital important for the method working at its best situation.

First, we would like to talk about the 4 neighbors system of our model. A pixel node is connected to his four neighbors, in our experiments, it is computational efficiency and the results are satisfactory. Of course, we can apply 8 neighbors or more neighbors

Please cite this article in press as: Z. Wang, H. Wang, Image smoothing with generalized random walks: Algorithm and applications, Appl. Soft Comput. J. (2016), http://dx.doi.org/10.1016/j.asoc.2016.01.003

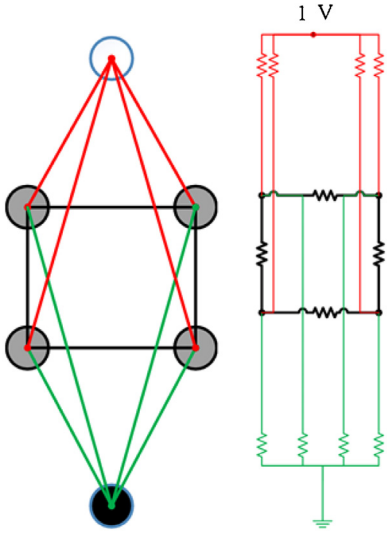

Fig. 3. Model used in the paper (2×2 image) and its corresponding electric circuits. Resistors represent the inverse of the corresponding edge weights. Fix the potential of terminal node **W** to unity (1 V) and set to zero (ground) the remaining terminal node **B**. the electric potentials calculated represent the probability that a random walker starting from each pixel node first reaches terminal node **W**.

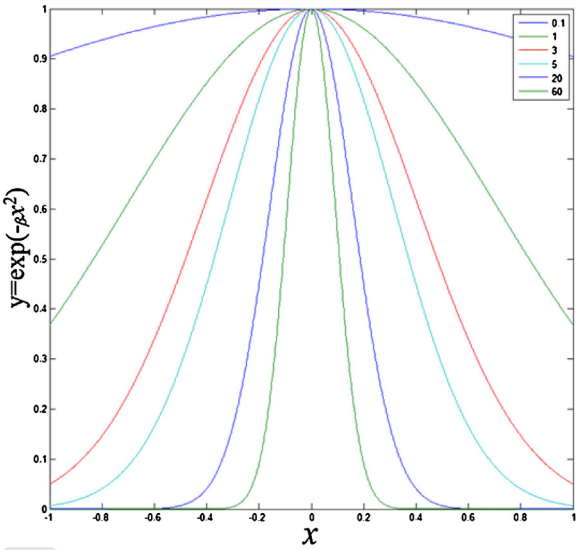

Fig. 4. Gaussian weighting function Eq. (1) with different β.

system, but computational burden increases much while the results do not improve that much. All the experimental results in this paper are acquired using 4 neighbors system [14].

In this paper, we use a typical Gaussian function for edge weight calculation. However, it is more appropriate to modify the function to texture information, filter coefficients of other image features when applying our methods to other specific problems [3]. Fig. 4 shows the plot of Gaussian function with different β. When β is small, for example 0.1, edge weights connecting to four neighbors have a similar value, which can be seen from Fig. 4, all the weights of edges are between 0.9 and 1, the intensity differences between neighboring pixels are ignored. The walker has almost the same probability to go to his four neighbors. In this situation, our method performs similar to a Gaussian filter. Fig. 5 shows the situation, when μ takes different values, our method acts like a Gaussian filter with different radius. With the increasement of β, our method gradually performs as an edge preserving filter. When β is appropriately big, the random walker will avoid crossing sharp intensity gradients. This makes our algorithm have a strong ability to preserve salient edges. Fig. 6 shows the situation. Smoothing kernel is calculated for each labeled pixel position. When β equals 0.1, it is similar to a Gaussian filter; When β equals 100, our smoothing method is edge aware; When β equals 50, it has a combined effects of 0.1 and 100. Fig. 7 shows a more detailed effect of varying parameters.

Let us consider a situation, if terminal node in our graph model is a kind of special node, whenever a random walker reaches it, the walker would be absorbed, he never walks again. For a walker walks on such a graph model, when he first reaches terminal nodes, he is absorbed. So the probability the walker first reaches **W** is the same as the probability the walker is absorbed by **W**, they share the same solution.

In our model, the walker has a solid possibility to be absorbed by terminal node in each step, and the probability is $p_t = \mu/(1+\mu)$ as computed in Eq. (5). So no matter where random walker starting from, the mean step number he will take before he is absorbed by terminal node is determined by μ. When μ is big enough, there is almost no smoothing effect. As random walker tends to directly go the way of terminal node direction. He is absorbed at the very first step. When μ decreases, the random walker starting from each pixel position tends to wander in a larger radius, random walker can take more steps before he is absorbed, the smoothing scale gradually improves. When μ tends to zero, the random walker starting from each pixel position tends to wander in the whole image lattice before he is absorbed, finally the probability that the walker first reaches terminal node **W** will no longer depend on his starting pixel position, it is a weighted result of all image pixels, the smoothing output will be a constant image.

4. Algorithm analysis

In this section, different properties of our algorithm are analyzed. This part will begin with establishing relationships with anisotropic diffusion [6] and discusses the relationship of our graph

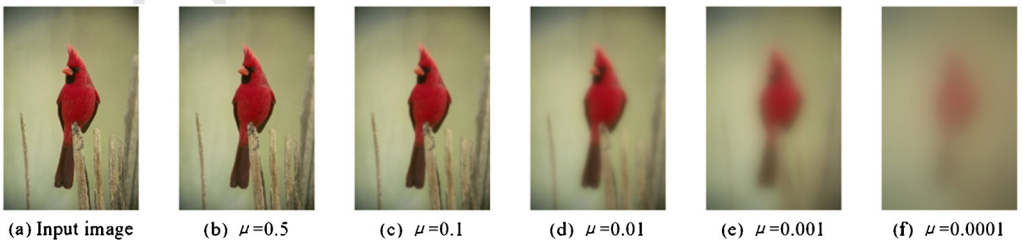

Fig. 5. Gaussian filter effect. Set $\beta = 0.1$, the left free parameter μ in weighting function controls smoothing scale. In this situation, when β is small, all edge weights calculate using Gaussian weighting function are basically equal. Random walkers walk on such a graph have no biased direction, in which case, proposed method has no edge preserving property.

