RANDOM WALKER WATERSHEDS: A NEW IMAGE SEGMENTATION APPROACH

Sundaresh Ram and Jeffrey J. Rodríguez

Department of Electrical and Computer Engineering
University of Arizona, Tucson, AZ 85721, USA
{ram1,jrod}@ece.arizona.edu

ABSTRACT

We propose a new graph-based approach for performing a multilabel, interactive image segmentation using the principle of random walks. Using the random walk principle, given a set of user-defined (or prelabeled) pixels as labels, one can analytically calculate the probability of walking from each unlabeled pixel to each labeled pixel, thereby defining a vector of probabilities for each unlabeled pixel. By efficiently combining this vector of probabilities obtained for each unlabeled pixel, they can be assigned to one of the labels using the watershed algorithm to obtain an image segmentation. We present quantitative and qualitative results, comparing our new algorithm with the original random walker image segmentation algorithm.

Index Terms— Image segmentation, graph theory, random walks, combinatorial Dirichlet problem, watersheds.

1. INTRODUCTION

Image segmentation is a very important task in many computer vision and image processing applications. The goal of segmentation is to simplify and/or group the image into various homogeneous regions that are more meaningful and easier to analyze. Segmentation is a widely researched topic and there have been numerous segmentation algorithms that have been proposed in the literature– e.g., thresholding methods [1], clustering methods [2], edge-based methods [3, 4], region growing methods [5], region splitting and merging methods [6], partial differential equation (PDE) based methods [7] and graph based methods [8–14]. Image segmentation algorithms can be broadly classified into two main categories: automated segmentation and interactive segmentation. While automated segmentation approaches have looked at partitioning the entire image into multiple homogeneous segments, interactive methods implicitly define the segmentation problem relative to a particular task of object localization and categorization. This paper addresses a graph-based segmentation approach based on the principle of random walks and thus we limit our review to only graph-based segmentation algorithms.

An interactive approach to image segmentation requires user guidance of the segmentation algorithm to define desired content to be extracted. They typically operate under one of two paradigms for guidance: 1) specification of pieces of boundary of the desired object to be segmented or a nearby complete boundary either inside or outside the objects that evolves to the desired object boundary [7], [8], and [11]; 2) Specification of a small set of pixels within the object belonging to it and a set of pixels for the background [13], [14]. Mortensen *et al.* [8] proposed the intelligent scissors algorithm, which follows the first paradigm for guidance. It uses Dijkstra's algorithm to compute the shortest path between the user guided points, and this path is treated as the object boundary. This algorithm is fast and works well when the object to be segmented has a high contrast compared to its surroundings, but performs poorly when the contrast is low and in the presence of noise. There exists a large family of active contours and level-set methods that have been developed for image segmentation [7]. These methods follow the first paradigm for guidance, i.e., a nearby complete boundary is initially specified either inside or outside the object, which then evolves the boundary to a local energy minimum. This usually requires specifying many different terms in the energy functional. These methods suffer from weighting the various terms in the energy functional and also, the initialization that the user provides needs to be close enough to the true boundary of the object to be segmented or else it will lead to errors in segmentation.

The well known graph-cuts image segmentation technique was originally devised by Boykov *et al.* [9], [10], which follows the second paradigm mentioned above for user guidance, by labeling a few nodes with distinct labels. The algorithm represents the image as a graph and performs a max-flow/min-cut analysis to find the minimum-weight cut between the various labels. Since the algorithm returns the smallest cut separating the labels, it will often return the cut that minimally separates the labels from the rest of the graph, if a small number of labels are used. Additionally, the multi-way graph-cuts problem is NP-hard, requiring the use of a good heuristic to obtain a decent solution. The graph-cuts algorithm has been extended in many different directions, including addressing the issues of speed and processing of color images. One popular extension is the lazy snapping technique proposed by Li *et al.* [12], which uses an initial watershed approach to reduce the number of nodes to be classified, thereby increasing the speed. Another popular extension is the grabcut technique developed by Rother *et al.* [11], which uses a color model to prevent the need for a large number of labels as a user guidance to attain an optimal multi-way cut. Since these improvements make use of the attributes of the original graph-cuts algorithm, we can expect to associate them with the same difficulties as that of the graph-cuts algorithm. Grady [13] proposed a random walker approach to interactive image segmentation formulated on a weighted graph, where the unlabeled pixels are assigned the label of the node to which it is most likely to send a random walker. This algorithm has shown to perform well on different types of images, but is strongly influenced by the placement of the labels within the image.

