

Labeling Irregular Graphs with Belief Propagation

Ifeoma Nwogu and Jason J. Corso

State University of New York at Buffalo
Department of Computer Science and Engineering
201 Bell Hall,
Buffalo, NY 14260
{inwogu, jcorso}@cse.buffalo.edu

Abstract. This paper proposes a statistical approach to labeling images using a more natural graphical structure than the pixel grid (or some uniform derivation of it such as square patches of pixels). Typically, low-level vision estimations based on graphical models work on the regular pixel lattice (with a known clique structure and neighborhood). We move away from this regular lattice to more meaningful statistics on which the graphical model, specifically the Markov network is defined. We create the irregular graph based on superpixels, which results in significantly fewer nodes and more natural neighborhood relationships between the nodes of the graph. Superpixels are a local, coherent grouping of pixels which preserves most of the structure necessary for segmentation. Their use reduces the complexity of the inferences made from the graphs with little or no loss of accuracy. Belief propagation (BP) is then used to efficiently find a local maximum of the posterior probability for this Markov network. We apply this statistical inference to finding (labeling) documents in a cluttered room (under moderately different lighting conditions).

1 Introduction

Our goal in this paper is to label (natural) images based on generative models learned from image data in a specific imaging domain, such as labeling an office scene as documents or background (see figure 1). It can be argued that object description and recognition are the key goals in perception. Therefore, the labeling problem of inscribing and affixing tags to objects in images (for identification or description) is at the core of image analysis. But Duncan et al. [3] describe how a discrete model labeling problem (where every point has only a constant number of candidate labels) is NP-complete. The conventional way of solving this discrete labeling in computer vision is by stochastic optimization such as simulated annealing [6]. These are guaranteed to converge to the global optimum under some conditions, but are extremely slow to converge.

However, some efficient approximations based on combinatorial methods have been recently proposed. One such approximation involves viewing the image labeling problem as computing marginalizations in a probability distribution over a Markov random field (MRF). Inspired by the successes of MRF graphs in image analysis, and tractable approximation solutions to inferencing using belief propagation (BP) [10] [9], several other low level vision problems such as denoising, super-resolution, stereo etc., have

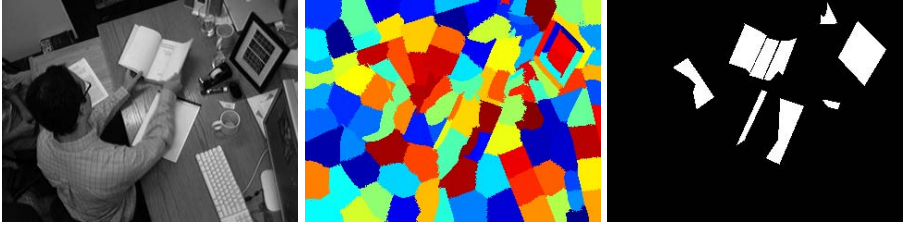


Fig. 1. On the left is a sample of the training data for an office scene with documents, the middle image is its superpixel representation (using normalized cuts, an over-segmentation of the original) and the right image is the manually labeled version of the original showing the documents as the foreground

been tackled by applying BP over Markov networks [5][15]. BP is an iterative sum-product algorithm, for computing marginals of functions on a graphical model. Despite recent advances, inference algorithms based on BP are still often too slow for practical use [4].

In this paper, we present an algorithmic technique that represents our image data with a Markov Random Field (MRF) graphical model defined on a more natural node structure, the superpixels. We infer the labels using belief propagation (BP) but get away from its drawbacks by substantially reducing the node structure of the graph. Thus, we reduce the combinatorial search space and improve the algorithmic running time while preserving the accuracy of the results.

Most stochastic models of images, are defined on the regular pixel grid, which is *not* a natural representation of visual scenes but rather an “artifact” of the image digitization process. We presume that it would be more natural, and more efficient, to work with perceptually meaningful entities called superpixels, obtained from a low-level grouping process [8], [16].

The organization of the paper is as follows: in section 2, we give a brief overview of BP irrespective of the graph structure and describe the process of inferring via message updates; in section 3, we describe our implementation of BP on an irregular graph structure and provide some justification to the use of superpixels; in section 4 we describe our experiments and provide quantitative and qualitative results and finally in section 5 we discuss some of the drawbacks of the technique, prescribe some means of improvement and discuss our plans for future work.