Please cite this article in press as: Z. Wang, H. Wang, Image smoothing with generalized random walks: Algorithm and applications, Appl. Soft Comput. J. (2016), http://dx.doi.org/10.1016/j.asoc.2016.01.003

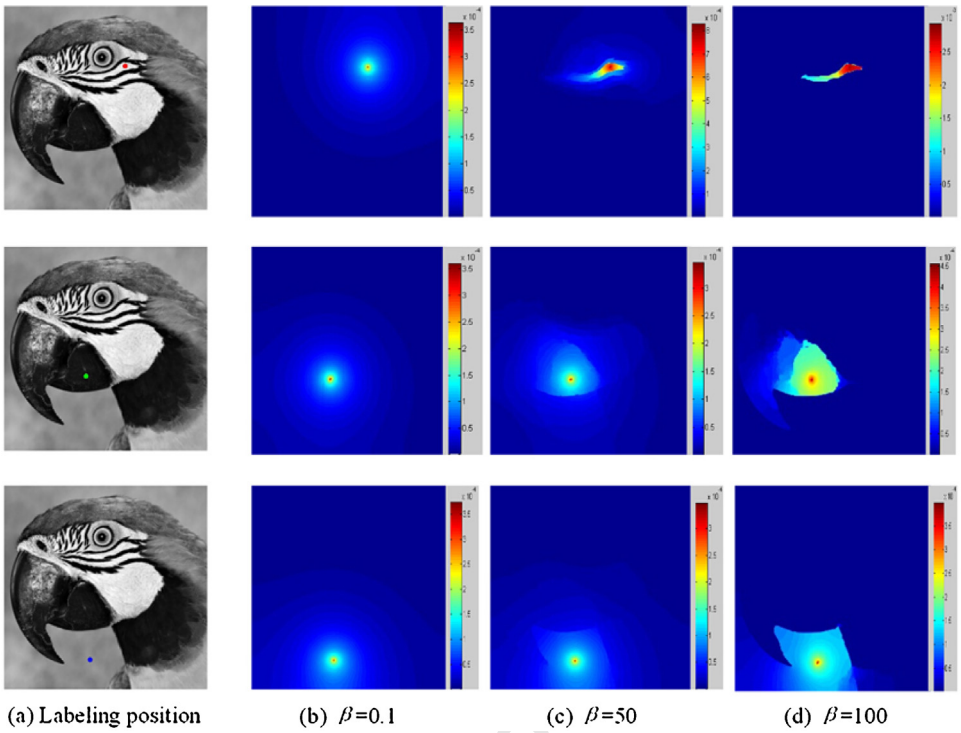

Fig. 6. Smoothing kernel of the proposed method with different β. $\beta = 0.1$, 50, 100 are computed for the labeled pixel in the Parrot image. $\mu = 1E{-}3$. When β equals 0.1, it is similar to a Gaussian filter; when β equals 100, the random walker will avoid to cross sharp intensity gradients, it has a strong ability to preserve salient edges; When β equals 50, it has a combined effects of $\beta = 0.1$ and $\beta = 100$.

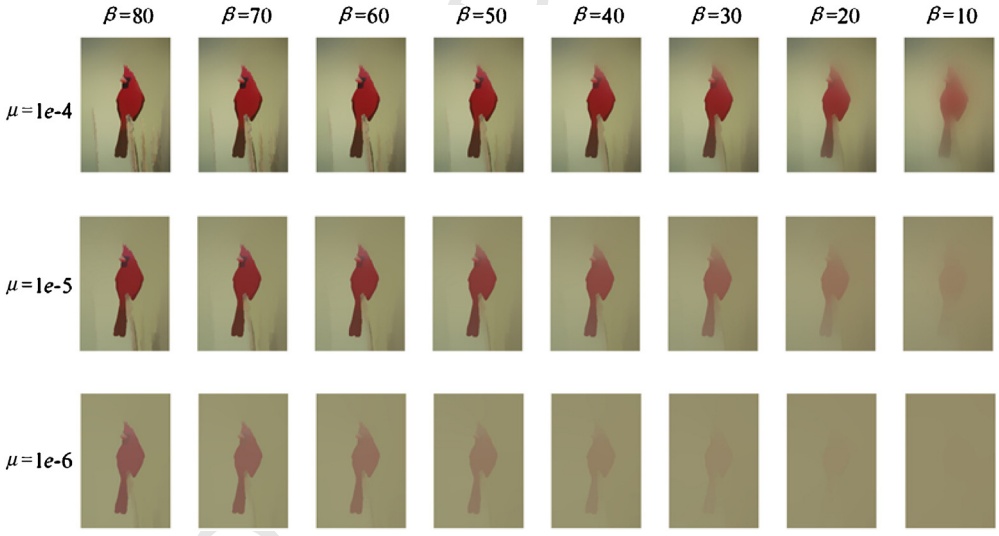

Fig. 7. Effect of varying parameters. μ controls the extent of smoothing; β controls edge preserving property.

smoothing algorithm to generalized random walk with restart (GRWR) and WLS [4,7].

4.1. Relationship to anisotropic diffusion

In fact, solving a system of equation (14) using Jacobi iteration can be thought as a kind of anisotropic diffusion.

$$I(k+1) = \frac{1}{1+\mu} D^{-1} W I(k) + \frac{\mu}{1+\mu} I(0), \tag{16}$$

where $I(0)$ is the initial state of iteration process (i.e. input image). An iterative solver of the proposed algorithm can be summarized as Algorithm 2. For each pixel

$$\begin{aligned} I_n(k+1) &= \frac{1}{1+\mu} \sum_{i=1}^{4} \frac{w_{ni}}{d_n} I_{ni}(k) + \frac{\mu}{1+\mu} I_n(0) \\ &= \frac{1}{1+\mu} \sum_{i=1}^{4} p_{ni} I_{ni}(k) + \frac{\mu}{1+\mu} I_n(0) \end{aligned} \tag{17}$$

where $I_{ni}(k)$ is the ith neighboring pixel of $I_n(k)$. While anisotropic diffusion can be discretized in a similar way:

Please cite this article in press as: Z. Wang, H. Wang, Image smoothing with generalized random walks: Algorithm and applications, Appl. Soft Comput. J. (2016), http://dx.doi.org/10.1016/j.asoc.2016.01.003

$$\begin{aligned} I_n(k+1) &= I_n(k) + \lambda \sum_{i=1}^{4} \{w_{ni}[I_{ni}(k) - I_n(k)]\} \\ &= I_n(k) - \lambda \sum_{i=1}^{4} [w_{ni} I_n(k)] + \lambda d_n \sum_{i=1}^{4} \left[\frac{w_{ni}}{d_n} I_{ni}(k)\right] \\ &= (1 - \lambda d_n) I_n(k) + \lambda d_n \sum_{i=1}^{4} \left[\frac{w_{ni}}{d_n} I_{ni}(k)\right] \\ &= (1 - \lambda d_n) I_n(k) + \lambda d_n \sum_{i=1}^{4} [p_{ni} I_{ni}(k)] \end{aligned} \tag{18}$$

Note that, both of above methods find their smoothing result in an iterative way, but there are some minor differences. First, our method obtains pixel information from the initial image in each iteration. Second, our method converges to a pre-defined solution, rather than to a constant image.

Algorithm 2. An iterative solver of the proposed algorithm

1: Input I, β, μ.
2: Calculate weights w_{ij} of each edges according to Eq. (1).
3: Calculate degree of each node.
4: **Do**

$$I_n(k+1) = I_n(0) + \frac{1}{(1+\mu)d_n} \sum_{i=1}^{4} w_{ni}(I_{ni}(k) - I_n(0))$$

(similar as anisotropic diffusion)
OR:

$$I(k+1) = \frac{1}{1+\mu} P I(k) + \frac{\mu}{1+\mu} I(0)$$

(similar as RWR)
Until $I(k)$ converges.
5: Output smoothed image $I(k)$.

