

A Bayesian Network Model for Automatic and Interactive Image Segmentation

Lei Zhang, *Member, IEEE*, and Qiang Ji, *Senior Member, IEEE*

Abstract—We propose a new Bayesian network (BN) model for both automatic and interactive image segmentation. A multilayer BN is constructed from an oversegmentation to model the statistical dependencies among superpixel regions, edge segments, vertices, and their measurements. The BN also incorporates various local constraints to further restrain the relationships among these image entities. Given the BN model and various image measurements, belief propagation is performed to update the probability of each node. Image segmentation is generated by the most probable explanation inference of the true states of both region and edge nodes from the updated BN. Besides the automatic image segmentation, the proposed model can also be used for interactive image segmentation. While existing interactive segmentation (IS) approaches often passively depend on the user to provide exact intervention, we propose a new active input selection approach to provide suggestions for the user's intervention. Such intervention can be conveniently incorporated into the BN model to perform *actively IS*. We evaluate the proposed model on both the Weizmann dataset and VOC2006 cow images. The results demonstrate that the BN model can be used for automatic segmentation, and more importantly, for actively IS. The experiments also show that the IS with active input selection can improve both the overall segmentation accuracy and efficiency over the IS with passive intervention.

Index Terms—Active labeling, Bayesian network (BN), image segmentation, interactive image segmentation.

I. INTRODUCTION

IMAGE segmentation is a difficult problem in computer vision. It aims at partitioning an image into constituent regions of interest. Many approaches have been proposed before, including the clustering method [1], region growing [2], active contours [3], normalized cuts [4], graph-cut-based approaches [5], Markov random fields (MRFs) [6], etc. These methods are basically data-driven approaches. The data-driven method sometimes fails to produce satisfactory segmentation when there are shadow, occlusion, cluttering, low contrast area, or noise in the image. Incorporation of contextual and prior information is very important to improve segmentation under such situations. Researchers have therefore incorporated global shape information, local image constraints such as smoothness, and other domain knowledge for image segmentation.

Manuscript received November 06, 2009; revised May 17, 2010; accepted February 10, 2011. Date of publication February 28, 2011; date of current version August 19, 2011. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Ying Wu.

L. Zhang is with UtopiaCompression Corporation, Los Angeles, CA 90064 USA (e-mail: leizhang2009@gmail.com).

J. Qiang is with the Department of Electrical, Computer, and Systems Engineering, Rensselaer Polytechnic Institute, Troy, NY 12180 USA.

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/TIP.2011.2121080

Some previous works have already tried to incorporate certain prior knowledge into image segmentation. Nguyen and Ji [7] incorporated smoothness and domain knowledge into the growing of region boundaries in the watershed algorithm. Chan and Zhu [8] incorporated a global shape prior into the active contours segmentation. Freedman and Zhang [9] incorporated the global shape prior in a level-set template into the graph-cut algorithm for interactive segmentation (IS). By incorporating the prior knowledge, these approaches successfully improve segmentation in cases, where the purely data-driven methods may fail. In addition, the MRF-based approach is often used to capture certain prior knowledge. It can encode the local spatial relationship among random variables. However, MRF is not effective in capturing the global spatial relationship or long-range interaction. An additional hidden layer is normally added to model such relationships [10], which makes the MRF model more complex and computationally inefficient.

The aforementioned methods usually incorporate the prior knowledge in a deterministic manner, ignoring the various uncertainties associated with image segmentation. Moreover, the incorporation of new information is often done in a batch mode, making it difficult to gradually incorporate new evidence and the user's intervention into the segmentation process. These methods, therefore, cannot effectively reuse the previous results when new information is available.

The fully automatic segmentation methods might also fail, even though they already integrate both data information and prior knowledge. The reason for this may be due to the complexity of image segmentation for real applications. Such difficulties as noise, occlusion, low contrast, and clutter make image segmentation very challenging. We believe that semiautomatic segmentation with the user's intervention (i.e., interactive image segmentation) is a possible solution to complex problems. The users can provide further clues for image segmentation because it is easy for them to know where the segmentation is wrong. If the segmentation algorithms can effectively integrate these valuable feedbacks, improved image segmentation can be achieved.

There are many works on interactive image segmentation [5],[9], [11]–[13]. They demonstrate the usefulness of the user's intervention for improving segmentation. However, these approaches mainly exploit a limited type of user interventions. The user typically gives hard constraints by fixing the labels of certain pixels or group of pixels, without considering the uncertainties in the user's intervention. Other types of users' interventions are rarely used in these works.

More importantly, these existing IS approaches often solely depend on the user to passively select the next intervention.

However, the user may not easily know which intervention is most effective and should be performed first. In interactive labeling domain, researchers have studied the active learning approach for helping the user select the label to be first corrected [14]–[17]. The active learning proceeds sequentially, with the learning algorithm actively asking for the labels of some instances from a user. The objective is to ask the user to label the most informative instance in order to reduce the total labeling cost and to accelerate the learning process. Similar strategies can be incorporated into the interactive labeling domain to form a new active labeling approach. For active labeling, the key is the sample selection strategy, which selects the next sample for the user to label based on a predefined criterion. A better sample selection criterion should simultaneously maximize the performance while minimizing the amount of human intervention.

For classification problems, the most commonly used active learning criterion is the close-to-boundary criterion [18]. Other criteria have also been proposed. For example, the expected information gain is a natural measure to select the next sample to label. Freund *et al.* [14] enhanced the query-by-committee method to obtain high expected information gain. Shen *et al.* [15] incorporated multicriteria (i.e., informativeness, representativeness, and diversity) for active learning. Kapoor *et al.* [16] also proposed the variance and uncertainty of a Gaussian process as the active learning criteria. In addition, Raghavan *et al.* [17] improved active learning by combining the instance-level feedback with the feature-level feedback.

In this paper, we aim at finding the object boundary from the edge map of an oversegmented image. We propose a Bayesian network (BN) [19], [20] to model this segmentation problem in a probabilistic way. Specifically, a multilayer BN is constructed to model regions, edge segments, vertices, their measurements, and their causal relationships. The BN also captures some local constraints to restrain the relationships among image entities. Given the BN, image segmentation is performed through a probabilistic inference. Thanks to the modeling power of BN, it is very easy to incorporate new constraints and measurements into the model. The human intervention can be easily added into the model as new evidence in an incremental, interactive, and sparing manner. This characteristic makes the proposed BN model very suitable for interactive image segmentation. In addition, we propose a new *active input selection* process to automatically generate a ranked list of candidates for the user to choose as the user's next intervention. This process can effectively reduce the total user's involvement for performance improvement. The impacts of newly input evidence are systematically propagated through the entire BN via belief propagation. In this way, constrained segmentation can be done in an incremental and interactive manner, fully utilizing the previous segmentation results.

II. RELATED WORK

BN is a powerful probabilistic graphical model that has been applied in computer vision. Early in 1990s, Sarkar and Boyer [21] applied BN to group low-level edge segments for high-level image understanding. Westling and Davis [22] applied BN

for the interpretation of complex scenes. Feng *et al.* [23] combined BN with neural network for scene segmentation. A trained neural network provides the local predictions for the class labels. These predications are then fused with the BN prior model for scene segmentation. Liu *et al.* [24] combine BN and MRF to form an image segmentation approach. The BN generates a probability map for all pixels. An MRF prior is further incorporated to produce the segmentation. Mortensen and Jia [25] used a two-layer BN model for image segmentation, which captures the relationships between edge segments and their junctions. Given a user-input seed path, they use the minimum-path spanning tree graph search to find the most likely object boundaries. Other than imposing a simple contour closure constraint, their method is basically a data-driven approach. They use the intensity gradient and local curvature as the edge's measurement. In addition, they encode a statistical similarity measure between the adjacent regions of an edge into its *a priori* probability, therefore, implicitly integrating region information. In their early study [26], they used a similar BN model for both automatic and IS. Their approach can find multiple nonoverlapping closed contours before any user's intervention is given. The user's input will serve as evidence to help select a single closed contour that encompasses the object of interest.

