Finding Deformable Shapes using Loopy Belief Propagation

James M. Coughlan[1] and Sabino J. Ferreira[2]

[1] Smith-Kettlewell Institute, 2318 Fillmore St., San Francisco, CA 94115, USA,
coughlan@ski.org
[2] Department of Statistics, Federal University of Minas Gerais UFMG, Av. Antonio Carlos 6627, Belo Horizonte MG, 30.123-970 Brasil,
sabino@est.ufmg.br

Abstract. A novel deformable template is presented which detects and localizes shapes in grayscale images. The template is formulated as a Bayesian graphical model of a two-dimensional shape contour, and it is matched to the image using a variant of the belief propagation (BP) algorithm used for inference on graphical models. The algorithm can localize a target shape contour in a cluttered image and can accommodate arbitrary global translation and rotation of the target as well as significant shape deformations, without requiring the template to be initialized in any special way (e.g. near the target).

The use of BP removes a serious restriction imposed in related earlier work, in which the matching is performed by dynamic programming and thus requires the graphical model to be tree-shaped (i.e. without loops). Although BP is not guaranteed to converge when applied to inference on non-tree-shaped graphs, we find empirically that it does converge even for deformable template models with one or more loops. To speed up the BP algorithm, we augment it by a pruning procedure and a novel technique, inspired by the 20 Questions (divide-and-conquer) search strategy, called "focused message updating." These modifications boost the speed of convergence by over an order of magnitude, resulting in an algorithm that detects and localizes shapes in grayscale images in as little as several seconds on an 850 MHz AMD processor.

1 Introduction

A promising approach to the detection and recognition of flexible objects involves representing them by deformable template models, for example, [7, 15, 17]. These models specify the shape and intensity properties of the objects. They are defined probabilistically so as to take into account the variability of the shapes and their intensity properties.

The flexibility of such models means that we have a formidable computational problem to determine if the object is present in the image and to find where it is located. In simple images, standard edge detection techniques may be sufficient to segment the objects from the background, though we are still faced with the difficult task of determining how edge segments should be grouped to form an object. In more realistic images, however, large segments of object boundaries will *not* be detected by standard edge detectors. It often seems impossible to do segmentation without using high level models like deformable templates [7]; an important challenge that remains is to find efficient algorithms to match deformable templates to images.

We have devised a Bayesian deformable template for finding shapes in cluttered grayscale images, and we propose an efficient procedure for matching the template to an image based on the belief propagation (BP) algorithm [11] used for inference on graphical models. The template is formulated as a Bayesian graphical model of a two-dimensional shape contour that is invariant to global translations and rotations, and which is designed to accommodate significant shape deformations.

BP can be applied directly to our deformable template model, yielding an iterative procedure for matching each part of the deformable template to a location in the image. Like the dynamic programming algorithm used in earlier related work [3], BP is guaranteed to converge to the optimal solution if the graphical model is tree-shaped (i.e. without loops). However, an important advantage of BP over dynamic programming is that it may also be applied to graphical models with one or more loops, which arise naturally in deformable template models. Although convergence is no longer guaranteed when loops are present, researchers have found empirically that BP does converge for a variety of graphical models with loops [10], and our experiments with deformable templates corroborate these findings.

Although BP can be applied straightforwardly to the deformable template model, an important additional contribution of our work is to augment BP by two procedures which speed it up by over an order of magnitude. The first procedure, called belief pruning, removes matching hypotheses that are deemed extremely unlikely from further consideration by BP. (This is very similar to the "beam search" technique used to prune states in hidden Markov models (HMM's) in speech recognition [9].) The speed-up of belief pruning is enhanced greatly by the second procedure, called "focused message updating." This procedure, inspired by the 20 questions (divide-and-conquer) search strategy [6], uses a carefully chosen sequence of "questions" to guide the operation of BP. The first stage of BP is devoted to matching a "key feature" of the template – corners or T-junctions which are relatively rare in the background clutter. A second key feature is chosen for BP to process next, in such a way that the *conjunction* of the two features is expected to be an even rarer occurrence in the background. By the time BP has processed these first two key features, substantial pruning has occurred – even though BP may still be far from convergence – which significantly speeds up the subsequent iterations of BP.

The result is a deformable template algorithm that detects and localizes shapes in grayscale images in as little as several seconds on an 850 MHz AMD processor. We demonstrate our algorithm on three deformable shape models – the letter "A," a stick-figure shape, and an open hand contour – for a variety of images, showing the ability of our algorithm to cope with a wide range of shape deformations, extensive background clutter, and low contrast between target and background.

Our work and related work on algorithms for deformable template matching [21], which emphasize the use of a generative Bayesian model that uses separate models to account for shape variability and for variations in appearance, may be compared with other recent work on object detection and shape matching. A variety of object detection techniques have been developed which find instances of deformable shapes by applying sophisticated tests to decide whether each local region in an image contains a target or belongs to the background. In [5, 13] these tests are based on a strategy of posing a series of questions designed to reject background regions as quickly as possible (similar in spirit to the 20 Questions strategy), while [12] relies on powerful statistical likelihood models of the appearance of target and background in order to discriminate between them. Although these object detection algorithms are fast and effective, they lack the explicit models of shape and appearance variability used in Bayesian deformable template models, and it seems difficult to extend these algorithms to highly articulated shapes. Finally, we cite recent work on point set matching [2] and matching using shape context [1], which are very effective matching algorithms for use with point sample targets. These algorithms are designed to perform highly robust matching in limited clutter, while our algorithm is intended to address the problem of visual search in more highly cluttered grayscale images.

2 Deformable Template

Our deformable template is designed to detect and localize a given shape amidst clutter in a grayscale image. The template is defined as a Bayesian graphical model consisting of a shape prior and an imaging model, and an inference algorithm based on BP is used to find the best match between the template and the image. More specifically, the shape prior is a graphical model that describes probabilistically what configurations (shapes) the template is likely to assume. The imaging model describes probabilistically how any particular configuration will appear in a grayscale image. Given an image, the Bayesian model assigns a posterior probability to each possible shape configuration. A BP-based algorithm is used to find the most likely configuration that optimally fits the image data.