4.2. Relationship to random walk with restart (RWR)

The solution to our optimization problem can be viewed as a solution to the combinatorial Laplace equation (with Dirichlet boundary conditions), it must satisfy two properties: (1) $0 \le F_n \le 1$, $\forall\, n$ (maximum/minimum principle) where n is pixel index. (2) The probability of each node assumes the weighted average of its neighboring nodes (the mean-value theorem)[3].

$$\begin{aligned} f_n = {} & \frac{1}{(1+\mu)\sum_{i=1}^{4} w_{ni}} \sum_{i=1}^{4} w_{ni} f_{ni} + \frac{\mu I_n \sum_{i=1}^{4} w_{ni}}{(1+\mu)\sum_{i=1}^{4} w_{ni}} \\ & \times 1 + \frac{\mu(1 - I_n)\sum_{i=1}^{4} w_{ni}}{(1+\mu)\sum_{i=1}^{4} w_{ni}} \times 0 \end{aligned} \tag{19}$$

It is written in a matrix form

$$f = \frac{1}{1+\mu} P f + \frac{\mu}{1+\mu} I, \tag{20}$$

where $P = D^{-1}W$, I is the input image, f is the smoothed output image. An iterative solver of the proposed algorithm similar as RWR can be summarized as Algorithm 2.

Note that above equation has a strong connection with random walk with restart. Random walk with restart (RWR) is used to develop a general method that can spot correlations across media [38]. After that, it has also been applied to interactive image segmentation, colorization [39,40]. The difference between RWR and RW is that, the random walker iteratively transmits to its neighborhood with the probability that is proportional to the edge weight between them in RW model, while the random walker has an additional restarting probability in RWR model [39].

Ham et al. [4] proposed generalized random walk with restart (GRWR) for depth up-sampling and interactive segmentation. Their proposed energy functional unifying the RW and RWR models is quite similar to us. Steady state solution via Gaussian-Jacobi iteration of GRWR can be formulated as follows:

$$f(k+1) = (1 - c) P f(k) + c I. \tag{21}$$

where $f(k)$ and $f(k+1)$ are the steady state probability of k and $k+1$ iteration, I is the initial state. Liu et al. [5] proposed a similar equation for image denoising from a perspective of regularization functions. All of our solution is exactly the same, but our starting point and application area is quite different.

4.3. Relationship to WLS

WLS achieves the goal of edge preserving smoothing through a compromise between data term (minimize the distance between input and output images) and regularization term (smoothing output in a spatially varying manner) [7]. Our method can also be seen as working in a similar way from the perspective of energy functional. There is a little difference, WLS regularizes the difference of input and output images as a whole, while our method regularizes the difference between each individual pixels. Next we will talk about details below.

First, WLS can be seen as seeking an f that minimize the following energy functional:

$$\begin{aligned} E_{WLS} &= (f - I)^T (f - I) + \lambda f^T L f \\ &= (f^T f - 2 I^T f) + \lambda f^T L f + const \end{aligned} \tag{22}$$

where f and I denote smoothed image and original image, and λ is a free parameter. From the last equality of Eq. (12), our proposed method can be considered as seeking an f that minimize the following energy functional:

$$\begin{aligned} E_{RW} &= -\mu f^T D I + \frac{1}{2} f^T ((1+\mu) D - W) f + const \\ &= \frac{1}{2} f^T L f + \left(\frac{\mu}{2} f^T D f - \mu f^T D I\right) + const \end{aligned} \tag{23}$$

Our method is closely connected with WLS, WLS was considered as a tradeoff between data term ($f^T f - 2 I^T f$) and regularization term ($f^T L f$). Our method has the same regularization term ($f^T L f$) with WLS, while our method has a regularization term ($f^T D f - 2 I^T D f$) taking the degree of each pixel node into consideration.

Furthermore, we explore the following energy functional to make our idea more clear.

$$\begin{aligned} E_{RW} &= (f - I)^T D (f - I) + \lambda f^T L f \\ &= (f^T D f - 2 I^T D f) + \lambda f^T L f + const \end{aligned} \tag{24}$$

As it turned out, it has the same data term and regularization term with our method, minimizing Eq. (24) will have the same effect as our algorithm. The only difference between Eq. (24) and Eq. (22) is that Eq. (24) takes each pixel into consideration.

Second, from the random walk view of our proposed random walk framework, WLS can be seen as calculating the probability of random walker starting from each pixel on a structure shown in Fig. 8(b). Where $w_{nW} = \mu I_n$, $w_{nB} = \mu(1 - I_n)$, which means the weight of edge connecting pixel node n with terminal node is changed.

We can validate our thoughts by calculating the probability that random walker starting from each pixel node position will first reach terminal node **W** on graph model Fig. 8(b). Similar to Eq. (12), in this situation, the probability can be calculated by minimizing following energy functional.

Please cite this article in press as: Z. Wang, H. Wang, Image smoothing with generalized random walks: Algorithm and applications, Appl. Soft Comput. J. (2016), http://dx.doi.org/10.1016/j.asoc.2016.01.003

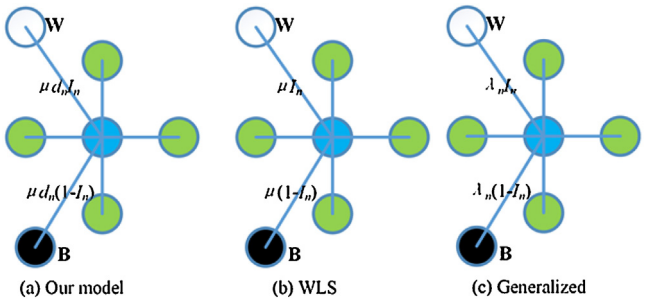

Fig. 8. Difference among our model, WLS [7] and a more generalized model. Take central blue pixel as an example to show graph weight assignment. WLS and some more generalized equation can be formulated as random walker walks on graph model (b) and (c), respectively. The difference between all these models is the weights assigned to edges connecting pixel nodes and terminal nodes. It gives a random walks view of these algorithms and applications.

$$\begin{aligned} E_{WLS} &= f^T \begin{bmatrix} -w_{1W} & -w_{1B} \\ \vdots & \vdots \\ -w_{nW} & -w_{nB} \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} + \frac{1}{2} f^T (L + \mu E) f + const \\ &= f^T \begin{bmatrix} -\mu I_1 \\ \vdots \\ -\mu I_n \end{bmatrix} + \frac{1}{2} f^T (L + \mu E) f + const \\ &= \left(-\mu f^T I + \frac{\mu}{2} f^T f \right) + \frac{1}{2} f^T L f + const \end{aligned} \tag{25}$$

Above equation has the same data term and regularization term as original energy functional equation (22), which means that the solution to the WLS optimization problem shares the same result as the proposed random walk problem on the graph model Fig. 8(b).