We propose a new approach for interactive image segmentation along the lines of the random walker algorithm. We notice from the random walker algorithm presented by Grady [13] that, given a number K of user defined (or prelabeled) pixels as labels, we can find the probability that a random walker starting at each unlabeled pixel reaches each of these K labels. Unlike the original random walker algorithm, we combine the K-tuple vector of probabilities obtained for each unlabeled pixel and segment the resulting image produced using the watershed algorithm. We describe our algorithm is detail and present both quantitative and qualitative results comparing it to the random walker algorithm in the following sections of this paper.

2. METHODS

The random walker algorithm, which involves placing random walkers at each unlabeled pixel and noting which labeled pixels they first arrive at is not practical due to the computation involved. It has been previously established [15], [13] that the probability a random walker first reaches a labeled pixel exactly equals the solution to the combinatorial Dirichlet problem with boundary conditions at the locations of the labeled pixels and labeled pixel in question fixed to unity while others are set to zero. Since solving the combinatorial Dirichlet problem is computationally reasonable, we use this combinatorial method to analytically compute the desired random walker probabilities.

A graph consists of a pair $G = (V, E)$ with vertices (nodes) $v \in V$ and edges $e \in E \subseteq V \times V$. An edge e, spanning two vertices, v_i and v_j, is denoted by e_{ij}. A weighted graph assigns a value to each edge called a weight. The weight of an edge e_{ij} is denoted by $w(e_{ij})$ or simply w_{ij}. The degree of a vertex $d_i = \sum w(e_{ij})$ for all edges e_{ij} incident on v_i. In order to interpret w_{ij} as the bias affecting a random walker's choice, we require that $w_{ij} > 0$. We also assume that our graph is undirected and connected. To represent the image structure in terms of random walker biases, we define a Gaussian weighting function given by

$$w_{ij} = \exp\left\{-\beta(g_i - g_j)^2\right\} \tag{1}$$

where g_i is the image intensity at pixel i. The value of β is the only free parameter in this algorithm.

2.1. Combinatorial Dirichlet problem

As mentioned earlier, placing a random walker at every unlabeled pixel to calculate the probability by noting when they first arrive at a particular label is computationally impractical. Established connections between random walks and potential theory provide us with a simple, convinient method for analytically computing the desired probabilities. The Dirichlet problem has the same solution as the desired random walker probabilities [15], [13] and so we solve this problem to find the random walker probabilites.The Dirichlet integral may be defined as

$$D\left[q\right] = \frac{1}{2}\int_{\Omega} \mid \nabla q \mid^2 d\Omega, \tag{2}$$

for a field q and region Ω. A harmonic function is a function that satisfies the Laplace equation $\nabla^2 q = 0$. The problem of finding a harmonic function subject to its boundary values is called the Dirichlet problem. We define a combinatorial Laplacian matrix L whose elements L_{ij} for vertices v_i and v_j are defined as

$$L_{ij} = \begin{cases} d_i & \text{if } i = j, \\ -w_{ij} & \text{if } v_i \text{ and } v_j \text{ are incident nodes,} \\ 0 & \text{otherwise,} \end{cases}$$