2.1 The Shape Prior

The variability of the template shape is modelled by the shape prior, which assigns a probability to each possible deformation of the shape. The shape is represented by a set of points $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N$ in the plane which trace the contours of the shape, and by an associated chain $\theta_1, \theta_2, \cdots, \theta_N$ of normal orientations which describe the direction of outward-pointing normal vectors at the points (see Figure (1)). Each point $\mathbf{x}_i$ has two components x_i and y_i. For brevity we define $\mathbf{q}_i = (\mathbf{x}_i, \theta_i)$, which we also refer to as "node" i. The *configuration* $\mathbf{Q}$, defined as $\mathbf{Q} = (\mathbf{q}_1, \mathbf{q}_2, \cdots, \mathbf{q}_N)$, completely defines the shape.

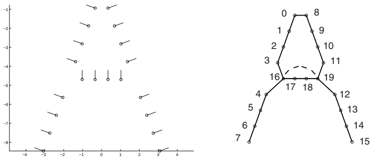

Fig. 1. Left, "letter A" template reference shape, with points drawn as circles and line segments indicating normal directions. Right, the associated connectivity graph of the template, with lines joining interacting points (the dashed line denotes a long-distance connection) and numbers labeling the nodes. Note that the connectivity graph has loops.

The shape prior is defined relative to a *reference shape* so as to assign high probability to configurations $\mathbf{Q}$ which are similar to the reference configuration $\tilde{\mathbf{Q}} = (\tilde{\mathbf{q}}_1, \tilde{\mathbf{q}}_2, \cdots, \tilde{\mathbf{q}}_N)$ and low probability to configurations that are not. This is achieved using a graphical model (Markov random field) which penalizes the amount of deviation in shape between $\mathbf{Q}$ and $\tilde{\mathbf{Q}}$ in a way that is invariant to global rotation and translation. (The scale of the shape prior is fixed and we assume knowledge of this scale when we execute our algorithm.)

Deviations in shape are measured by the geometric relationship of pairs of points $\mathbf{q}_i$ and $\mathbf{q}_j$ on the template, and are expressed in terms of *interaction energies* $U_{ij}(\mathbf{q}_i, \mathbf{q}_j)$. Low interaction energies occur for highly probable shape configurations, for which the geometric relationships of pairs of points tend to be faithful to the reference shape $\tilde{\mathbf{Q}}$, and high interaction energies are obtained for improbable configurations (the precise connection to probabilities is formulated in Equation (2)). Two kinds of shape similarities are used to calculate $U_{ij}(\mathbf{q}_i, \mathbf{q}_j)$. First, the relative orientation of θ_i and θ_{i+1} should be similar to that of $\tilde{\theta}_i$ and $\tilde{\theta}_{i+1}$, meaning that we typically expect $\theta_j - \theta_i \approx \tilde{\theta}_j - \tilde{\theta}_i$. This motivates the inclusion in the interaction energy $U_{ij}(\mathbf{q}_i, \mathbf{q}_j)$ of the following term:

$$U_{ij}^C(\mathbf{q}_i, \mathbf{q}_j) = \sin^2\left(\frac{\theta_j - \theta_i - C_{ij}}{2}\right)$$

where $C_{ij} = \tilde{\theta}_j - \tilde{\theta}_i$. This energy attains a minimum when $\theta_j - \theta_i = C_{ij}$ (and a maximum when $\theta_j - \theta_i = C_{ij} + \pi$).

Second, we note that the location of point $\mathbf{x}_j$ relative to $\mathbf{q}_i$ is also invariant to global translation and rotation. As a result, the location $\mathbf{x}_j$ *relative* to $\mathbf{q}_i$ should typically be similar to the location $\tilde{\mathbf{x}}_j$ *relative* to $\tilde{\mathbf{q}}_i$ (and in fact if $\mathbf{q}_i$ is known then it is possible to predict the approximate location of $\mathbf{x}_j$). In other words, $\tilde{\mathbf{x}}_i$ and $\tilde{\theta}_i$ define a local coordinate system, and the coordinates of $\tilde{\mathbf{x}}_j$ in that coordinate system are invariant to global translation and rotation. If we define the unit normal vectors $\mathbf{n}_i = (\cos\theta_i, \sin\theta_i)$ and $\tilde{\mathbf{n}}_i = (\cos\tilde{\theta}_i, \sin\tilde{\theta}_i)$ and vectors perpendicular to them $\mathbf{n_i}^{\perp} = (-\sin\theta_{\mathbf{i}}, \cos\theta_{\mathbf{i}})$ and $\tilde{\mathbf{n}}_i^{\perp} = (-\sin\theta_i, \cos\theta_i)$, then the dot product of $\mathbf{x}_j - \mathbf{x}_i$ with $\mathbf{n_i}$ and $\mathbf{n_i}^{\perp}$ should have values similar to the corresponding values for the reference shape: $(\mathbf{x}_j - \mathbf{x}_i) \cdot \mathbf{n}_i \approx (\tilde{\mathbf{x}}_j - \tilde{\mathbf{x}}_i) \cdot \tilde{\mathbf{n}}_i$ and $(\mathbf{x}_j - \mathbf{x}_i) \cdot \mathbf{n}_i^{\perp} \approx (\tilde{\mathbf{x}}_j - \tilde{\mathbf{x}}_i) \cdot \tilde{\mathbf{n}}_i^{\perp}$. Now we can define the remaining two terms in $U_{ij}(\mathbf{q}_i, \mathbf{q}_j)$, the energies

$$U_{ij}^{A}(\mathbf{q}_i, \mathbf{q}_j) = [(\mathbf{x}_j - \mathbf{x}_i) \cdot \mathbf{n}_i - A_{ij}]^2$$

and

$$U_{ij}^{B}(\mathbf{q}_i, \mathbf{q}_j) = [(\mathbf{x}_j - \mathbf{x}_i) \cdot \mathbf{n}_i^{\perp} - B_{ij}]^2$$

where $A_{ij} = (\tilde{\mathbf{x}}_j - \tilde{\mathbf{x}}_i) \cdot \tilde{\mathbf{n}}_i$ and $B_{ij} = (\tilde{\mathbf{x}}_j - \tilde{\mathbf{x}}_i) \cdot \tilde{\mathbf{n}}_i^{\perp}$. The full interaction energy is then given as:

$$U_{ij}(\mathbf{q}_i, \mathbf{q}_j) = \frac{1}{2}\{K_{ij}^{A} U_{ij}^{A}(\mathbf{q}_i, \mathbf{q}_j) + K_{ij}^{B} U_{ij}^{B}(\mathbf{q}_i, \mathbf{q}_j) + K_{ij}^{C} U_{ij}^{C}(\mathbf{q}_i, \mathbf{q}_j)\} \qquad (1)$$

where the non-negative coefficients K_{ij}^{A}, K_{ij}^{B} and K_{ij}^{C} define the strengths of the interactions and are set to 0 for those pairs i and j with no direct interactions (the majority of pairs). Higher values of K_{ij}^{A}, K_{ij}^{B} and K_{ij}^{C} produce a stiffer (less deformable) template.