4.4. Relationship to a more generalized model

If we replace D of Eq. (24) with a more general diagonal matrix Λ which is encoded with the weights of other constraints. Let us assume Λ is a diagonal matrix, with element $\{\lambda_1, \lambda_2, \ldots, \lambda_n\}$ along the diagonal.

$$\begin{aligned} E_{generalized} &= (f - I)^T \Lambda (f - I) + f^T L f \\ &= (f^T \Lambda f - 2 I^T \Lambda f) + f^T L f + const \end{aligned} \tag{26}$$

The energy functional can be found in guided image smoothing, super resolution, image matting, image dehazing and many other applications [17,42–44]. Next we will calculate the probability that random walker first reaches terminal node **W** on a graph model as shown in Fig. 8(c).

$$\begin{aligned} E_{generalized} &= f^T \begin{bmatrix} -w_{1W} & -w_{1B} \\ \vdots & \vdots \\ -w_{nW} & -w_{nB} \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} + \frac{1}{2} f^T (\Lambda + L) f + const \\ &= f^T \begin{bmatrix} -w_{1W} \\ \vdots \\ -w_{nW} \end{bmatrix} + \frac{1}{2} f^T (\Lambda + L) f + const \\ &= -f^T \Lambda I + \frac{1}{2} f^T (\Lambda + L) f + const \\ &= \left(\frac{1}{2} f^T \Lambda f - I^T \Lambda f \right) + \frac{1}{2} f^T L f + const \end{aligned} \tag{27}$$

The result shows that when we calculate the probability of random walk problem on the graph model Fig. 8(c), we are actually minimizing Eq. (26). These applications are highly related to our proposed framework. It opens the possibility for a hardware (e.g. VLSI) implementation of all these applications.

5. Applications and results

To validate our algorithm, we have tested our method on hundreds of related images, from most-used images in the area to all kinds of image libraries. Most of our experiments are conducted on Berkeley Segmentation Data Set and Benchmarks 500 (BSDS500). The dataset consists of 500 different natural images. However, we would like to stress that next experiments do not aim to give state-of-the-art results, and instead concentrate on demonstrating how the proposed algorithm can be harnessed directly to a variety of applications.

As edge preserving image smoothing is one of the most fundamental work in image processing area, it finds lots of interesting applications in the field of computer vision. For some problems, such as image enhancement, edge detection, image denoising and HDR tone mapping, our method helps to make a satisfactory result. We show in this section how flattening, edge detection and other applications can be effectively addressed using the proposed method. Different tasks are characterized by small variations in term of task complexity. We briefly describe these tools and show some results in this section.

5.1. Flattening

5.1.1. Skin smoothing

For skin smoothing, it is desired to smooth out freckles while important contours are preserved. Skin smoothing is an application which requires the algorithm should be able to generate a high precise result. We applied our smoothing method to smooth skin texture and compared with four other algorithms.

Fig. 9 shows our result. Some image smoothing works in some particular situation, for example, local extreme due to its number Q5 of smoothing scale is limited, so it can hardly have a good result in such situation. Over-sharpening often happens in L0 algorithm when remove details. Our smoothing method together with WLS and guided filter are suitable for this application, above algorithm analysis has shown that our method has a similar principle foundation. Experimental results show that our method gives the lady a beautiful face. Notice how fine details in the lady's face are preserved (e.g., the pupil, the hair), while the skin texture is effectively smoothed.

5.1.2. Compression artifacts removal

Image compressed at low bit rates using current popular image standard will bring the annoying visual artifact. As the compression artifacts are always near the edges, they are strongly correlated with edges, general denoising approaches are not suitable in this application. Our method is compared with L0 regularization, which defeats many state-of-art algorithms in PSNR and SSIM comparison [15].

Fig. 10 shows our result. As we have seen, L0 regularization can remove most artifacts of (a) as shown in (b), it achieves quite good result. However, when take a careful look at the close-up images, there still exist some annoying artifacts. (c) Gives our result, our method creates a cleaner contour line and gives a cleaner reconstruction.

Please cite this article in press as: Z. Wang, H. Wang, Image smoothing with generalized random walks: Algorithm and applications, Appl. Soft Comput. J. (2016), http://dx.doi.org/10.1016/j.asoc.2016.01.003

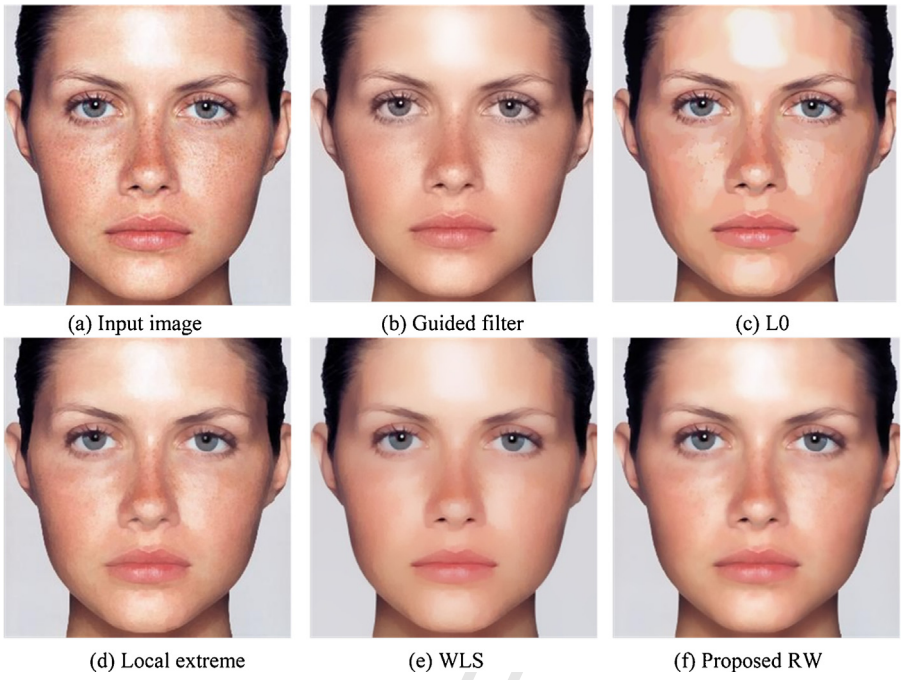

Fig. 9. Skin smoothing. (a) Input image. (b) Guided filter [17]. (c) The L0 regularized method [15]. (d) Local extreme [16]. (e) WLS [7]. (f) The proposed RW method. We adjust parameters of each algorithm to present the best result of each algorithm. (f) The result of our proposed RW method, the skin texture is effectively smoothed, while pupil and hair are preserved.