We also define an $m \times n$ edge-node incidence matrix A whose elements $A_{e_{ij}v_k}$ for each vertex v_k and edge e_{ij} are defined as

$$A_{e_{ij}v_k} = \begin{cases} +1 & \text{if } i = k, \\ -1 & \text{if } j = k, \\ 0 & \text{otherwise,} \end{cases}$$

The matrix A acts as a combinatorial gradient operator and the matrix $A^\top$ as a combinatorial divergence. The isotropic combinatorial Laplacian is given by $L = A^\top A$. We define the $m \times m$ constitutive matrix C as the diagonal matrix with the weights of each edge along the diagonal. The constitutive matrix may be interpreted as representing a metric, in the sense that it defines a weighted inner product on the vector space of functions defined on the edge set. In this sense, the combinatorial Laplacian generalizes to the combinatorial Laplace-Beltrami operator via $L = A^\top CA$. The combinatorial formulation of the Dirichlet integral is then given by

$$D[x] = \frac{1}{2}(Ax)^\top C(Ax) = \frac{1}{2}x^\top Lx = \frac{1}{2}\sum_{e_{ij}} w_{ij}\,(x_i - x_j)^2 \tag{3}$$

If we partition the vertices into two sets, V_m representing the prelabeled nodes or user defined labels and V_u representing the unlabeled nodes, then $V_m \cup V_u = V$ and $V_m \cap V_u = \emptyset$. We can decompose (3) as

$$\begin{aligned} D[x_U] &= \frac{1}{2}\left[\begin{array}{cc} x_M^T & x_U^T \end{array}\right]\left[\begin{array}{cc} L_M & B \\ B^T & L_U \end{array}\right]\left[\begin{array}{c} x_M \\ x_U \end{array}\right] \\ &= \frac{1}{2}\left(x_M^T L_M x_M + 2x_U^T B^T x_M + x_U^T L_U x_U\right) \end{aligned} \tag{4}$$

where x_M and x_U correspond to the probabilities of the labeled and unlabeled pixels, respectively. For each unlabeled pixel

$$v_{u_i} \in V_u, \quad X = \left[x_u^j, \quad 0 < j \leq K\right]^\top$$

where x_u^j is the probability that random walker, starting at v_{u_i} will first reach the j^{th} labeled pixel, $v_{m_j} \in V_m$, and K is the total number of labels the user specifies. For each labeled pixel

$$v_{m_i} \in V_m, \quad M = \left[\delta_i^j, \quad 0 < j \leq K\right]^\top$$

Differentiating $D[x_U]$ with respect to x_U and finding critical points yields

$$L_U x_U = -B^\top x_M. \tag{5}$$

Equation (5) represents a system of linear equations with $|V_u|$ unknowns. The solution to the combinatorial Dirichlet problem for the label j may then be found by solving