Although these researchers have successfully applied BN in their specific problems, most of them only use a simple BN structure (typically a naïve BN). For complex problems, this model may not be expressive enough to model many different kinds of image entities and their relationships. How to effectively capture these relationships using a BN is crucial to solve these difficult problems.

In our latest study on automatic image segmentation [27], we introduce a unified graphical model combining the directed graphical model with the undirected graphical model. It allows capturing more complex and heterogeneous relationships among image entities. Compared with the BN model proposed in this paper, the unified graphical model is more expressive and more powerful. But the unified model was only used for automatic segmentation. In contrast, this paper focuses on interactive image segmentation with the human in the loop. This paper also integrates an active learning approach into the interactive image segmentation, which is new in this domain.

Other than automatic image segmentation, several semi-automatic segmentation approaches have been proposed before. Intelligent scissors [13] allow the user to give seeds to guide the segmentation algorithm to find the low-cost path from the cursor position to the last seed position. For textured images, the low-cost path may be sensitive to the cursor position, and more seeds are needed to achieve a satisfactory result. Boykov and Jolly [5] modeled segmentation as an energy minimization problem and used the graph-cut technique to solve the optimization problem. The user provides some hard constraints that indicate certain pixels to be parts of the object or the background. "GrabCut" [11] extends the graph-cut segmentation [5] by reducing the human intervention to dragging a rectangle around the desired object. Freedman and Zhang [9] integrate a level-set-based shape prior into graph-cut-based segmentation. Similar to [5], the user marks some pixels as parts of the object or the background. An energy term related to the difference between the

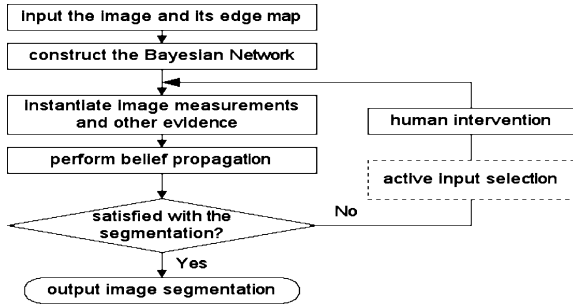


Fig. 1. Flowchart of the proposed segmentation approach.

segmentation shape and the prior shape is added into the energy function. Incorporating such shape priors brings the benefit of handling blurred edges or multiple similar objects close to one another. “Lazy snapping” [12] provides a user-friendly tool for IS, which separates coarse and fine-scale processing to improve the performance. Duchenne *et al.* [28] used the user-supplied seeds provided as a set of broad brush strokes as known pixel-wise labels and inferred the remaining pixel labels by a statistical transductive inference.

Although these interactive approaches are somewhat successful, most of them passively depend on the user to decide the next intervention and perform segmentation in a batch mode. However, sometimes it might be difficult for the user to select the most important intervention. An automatic process that can suggest a useful user intervention and reduce the user’s involvement would be more desirable. In addition, these existing approaches typically only integrate one type of the user’s intervention, and the intervention is often done in an ad hoc way, often ignoring the current segmentation results. Moreover, the user’s intervention is often provided as hard constraints without considering their uncertainties, and is usually provided before the segmentation starts. Mimicking the human’s decision process, it may be more beneficial to be able to provide different types of interventions, to account for their uncertainties, to provide interventions at any time of the segmentation process, and to integrate the user’s intervention with current segmentation results incrementally.

III. OVERVIEW OF THE APPROACH

To realize the goal of both automatic and interactive image segmentation, we need a model that can conveniently integrate image measurements, take the user’s input, and systematically propagate its impact on the segmentation results. For this purpose, we propose a multilayer BN model to find object boundaries according to measurements of regions (superpixels), edges, vertices (junctions), and local constraints. The local constraints capture our assumption that the object boundary should be locally smooth and have simple connectivity. Image segmentation is modeled as inferring the optimal states (e.g., on the object boundary or not) of these hidden variables from various measurements subject to constraints.

The flowchart of our approach is illustrated in Fig. 1. Given the edge map of an oversegmented image, we construct a multilayer BN to capture the local image regions, edge segments,

vertices, and their relationships. Various image measurements are then acquired, based on which an initial automatic segmentation is obtained through a probabilistic inference. For interactive image segmentation, the user examines the initial segmentation and provides appropriate interventions with/without the active input selection process. The influence of these user interventions on the states of other image entities is estimated through belief propagation. Finally, given the updated belief for each node, a probabilistic inference is performed to find the most probable explanation (MPE) of the region and edge nodes. Based on the MPE result, the most likely configuration of the superpixel regions and the edge segments form the final segmentation.

IV. IMAGE SEGMENTATION FRAMEWORK

We use BN as the basic framework for image segmentation. We construct the BN model from an oversegmentation of the image. Fig. 2 shows a concrete example. The edge map in the oversegmentation consists of superpixel regions (i.e., the regional blobs), edge segments, and vertices. The multilayer BN models the causal relationships among the superpixel regions $\{R_i\}_{i=1}^L$, edge segments $\{E_j\}_{j=1}^N$, vertices $\{V_k\}_{k=1}^S$, and angles $\{\theta_{ij}\}$. Angle nodes are introduced to mainly enforce the smooth connection between adjacent edges.

A. Regions, Edges, Vertex Nodes, and Their Measurements

We build the BN model based on an edge map in the oversegmentation of an image, which can be generated by any standard image segmentation approach such as watershed segmentation, normalized cuts, anisotropic diffusion, etc. Fig. 2(b) shows a small part of the edge map in the oversegmentation. The edge map consists of superpixel regions, edge segments, and vertices. Fig. 2(c) illustrates a multilayer BN that models the statistical relationships among these image entities. Specifically, the region layer contains all superpixel regions. The edge layer contains all edge segments, while the vertex layer contains all vertices. A vertex is the place, where three or more edges intersect with each other. The angle layer will be explained in Section IV-B.

The parents of an edge node are the two adjacent regions along this edge. If the parents of an edge E_j have different labels, it is more likely that there is a true object boundary between them, i.e., $E_j = 1$. Similarly, the edge nodes and the vertex nodes are causally linked. The parents of a vertex node are those edge segments that intersect at this vertex. The region nodes assume binary labels, i.e., foreground or background. Each edge node is a binary node, and its true state means that this edge segment belongs to the object boundary. The vertex node also assumes binary values (true or false).