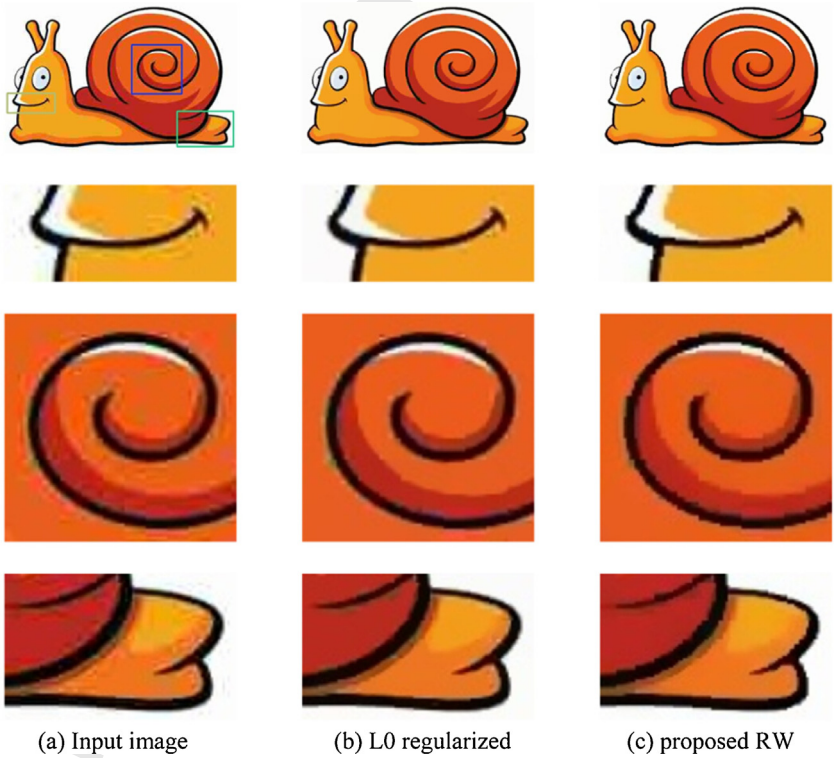

Fig. 10. Clip-art JPEG compression artifact remove result. (a) Input image. (b) The L0 regularized method [15]. (c) Proposed RW. The last three rows are the close-ups of images in 1st row. The artifact remove effect of (b) is quite similar to the one in (c) produced by our proposed algorithm on parts of the image but much less noticeable on other parts, such as the artifacts near important contours.

5.1.3. Edge detection

Our method is able to suppress low-amplitude details, while preserving salient edges. In the example shown in Fig. 11(a), the top image has many waves, and the bottom one has many hairline cracks, directly applying the Canny edge detector to the original image produces a problematic result as shown in Fig. 11(b). Many unwanted detail edges appear in the final result which greatly affects visual pleasure. Our smoothing method can remove tiny

Please cite this article in press as: Z. Wang, H. Wang, Image smoothing with generalized random walks: Algorithm and applications, Appl. Soft Comput. J. (2016), http://dx.doi.org/10.1016/j.asoc.2016.01.003

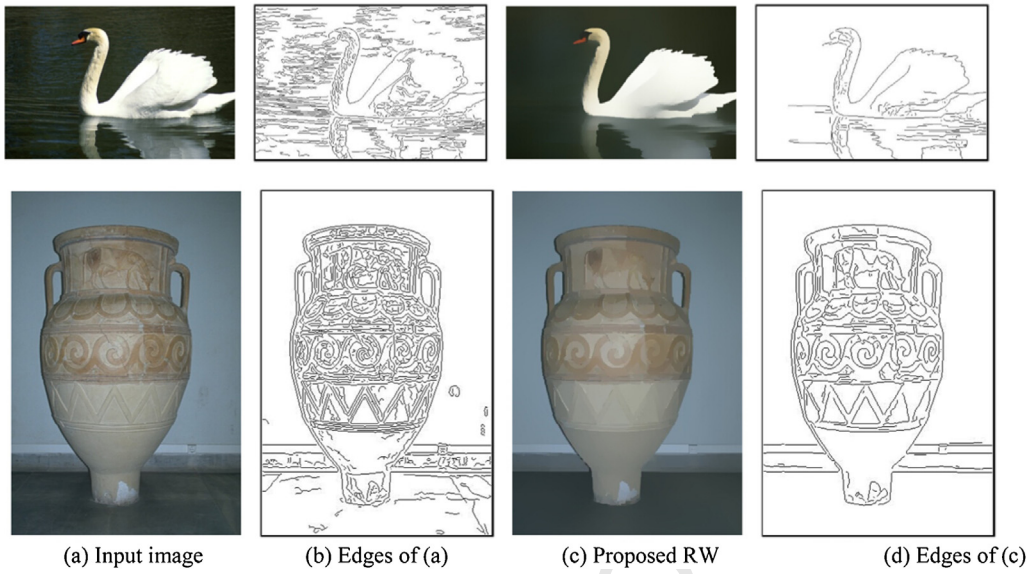

Fig. 11. Smoothing for edge detection. (b) Results by directly applying the Canny edge detector; (c) the smoothed image using the proposed method; (d) the edge computed by the same edge detector.

details of the original image as shown in (c) which gives people a visual esthetic feeling. The edges in (d) computed by the same edge detector are much better.

5.2. Image decomposition

Image is often decomposed into a piecewise smooth base layer and one or more detail layers in computational photography [7]. For example, detail enhancement can be done by combining together boosted base layer and base layer [45]; combining a compressed base layer and the detail layer can be used for HDR tone mapping; image stylization discards details layer while the base layer is further processed to achieve a stylized look [46]. Our method may be also used in place of the BLF, WLS, L0 regularization and other smoothing methods based decomposition applications.

5.2.1. Detail enhancement

Given the input image, using our decomposition method, we compute the detail magnification results by only magnify the detail layer. Fig. 12 shows the magnification results of L0 regularization, local extreme [16] and our method. (a) A widely used flower image for image enhancement, (b)–(d) show similar results of the three methods. Fig. 13 gives a comparison with a close examination. As shown in (b)–(c), results obtained from L0 regularization and local extreme may suffer from artifacts along some of the edges. In contrast, edges in our result appear much better.

As our edge preserving smoothing method is able to smooth the image at arbitrary scales, it can create multi-scale smoothing results, so an image can also be decomposed into several layers and then manipulate image information on multiple layer similar as [7].

5.2.2. Stylization

Stylization aims to produce digital imagery with a wide variety of effects not focused on photorealism [47]. In this application, the base layer is further processed to achieve a stylized abstract look. We adopt the framework of Ref. [46]. The framework is proposed for stylization for efficient visual communication. The basic workflow of the framework, readers can refer to their paper. Our smoothing method is perfect for this application. Fig. 14 illustrates an example of our smoothing result and bilateral filtering [48] to produce abstracted results.

5.2.3. HDR tone mapping

HDR tone mapping is another popular application. There exist many different tone mapping methods [49–52]. Most of the recent operators are capable of effectively mapping HDR radiance maps into a displayable low dynamic range image. Our method avoids haloing and other artifacts that may happen in other methods. According to the standard tone mapping strategy of compressing base layer while preserving the detail layer proposed by Durand and Dorsey [53], edge preserving image smoothing method is used to decompose an image into base layer and detail layer. We use our RW based smoothing framework to replace the bilateral filter in their original work. Farbman et al.'s implementation is used in our experiments [41].

Fig. 15 compares the result of a tone mapping operator implemented using our RW based method and other popular methods.

Q10 **Fig. 12.** Detail exaggeration using our tool. The detail enhanced results (3×). Parameters: L0 ($\lambda = 3E{-}2$), local extreme (17), and proposed RW ($\mu = 0.005$, $\beta = 9$).