$$L_U x_u^j = -B^\top x_m^j \tag{6}$$

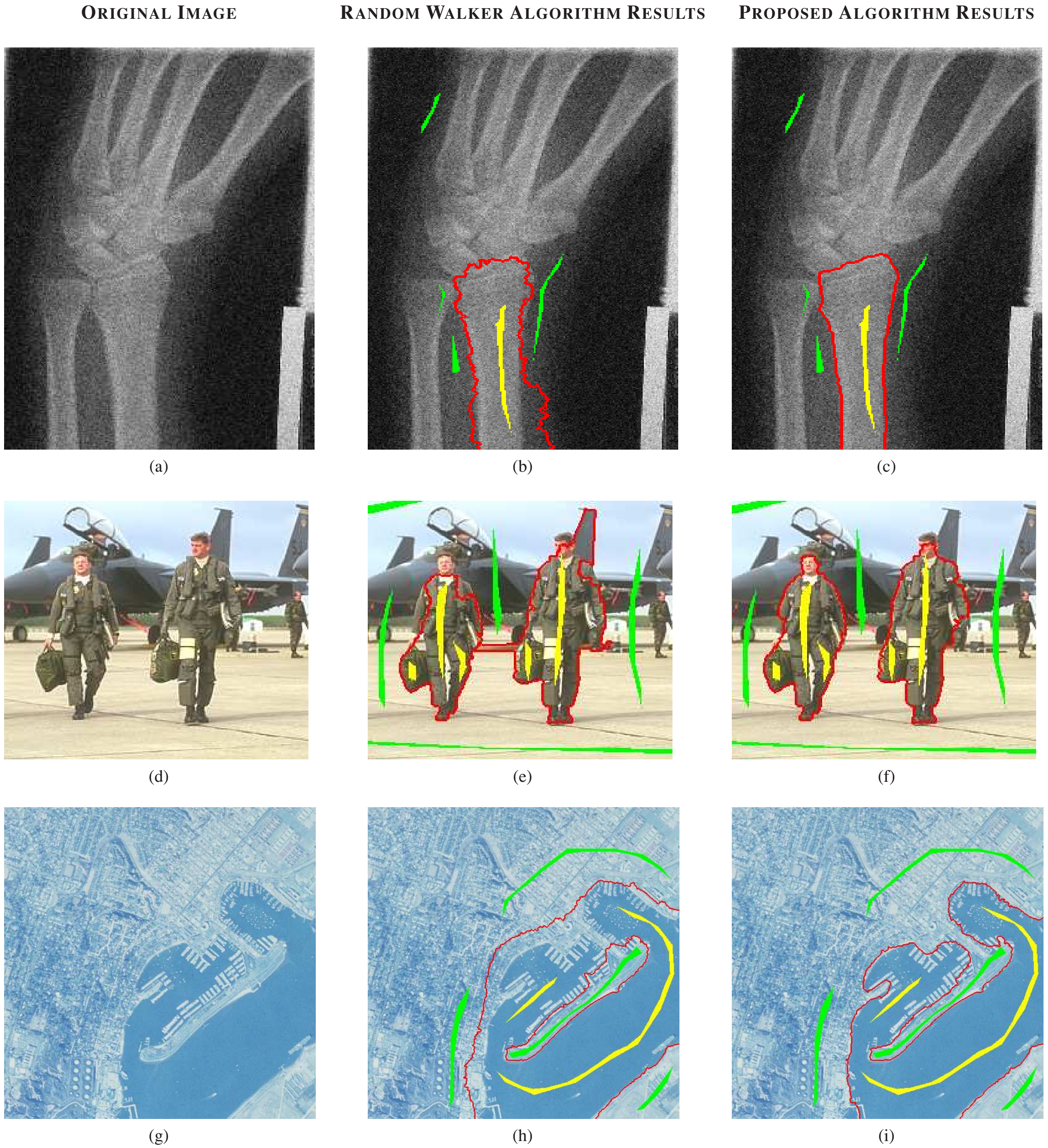
for one label or

$$L_U X = -B^\top M \tag{7}$$

for all labels, where X has K columns taken by each x_u and M has columns taken by each x_m. Since the probabilities at any node will sum to unity, i.e.

$$\sum_j x_u^j = 1 \quad \forall \quad v_{u_i} \in V_u \tag{8}$$

only $K - 1$ sparse linear systems need to be solved, where K is the total number of labels.



ORIGINAL IMAGE

RANDOM WALKER ALGORITHM RESULTS

PROPOSED ALGORITHM RESULTS

(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

Fig. 1: Example segmentations using the foreground (yellow) and background (green) seeds. **Top Row**: random noisy medical image, **Middle Row**: an image from the Berkeley segmentation dataset. **Bottom Row**: an image from the USC SIPI image database. The resultant segmentation boundaries are overlaid onto the original image in red color.