The region nodes, edge nodes, and vertex nodes have their image measurements. The measurements of regions can be any feature vector extracted from the statistics of the superpixel region. In this paper, we use the average CIELAB color as the region features. The conditional probability $P(M_{R_i} | R_i)$ is modeled as a mixture of Gaussians (MOGs), which is learned from the training data. The conditional probability $P(E_j | \text{pa}(E_j))$ models the statistical relationships between the region labeling and the edge state, where $\text{pa}(E_j)$ denotes the parent nodes of E_j . In general, the edge E_j is more likely to be a true object

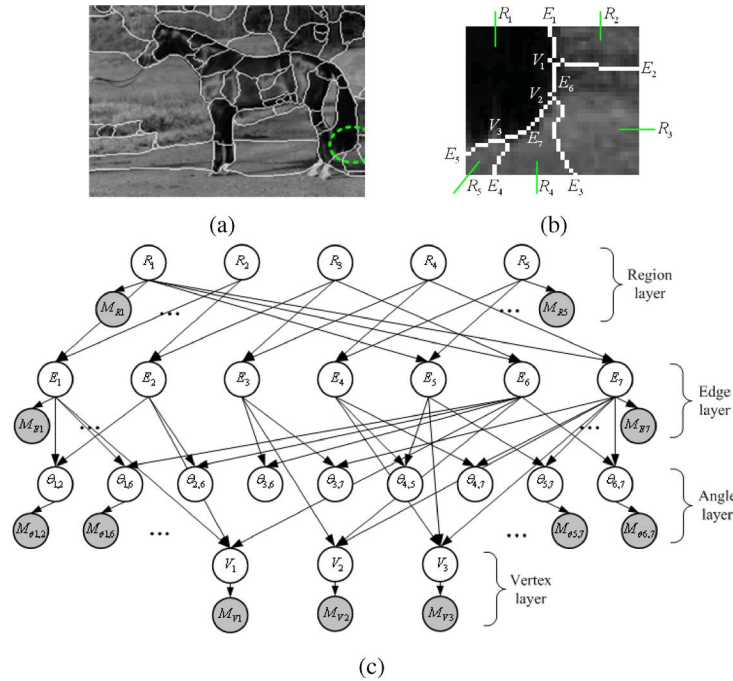


Fig. 2. BN framework that models the relationships among the superpixel regions, edge segments, vertices, angles, and their image measurements. (a) Example of the initial oversegmentation. (b) Partial edge map corresponding to the area inside the dotted line in (a). (c) BN model corresponding to (b).

boundary when its parents are assigned different labels. However, this relationship itself has uncertainty. We encode this relationship and its uncertainty by defining the conditional probability $P(E_j | \text{pa}(E_j))$ as follows:

$$P(E_j = 1 | \text{pa}(E_j)) = \begin{cases} 0.8, & \text{if the parent region nodes} \\ & \text{have different labels} \\ 0.2, & \text{otherwise.} \end{cases} \quad (1)$$

In this paper, we use the average intensity gradient magnitude as the measurement of each edge segment. More advanced statistical edge measurements can also be used. The measurement of the edge node E_j is denoted as M_{E_j} . Another layer is constructed to relate the edge nodes to their measurements (i.e., the shaded circles), as shown in Fig. 2(c). The measurement nodes are continuous nodes. The conditional probability $P(M_{E_j} | E_j)$ is parameterized using Gaussian distributions, which are learned from training data.

Similarly, each vertex node is also associated with a measurement node. The M_{V_k} node in Fig. 2(c) is the measurement of a vertex V_k . We use the Harris corner detector [29] to calculate this measurement. Let $I(x, y)$ denote the grayscale image. The Harris matrix A is given by

$$A = \begin{bmatrix} \left(\frac{\partial I}{\partial x}\right)^2 & \frac{\partial I}{\partial x} \frac{\partial I}{\partial y} \\ \frac{\partial I}{\partial x} \frac{\partial I}{\partial y} & \left(\frac{\partial I}{\partial y}\right)^2 \end{bmatrix}. \quad (2)$$

Given the matrix A , the strength of a corner is determined by a corner response function Res , i.e.,

$$Res = \det(A) - k \cdot \text{trace}(A)^2 \quad (3)$$

where k is set to 0.04, as suggested in [29]. The vertex measurement M_{V_k} is currently discretized according to the corner

response Res calculated by (3). If the corner response is above a threshold (empirically fixed as 1000) and it is a local maximum, a corner is detected and the measurement M_{V_k} becomes true. If no corner is detected, the measurement M_{V_k} becomes false.

The conditional probability that quantifies the statistical relationship between the vertex label V_k and its measurement M_{V_k} can be empirically modeled as follows:

$$\begin{aligned} P(M_{V_k} = 1 | V_k = 1) &= 0.99 \\ P(M_{V_k} = 1 | V_k = 0) &= 0.1. \end{aligned} \quad (4)$$

This definition basically means that the measurement uncertainty is low. These numbers may vary, depending on the quality of the used corner detector.

B. Local Smoothness Constraint

The boundary of a natural object is usually smooth. It is, therefore, less likely that sharp corners exist at the boundary of a natural object. We incorporate the local smoothness constraint by penalizing sharp corners between the edges. A sharp corner is defined as an angle that is less than a threshold. To impose this constraint, a new angle node $\theta_{i,j}$ is introduced to model the relationship between two intersecting edges E_i and E_j . It is a binary node with the true state meaning that the local smoothness constraint is violated by these two edges. The relationships among edge nodes, angle nodes, and their measurements are illustrated in Fig. 3. The conditional probability table (CPT) between an angle node and its measurement $M_{\theta_{i,j}}$ is similarly defined as (4).

The angle measurement $M_{\theta_{i,j}}$ is currently discretized according to the threshold $\pi/6$. If the angle is smaller than $\pi/6$, the measurement $M_{\theta_{i,j}}$ becomes 1 (true). Its parent angle node $\theta_{i,j}$ therefore has a high probability to be true (i.e., a sharp

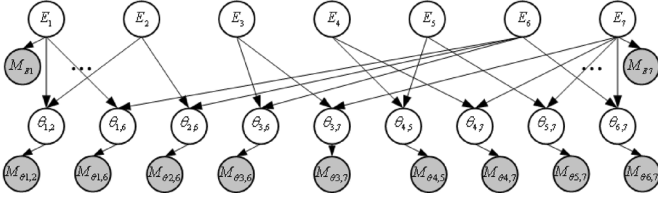


Fig. 3. BN with angle nodes to impose the local smoothness constraint.

corner), and the local smoothness constraint will be enforced. Under such condition, it is less likely that both edges E_i and E_j belong to the object boundary. We add a penalty for such edges by the links between angle nodes and edge nodes. To enforce the smoothness constraint, a CPT is defined to specify their relationships as follows:

$$P(\theta_{ij} = 1 | E_i, E_j) = \begin{cases} 0.2, & \text{if both } E_i \text{ and } E_j \text{ are true} \\ 0.5, & \text{otherwise.} \end{cases} \quad (5)$$

This CPT definition effectively reduces the probability of a pair of edge segments to be both on the object boundary if they form a sharp corner (i.e., $\theta_{ij} = 1$).

C. Simple Connectivity Constraint

In general, the boundary of an object should be simply connected, i.e., an edge segment should intersect with at the most one edge segment at its end point. This constraint is imposed by defining a CPT between the edge nodes and the related vertex node as follows:

$$P(V_k = 1 | \text{pa}(V_k)) = \begin{cases} 1, & \text{if exactly two parent } E \text{ nodes are true} \\ 0.3, & \text{if none of the parent } E \text{ nodes is true} \\ 0, & \text{otherwise} \end{cases} \quad (6)$$

where $\text{pa}(V_k)$ denotes the parent edge nodes of a vertex. We set the second entry as 0.3 because it is possible that none of the parent edge segments is the true object boundary. For example, there can be vertices detected in the background edge map. However, the conditional probability for this case shall be smaller than the case that exactly two parent edge nodes are true. With the above CPT definition, if V_k becomes 1 (true), it is most likely that exactly two parent edges are true boundary. The simple connectivity constraint is therefore enforced.