Please cite this article in press as: Z. Wang, H. Wang, Image smoothing with generalized random walks: Algorithm and applications, Appl. Soft Comput. J. (2016), http://dx.doi.org/10.1016/j.asoc.2016.01.003

Fig. 13. Detail enhancement. L0 and local extreme based detail layer results in halos (b) and artifacts (c) along the high-contrast edges, while our proposed RW (d) suffers from no this problem.

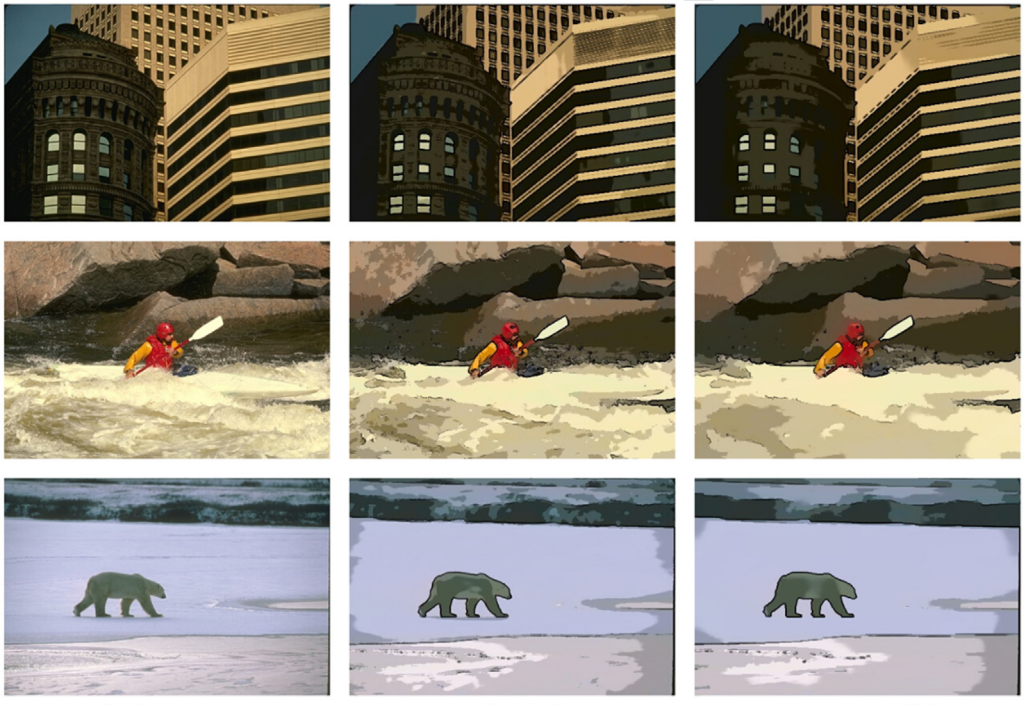

Fig. 14. Abstraction examples. (a) Input image. (b) Results of BLF [48]. (c) Proposed RW.

Our results show that the proposed method is able to generate a competitive tone mapping result.

5.3. Test for computation complexity

In Section 3, we analyze the time complexity of the algorithm theoretically. The runtime of different methods is tested in this section. The methods for comparison consist of L0 method, WLS method, the proposed RW method, and local extrema method. All the algorithms run on the same platform (CPU: Intel i5-3470, Memory: 10 GB, and Software: Windows 10 + Matlab2015a). Images in Fig. 16 are employed to test the runtime of algorithm.

The experimental results of image smoothing are listed in Tables 1 and 2. From the Table 1, we can see that although test image with the same size (200×300) have different textures, the runtime of each method keeps almost unchanged. It is thus clear that image content is not the major factor that affects the running time of algorithms.

If its size (image resolution) is changed for the same image, the runtime of each method will change accordingly. For example, Table 2 shows the runtime of Tulips image under different image

Table 1
Comparison with different algorithms (image size: 200×300, time unit: second).

Test images	L0	WLS	Proposed RW	Local extrema
Tulips	0.29	0.43	0.44	29.63
Rock	0.29	0.43	0.44	29.67
Flower	0.29	0.43	0.44	29.14
Statue	0.29	0.43	0.44	29.59

Table 2
Runtime of Tulips under different image resolution (time unit: second).

Image size	L0	WLS	Proposed RW	Local extrema
100×150	0.13	0.10	0.10	7.37
200×300	0.29	0.43	0.44	29.63
400×600	1.49	2.05	2.21	120.89
800×1200	6.53	10.26	10.50	508.47

Please cite this article in press as: Z. Wang, H. Wang, Image smoothing with generalized random walks: Algorithm and applications, Appl. Soft Comput. J. (2016), http://dx.doi.org/10.1016/j.asoc.2016.01.003

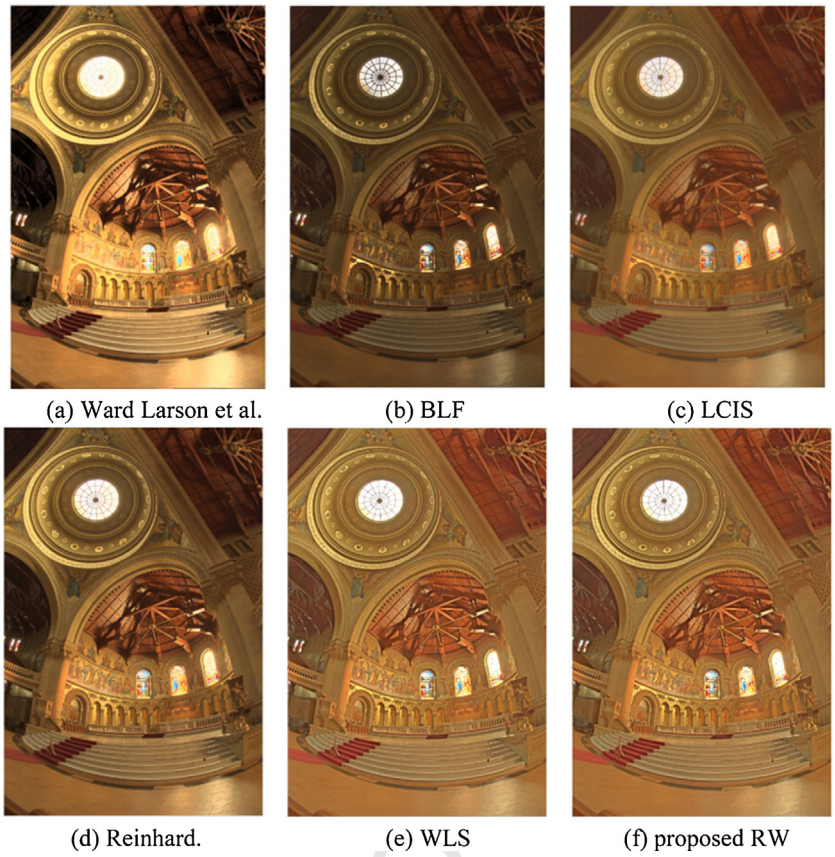

Fig. 15. Comparison with other methods. (a) Ward Larson et al. [54]. (b) BLF [53]. (c) LCIS [49]. (d) Reinhard [51]. (e) WLS [7].