Noting that in general $U_{ij}(\mathbf{q}_i, \mathbf{q}_j) \neq U_{ji}(\mathbf{q}_j, \mathbf{q}_i)$, we symmetrize the interaction energy as follows: $U_{ij}^{sym}(\mathbf{q}_i, \mathbf{q}_j) = U_{ij}(\mathbf{q}_i, \mathbf{q}_j) + U_{ji}(\mathbf{q}_j, \mathbf{q}_i)$. We use the symmetrized energy to define the shape prior:

$$P(\mathbf{Q}) = \frac{1}{Z} \prod_{i<j} e^{-U_{ij}^{sym}(\mathbf{q}_i, \mathbf{q}_j)} \qquad (2)$$

where Z is a normalization constant and the product is over all pairs i and j, with the restriction $i < j$ to eliminate double-counting and self-interactions. Note that the prior is a Markov random field or graphical model that has pairwise connections between all pairs i and j whose coefficients are non-zero. The values of the coefficients were chosen experimentally by stochastically sampling the prior using a Metropolis MCMC sampler, i.e. generating samples from the prior distribution to illustrate what shapes have high probability (see Figures (2,4) for examples). Learning techniques such as maximum likelihood estimation could be employed to determine more accurate values.

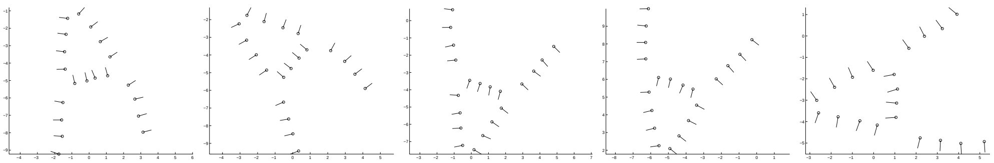

Fig. 2. Stochastic samples from shape prior.

2.2 The Imaging Model

The imaging (likelihood) model explains what image data may be expected given a specific configuration. Two forms of image data are derived from the raw grayscale image $I(\mathbf{x})$: an edge map $I_e(\mathbf{x})$ to provide local evidence for the presence or absence of an object boundary at each pixel, and an orientation map $\phi(\mathbf{x})$ to provide estimates of the orientation of edges throughout the image. The filter that defines $I_e(\mathbf{x})$ was chosen appropriate to the problem domain, but in this section we will assume it is the magnitude of the image gradient: $I_e(\mathbf{x}) = |\mathbf{G} \star \nabla \mathbf{I}(\mathbf{x})|$ where $G(.)$ is a smoothing Gaussian. The orientation map $\phi(\mathbf{x})$ is calculated as $\arctan(g_y/g_x)$ where $(g_x, g_y) = G \star \nabla I(\mathbf{x})$. (Both filters can be evaluated very rapidly on an entire image.)

First consider the behavior of the edge map. The likelihood model quantifies the tendency for edge strength values to be high *on* edge and low *off* edge. Following the work of Geman and Jedynak [6] and Konishi *et al.* [8], we define two conditional distributions of edge strength that are empirically obtained: $P_{on}(I_e(\mathbf{x})) = P(I_e(\mathbf{x})|\mathbf{x}\, ON\, edge)$ and $P_{off}(I_e(\mathbf{x})) = P(I_e(\mathbf{x})|\mathbf{x}\, OFF\, edge)$. See Figure (3) for samples of P_{on} and P_{off} in a particular domain. Note that P_{off} is a simple model of the background which expresses the fact that edge pixels are rare off the target shape. (A more realistic background model, which would capture the fact that background clutter is structured, seems unnecessary for our application.)

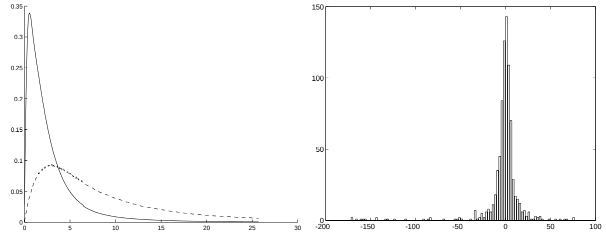

Fig. 3. Empirical distributions that make up the imaging model. Left panel shows $P_{on}(I_e) = P(I_e|\, ON\, edge)$, dashed line, and $P_{off}(I_e) = P(I_e|\, OFF\, edge)$, solid line, as a function of $I_e(\mathbf{x}) = |G\star\nabla I(\mathbf{x})|$. Note that $P_{on}(.)$ peaks at a higher edge strength than does $P_{off}(.)$, since true edge boundaries tend to have higher image gradient magnitude. Right panel shows $P_{ang}(\phi|\theta)$, the distribution of measured edge direction ϕ (obtained from the $G \star \nabla I(\mathbf{x})$ direction) given the true edge normal direction θ, plotted as a function of the angular error $\phi - \theta$ (in degrees). Note the sharp fall-off about 0 (the plot is periodic with period 180°), showing that the angular error tends to be small.

Next we turn to the orientation map. Since $\phi(\mathbf{x})$ estimates the angle that ∇I makes with the x-axis, we can expect that on a true object boundary the value of $\phi(\mathbf{x})$ should point roughly *perpendicular* to the tangent of the boundary. (For instance, if the intensity is uniformly high on one side of a straight boundary and uniformly low on the other, then $\phi(\mathbf{x})$ should describe the vector perpendicular to the boundary that points from the darker to the lighter region.) Denoting the true normal orientation of the boundary as θ, we expect that $\phi(\mathbf{x})$ is approximately equal to either θ or $\theta + \pi$. This relationship between θ and $\phi(\mathbf{x})$ may also be quantified as a conditional distribution $P_{ang}(\phi|\theta)$, which was measured in previous work [3], see Figure (3). If $\mathbf{x}$ is not on an edge then we may assume that the distribution of $\phi(\mathbf{x})$ is uniform in all directions: $U(\phi) = 1/2\pi$.