2 Background on Belief Propagation (BP)

A Markov network graphical model is especially useful for low and high level vision problems [5], [4] because the graphical model can explicitly express relationships between the nodes (pixels, patches, superpixels etc). We consider the undirected pairwise MRF graph $G = (V, E)$, V denoting the vertex (or node) set and E , the edge set. Each node $i \in V$ has a set of possible states C and also is affiliated with an observed state r_i . Given an observed state r_i , the goal is to infer some information about the states $c_i \in C$.

The edges in the graph indicate statistical dependencies. In our document-labeling problem, the hidden states C are (1) the documents and (2) the office background.

In general, MRF models of images are defined on the pixel lattice, although this restriction is not imposed by the definition of MRF. Pairwise-MRF models are well suited to our labeling problem because they define (1) the relationship between a node's states and its observed value and (2) the relationship within a clique (a set of pairwise adjacent nodes). We assume in here that the energies due to cliques greater than two are zero.

If these two relationships can be defined statistically as are probabilities, then we can define a joint probability function over the entire graph as:

$$p(c, r) = \frac{1}{Z} p(c_1, c_2 \cdots c_n, r_1, \cdots r_n) \tag{1}$$

$$= \frac{1}{Z} p(c_1) p(r_1|c_1) p(r_2|c_2) p(c_2|c_1) \cdots p(r_n|c_n) p(c_n|c_{n-1}) \tag{2}$$

Z is a normalization constant such that $\sum_{c_1 \in C_1, \dots, c_n \in C_n} p(c_1, c_2, \dots, c_n) = 1$. It is important to mention that the derivations in this section are done over a simplified graph (the chain) but the solutions generalize sufficiently to more complex graph structures.

If we let $\psi_{ab}(c_a, c_b) = p(c_a)p(c_b|c_a)$ and $\phi_a(c_a, r_a) = p(r_a|c_a)$ then the marginal probability at any of the nodes is given by:

$$p(c_i) = \frac{1}{Z} \sum_{c_1} \sum_{c_2} \cdots \sum_{c_{i-1}} \sum_{c_{i+1}} \cdots \sum_{c_n} \psi_{1,2}(c_1, c_2) \tag{3}$$

$$\psi_{2,3}(c_2, c_3) \cdots \psi_{n-1,n}(c_{n-1}, c_n)$$

$$\phi_1(c_1, r_1) \phi_2(c_2, r_2) \cdots \phi_n(c_n, r_n)$$

$$\text{Let } f(c_2) = \sum_{c_1} \psi_{1,2}(c_1, c_2)$$

$$f(c_3) = \sum_{c_2} \psi_{2,3}(c_2, c_3) f(c_2)$$

$$\vdots$$

$$f(c_n) = \sum_{c_n} \psi_{n-1,n}(c_{n-1}, c_n) f(c_{n-1})$$
(4)

The last line in equation (4) shows a recursive definition which we later take advantage of in our implementation. The equation shows how functions of probabilities are propagated from one node to the next. The ‘‘probabilities’’ are now converted to functionals (functions of the initial probabilities).

If we replace the functional $f(c_i)$ with the **message** property m_{ij} where i is the node to which the message are propagated and j is the node from which the message originates, then we can define our marginal probability at a node in terms of message updates. Also, if we replace our probability at a node $p(c_i)$ by the belief at the node $b(c_i)$ (since the computed values are no longer strictly probability distributions), then we can rewrite equation (4) as,

$$b_i(c_i) = \phi_i(c_i, r_i) \prod_{j \in \mathcal{N}(i)} m_{ij}(c_i) \tag{5}$$

where $\mathcal{N}(i)$ is the neighborhood of i . Equation (5) above shows how the derived functions of probabilities (or messages) are propagated along a simplified graph. Under the assumption that our solution so far generalizes to a more complex graph structure, we can now extend our derivation to the joint probability on an MRF graph given as:

$$p(c) = \frac{1}{Z} \prod_{i,j} \psi(c_i, c_j) \prod_k \phi(c_k, r_k) \tag{6}$$

The joint probability on the MRF graph is described in terms of two **compatibility functions**, (1) between the states and observed nodes and (2) between neighboring nodes.