D. Complete BN Model

Given the BN components that model different parts of the segmentation problem, we combine them to form the complete BN model for image segmentation. In the model shown in Fig. 2(c), the R nodes represent all regions. The E nodes represent all edges. The θ nodes impose the local smoothness constraint on edges. The V nodes impose the connectivity constraint. The M_R , M_E , M_θ , and M_V nodes are the measurements of regions, edges, angles, and vertices, respectively. All nodes are binary nodes except the measurement nodes M_R

and M_E . Besides the natural causal relationships among image entities, local constraints are also imposed on the relationships among edges to encourage smooth and simple connections. We can calculate the joint probability distribution of all nodes as follows:

$$\begin{aligned} P(R, M_R, E, M_E, \theta, M_\theta, V, M_V) &= \prod_{i=1}^L [P(R_i)P(M_{R_i} | R_i)] \\ &\times \prod_{j=1}^N [P(E_j | \text{pa}(E_j))P(M_{E_j} | E_j)] \\ &\times \prod_{i=1, j \in \Omega_i}^N [P(\theta_{ij} | E_i, E_j)P(M_{\theta_{ij}} | \theta_{ij})] \\ &\times \prod_{k=1}^S [P(V_k | \text{pa}(V_k))P(M_{V_k} | V_k)] \end{aligned} \quad (7)$$

where $\text{pa}()$ denotes the parents of a node. $P(R_i)$ is the prior probability of R_i . Since we have no further prior information about the region labeling, $P(R_i)$ is set as a uniform distribution, which means that there is no bias for the region label. Ω_i denotes the set of edges that intersect with the edge E_i . $\text{pa}(V_k)$ denotes the parent edge nodes of the vertex V_k . L is the total number of regions. N is the total number of edge nodes and S is the total number of vertex nodes. The factorization of this joint probability is achieved by using the conditionally independent relationships among the nodes, which are implied by the BN structure and can be ascertained by the d -separation rules [19].

Given the measurements of regions, edges, vertices, and angles, the most probable states of hidden variables can be inferred by MPE inference, i.e.,

$$E^*, R^* = \arg \max_{E, R} P(E, R | M_R, M_E, M_\theta, M_V). \quad (8)$$

We employ the junction tree method [20] to perform the exact MPE inference. In the MPE result, the superpixel regions and the edge segments with their most likely states form the estimated object region and boundary.

V. INTERACTIVE IMAGE SEGMENTATION

Fully automatic segmentation may not produce correct segmentation when the image data information cannot discriminate the boundary edges from nonboundary edges. Under such situations, the user's intervention can provide additional information to aid the automatic approach to produce correct segmentation. Existing IS approaches usually passively require the user to provide exact information such as labels for some edges, pixels, or regions. In contrast, we propose a new active input selection process that can improve both efficiency and accuracy of the IS.

A. Interactive Image Segmentation

For the interactive image segmentation, the user's interactions are incorporated as new evidence by instantiation of some nodes. Let U denote the set of variables that are instantiated by

the user's input and X denote the hidden variables (E, R) excluding U . Their hidden states can be inferred as follows:

$$X^* = \arg \max_X P(X | U, M_R, M_E, M_\theta, M_V). \quad (9)$$

Compared to the existing IS approaches, the intervention in our model can be given anytime, of different types, and can be applied incrementally. Moreover, the effect of the intervention on the states of other hidden nodes are systematically estimated via a principled probabilistic inference.

B. Active Input Selection

The active input selection process helps the user to identify the next best intervention to take in order to minimize the overall user's annotations and to simultaneously improve the segmentation. Given the current segmentation result, the user first identifies the most problematic area using a mouse click. The image region within a certain radius around the clicked point will be considered for analysis. The computer program then searches for the image entities (regions, edges, vertices, and angles) within this area and calculates a ranked list of these image entities according to certain criterion. We propose to use the mutual information (MI) to help identify the next best action to take. MI measures the potential of a single user intervention to reduce the overall uncertainty of the BN segmentation model. Using MI, the system will rank a list of possible user's interventions on the identified entities based on their contributions to improve subsequent image segmentation. The user can then select the one from the ranked list that can maximally reduce the overall uncertainty of the model, and therefore, maximally improve the subsequent overall segmentation. In our current implementation, the entities are listed in a descending order according to their MI with the BN model.

The MI between a single user intervention and the BN model can be calculated as follows:

$$I(\mathbf{E}, \mathbf{R}, \mathbf{V}, \theta; U_i) = \sum_{\mathbf{E}, \mathbf{R}, \mathbf{V}, \theta} P(\mathbf{E}, \mathbf{R}, \mathbf{V}, \theta, U_i) \times \log \frac{P(\mathbf{E}, \mathbf{R}, \mathbf{V}, \theta, U_i)}{P(\mathbf{E}, \mathbf{R}, \mathbf{V}, \theta)} \quad (10)$$

where U_i denotes the intervention that will be given next time and $(\mathbf{E}, \mathbf{R}, \mathbf{V}, \theta)$ are all the hidden nodes excluding U_i .

Using (10), a ranked list of actions to take next can be generated. With the ranked list of candidate interventions, the user can select the identified image entity that has the highest rank (i.e., the maximum MI) and is not correctly labeled to act upon next. Our current implementation automatically selects the top-ranked entity and flips its current state if the user decides its current state is incorrect. Once this intervention is input into the BN model, a new MPE inference is performed to propagate its impact and to produce a new segmentation. The aforementioned process can be iterated to incrementally improve the segmentation result.

In terms of actively determining the next best intervention, our method resembles some active learning techniques. There are two main categories of active learning methods. The first category selects the most informative sample (typically a sample with the largest uncertainty) for labeling, independent of other

unlabeled samples. The second one selects a sample that minimizes the uncertainties on the remaining unlabeled samples. Our method is similar to the second kind in that we are also looking for an image entity (region, edge, and vertex) for the user to annotate so that the annotation of this image entity would minimize the uncertainties on other image entities. The key difference between our approach and the conventional active learning method is that we estimate the expected uncertainties on the remaining image entities jointly via belief propagation in the BN. The active learning approach, on the other hand, needs to retrain a classifier using the newly labeled data and then assess each sample's uncertainty using the newly trained classifier individually. In addition, our active learning can handle heterogeneous data because region, edge, and vertex entities in our cases are of different types, while the conventional ones usually only handle homogeneous samples.

VI. EXPERIMENTS

A. Automatic Image Segmentation

We first tested our model for automatic segmentation on two sets of testing images: 1) 110 images from the Weizmann horse dataset [30] and 2) 45 cow images from the VOC2006 database [31]. The Weizmann dataset includes the side views of many horses that have different appearances and poses, which makes it challenging to segment them. In order to learn the MOGs distributions for the region measurements, we use 60 other horse images for training. We use the Edgeflow-based anisotropic diffusion software [32] to generate the oversegmented edge map of each image. The superpixel regions, edge segments, and vertices are automatically extracted from this edge map to construct the BN model. For the training process, we collect the average CIELAB color of each region as the region measurements. The likelihood model of these measurements are learnt by MOG analysis. We also collect the average gradient magnitude of the edge segments from the training data and learn the likelihood model of the edge measurements.

Other conditional probabilities are preset and fixed in all our experiments (see Section IV). We empirically set these parameters due to several reasons. First, we can directly define these CPTs according to their conceptual meaning. Second, some previous work [33] shows that the performance of a BN is not very sensitive to accurate parameter setting. Third, we have changed some CPTs within a range of $\pm 10\%$ – 20% . The segmentation results did not change very much, which echoes with the observations in [33]. Fourth, this parameterization allows us to apply the model to segment other images without much reparameterization.

It should be noted that our BN model requires limited training to learn the likelihood models. The model can be extended to multiclass segmentation as well, where only the region measurements are object class dependent and should be learned for each class. This property allows the model to be easily generalized with little retraining process.

During testing, given all the measurements and the parameterized BN model, image segmentation can be performed using the process described in Section IV-D. Fig. 4 shows some example segmentation results of the Weizmann horse images. By



Fig. 4. Examples of image segmentation results arranged in two groups of two rows. In each group, the first row includes the test images and the second row includes the segmentation masks produced by the proposed approach.