Fig. 16. Images for run-time test.

resolution. Obviously, this is the main factor slowing down the algorithm running because large image (high resolution) usually has more data, which will cost more time.

For the speed of algorithms in Table 2, L0 method is the fastest, and local extrema method is the slowest. WLS and the proposed RW have almost equivalent speed, accurately speaking, WLS is slightly faster than the proposed RW. Most of the time (about 70%) of the proposed algorithm are spent in the process of solving the equation system. In fact, image smoothing does not require a precision result of the equation system. A faster speed can be obtained using iterative solver. This is our future work.

6. Conclusion

In this work, we proposed a novel algorithm for edge preserving image smoothing. Unlike previous methods, our method has a clear physical meaning, which turned image smoothing to a random walk problem. A generalized random walks framework was proposed to solve the image smoothing problem in this paper, which provides a new perspective for the problem. Furthermore, the proposed framework has many equivalences with electric circuits, and some image processing techniques are showed to be highly related to our method, which opens the possibility for a hardware (e.g., VLSI) implementation of these algorithms. Finally, comparison with some other techniques, experiments demonstrate that our algorithm generates high quality results at low computational cost and does not suffer from some drawbacks of other previous approaches. Our future work will improve the speed of our method further and apply it into more applications.

Uncited references

Q7

[18,21,28].

Acknowledgments

Authors would like to thank L. Grady and Z. Farbman et al. for providing the source code of their algorithms. The authors would also like to thank the reviewers for their valuable comments. This work was jointly supported by National Natural Science Foundation Q8 of China (Grant No. 61201421), China Postdoctoral Science Foundation (Grant No. 2013M532097), and Fundamental Research Funds for the Central Universities (lzujbky-2014-52).

References

Q9

[1] Y. Boykov, O. Veksler, R. Zabih, Fast approximate energy minimization via graph cuts, IEEE Trans. Pattern Anal. Mach. Intell. 23 (2001) 1222–1239, http://dx.doi.org/10.1109/34.969114.
[2] L. Grady, G. Funka-Lea, Multi-label image segmentation for medical applications based on graph-theoretic electrical potentials, in: Computer Vision and Mathematical Methods in Medical and Biomedical Image Analysis, 2004, pp. 230–245, http://dx.doi.org/10.1007/978-3-540-27816-0_20.