We illustrate the message propagation process to node 2 in a five-node graph in figure(2). In this simple example, the belief at a node i can now be given as:

$$\begin{aligned} b_i(c_i) &= \sum_{c_j \in C_j, 1 \leq j \leq 5, j \neq i} p(c_1, \dots, c_5); \tag{7} \\ &= \phi_i(c_i, r_i) \prod_{j \in \mathcal{N}(i)} m_{ji}(c_i) \\ m_{ji}(c_i) &= \sum_{c_j \in C_j} \phi_j(c_j, r_j) \psi_{ji}(c_j, c_i) \prod_{k \in \mathcal{N}(j), j \neq i} m_{kj}(c_j) \end{aligned}$$

Unfortunately, the complexity of general belief propagation is exponential in the size of the largest clique. In many computer vision problems, belief propagation is prohibitively slow. The high-dimensional summation in equation (3) has a complexity of $\mathcal{O}(nM^k)$, where M is the number of possible labels for each variable, k is the maximum clique size in the graph and n is the number of nodes in the graph. By using the message updates, the complexity of the inference (for a non-loopy graph as derived above) is reduced to $\mathcal{O}(nkM^2)$. By extending this derivation to a more complex graph structure, the convergence property of the inference algorithm is removed and it is no longer guaranteed to converge. But in practice the algorithm consistently gives a good solution. Also, by significantly reducing n , we further reduce the algorithmic time.

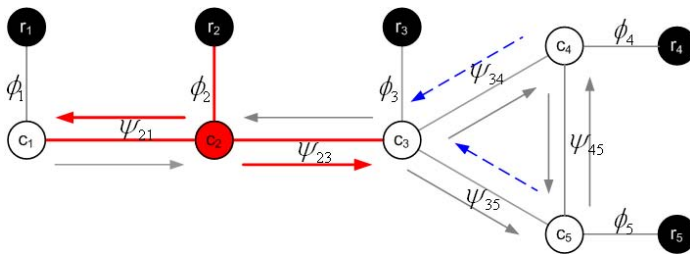


Fig. 2. An example of computing the messages for node 2, involving only $\phi_2, \psi_{2,1}, \psi_{2,3}$ and any messages to node 2 from its neighbors' neighbors

3 Labeling Images Using BP Inference

When doing labeling, the images are first abstracted into a superpixel representation (described in more detail in section (3.1)). The Markov network is defined on this irregular graph, and the compatibility functions are learned from labeled training data (section 3.2). The BP is used to infer the labels on the image.

3.1 The Superpixel Representation

First, we need to choose a representation for the image and scene variables. The image and scenes are arrays of single-valued superpixels. A superpixel is a homogenous segment obtained from a low-level grouping of the underlying pixels. Although irregular when compared to the pixel-grid, we choose the superpixel grid because we believe that it is representationally more efficient: i.e. pairwise constraints exist between entire segments, while they only exist for adjacent pixels on the pixel-grid. For a local model such as the MRF model, this property is very appealing in that we can model much longer-range interactions between superpixels segments. The use of superpixels is also computationally more efficient because it reduces the combinatorial complexity of images from hundreds of thousands of pixels to only a few hundred superpixels.

There are many different algorithms that generate superpixels including the segregated weighted algorithm (SWA) [12],[2], normalized cuts [13], constrained Delaunay triangulation [11] etc. It is very important to use near-complete superpixel maps to ensure that the original structures in the images are conserved. Therefore, we use the region segmentation algorithm normalized cuts [13], which was empirically validated and presented in [8]. Figure (3) shows an example of a natural image with its superpixel representation. For building superpixels using normalized cuts, the criterion for partitioning the graph are (1) to minimize the sum of weights of connections across the groups and (2) to maximize the sum of weights of connections within the groups. For completeness, we now give a brief overview of the normalized cuts process.