TABLE I
QUANTITATIVE COMPARISON OF RESULTS. THE IS RESULT IS THE FIRST ROUND OF IS WITH ACTIVE INPUT SELECTION, WHERE AT THE MOST ONE HUMAN INTERVENTION IS ALLOWED FOR EACH IMAGE

approaches	segmentation accuracy
Cour et al. [34]	94.2%
Winn et al. [35]	93.1%
Ren et al. [36]	91.0%
this paper (automatic segmentation)	93.7%
this paper (interactive segmentation)	95.6%

visual inspection, we achieved reasonable results on these images. The errors mainly come from the appearance changes and the cluttering of background objects (e.g., the shadow) that have similar appearances as the horses.

To quantitatively evaluate the segmentation performance, we calculated the percentage of correctly labeled pixels (i.e., segmentation accuracy). Table I shows our results compared to three related works that also performed experiments on the Weizmann dataset [34]–[36]. Our automatic segmentation performance is ranked the second and just lower than [34], without using any shape prior, while both [34] and [35] used shape constraints. Table I also lists the performance of our IS with at most one active intervention per image. Our IS performance surpasses other approaches, which shows that even very limited user's intervention can significantly improve the segmentation performance.

Previously, we mention that the BN segmentation is not very sensitive to the accurate setting of BN parameters. To empirically study this issue, we performed two sets of experiments on the Weizmann dataset. First, we changed the CPTs of all edge nodes. Specifically, we changed the CPT $P(E_{ij} = 1 | R_i \neq R_j)$ from 0.7 to 0.9 and all related CPT entries for all edge nodes. We redid image segmentation on the same set of Weizmann images. The results are summarized in Fig. 5. We observed that although the CPT values changed within $\pm 10\%$ with respect to the preset value (0.8), the segmentation accuracy only slightly changed (less than 0.5%). These results demonstrated the insensitivity of BN segmentation to the accurate setting of BN parameters.

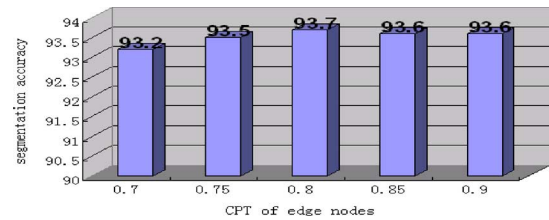


Fig. 5. Change of segmentation performance with respect to the change of all edge nodes' CPTs. The horizontal axis is the value of CPT entry $P(E_{ij} = 1 | R_i \neq R_j)$. The vertical axis is the average segmentation accuracy for the experiments on the Weizmann images.

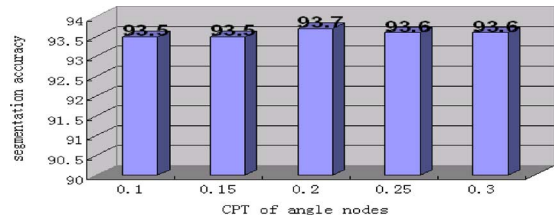


Fig. 6. Change of segmentation performance with respect to the change of all angle nodes' CPTs. The horizontal axis is the value of CPT entry $P(\theta_{ij} = 1 | E_i = 1, E_j = 1)$. The vertical axis is the average segmentation accuracy for the experiments on the Weizmann images.

To further study this issue, we also changed the CPT of the angle nodes $P(\theta_{ij} = 1 | E_i = 1, E_j = 1)$ from 0.1 to 0.3 and all related CPT entries. We did similar experiments as above and summarized the segmentation accuracy in Fig. 6. We observed a quite similar phenomenon. The segmentation accuracy changed marginally (less than 0.2%) in this set of experiments.

From the aforementioned experiments, we empirically demonstrate that the BN segmentation model is not very sensitive to the accurate parameter setting. Our results also agree with the findings in other researchers' prior work [33]. Although it might be difficult to theoretically prove this phenomenon, we speculate that since our BN segmentation is based on the MPE inference, it searches for the optimal configuration based on the junction tree algorithm. Although the change of certain BN parameters might change the exact value of the joint probability under certain configuration, it may not change the order of the joint probability under different configurations.



Fig. 7. Examples of image segmentation results on the cow images from the VOC2006 database, where the first row shows the original images, while the second row displays the segmentation results.

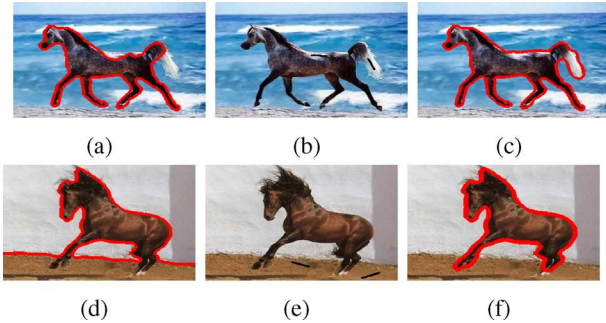


Fig. 8. Image segmentation with the user’s intervention. The three columns are the initial automatic segmentation results, the images with black strokes input by the user to indicate either foreground seeds (b) or background seeds (e), and the superimposed object boundary produced by image segmentation with the user’s intervention, respectively.

Hence, the MPE configuration will not change in many cases and the segmentation result remains the same.

Besides the Weizmann dataset, we also tested the automatic segmentation on the cow images from the VOC2006 database. We randomly select 50 cow images from this database to train the BN model (to mainly learn the region likelihood models) and then test it on a different set of 45 cow images. Example results are shown in Fig. 7. We achieved an overall accuracy of 94.5% for the automatic segmentation.

B. Passively IS

We notice that automatic segmentation may have difficulty in segmenting low contrast image areas, where the edge measurements cannot differentiate true edges from those edges that are not object boundary. The regions along these edges also have similar features so that they tend to be assigned the same label. Hence, both the edge information and the region information cannot overcome this difficulty. Under such cases, human intervention is needed for further improvement.

Fig. 8 shows two examples, where the automatic segmentation has difficulty in correctly segmenting some body parts. The user can use strokes to indicate some foreground or background seeds. The superpixels or edges crossed by the strokes are then labeled accordingly. The new evidence is taken into the BN by instantiating the related nodes. The impact of the new evidence on other nodes can be automatically estimated through belief propagation, which updates the states of the whole BN to produce the improved segmentation.

Due to its convenience to incorporate new evidence, the BN model allows the user to very easily interact with the segmentation process. The intervention can be performed at any stage of the segmentation process. It can be made before the image segmentation starts or added during the segmentation process. Moreover, various types of interventions can be selectively performed by the user. The user’s interventions can include but are not limited to the following types:

- 1) changing the state of an edge/region node;
- 2) giving inputs as foreground or background seeds;
- 3) selectively enforcing the local constraints in some places;
- 4) overriding the image measurements;
- 5) changing the uncertainties of some image measurements.

In other words, the user can change the state of any node in the BN, based on his knowledge about the object boundary. Given any input from the user, belief propagation using the junction tree method [20] is performed to propagate the impact of the user’s input and to update the beliefs of all hidden variables. The beauty of belief propagation includes: 1) it is done incrementally based on the current state of the model, without having to restart from the scratch again, and 2) it can extend the impact of the user’s input to any node in the BN in a principled way, not only limited to the nearby nodes.

We give an example in Fig. 9 to illustrate these characteristics. This figure shows the segmentation results at two stages. The user incrementally gives two kinds of interventions by indicating some foreground/background seeds using strokes and relaxing the connectivity constraint on a vertex by manipulating its CPTs. With two-step interventions, the framework gradually improves the segmentation results. It can be also observed that one human input may change the states of the whole network and produce the new segmentation.