The imaging model is a distribution of the edge map and image orientation data across the entire image, conditioned on the template shape configuration $\mathbf{Q}$. Defining $\mathbf{d}(\mathbf{x}) = (I_e(\mathbf{x}), \phi(\mathbf{x}))$ and letting $\mathbf{D}$ denote the values of $\mathbf{d}(\mathbf{x})$ across the entire image, we can express the likelihood model as a distribution that factors over every pixel:

$$P(\mathbf{D}|\mathbf{Q}) = \prod_{\substack{all\ pixels \\ \mathbf{x}}} P(\mathbf{d}(\mathbf{x})|\mathbf{q}_1 \cdots \mathbf{q}_N) \tag{3}$$

where $P(\mathbf{d}(\mathbf{x})|\mathbf{q}_1 \cdots \mathbf{q}_N)$ is set to $P_{on}(I_e(\mathbf{x}))P_{ang}(\phi(\mathbf{x}) - \theta_i)$ if there exists an i such that $\mathbf{x} = \mathbf{x_i}$ – i.e. if $\mathbf{x}$ lies somewhere on the configuration shape – or to $P_{off}(I_e(\mathbf{x}))U(\phi(\mathbf{x}))$ otherwise. In other words, pixels *on* the target contour have edge strengths distributed according to $P_{on}(I_e)$ and orientations drawn from $P_{ang}(\phi|\theta)$; pixels *off* the target contour have edge strengths distributed according to $P_{off}(I_e)$ and orientations drawn from $U(\phi)$.

2.3 The Posterior Distribution

The shape configuration $\mathbf{Q}$ is determined by the posterior distribution $P(\mathbf{Q}|\mathbf{D}) = P(\mathbf{Q})P(\mathbf{D}|\mathbf{Q})/P(\mathbf{D})$. As an approximation we can assume that no points in the template $\mathbf{q}_i$ and $\mathbf{q}_j$ map to the same point $\mathbf{x}$ in the image (a reasonable approximation since the prior would make such a deformation unlikely), so we can re-express the likelihood function in a more convenient form:

$$P(\mathbf{D}|\mathbf{Q}) = \prod_{i=1}^{N} \frac{P_{on}(I_e(\mathbf{x}_i))P_{ang}(\phi(\mathbf{x}_i) - \theta_i)}{P_{off}(I_e(\mathbf{x}_i))U(\phi(\mathbf{x}_i))} F(\mathbf{D}) \tag{4}$$

where $F(\mathbf{D}) = \prod_{all\ pixels\ \mathbf{x}} P_{off}(I_e(\mathbf{x}))U(\phi(\mathbf{x}))$ is a function *independent* of $\mathbf{Q}$. Thus the posterior can be written as:

$$P(\mathbf{Q}|\mathbf{D}) \propto P(\mathbf{Q}) \prod_{i=1}^{N} \frac{P_{on}(I_e(\mathbf{x}_i))}{P_{off}(I_e(\mathbf{x}_i))} \frac{P_{ang}(\phi(\mathbf{x}_i) - \theta_i)}{U(\phi(\mathbf{x}_i))} \tag{5}$$

Note that the ratio $P_{on}(I_e(\mathbf{x}_i))/P_{off}(I_e(\mathbf{x}_i))$ will be high for pixels with high image gradients, and that $P_{ang}(\phi(\mathbf{x}_i) - \theta_i)/U(\phi(\mathbf{x}_i))$ will be high where the image gradient is approximately aligned parallel with (or 180° opposite to) the direction specified by θ_i.

Finally, we can express the posterior as a Markov random field with pairwise interactions (i.e. a graphical model):

$$P(\mathbf{q}_1, \cdots, \mathbf{q}_N|\mathbf{D}) = \frac{1}{Z} \prod_i \psi_i(\mathbf{q}_i) \prod_{i<j} \psi_{ij}(\mathbf{q}_i, \mathbf{q}_j) \quad (6)$$

where $\psi_i(\mathbf{q}_i)$ is the local evidence for $\mathbf{q}_i$ (from the likelihood function, Equation (4)) and $\psi_{ij}(\mathbf{q}_i, \mathbf{q}_j)$ is the compatibility (or binary potential) between $\mathbf{q}_i$ and $\mathbf{q}_j$ (from the shape prior, Equation (2)).

3 Inference by Efficient Belief Propagation

The posterior distribution expresses our knowledge of what shape configurations are most likely given the image data. One way to estimate the most likely shape configuration is to calculate the MAP (maximum a posterior) estimate, i.e. $\mathbf{Q}^* = \arg\max_{\mathbf{Q}} P(\mathbf{Q}|\mathbf{D})$. However, we find empirically in our application that the MAP marginal estimate suffices, which is the MAP estimate applied separately to each variable $\mathbf{q}_i$:

$$\mathbf{q}_i^* = \arg\max_{\mathbf{q}_i} P(\mathbf{q}_i|\mathbf{D}) \quad (7)$$

for all i, where $P(\mathbf{q}_i|\mathbf{D}) = \int d\mathbf{q}_1 d\mathbf{q}_2 \ldots d\mathbf{q}_{i-1} d\mathbf{q}_{i+1} \ldots d\mathbf{q}_N P(\mathbf{Q}|\mathbf{D})$ is the marginal posterior distribution of $\mathbf{q}_i$. (If the posterior is sufficiently peaked about its mode then the MAP marginal estimate will approach the MAP estimate, and we speculate that this is the case for our application.) We can estimate the MAP marginal using the BP (or alternate algorithms such as CCCP [20]). (An alternate version of BP, the max-product belief propagation algorithm, may also be used to estimate the MAP directly.) We note that the MAP marginal estimate may not suffice for multiple targets (which produce multiple modes in the posterior), but techniques similar to those used to find multiple targets in [3] may be adapted to solve this problem in future research.

If the connectivity of the Markov random field contains no loops (cycles) – i.e. the associated connectivity graph forms a tree – then algorithms such as BP are guaranteed [11] to find the exact marginals of the posterior in an amount of time linear in N. These algorithms are not guaranteed to find a correct solution when the connections have loops, which will naturally arise for the shapes we will consider in this paper. However, recent work shows that, empirically, BP works successfully in certain domains even when loops are present [10, 16], and our experimental results corroborate this result.