2.2. Combining probabilities and watershed

Once we obtain the K-tuple vector of probabilities for each unlabeled pixel, we combine this vector of probabilities into one value by taking the product of all the probabilities in the vector, in order to obtain a resultant image R:

$$R = \prod_{j} x_u^j \tag{9}$$

The resultant image R will have maximum values in the areas where the probabilities x_u^j are equal for every $0 < j \leq K$, i.e., when an unlabeled pixel has equal probability to reach any of the K labels. Since the probability of an unlabeled pixel to first reach any labeled pixel decreases as we move away from this labeled pixel, we see a local ridge formation around each labeled pixel in the resultant image generated by (9). We grow the labeled pixel regions in all directions until they reach their corresponding ridge locations in all directions in the image R, thereby extending the original labeled pixel region size fed by the user, to a larger size. Finally, we invert this image R and perform a marker-controlled watershed transform on this inverted image, where the labeled pixel regions act as markers. This leads to an improved image segmentation with more accurate delineations between the objects boundaries.

3. EXPERIMENTS AND RESULTS

To test and evaluate the performance of the proposed algorithm in comparison with the original random walker segmentation algorithm, we made use of images from the Berkeley segmentation dataset [16] and the USC SIPI image database.

3.1. Quantitative results

Fig. 1 shows example segmentation results for the proposed method and the random walker segmentation method on the “x-ray image",“ men image", and “ariel image" in the top, middle and bottom rows, respectively. We show the results for both gray-scale and color images. The value of the one free parameter, β in (1), was kept constant for all the images and set to $\beta = 100$ for both the methods, despite the different characteristics of the images. The “x-ray image" shown on the top row in Fig. 1 is a noisy image, and a visual comparison of the segmentation boundaries which are overlaid onto the original image shown in red color in Figs. 1b&c clearly indicates that our method is more robust to noise in comparison to the original random walker algorithm. A multilabel segmentation is performed for the “men image" shown on the middle row in Fig. 1. We observe from Fig. 1e that the random walker segmentation method gets confused and produces an erroneous result due to the color similarities between the object and the background shown on the top right hand side of the image, whereas our proposed technique performs well as shown in Fig. 1f. Finally, from the “ariel image" shown in the bottom row of Fig. 1 we can see how the placement of the label affects the result of the random walker algorithm but not our proposed technique. The isthmus which belongs to the land region has been segmented as a part of the water region by the random walker segmentation method, where as our proposed method is able to identify that it is a land region and has segmented it correctly.

3.2. Qualitative results

In order to obtain an objective measure of performance for the two segmentation methods under comparison here, we validated the results of the three images shown in Fig. 1 using the Dice evaluation [6] based on the mutual overlap between the supervised segmentation methods and manual segmentation. The manual segmentations of the “men image" an “ariel image" are known to us since they belong to the Berkeley segmentation dataset and the USC SIPI image database, respectively. We carefully performed manual segmentation for the “x-ray image" for validation. The Dice performance metric is given by

$$O = \frac{\#\left(R_1 \cap R_2\right)}{\#\left(R_1\right) + \#\left(R_2\right)} \tag{10}$$

Table 1: DICE EVALUATION OF PERFORMANCE OF THE SEGMENTATION ALGORITHMS

	x-ray image	**men image**	**ariel image**
Our Method	0.965	0.903	0.916
RW Algorithm	0.868	0.812	0.804

where R_1 and R_2 are the automatically segmented region and the manually segmented region, respectively. Here $\#(\cdot)$ represents the number of pixels in the region. Table 1 shows the segmentation accuracy as a percentage for the three images. We observe from Table 1 that our proposed algorithm produces higher segmentation accuracy as compared to the random walker segmentation algorithm for all three images.