[3] L. Grady, Random walks for image segmentation, IEEE Trans. Pattern Anal. Mach. Intell. 28 (2006) 1768–1783, http://dx.doi.org/10.1109/TPAMI.2006.233.
[4] B. Ham, D. Min, K. Sohn, A generalized random walk with restart and its application in depth up-sampling and interactive segmentation, IEEE Trans. Image Process. 22 (2013) 2574–2588, http://dx.doi.org/10.1109/TIP.2013.2253479.
[5] G. Liu, X. Zeng, Y. Liu, Image denoising by random walk with restart Kernel and non-subsampled contourlet transform, IET Signal Process. 6 (2012) 148, http://dx.doi.org/10.1049/iet-spr.2010.0265.
[6] P. Perona, J. Malik, Scale-space and edge detection using anisotropic diffusion, IEEE Trans. Pattern Anal. Mach. Intell. 12 (1990) 629–639, http://dx.doi.org/10.1109/34.56205.
[7] Z. Farbman, R. Fattal, D. Lischinski, R. Szeliski, Edge-preserving decompositions for multi-scale tone and detail manipulation, ACM Trans. Graph. 27 (2008) 1, http://dx.doi.org/10.1145/1399504.1360666.
[8] M.J. Black, G. Sapiro, D.H. Marimont, D. Heeger, Robust anisotropic diffusion, IEEE Trans. Image Process. 7 (1998) 421–432, http://dx.doi.org/10.1109/83.661192.
[9] C. Lopez-Molina, M. Galar, H. Bustince, B. De Baets, On the impact of anisotropic diffusion on edge detection, Pattern Recognit. 47 (2014) 270–281, http://dx.doi.org/10.1016/j.patcog.2013.07.009.
[10] L.I. Rudin, S. Osher, E. Fatemi, Nonlinear total variation based noise removal algorithms, Physica D: Nonlinear Phenom. 60 (1992) 259–268, http://dx.doi.org/10.1016/0167-2789(92)90242-F.
[11] T. Chan, S. Esedoglu, F. Park, et al., Recent developments in total variation image restoration, in: Mathematical Models of Computer Vision, 2005, p. 17, doi:10.1.1.101.2342.
[12] M. Elad, On the origin of the bilateral filter and ways to improve it, IEEE Trans. Image Process. 11 (2002) 1141–1151, http://dx.doi.org/10.1109/TIP.2002.801126.
[13] S. Srisuk, Bilateral filtering as a tool for image smoothing with edge preserving properties, in: Electrical Engineering Congress (iEECON), IEEE, 2014, pp. 1–4, http://dx.doi.org/10.1109/iEECON.2014.6925940.
[14] D. Min, S. Choi, J. Lu, B. Ham, K. Sohn, M. Do, Fast global image smoothing based on weighted least squares, IEEE Trans. Image Process. 7149 (2014) 1–15, http://dx.doi.org/10.1109/TIP.2014.2366600.
[15] L. Xu, C. Lu, Y. Xu, J. Jia, Image smoothing via L 0 gradient minimization, Proc. 2011 SIGGRAPH Asia Conf. – SA '11 30 (2011) 1, http://dx.doi.org/10.1145/2024156.2024208.
[16] M.I.T. Csail, Edge-preserving multiscale image decomposition based on local extrema, ACM SIGGRAPH Asia (2009) 1–9, http://dx.doi.org/10.1145/1618452.1618493.
[17] K. He, J. Sun, X. Tang, Guided image filtering, IEEE Trans. Pattern Anal. Mach. Intell. 35 (2013) 1397–1409, http://dx.doi.org/10.1109/TPAMI.2012.213.
[18] H. Tabout, Y. Chahir, A. Souissi, A. Sbihi, Random walks based segmentation approach for image retrieval, in: 2008 Third Int. Conf. Pervasive Comput. Appl., 2008, pp. 593–597, http://dx.doi.org/10.1109/ICPCA.2008.4783681.
[19] G.F. Lawler, V. Limic, Random Walk: A Modern Introduction, Cambridge University Press, 2010, http://dx.doi.org/10.1017/CBO9780511750854.
[20] S.T. Gries, Particle movement: a cognitive and functional approach, Cogn. Linguist. 10 (1999), http://dx.doi.org/10.1515/cogl.1999.005.
[21] E.A. Codling, M.J. Plank, S. Benhamou, Random walk models in biology, J. R. Soc. Interface 5 (2008) 813–834, http://dx.doi.org/10.1098/rsif.2008.0014.
[22] R. Ratcliff, Diffusion and random walk processes, Int. Encycl. Soc. Behav. Sci. 6 (2001) 3668–3673.
[23] L. Yen, D. Vanvyve, F. Wouters, F. Fouss, M. Verleysen, M. Saerens, Clustering using a random walk based distance measure, in: Proc. 13th Symp. Artif. Neural Networks ESANN 2005, 2005, pp. 27–29, doi:10.1.1.59.9581.
[24] J. Cui, H. Liu, J. He, P. Li, X. Du, P. Wang, TagClus: a random walk-based method for tag clustering, Knowl. Inf. Syst. 27 (2011) 193–225, http://dx.doi.org/10.1007/s10115-010-0307-y.
[25] M. Chen, J. Liu, X. Tang, Clustering via random walk hitting time on directed graphs, in: AAAI '08 Proceedings of the 23rd National Conference on Artificial Intelligence, 2008, pp. 616–621.
[26] H. Qui, E.R. Hancock, Clustering and embedding using commute times, IEEE Trans. Pattern Anal. Mach. Intell. 29 (2007) 1873–1890, http://dx.doi.org/10.1109/TPAMI.2007.1103.
[27] D. Dolgopyat, G. Keller, C. Liverani, Random walk in Markovian environment, Ann. Probab. 36 (2008) 1676–1710, http://dx.doi.org/10.1214/07-AOP369.
[28] D. Jin, B. Yang, C. Baquero, D. Liu, D. He, J. Liu, A Markov random walk under constraint for discovering overlapping communities in complex networks, J. Stat. Mech. Theory Exp. (2011) P05031, http://dx.doi.org/10.1088/1742-5468/2011/05/P05031.
[29] L. Grady, T. Schiwietz, S. Aharon, Random walks for interactive alpha matting, in: VIIP 2005, 2005, doi:10.1.1.220.8725.
[30] Z. Ji, Y. Su, Y. Pang, X. Qu, Diversifying the image relevance reranking with absorbing random walks, in: 2011 Sixth Int. Conf. Image Graph., 2011, pp. 981–986, http://dx.doi.org/10.1109/ICIG.2011.113.
[31] J.G. Amar, The Monte Carlo method in science and engineering, Comput. Sci. Eng. 8 (2006), http://dx.doi.org/10.1109/MCSE.2006.34.
[32] P.G. Doyle, J.L. Snell, Random Walks and Electric Networks, 2006, doi:10.1.1.169.8044.
[33] F. Harary, Graphs and matrices, SIAM Rev. 9 (1967) 83–90, http://dx.doi.org/10.1137/1009003.
[34] J. Dodziuk, Difference equations, isoperimetric inequality and transience of certain random walks, Trans. Am. Math. Soc. 284 (1984) 787, http://dx.doi.org/10.2307/1999107.
[35] U. Luxburg, A tutorial on spectral clustering, Stat. Comput. 17 (2007) 395–416, http://dx.doi.org/10.1007/s11222-007-9033-z.
[36] N. Biggs, Algebraic potential theory on graphs, Bull. Lond. Math. Soc. 29 (1997) 641–682, http://dx.doi.org/10.1112/S0024609397003305.
[37] P. Tetali, Random walks and the effective resistance of networks, J. Theor. Probab. 4 (1991) 101–109, http://dx.doi.org/10.1007/BF01046996.
[38] J.-Y. Pan, H.-J. Yang, C. Faloutsos, P. Duygulu, Automatic multimedia cross-modal correlation discovery, in: Proc. 2004 ACM SIGKDD Int. Conf. Knowl. Discov. Data Min. – KDD '04, 2004, p. 653, http://dx.doi.org/10.1145/1014052.1014135.
[39] T. Kim, K. Lee, S. Lee, Generative image segmentation using random walks with restart, in: Comput. Vision – ECCV 2008, 2008, pp. 264–275, http://dx.doi.org/10.1007/978-3-540-88690-7_20.
[40] T.H. Kim, K.M. Lee, S.U. Lee, Edge-preserving colorization using data-driven random walks with restart, in: Proc. – Int. Conf. Image Process (ICIP), 2009, pp. 1661–1664, http://dx.doi.org/10.1109/ICIP. 2009.5413394.
[41] http://www.cs.huji.ac.il/~danix/epd/.
[42] A. Zomet, S. Peleg, Multi-sensor super-resolution, in: Sixth IEEE Work. Appl. Comput. Vision 2002. (WACV 2002). Proceedings, 2002, http://dx.doi.org/10.1109/ACV.2002.1182150.
[43] K. He, J. Sun, X. Tang, Single image haze removal using dark channel prior, IEEE Trans. Pattern Anal. Mach. Intell. 33 (2011) 2341–2353, http://dx.doi.org/10.1109/TPAMI.2010.168.
[44] A. Levin, D. Lischinski, Y. Weiss, A closed-form solution to natural image matting, IEEE Trans. Pattern Anal. Mach. Intell. 30 (2008) 228–242, http://dx.doi.org/10.1109/TPAMI.2007.1177.
[45] R. Fattal, M. Agrawala, S. Rusinkiewicz, Multiscale shape and detail enhancement from multi-light image collections, ACM Trans. Graph. 26 (2007) 51, http://dx.doi.org/10.1145/1276377.1276441.
[46] H. Winnemöller, S.C. Olsen, B. Gooch, Real-time video abstraction, ACM Trans. Graph. 25 (2006) 1221, http://dx.doi.org/10.1145/1141911.1142018.
[47] E.S.L. Gastal, M.M. Oliveira, Domain transform for edge-aware image and video processing, ACM Trans. Graph. 30 (2011) 1, http://dx.doi.org/10.1145/2010324.1964964.
[48] C. Tomasi, R. Manduchi, Bilateral filtering for gray and color images, in: Sixth Int. Conf. Comput. Vis. (IEEE Cat. No. 98CH36271), 1998.
[49] J. Tumblin, G. Turk, LCIS: a boundary hierarchy for detail-preserving contrast reduction, in: Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques, 1999, pp. 83–90, http://dx.doi.org/10.1145/311535.311544.
[50] M. Seeman, P. Zemčík, R. Juránek, A. Herout, Fast bilateral filter for HDR imaging, J. Vis. Commun. Image Represent. 23 (2012) 12–17, http://dx.doi.org/10.1016/j.jvcir.2011.07.012.
[51] E. Reinhard, M. Stark, P. Shirley, J. Ferwerda, Photographic tone reproduction for digital images, ACM Trans. Graph. 21 (2002), http://dx.doi.org/10.1145/566654.566575.
[52] R. Fattal, D. Lischinski, M. Werman, Gradient domain high dynamic range compression, ACM Trans. Graph. 21 (2002), http://dx.doi.org/10.1145/566654.566573.
[53] F. Durand, J. Dorsey, Fast bilateral filtering for the display of high-dynamic-range images, ACM Trans. Graph. 21 (2002), http://dx.doi.org/10.1145/566654.566574.
[54] G.W. Larson, LogLuv encoding for full-gamut, high-dynamic range images, J. Graph. Tools 3 (1998) 15–31, http://dx.doi.org/10.1080/10867651.1998.10487485.

Please cite this article in press as: Z. Wang, H. Wang, Image smoothing with generalized random walks: Algorithm and applications, Appl. Soft Comput. J. (2016), http://dx.doi.org/10.1016/j.asoc.2016.01.003