In this section we describe how the BP algorithm is used to compute the MAP marginals of the posterior distribution. We note that BP requires the use of discrete variables, so it is first necessary to quantize the configuration variable $\mathbf{Q}$ by restricting it to assume only a discrete set of values (referred to as the "state space"). In principle this quantization can be made as fine as necessary to approximate the original posterior arbitrarily well, but for computational reasons we will use as coarse a quantization as possible.

We will require that $q_i \in S$, where S is a finite, discrete state space, for all $i = 1, \ldots, N$. In general we will define S according to an *adaptive quantization* scheme to be the set of all triples (x, y, θ) such that (x, y) are the coordinates of pixels on the image lattice (or only those pixels having sufficiently high edge values), and θ belongs to a sampling of orientations in the range $[0, 2\pi)$. See Section (4) for details.

3.1 The Standard BP Algorithm

The Markov Random Field framework represents the template as a set of nodes with a particular pairwise connectivity structure and a posterior probability distribution given by Equation (6). In order to estimate the marginal distributions the BP algorithm introduces a set of *message* variables, where $m_{ij}(\mathbf{q}_j)$ corresponds to the "message" that node i sends to node j about the degree of its belief that node j is in state $\mathbf{q}_j$. The BP algorithm to update the messages is given by:

$$m_{ij}^{(t+1)}(\mathbf{q}_j) = \frac{1}{Z_{ij}} \sum_{\mathbf{q}_i} \psi_{ij}(\mathbf{q}_i, \mathbf{q}_j) \psi_i(\mathbf{q}_i) \prod_{k \in N(i) \backslash j} m_{ki}^{(t)}(\mathbf{q}_i) \tag{8}$$

where $Z_{ij} = \sum_{\mathbf{q}_i} m_{ij}^{(t+1)}(\mathbf{q}_i)$ is a normalization factor and the neighborhood $N(i)$ denotes the set of nodes directly coupled to i (excluding i itself), and the superscripts $(t+1)$ and t are discrete time indices. The product is over all nodes k directly coupled to i except for j. The messages $m_{ij}^{(0)}(.)$ may be initialized to uniform values (see next subsection for details). The updates can be done in parallel for *all* connected pairs of nodes i and j, or they can be applied to different *subsets* of pairs of nodes at different times, without changing the fixed point of Equation (8). We will exploit this freedom in our focused message updating scheme, described in section (3.3), in order to boost the speed of convergence.

The *belief variable* $b_i(\mathbf{q}_i)$ is an estimate of the marginal distribution $P(\mathbf{q}_i|\mathbf{D})$ derived from the message variables as follows (omitting the time superscripts for simplicity):

$$b_i(\mathbf{q}_i) = \frac{1}{Z_i'} \psi_i(\mathbf{q}_i) \prod_{k \in N(i)} m_{ki}(\mathbf{q}_i) \tag{9}$$

where Z_i' is a normalization factor chosen so that $\sum_{\mathbf{q}_i} b_i(\mathbf{q}_i) = 1$. If the connectivity graph is a tree, then the belief variables $b_i(\mathbf{q}_i)$ are guaranteed to equal the true marginals $P(\mathbf{q}_i|\mathbf{D})$ when the update equations have converged to a fixed point[11].

In general the computational cost of performing a message update is proportional to $|S|^2$, where $|S|$ is the size of the set S, since the sum over $\mathbf{q}_i$ in Equation (8) must be evaluated for all $\mathbf{q}_j$ on the right-hand side of the equation. However, we exploit the fact that the compatibility function $\psi_{ij}(\mathbf{q}_i, \mathbf{q}_j)$ is vanishingly small for most combinations of $\mathbf{q}_i$ and $\mathbf{q}_j$. We have experimented with exploiting this "sparsity" constraint in two ways. One way is to consider only all (x_j, y_j) that are sufficiently close to (x_i, y_i) (no more than a few times the distance between (x_i, y_i) and (x_j, y_j) on the reference shape), i.e. all $\mathbf{q}_j$ within a ball B centered on $\mathbf{q}_i$. The other is to use a smaller ball B centered on a prediction of $\mathbf{q}_j$ given $\mathbf{q}_i$, using the reference shape to make the prediction. Both of these schemes reduce the computational cost of performing a message update from $|S|^2$ to $|S||B|$.

3.2 Belief Pruning

We initialize the messages to uniform values: $m_{ij}^{(0)}(\mathbf{q}_i) = 1/|S|$, which corresponds to all of the beliefs being initialized to uniform distributions across S. We then perform message updates followed by calculations of the beliefs, and iterate this process. If at any point the (normalized) belief of a state $\mathbf{q}_i$ associated with node i falls below a threshold value ϵ (which we typically set to $\epsilon = 10^{-8}/|S|$), we *remove this state from further consideration* for node i. More precisely, if $b_i^{(t)}(\mathbf{q}_i)$ is the belief of node i in state $\mathbf{q}_i$ at time t, then we define a *pruned state space* $S_i^{(t)}$ for node i:

$$S_i^{(t)} = \{\mathbf{q} \in S | b_i^{(t)}(\mathbf{q}) > \epsilon\}$$

Using this notation we can write the following message update equation:

$$m_{ij}^{(t+1)}(\mathbf{q}_j) = \frac{1}{Z_{ij}} \sum_{\mathbf{q}_i \in S_i^{(t)}} \psi_{ij}(\mathbf{q}_i, \mathbf{q}_j)\psi_i(\mathbf{q}_i) \prod_{k \in N(i) \setminus j} m_{ki}^{(t)}(\mathbf{q}_i) \tag{10}$$

for all $\mathbf{q}_j \in S_j^{(t)}$. The expression for the beliefs is unchanged from Equation (9) except that $b_i^{(t)}(\mathbf{q}_i)$ is only evaluated for $\mathbf{q}_i \in S_i^{(t)}$, and the normalization factor Z_i' is chosen so that $\sum_{\mathbf{q}_i \in S_i^{(t)}} b_i(\mathbf{q}_i) = 1$.