It should be noted that the different types of interventions mentioned previously require different levels of the user’s understanding and manipulation of the BN model. For example, it is very easy to change the state of binary nodes by switching their states. However, relaxing the local constraints by changing the CPT values requires a deep understanding of the meaning of those CPTs. Changing the image measurements might be even harder since it is not easy for the user to mentally estimate those measurements. In our current implementation, we allow the user to interactively change the states of region nodes, edge nodes, vertex nodes, and their CPTs.

Finally, the IS demonstrated so far solely depends on the user to select next interventions. We therefore call it as *passively IS*.



Fig. 9. Example of incremental IS. Three black strokes are input as foreground/background seeds in (a). The BN model then produces the result in (b). The connectivity constraint is relaxed in one vertex in (c), which is highlighted by the white circle. Other small black dots indicate other vertices. The model finally produced the segmentation result in (d). Please note that several leg parts are corrected simultaneously in (d).

TABLE II
AVERAGE ERROR RATES AFTER DIFFERENT ROUNDS OF THE USER'S INTERVENTIONS. THE NUMBERS IN THE BRACKETS ARE THE CUMULATIVE HUMAN INTERVENTIONS THAT HAVE BEEN GIVEN TO 110 HORSE IMAGES

	initial	round 1	round 2	round 3	round 4
active	6.3%	4.41% (94)	3.55% (171)	3.08% (223)	2.56% (287)
passive	6.3%	4.7% (101)	3.74% (187)	2.94% (261)	2.58% (325)
GrabCut	8.8%	6.8% (94)	5.8% (178)	5.1% (258)	4.7% (333)

TABLE III
AVERAGE ERROR RATES AFTER DIFFERENT ROUNDS OF THE USER'S INTERVENTIONS. THE NUMBERS IN THE BRACKETS ARE THE CUMULATIVE HUMAN INTERVENTIONS THAT HAVE BEEN GIVEN TO 45 COW IMAGES

	round 1	round 2	round 3	round 4	round 5
active	3.3% (41)	1.9% (69)	1.4% (87)	1.14% (100)	1.08% (106)
passive	4.4% (41)	2.8% (73)	2.2% (97)	1.8% (117)	1.36% (133)

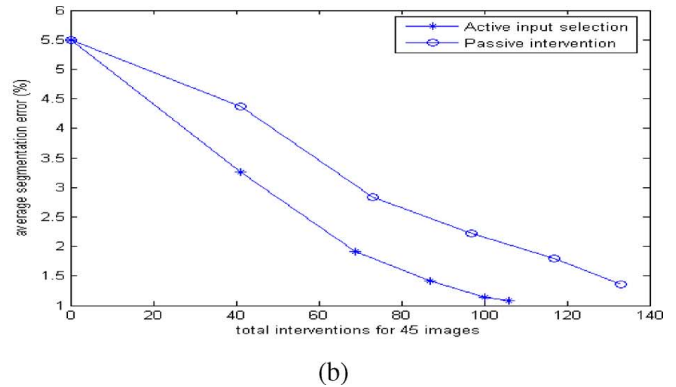
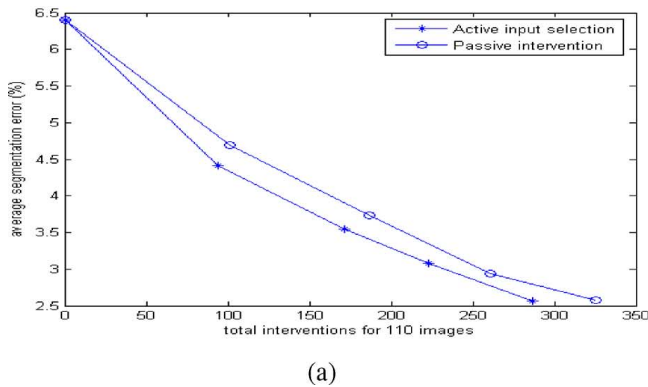


Fig. 10. Decrease of average segmentation errors with respect to the cumulative human interventions. (a) For 110 images from the Weizmann dataset. (b) For 45 images from the VOC2006 dataset.

C. Actively IS

Besides the fully automatic segmentation and the passively IS, we have performed IS with active input selection on both sets of testing images. In each round of IS, we allow the user to give at the most one human intervention per image. Here, we focus on the first type of intervention, i.e., changing the state of a region/edge node. In each round, all testing images are checked once to identify whether it requires the user's input to improve segmentation or not. The average segmentation error is calculated when a round of IS is completed. This process is iterated by several rounds to gradually improve the segmentation.

In the IS with active input selection, the computer program calculates a ranked list of candidates for the user's next intervention. The user then gives intervention based on this candidate list. In contrast, in the passively IS, the user just arbitrarily selects the next intervention based on his own judgment.

We compare the performance of both strategies through the overall segmentation error rate. These results are summarized in Tables II and III. In both tables, we observe that the user's intervention can significantly reduce the error rate. For example, the error rate of horse images decreases from 6.3% to 4.4% with only one round of active intervention. We also observe that the actively IS can improve both the IS accuracy and efficiency.

Fig. 10 shows the change of error rates with respect to the number of cumulative interventions given to all images. Compared to the passive intervention, the active intervention can reduce the segmentation error faster. It requires 287 active interventions for 110 horse images to reduce the average error rate to 2.56%, while the passive intervention requires 325 interventions to reduce the average error rate to 2.58%. With comparable error rates, the active intervention is about 12% more efficient than the passive intervention.

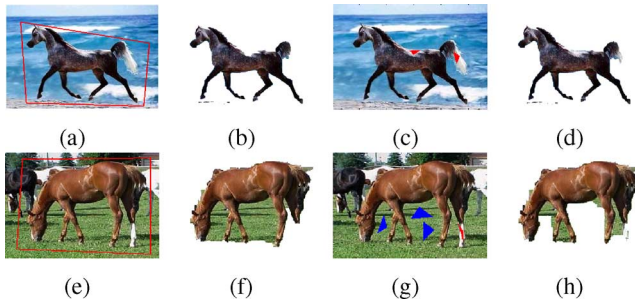


Fig. 11. GrabCut IS. The four columns are the raw images with the polygon input by the user, the initial segmentation, the additional user's inputs of foreground seeds (red polygons) or background seeds (blue polygons), and the segmentation results after all users' inputs, respectively.

For the cow dataset, the difference is even more significant. For example, the error rate reduces to 1.4% with 87 cumulative interventions for 45 cow images using the active intervention, while the error rate reduces to 1.36% with 133 passive interventions. The active intervention is about 35% more efficient than the passive intervention. These results demonstrate that the active input selection process can reduce the total user's involvement for IS and make the IS more efficient.

D. Comparison With GrabCut IS

In addition to the aforementioned experiments, we also compare our approach with the popular GrabCut IS [11]. We downloaded a publicly shared GrabCut software [37]. This software can initially do GrabCut segmentation with a user's input bounding polygon. It also allows further adding foreground/background seeds using polygons to refine the segmentation results, which in principle is very similar to the Lazy-Snapping technique [12]. We use this GrabCut software to segment the Weizmann horse images used in this paper. For each image, we first give a polygon and perform GrabCut to segment the image. If there are still significant segmentation errors, we further give additional foreground/background seeds to gradually improve the segmentation. We maximally give four additional inputs for each image.

We summarize the quantitative experimental results in the third row of Table II. From this table, we found that our approach achieved generally better performance than the GrabCut approach. Our automatic segmentation achieved 6.3% error rate and the GrabCut approach with the polygon input achieved 8.8% error rate. In addition, our approach more effectively and efficiently improves the segmentation when additional users' inputs are given. For the actively IS, our total number of the user's inputs for each round of IS is smaller than that of the GrabCut segmentation. Please note that for the initial segmentation, the GrabCut approach actually already used one human input (i.e., the polygon) for each image but we did not count it in.