We begin by defining a similarity matrix as $S = [S_{ij}]$ over an image $I(i, j)$. A similarity matrix is a matrix of scores which express the similarity between any two points in an image. If we define a graph $G(V, E)$ where the node set is defined as the relationship ij between nodes, and all edges $e \in E$ have equal weight, we can define the degree of a node as $d_i = \sum_j S_{ij}$ and the volume of a set in the graph as $vol(A) = \sum_{i \in A} d_i, A \subseteq V$. The cuts in the graph are therefore $cut(A, \bar{A}) = \sum_{i \in A, j \in \bar{A}} S_{ij}$. Given these definitions, normalized cuts are described as the solution to:

$$N_{cut} = cut(A, B) \left(\frac{1}{vol(A)} + \frac{1}{vol(B)} \right) \quad (8)$$

Our implementation of superpixel generation implements an approximate solution using spectral clustering methods. The resulting superpixel representation is an over-segmentation of the original image. We define the Markov network over this representation.

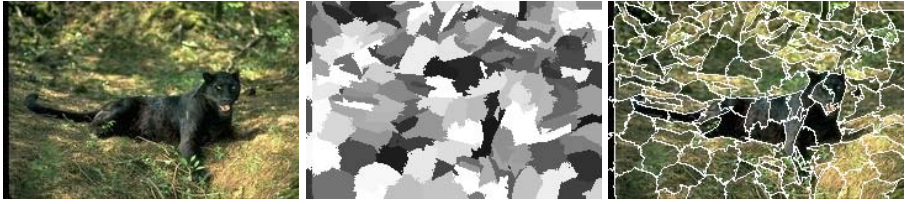


Fig. 3. On the left is an example of a natural image, the middle image is the superpixel representation (using normalized cuts, an over-segmentation of the original) and the right image the superimposition of the two

3.2 Learning the Compatibility Functions

The model we assume for the document labeling images is generated from the training images. The joint probability distribution is modeled using the image data and their corresponding labels. The joint probability is expressed in terms of the two compatibility functions defined in equation(6).

If we let r_i represent a superpixel segment in the training image and c_i , its corresponding label. The first compatibility function $\phi(c_i, r_i)$ can be computed by learning a mixture of Gaussian models. The resulting Gaussian distributions will represent either the document class or the background class. So although we have two real-life classes (document and background), the number of states to be input to the BP problem will have increased based on the output of the Gaussian Mixture Model (GMM), i.e. each component of the GMM represents a distinct label.

The second compatibility function $\psi(c_i, c_j)$ relates the superpixels to each other. We use the simplest interacting Pott's model where the function takes one of 2 values $-1, +1$ and the interactions exists only for amongst neighbors with the same labels. Our compatibility function between superpixels is therefore given as:

$$\psi(c_i, c_j) := \begin{cases} +1 & \text{if } c_i \text{ and } c_j \text{ have the same initial label values,} \\ -1 & \text{otherwise} \end{cases} \quad (9)$$

So given a new image of a cluttered room, we can extract the documents in the image by using the steps given in section (3.3). The distribution of the superpixels $r_i \in R$ given the latent variables $c_i \in C$ can therefore be modeled graphically as:

$$P(R, C) \propto \prod_i \phi(r_i, c_i) \prod_{(c_j, c_k)} \psi(c_j, c_k) \quad (10)$$

Equation (10) can also be viewed as the pairwise-MRF graphical representation of our labeling problem, which can be solved using BP with the two parts of equation (7).

3.3 Putting All Together...

The general strategy for designing the label system can therefore be described as:

1. Use the training data to learn the latent parameters of the system. The number of resulting latent parameter sets will give the number of states required for the inference problem.

2. Using the number of states obtained in the previous steps, design compatibility functions such that eventually, only a single state can be allocated to each superpixel.
3. For the latent variable c_i associated with every superpixel i , use the BP algorithm to choose its best state.
4. If the state values correspond to labeling classes (as in the case of our document labeling system), the selected state variables are converted to their associated class labels

4 Experiments, Results and Discussion

The first round of experiments consisted of testing the BP labeling algorithm on synthetically generated image data, whose values were samples drawn from a known distribution. We first generated synthetic scenes by drawing samples from Gaussian distribution functions, and then added noise to the resulting images. These two datasets (clean and noisy images) represented our observations in a controlled setting. To add $X\%$ noise, we randomly selected unique $X\%$ of the pixels in the original image and the pixel values were replaced by a random number between 0 and 255;

The scene (or hidden parameters) were represented by the parameters of our generating distributions. We modeled the relationships between the scenes and observations with a pairwise-Markov network and used belief propagation (BP) to find the local maximum of the posterior probability for the original scenes.