3.3 Focused Message Updating

The BP message update equation can be executed either by updating all the messages $m_{ij}(\mathbf{q}_i)$ (we are omitting time superscripts in this section for convenience) in parallel or by updating certain messages before others. The idea of focused message updating is to update messages in a particular sequence so as to speed up BP. The sequence of message updates corresponds to a particular sequence of nodes $i_1, i_2, \ldots i_M$. The first messages to be updated are $m_{i_1,i_2}(\mathbf{q}_{i_2})$, followed by a calculation of the belief $b_{i_2}(\mathbf{q}_{i_2})$. This causes the state space S_{i_2}, which was initialized to the full space S, to be pruned. Then messages $m_{i_2,i_3}(\mathbf{q}_{i_3})$ are updated, and since S_{i_2} has just been pruned this update will be faster than the first update. Additional pruning will occur when $b_{i_3}(\mathbf{q}_{i_3})$ is calculated, and the process repeats until $m_{i_{M-1},i_M}(\mathbf{q}_{i_M})$ is updated. Subsequent sequences of updates are performed until messages $m_{ij}(\mathbf{q}_j)$ have been updated for all neighboring pairs i and j at least once and the updates have converged. In practice, we have obtained good solutions for our deformable template by calculating the beliefs after a sufficiently long *fixed* sequence of updates, without needing to check for convergence to decide when to halt the updates.

For our deformable template the initial sequence of nodes has been chosen consistent with the 20 Questions (divide-and-conquer) search strategy [5,13]. Nodes corresponding to "key features" of the target shape – portions of the contour, such as T-junctions and corners, that are less likely than other portions to appear in the background clutter – are visited in the message updates first. This can be illustrated in the case of the letter "A" template, whose connectivity graph is shown in Figure (1). We begin by passing messages from node 3 to 16, which is a pair of neighbors chosen as defining a portion of a T-junction on the left side of the letter. The normals corresponding to these neighboring nodes are roughly perpendicular to each other, and we expect this configuration to be somewhat rarer in background clutter than the straight-line configurations that predominate in the structure of the letter "A" (e.g. nodes 0 through 3 lie along a roughly straight section).

Continuing the 20 Questions strategy, the best portion of the template to search for next should be the other T-junction (on the right side of the letter), because we expect that the *conjunction* of the two T-junction features should be quite rare in the background. Accordingly we next pass messages from node 16 to 19 (via the long-distance connection) and then from 19 to 11, and by the time messages are passed to node 11 a significant amount of pruning has taken place. Messages are then passed in the opposite direction, from nodes 11 to 19 to 16 to 3, so that the state space at all four of those nodes has been substantially reduced, and we can treat these four nodes as a foundation from which we can send out messages to the rest of the nodes in the template. There are many possible ways to continue the updates, and we refer to the technical report [4] for details of the procedure we use. Briefly, the bridge of the letter is traversed, followed by the two legs, followed by the main loop at the top of the letter, and then the entire process is repeated.

We experimented with removing the long distance connection from the letter "A" prior (which required slight modifications of the message update sequence) and found that, although BP still converged to the correct solution, it took about 50 percent longer to do so. This shows that the long-distance connection causes substantial pruning, and since the bulk of the computations take place in the message updates among the first several pairs of nodes, increased pruning early on can speed up the overall convergence significantly. Empirically we found that the combined use of belief pruning and focused message updating speeds up BP by a factor of about 30.

4 Experimental Results

We tested our deformable template on three shape models: the letter "A," a stick-figure shape and the contour of an open hand. For each shape, the shape prior was created by fixing a suitable reference shape and a sparse connectivity structure. The coefficients K^A_{ij}, K^B_{ij} and K^C_{ij} from Equation (1) were then chosen so that the stochastic samples from the shape prior represented the desired shape variability reasonably well (see Figures (1,2,4) to see the reference shapes and stochastic samples for the letter "A," stick-figure and hand shapes).

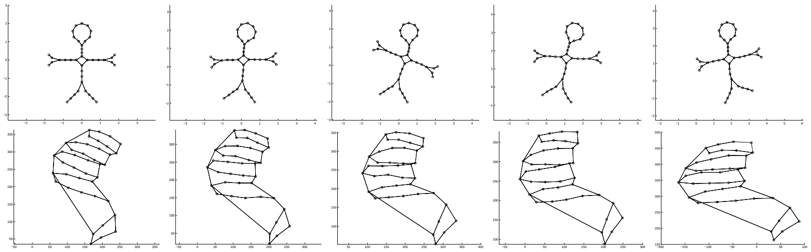

Fig. 4. Stick-figure shape on top row, hand shape on bottom row. Reference shapes are shown in leftmost column with lines showing the associated connectivity graph, followed by stochastic shape samples in other columns. In the hand shape sample figures, the thumb is shown on the bottom, and long-distance connections are shown between adjacent fingertips.

The letter "A" and stick-figure templates were tested on grayscale images of a whiteboard with handwritten characters and shapes (see Figures (5,6,7) for sample results), and the hand template was tested on grayscale indoor images taken of the hand. The images, which were taken by a digital camera at a resolution of 1280 x 1000 pixels, were decimated by an appropriate amount (typically by a factor of 4 in each dimension) to reduce the size of the state space S. The edge map $I_e(\mathbf{x})$ and orientation map $\phi(\mathbf{x})$ were then computed on the decimated images, which took less than a second of CPU time.

To reduce the state space S further, we *pre-pruned* the images by removing pixels that were unlikely to be edges: any pixel location $\mathbf{x}$ such that the ratio $P_{on}(I_e(\mathbf{x}))/P_{off}(I_e(\mathbf{x})) < 0.5$ was removed from consideration in the state space S. Finally, we experimented with the simplest possible quantization of template angles: for each pixel location (x, y) remaining after pre-pruning, the triples $(x, y, \phi(x, y))$ and $(x, y, \phi(x, y) + \pi)$ were included in S. A finer sampling of angles in the range $[0, 2\pi)$, perhaps including several angles focused in the neighborhood of $\phi(x, y)$ and $\phi(x, y) + \pi$, may be more desirable, though this has not yet been tested. Finally, the template was scaled manually to match the scale of the target shape in each image before running BP (it sufficed to set the scale to a precision of about ±10%.)

For the whiteboard images, the usual edge map $I_e(\mathbf{x})$, based on the magnitude of the image gradient (see section (2.2)) was replaced by a Laplacian operator in order to minimize the number of edge responses: the image gradient is high on both sides of a handwritten stroke, whereas the magnitude of the Laplacian attains a high value only in the center of the stroke. Since the Laplacian map had high contrast on the whiteboard images, we were able to safely quantize its values into two bins, so that the $P_{on}(I_e(\mathbf{x}))/P_{off}(I_e(\mathbf{x}))$ ratio assumed either a low or a high value (see [4] for details).