In Fig. 11, we showed two typical GrabCut segmentation results. Compared with our results shown in Figs. 8 and 9, our approach can more effectively correct those misclassified areas. We think that this may be due to two major reasons: first, our approach is a superpixel-based approach, while the GrabCut is a pixel-based approach. When new intervention is given, our

approach can correct the error more effectively. Second and more importantly, our BN-based approach quickly propagates the impact of new intervention throughout the network and corrects more erroneous areas. This phenomenon has been demonstrated in Fig. 9(c) and (d). In contrast, the GrabCut approach propagates the impact of new intervention slowly, as shown in Fig. 11(h). With four polygons input by the user, it still did not roughly find the correct object boundary.

Although our approach achieves better segmentation accuracy, we did find some of its shortcomings compared to the GrabCut approach. First, our approach is a superpixel-based approach. Its performance may be limited by the granularity of those superpixels. If one wants to refine the segmentation to the single pixel level or subpixel level, we need to integrate some pixel-based approach with our approach for fine-grained refinement. Considering this aspect, we think our approach is mainly useful for those problems that may not require the single pixel level segmentation accuracy.

Second, the GrabCut approach is running faster than ours. In an old laptop (IBM T42, Pentium M 1.7 GHz CPU, 1.5 GB RAM), the average running time for the initial GrabCut segmentation is about 36 s per image, which is about 1/3 of the average running time (112 s) of the initial automatic segmentation by our approach. But we did find that the speed of GrabCut segmentation significantly depends on the implementation. We have tried to use another GrabCut code [38], which requires about twice the running time of the GrabCut software we finally chose. In addition, once the initial segmentation is produced, further IS by both approaches is fast. It usually takes our approach 1–4 s to finish one step of IS.

Our prototype is implemented in MATLAB 7.0. The most time-consuming part in our approach is building the BN model and converting it to a junction tree for exact MPE inference, whose computation depends on the complexity (e.g., the number of nodes, tree width) of the model. We can use C++ implementation and approximate inference approaches such as loopy belief propagation to further speed up our system.

E. Weak- and Self-Learning of Region Distributions

In our BN model, we learned the distributions of image measurements using supervised learning. Although the distributions of edge measurements, vertex measurements, and angle measurements are usually very general, the distributions of region measurements may depend on the used dataset. We performed two sets of additional experiments to study the influence of region measurement distributions on the segmentation performance.

First, we used the region likelihood models learned from the VOC2006 cow dataset to segment the Weizmann horse images. We call this as “*weak-learning*” of the region measurement distributions since we used a completely different dataset for training. We performed automatic segmentation on the same set of Weizmann images and found that the average labeling accuracy decreased from 93.7% to 90.7%. This is a moderate performance decrease but can be easily understood because we used a different dataset to train the model.

Second, in order to improve the generalization of our IS approach, we modified the code so that we could leverage the

polygon used by the GrabCut approach to do “self-learning” of the region likelihood models. Specifically, the polygon used in the GrabCut actually tells us that the exterior area should be the background but the interior area could be the mixture of both foreground and background. We used the exterior pixels to learn the region likelihood model for the background area. Once we learned the region likelihood model for the background, we used it to label the interior area and found the area with low background probability. These areas have high likelihood to be the foreground since they are significantly different from the background. We used these areas to learn the region distributions for the foreground. In this way, we can adaptively learn the region distributions for both the foreground and background in each individual image.

In addition, the input polygon tells us the background area. We also use it to adjust the prior probability of region nodes. Specifically, if $x\%$ pixels of a superpixel region is outside of the polygon, we adjust its prior probability of being the background to $x\%$. Similarly, we also adjust the evidence of the edge nodes.

We used the self-learning approach to learn the region distribution for each image, given the polygon used by the GrabCut approach. We redid the experiments on the Weizmann images and found this self-learning approach performed well compared to our original automatic segmentation. We achieved about 95.3% average labeling accuracy in these experiments, which is better than our original automatic segmentation performance (93.7%). This performance is also better than the GrabCut approach that uses the same polygon as the input to produce the initial segmentation. Finally, we want to mention that we did not use the iterative self-learning to gradually improve the estimation of foreground/background region distributions, although this strategy is usually used by the GrabCut approach and may further improve the accuracy.

VII. SUMMARY

In this paper, we first propose a model-based segmentation approach based on BN. The BN systematically encodes the relationships among regions, edge segments, vertices, angles, and their measurements. The BN also models the local smoothness constraint and the simple connectivity constraint. Given the BN model, image segmentation is performed through belief propagation and probabilistic inference. The final segmentation is achieved by finding the MPE result that is consistent with all the measurements and constraints.

The proposed model is demonstrated to be able to perform fully automatic image segmentation, comparable to or outperforming several related methods. In addition, we further extend it to be used for interactive image segmentation. Specifically, we propose a new active input selection process to perform actively IS using the proposed BN model. This active selection can effectively improve the overall segmentation accuracy and reduce the total user’s involvement. We demonstrate the concept of actively IS and compare it with passively IS. Our experimental results demonstrate the promising capability of the proposed BN model for both automatic image segmentation and effective interactive image segmentation.

The advantages of using a BN to model segmentation problems can be attributed to several aspects. First, it provides a

systematic way to model the image segmentation problem in a probabilistic manner. Second, it is very convenient to incorporate new knowledge into the model due to the expressive and inference power of BNs. Third, the user’s intervention can be easily incorporated into the BN as new evidence in an incremental way. Unlike other IS methods, multiple types of the user’s interventions can be added at any stage of the segmentation process. The impact of new evidence will be propagated throughout the whole network in a principled manner. These characteristics are very important, especially for the segmentation of complex images.

Finally, although our BN model is focused on the figure/ground segmentation problem, it can be easily extended to deal with multiclass segmentation. This extension can be done in two ways. First, we can design a series of BN models for segmenting each class versus the background. Applying these models will generate a multiclass segmentation. Second, we can extend the binary nodes (especially the region nodes) in the BN to discrete nodes with multiple states, allowing distinguishing multiple object classes. The local constraint such as the connectivity constraint should be changed accordingly since there could be complex boundaries in a multiclass segmentation. We plan to study this extension in future.