Figure (4) shows the results obtained from running the process on the synthetic data. We also present a graph showing the sum-of squared-differences (SSD) between the ground-truth data and varying levels of noise in figure (5).

We then extended this learning based scene-recovery approach to finding documents in a cluttered room (under moderately different lighting conditions). Documents were labeled in images of a cluttered room and used in training to obtain the prior and conditional probability density functions. The labeled images were treated as scenes and the goal was to infer these scenes given a new image from the same imaging domain (pictures of offices) but not from the training set. For inferring scenes from given observations, the computed distributions were used as compatibility functions in the BP message update process.

We learned our distributions from the training data using an EM-GMM algorithm (Expectation Maximization for Gaussian Mixture Models) on 50 office images. The training images consisted of images with different documents in a cluttered background, all taken in one office. The document data was modeled with a mixture of three Gaussian distributions while the background data was modeled with two Gaussian distributions. The resulting parameters (mean μ_i , variance σ_i and prior probability p_i) from training are:

- class 1: $\mu_1 = 6.01$; $\sigma_1 = 9.93$; $p_1 = 0.1689$
- class 2: $\mu_2 = 86.19$; $\sigma_2 = 147.79$; $p_2 = 0.8311$
- class 3: $\mu_3 = 119.44$; $\sigma_3 = 2510.6$; $p_3 = 0.5105$
- class 4: $\mu_4 = 212.05$; $\sigma_4 = 488.21$; $p_4 = 0.2203$
- class 5: $\mu_5 = 190.98$; $\sigma_5 = 2017$; $p_5 = 0.2693$

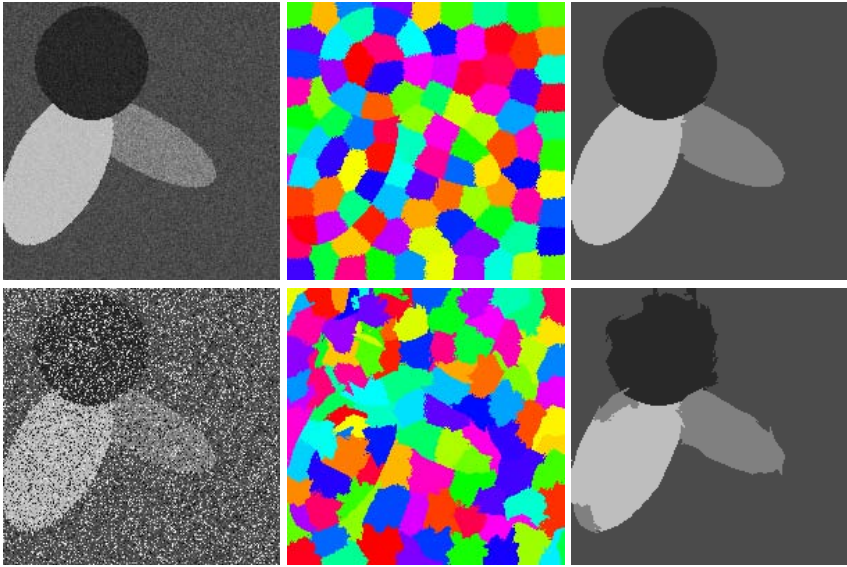


Fig. 4. Top row: the left column shows an original synthetic image created from samples from Gaussian distributions, the middle column is its near-correct superpixel representation and the right column shows the resulting labeled image. Bottom row: the left column shows a noisy version of the synthetic image, the middle column is its superpixel representation and the right column also shows the resulting labeled image.

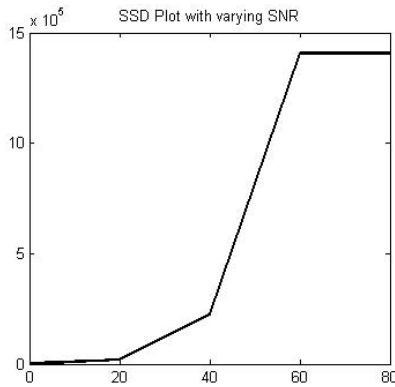


Fig. 5. Quantitative display of how the error increases exponentially with increasing noise

The resulting five classes were then treated as the states that any variable $c_i \in C$ in the MRF graph could take. Classes 1 and 2 correspond to the background while classes 3,4 and 5 are the documents. Unfortunately, because the models were trained separately, the prior probabilities are not reflective of the occurrences in the entire data, only in the background/document data alone.