4. CONCLUSIONS

Due to the varying imaging conditions and complexity of typical images, fully automated segmentation methods are not always very reliable. Thus, there is a need for a user-interactive image segmentation method, especially in the medical field where the pathologist or radiologist may want to participate in the image segmentation. The work presented in this paper has focused on the formulation of a new graph theoretic image segmentation method —“random walker watershed" segmentation—which works on the principle of random walks. The random walker segmentation approach proposed by Grady [13] also focuses on the principle of random walks but differs from our approach in the sense that the final label given to an unlabeled pixel is the value that maximizes the vector of probabilities of the unlabeled pixel, whereas our approach combines these probabilities and generates a resulting image which is then segmented using the watershed algorithm. The advantage of doing so is that the labeled pixel regions given as inputs to the segmentation algorithm could be placed anywhere within the object of interest in order to accurately segment and delineate the object from the rest of the image. The algorithm proposed here has only one parameter β, and all segmentations shown in this paper were performed by fixing this parameter. Finally, the algorithm simply requires solution to a sparse, symmetric, positive-definite system of equations, which is straightforward to implement efficiently. Additionally, interactive editing of the segmentation generally results in even faster computation time since the previous solution may be used as an initial solution for an iterative matrix solver. We show qualitative results in Fig. 1 and quantitative results in Table 1. According to the results, our proposed algorithm performs better in comparison to the random walker segmentation algorithm for the performance metric considered.

5. REFERENCES

[1] M. Sezgin and B. Sankur, “Survey over image thresholding techniques and quantitative performance evaluation,” *J. Electron Imaging*, vol. 13, no. 1, pp. 146–165, Jan. 2004.

[2] D. Comaniciu and P. Meer, “Mean shift: a robust approach towards feature space analysis,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 5, pp. 603–619, May 2002.

[3] J. Canny, “A computational approach to edge detection,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. PAMI-8, no. 6, pp. 679–698, Nov. 1986.

[4] M. Tabb and N. Ahuja, "Multiscale image segmentation by integrated edge and region detection," *IEEE Trans. Image Process.*, vol. 6, no. 5, pp. 642–655, May 1997.

[5] S.A. Hojjatoleslami and J. Kittler, "Region growing: a new approach," *IEEE Trans. Image Process.*, vol. 7, no. 7, pp. 1079–1084, Jul. 1998.

[6] M. Sonka, V. Hlavac, and R. Boyle, *Image Processing, Analysis, and Machine Vision*, chapter 5, pp. 176–211, Brooks/Cole-Thompson Learning, Pacific Groove, CA, 3rd edition, 2007.

[7] A. Mitiche and I. Ben, *Variational and level set methods in image segmentation*, Springer Topics in Signal Processing. Springer-Verlag, Berlin-Heidelberg, Germany, 2011.

[8] E.N. Mortensen and W.A. Barrett, "Interactive segmentation with intelligent scissors," *Graphical Models and Image Processing*, vol. 60, no. 5, pp. 349–384, June 1998.

[9] Y.Y. Boykov and M.-P. Jolly, "Interactive graph cuts for optimal boundary and region segmentation of objects in n-d images," in *Proc. IEEE Int. Conf. Computer Vision*, 2001, pp. 105–112.

[10] Y.Y. Boykov and V. Kolmogorov, "An experimental comparison of min-cut/max- flow algorithms for energy minimization in vision," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 9, pp. 1124–1137, Sept. 2004.

[11] C. Rother, V. Kolmogorov, and A. Blake, "Grabcut – interactive foreground extraction using iterated graph cuts," *ACM Trans. Graphics*, vol. 23, no. 3, pp. 309–314, Aug. 2004.

[12] Y. Li, J. Sun, C.-K. Tang, and H.-Y. Shum, "Lazy snapping," in *Proc. IEEE Int. Conf. Computer Graphics and Interactive Techniques*, 2004, pp. 303–308.

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

[14] B. Peng, D. Zhang, and J. Yang, "Image segmentation by iterated region merging with localized graph cuts," *Pattern Recognition*, vol. 44, no. 11, pp. 2527–2538, Nov. 2011.

[15] P. Doyle and L. Snell, "Random walks and electric networks," *Math. Assoc. of Am.*, , no. 22, 1984.

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