REFERENCES

- [1] D. Comaniciu and P. Meer, “Mean shift: A robust approach toward feature space analysis,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 5, pp. 603–619, May 2002.
- [2] L. Vincent and P. Soille, “Watersheds in digital spaces: An efficient algorithm based on immersion simulations,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 13, no. 6, pp. 583–598, Jun. 1991.
- [3] V. Caselles, R. Kimmel, and G. Sapiro, “Geodesic active contours,” *Int. J. Comput. Vis.*, vol. 22, no. 1, pp. 61–79, Feb. 1997.
- [4] J. Shi and J. Malik, “Normalized cuts and image segmentation,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 8, pp. 888–905, Aug. 2000.
- [5] Y. Boykov and M.-P. Jolly, “Interactive graph cuts for optimal boundary and region segmentation of objects in N-D images,” in *Int. Conf. Comput. Vis.*, 2001, vol. 1, pp. 105–112.
- [6] S. Geman and D. Geman, “Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. PAMI-6, no. 6, pp. 721–741, Nov. 1984.
- [7] H. Nguyen and Q. Ji, “Improved watershed segmentation using water diffusion and local shape priors,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2006, pp. 985–992.
- [8] T. Chan and W. Zhu, “Level Set Based Shape Prior Segmentation,” in *Proc. 2005 IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, 2005, pp. 1164–1170.
- [9] D. Freedman and T. Zhang, “Interactive graph cut based segmentation with shape priors,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2005, pp. 755–762.
- [10] J. Winn and J. Shotton, “The layout consistent random field for recognizing and segmenting partially occluded objects,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2006, pp. 37–44.
- [11] C. Rother, A. Blake, and V. Kolmogorov, “Grabcut: Interactive foreground extraction using iterated graph cuts,” in *Proc. SIGGRAPH*, 2004, pp. 309–314.
- [12] Y. Li, J. Sun, C.-K. Tang, and H.-Y. Shum, “Lazy snapping,” in *Proc. SIGGRAPH*, 2004, vol. 23, no. 3, pp. 303–308.
- [13] E. N. Mortensen and W. A. Barrett, “Intelligent scissors for image composition,” in *Proc. SIGGRAPH*, 1995, pp. 191–198.
- [14] Y. Freund, H. Seung, E. Shamir, and N. Tishby, “Selective sampling using the query by committee algorithm,” *Mach. Learning*, vol. 28, pp. 133–168, 1997.
- [15] D. Shen, J. Zhang, J. Su, G. Zhou, and C.-L. Tan, “Multi-criteria-based active learning for named entity recognition,” presented at the 42nd Annu. Meeting Assoc. Comput. Linguist., Stroudsburg, PA, 2004.
- [16] A. Kapoor, K. Grauman, R. Urtasun, and T. Darrell, “Active learning with Gaussian processes for object categorization,” in *Proc. IEEE Int. Conf. Comput. Vis.*, 2007, pp. 1–8.

- [17] H. Raghavan, O. Madani, and R. Jones, "Active learning with feedback on both features and instances," *J. Mach. Learning Res.*, vol. 7, pp. 1655–1686, 2006.
- [18] S. Tong and D. Koller, "Support vector machine active learning with applications to text classification," *J. Mach. Learning Res.*, vol. 2, pp. 45–66, 2000.
- [19] J. Pearl, *Probabilistic Reasoning In Intelligent Systems: Networks of Plausible Inference*. San Mateo, CA, : Morgan Kaufmann, 1988.
- [20] F. V. Jensen, *Bayesian Networks and Decision Graphs*. New York, : Springer-Verlag, 2001.
- [21] S. Sarkar and K. L. Boyer, "Integration, inference, and management of spatial information using Bayesian networks: Perceptual organization," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 15, no. 3, pp. 256–274, Mar. 1993.
- [22] M. F. Westling and L. S. Davis, "Interpretation of complex scenes using Bayesian networks," in *Proc. Asian Conf. Comput. Vis.*, 1998, vol. 2, pp. 201–208.
- [23] X. Feng, C. Williams, and S. Felderhof, "Combining belief networks and neural networks for scene segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 4, pp. 467–483, Apr. 2002.
- [24] F. Liu, D. Xu, C. Yuan, and W. Kerwin, "Image segmentation based on Bayesian network-Markov random field model and its application on in vivo plaque composition," in *Int. Symp. Biomed. Imag.*, 2006, pp. 141–144.
- [25] E. N. Mortensen and J. Jia, "Real-time semi-automatic segmentation using a Bayesian network," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2006, pp. 1007–1014.
- [26] E. Mortensen and J. Jia, "A Bayesian network framework for real-time object selection," in *Workshop Percept. Org. Comput. Vis.*, 2004, p. 44.
- [27] L. Zhang and Q. Ji, "Image segmentation with a unified graphical model," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 8, pp. 1406–1425, Aug. 2010.
- [28] O. Duchenne, J. Audibert, R. Keriven, J. Ponce, and F. Ségonne, "Segmentation by transduction," in *IEEE Conf. Comput. Vis. Pattern Recognit.*, 2008, pp. 1–8.
- [29] C. Harris and M. Stephens, "A combined corner and edge detector," in *4th Alvey Vis. Conf.*, 1988, pp. 147–152.
- [30] E. Borenstein, E. Sharon, and S. Ullman, "Combining top-down and bottom-up segmentation," in *CVPR Workshop Perceptual Org. Comput. Vis.*, 2004, p. 46.
- [31] M. Everingham, "The PASCAL visual object classes challenge 2006," University of Oxford [Online]. Available: <http://www.pascal-network.org/challenges/VOC/databases.html>
- [32] B. Sumengen and B. S. Manjunath, "Edgeflow-driven variational image segmentation: Theory and performance evaluation," Univ. California, Santa Barbara, Tech. Rep. 2005 [Online]. Available: <http://barissumengen.com/seg/>
- [33] M. Pradhan, M. Henrion, G. M. Provan, B. D. Favero, and K. Huang, "The sensitivity of belief networks to imprecise probabilities: An experimental investigation," *Artif. Intell.*, vol. 85, no. 1–2, pp. 363–397, 1996.
- [34] T. Cour and J. Shi, "Recognizing objects by piecing together the segmentation puzzle," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2007, pp. 1–8.
- [35] J. Winn and N. Jovic, "LOCUS: Learning object classes with unsupervised segmentation," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2005, pp. 756–763.
- [36] X. Ren, C. C. Fowlkes, and J. Malik, "Cue integration in figure/ground labeling," in *Proc. Adv. Neural Inf. Process. Syst.*, 2005, pp. 1121–1128.
- [37] D. Irena and S. Aviad, Implementing the GrabCut segmentation technique [Online]. Available: http://www1.idc.ac.il/toky/CompPhoto-09/Projects/Stud_projects/IrenaAviad/W eb/index.htm
- [38] M. Gupta and K. Ramnath, Interactive segmentation tool-box [Online]. Available: <http://www.cs.cmu.edu/mohitg/segmentation.htm>



Lei Zhang (M'09) received the Ph.D. degree in electrical engineering from Rensselaer Polytechnic Institute, Troy, NY, in 2009.

He is currently a Senior Research and Development Scientist at UtopiaCompression Corporation in Los Angeles, CA. He has designed different probabilistic graphical models for solving many problems, including image segmentation, upper body tracking, facial expression recognition, human activity recognition, medical image processing, multimodal sensor fusion, etc. He is the author or coauthor of more than

20 papers in several top journals, international conferences, and book chapters in different domains. His research interests include machine learning, computer vision, pattern recognition, and image processing.

Dr. Zhang is a reviewer for several top computer vision and image processing journals. He is a full member of Sigma Xi, The Scientific Research Society.



Qiang Ji (SM'04) received the Ph.D. degree in electrical engineering from the University of Washington, Seattle, in 1998.

He is currently a Professor in the Department of Electrical, Computer, and Systems Engineering, Rensselaer Polytechnic Institute (RPI), Troy, NY, where he is also the Director of the Intelligent Systems Laboratory. Recently, he was a Program Director at the National Science Foundation (NSF), where he managed NSF's computer vision and machine learning programs. He also held teaching

and research positions with the Beckman Institute at the University of Illinois at Urbana-Champaign, the Robotics Institute at Carnegie Mellon University, the Department of Computer Science at University of Nevada at Reno, and the U.S. Air Force Research Laboratory. His research interests include computer vision, probabilistic graphical models, information fusion, and their applications in various fields. He is the author or coauthor of more than 150 papers in peer-reviewed journals and conferences.

Prof. Ji is an Editor of several related IEEE and international journals. He has been as a Chair, Technical Area Chair, and Program Committee Member in numerous international conferences/workshops. His research has been supported by major governmental agencies including NSF, National Institutes of Health, Defense Advanced Research Projects Agency, Office of Naval Research, Army Research Office, and Air Force Office of Scientific Research, as well as by major companies including Honda and Boeing.