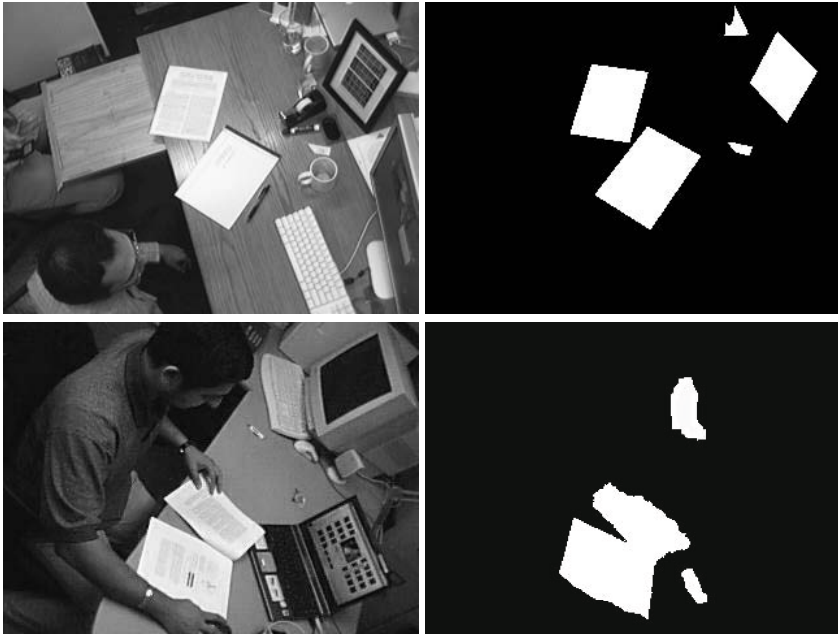


Fig. 6. The top row shows sample of training data and the corresponding labeled image; the bottom row shows a testing image (taken at a different time, in a different office. The output of the detection is shown.

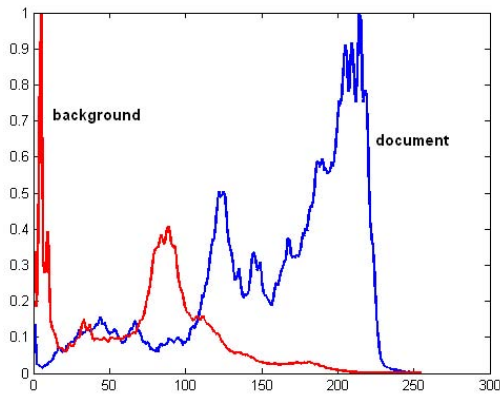


Fig. 7. The distributions of the foreground document and cluttered background classes

For testing, a new image was taken in a completely different office under moderately different lighting conditions and the class labels were extracted. The pictorial results of labeling the real-life rooms with documents are shown in figure (6).

We observed that even with applying relatively weak models (the synthetic images are no longer strongly coupled to the generating distribution), we were able to successfully

recover a labeled image for both synthetic and real-life data. A related drawback we faced was the simplicity of our representation. We used grayscale values as the only statistic in our document-finding system and this (as seen in figure (6)b), introduces artifacts into our final labeling solution. Superpixels whose intensity values are close to those of the trained documents can be mis-labeled.

Also, we observed that the use of superpixels reduced the number of nodes significantly, thus reducing the computational time. Also, the segmentation results of our low noise synthetic images and the real-life data were promising with superpixels. A drawback though is the limitation imposed by the superpixel representation. Although we used a well tested and efficient superpixel implementation, we found that as the noise levels increased in the images, the superpixels became more inaccurate and the errors obtained in the representation were propagated into the system. Also, due to the loops in the graph, it does not converge if run long enough, but we can still sufficiently recover the true solution from the graphical structure.