Results for the "A" template are shown in Figures (5) and (6). Figure (5) illustrates the deformable template's ability to handle significant shape deformations (left panel) and global rotations (right panel). Figure (6) demonstrates that matching is successful even with substantial amounts of clutter. Figure (7) shows successful matches for the stick-figure template, which demonstrates the high amount of variability that is allowed (in this case in the arms of the stick-figure). Finally, Figure (8) shows that the deformable template can detect shapes even when the target has relatively low contrast relative to the background.

The CPU time needed to perform the deformable template matching by BP is roughly proportional to the number of nodes on the template and the number of pixels that survive pre-pruning. However, how much pruning occurs during BP, as well as when it occurs, also affects the speed significantly, and these factors vary from image to image. On an 850 MHz AMD processor, the images in Figure (5) took 2.8 and 21 seconds of CPU time, respectively; the total CPU time divided by the number of initial states $|S|$ for both was on the order of 1 msec per state. The times for the stick-figure and hand results were not measured precisely but ranged from several seconds to half a minute for the stick figures to a few minutes for the hands.

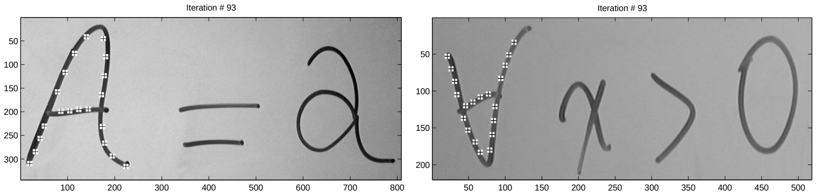

Fig. 5. Two sample letter "A" results, with black-and-white crosses drawn to indicate the location of the "A" shape as estimated by the deformable template algorithm. In the example on the left, note that the template locates the upper-case "A" and not the lower-case version (a separate shape prior would need to be used to detect lower-case "A"'s). On the right, note that the template is rotation-invariant and thus is able to detect the upside-down "A."

5 Conclusion

We have described a promising algorithm for detecting flexible objects in real images. The algorithm can handle substantial shape deformations, large amounts of background clutter and low contrast between target and background.

It is desirable to speed up the algorithm further and to deal with certain limitations of the model. Alternatives to BP for estimating posterior marginals, such as CCCP [20] and tree reparameterization [14], enjoy certain advantages over BP (for instance, convergence to a locally optimal solution is guaranteed for CCCP even on a loopy graph) and may prove faster than BP if techniques

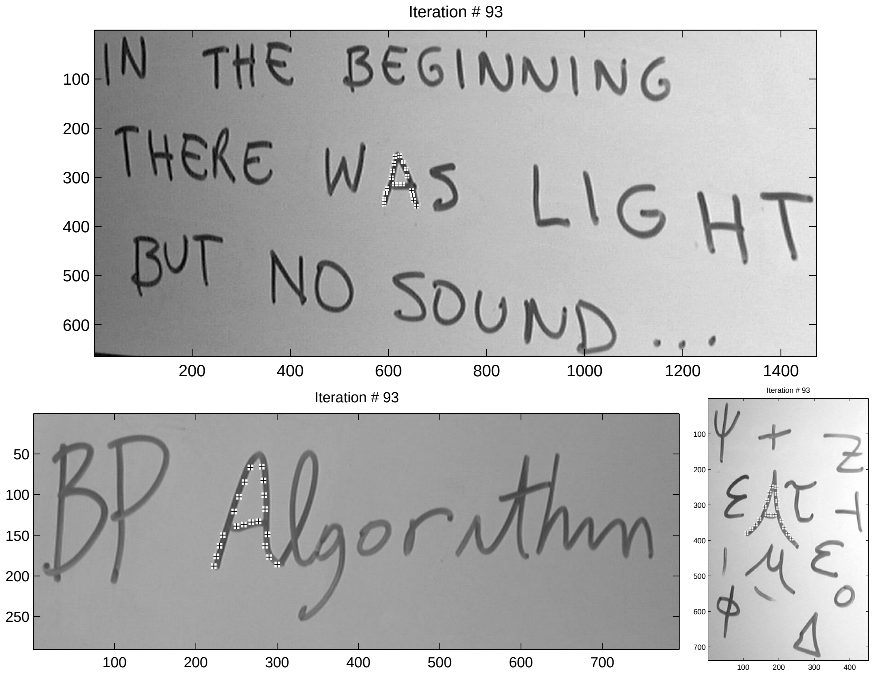

Fig. 6. Letter " A" located by deformable template algorithm in images with considerable background clutter.

analogous to belief pruning and focused message updating can be adapted to them. We expect that augmenting the existing edge and orientation maps by other low-level cues such corner, T-junction and endstop detectors would significantly speed up template matching by providing a quick method for zeroing in on key features.

The biggest limitation of the current model seems to be the lack of scale invariance. One way to overcome this limitation is to use a scale-invariant shape representation. Another solution may be to use a hierarchical representation that detects parts of the target shape – such as straight-line segments and fingertips in the case of a hand shape – at a range of scales before grouping them together.

Acknowledgements

We would like to thank Alan Yuille for his useful comments and suggestions. Both authors were supported by the National Institute of Health (NEI) grant number RO1-EY 12691-01. Sabino Ferreira was also supported by the CNPq-Brazilian National Council of Research.

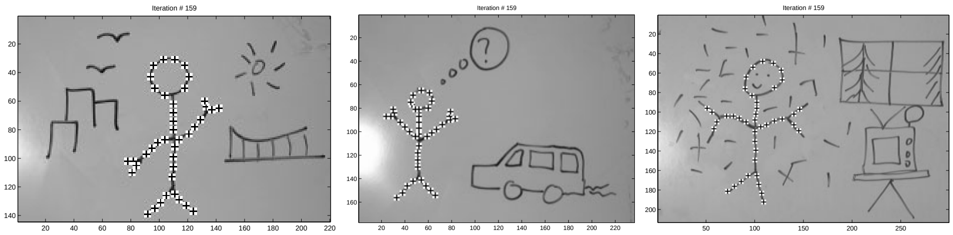

Fig. 7. Stick-figure matching results. Note the high variability of the shape, especially in the arms.