5 Conclusion

In this paper, we have proposed a way of labeling irregular graphs generated by an oversegmentation of an image, using BP inferences on MRF graphs. Because a common limitation of graph models in low level image processing is often due to intractable node size on the graph, we have reduced the computational intensity of the graph model by introducing the use of superpixels, without any loss of generality on the definition of the graph. We reduced the number of node variables from orders of tens of thousands of pixels to about a hundred superpixels.

Furthermore, we define compatibility functions for inference based on learning the statistical distributions of the real-life data.

In the future, we intend to base our statistics on more definitive features of the images (other than simply grayscale values) to model the real-life document and background data.. These could include textures at different scales, and other scale-invariant measurements. We also plan to investigate the use of stronger inference methods by relaxing the assumption that the cliques in our MRF graphs are only of size 2.

References

1. Cao, H., Govindaraju, V.: Handwritten carbon form preprocessing based on markov random field. In: Proc. IEEE Conf. Comput. Vision And Pattern Recogniton (2007)
2. Corso, J.J., Sharon, E., Yuille, A.L.: Multilevel Segmentation and Integrated Bayesian Model Classification with an Application to Brain Tumor Segmentation. In: Larsen, R., Nielsen, M., Sporring, J. (eds.) MICCAI 2006, Part II. LNCS, vol. 4191, pp. 790–798. Springer, Heidelberg (2006)
3. Duncan, R., Qian, J., Zhu, B.: Polynomial time algorithms for three-label point labeling. In: Wang, J. (ed.) COCOON 2001. LNCS, vol. 2108, p. 191. Springer, Heidelberg (2001)
4. Felzenszwalb, P., Huttenlocher, D.: Efficient belief propagation for early vision. In: Proc. IEEE Conf. Comput. Vision And Pattern Recogn. (2004)
5. Freeman, W.T., Pasztor, E.C., Carmichael, O.T.: Learning low-level vision. International Journal of Computer Vision 40(1), 25–47 (2000)

6. Geman, S., Geman, D.: Stochastic relaxation, gibbs distributions, and the bayesian restoration of images. *IEEE Trans. on Pattern Analysis and Machine Intelligence* 6, 721–741 (1984)
7. Luo, B., Hancock, E.R.: Structural graph matching using the em algorithm and singular value decomposition. *IEEE Trans. Pattern Anal. Mach. Intell.* 23(10), 1120–1136 (2001)
8. Mori, G., Ren, X., Efros, A.A., Malik, J.: Recovering human body configurations: Combining segmentation and recognition. In: *Proc. IEEE Conf. Comput. Vision And Pattern Recogn.*, vol. 2, pp. 326–333 (2004)
9. Murphy, K.P., Weiss, Y., Jordan, M.I.: Loopy belief propagation for approximate inference: An empirical study. In: *Proceedings of Uncertainty in AI*, pp. 467–475 (1999)
10. Pearl, J.: *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann, San Francisco (1988)
11. Ren, X., Fowlkes, C.C., Malik, J.: Scale-invariant contour completion using conditional random fields. In: *Proc. 10th Int'l. Conf. Computer Vision*, vol. 2, pp. 1214–1221 (2005)
12. Sharon, E., Brandt, A., Basri, R.: Segmentation and boundary detection using multiscale intensity measurements. In: *Proc. IEEE Conf. Comput. Vision And Pattern Recogn.*, vol. 1, pp. 469–476 (2001)
13. Shi, J., Malik, J.: Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 22(8), 888–905 (2000)
14. Szeliski, R., Zabih, R., Scharstein, D., Veksler, O., Kolmogorov, V., Agarwala, A., Tappen, M.F., Rother, C.: A comparative study of energy minimization methods for markov random fields. In: Leonardis, A., Bischof, H., Pinz, A. (eds.) *ECCV 2006*. LNCS, vol. 3952, pp. 16–29. Springer, Heidelberg (2006)
15. Tappen, M.F., Russell, B.C., Freeman, W.T.: Efficient graphical models for processing images. In: *Proc. IEEE Conf. Comput. Vision And Pattern Recogniton*, pp. 673–680 (2004)
16. Yu, S., Shi, J.: Segmentation with pairwise attraction and repulsion. In: *Proceedings of the 8th IEEE International Conference on Computer Vision (ICCV 2001)* (July 2001)