References

1. S. Belongie, J. Malik and J. Puzicha, "Shape Matching and Object Recognition Using Shape Contexts," accepted for publication in PAMI.
2. H. Chui and A. Rangarajan, "A new algorithm for non-rigid point matching," IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 2000.
3. J. Coughlan, A.L. Yuille, C. English, D. Snow. "Efficient Deformable Template Detection and Localization without User Initialization." Computer Vision and Image Understanding, Vol. 78, No. 3, pp. 303-319. June 2000.
4. J. Coughlan and S. Ferreira. "Finding Deformable Shapes using Loopy Belief Propagation." Technical report in preparation.
5. F. Fleuret and D. Geman, "Coarse-to-fine face detection," Inter. Journal of Computer Vision, 41, 85-107, 2001.
6. D. Geman. and B. Jedynak. "An active testing model for tracking roads in satellite images". *IEEE Trans. Patt. Anal. and Machine Intel.* Vol. 18. No. 1, pp 1-14. January. 1996.
7. U. Grenander, Y. Chow and D. M. Keenan, *Hands: a Pattern Theoretic Study of Biological Shapes,* Springer-Verlag, 1991.
8. S. Konishi, A. L. Yuille, J. M. Coughlan, and S. C. Zhu. "Fundamental Bounds on Edge Detection: An Information Theoretic Evaluation of Different Edge Cues." *Proc. Int'l conf. on Computer Vision and Pattern Recognition*, 1999.
9. B. Lowerre and R. Reddy, "The Harpy Speech Understanding System." In **Trends in Speech Recognition**. Ed. W. Lea. Prentice Hall. 1980.
10. K.P. Murphy,Y. Weiss and M.I. Jordan. "Loopy belief propagation for approximate inference: an empirical study". In Proceedings of Uncertainty in AI. 1999.
11. J. Pearl. **Probabilistic Reasoning in Intelligent Systems.** Morgan Kaufman. 1988.
12. H. Schneiderman and T. Kanade. "A Statistical Method for 3D Object Detection Applied to Faces and Cars". IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2000).
13. P. Viola and M. Jones. "Rapid Object Detection using a Boosted Cascade of Simple Features," CVPR 2001.
14. M. Wainwright, T. Jaakkola and A. Willsky. "Tree-based reparameterization for approximate estimation on loopy graphs." To appear in NIPS 2001.
15. L. Wiscott and C. von der Marlsburg, "A Neural System for the Recognition of Partially Occluded Objects in Cluttered Scenes". Neural Computation. 7(4):935-948. 1993.

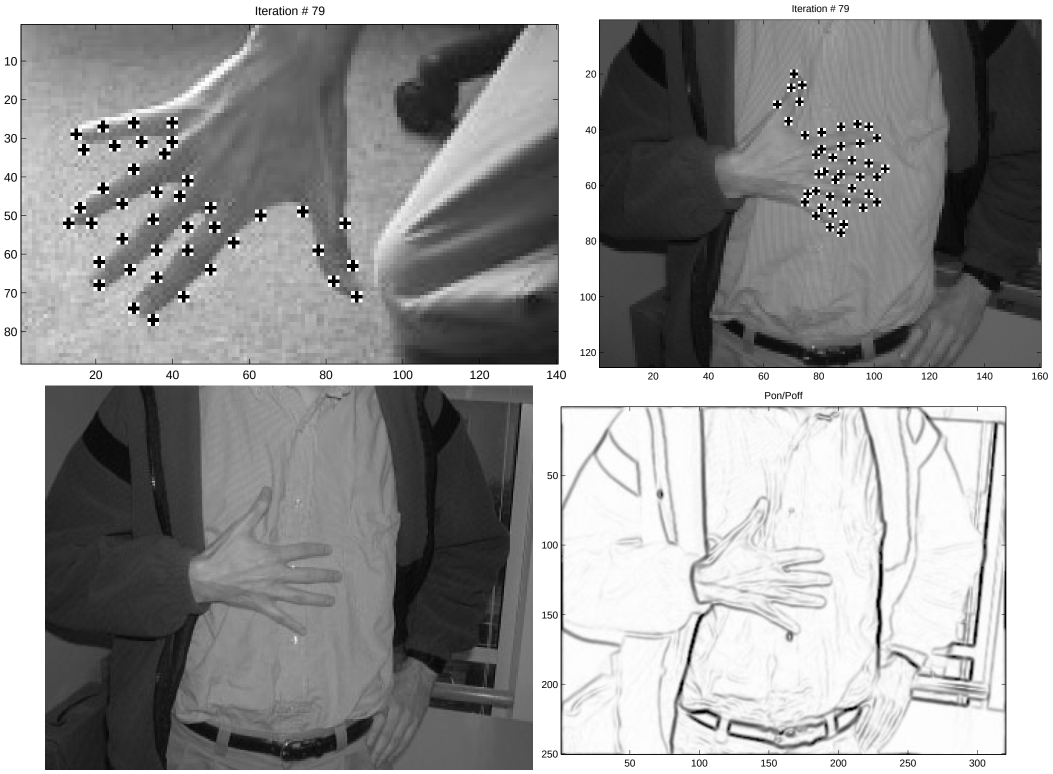

Fig. 8. Top row, results of matching hand template in images. Matching succeeds even for fairly low contrast between the target against background, as shown for second result in bottom row (original image on left, $P_{on}(I_e(\mathbf{x}))/P_{off}(I_e(\mathbf{x}))$ ratio on right).

16. J.S. Yedidia, W.T. Freeman, Y. Weiss. “Bethe Free Energies, Kikuchi Approximations, and Belief Propagation Algorithms”. 2001. MERL Cambridge Research Technical Report TR 2001-16.
17. A. L. Yuille, “Deformable Templates for Face Recognition”. *Journal of Cognitive Neuroscience.* Vol 3, Number 1. 1991.
18. A.L. Yuille and J. Coughlan. “Twenty Questions, Focus of Attention, and A*: A theoretical comparison of optimization strategies.” In *Proceedings EMMCVPR' 97.* Venice. Italy. 1997.
19. A.L. Yuille and J. Coughlan. ”Visual Search: Fundamental Bounds, Order Parameters, and Phase Transitions.” IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI), Vol. 22, No. 2, pp. 1-14. February 2000.
20. A. L. Yuille, “CCCP Algorithms to Minimize the Bethe and Kikuchi Free Energies: Convergent Alternatives to Belief Propagation.” Neural Computation. In press.
21. S. C. Zhu, R. Zhang, and Z. W. Tu, ”Integrating Top-down/Bottom-up for Object Recognition by Data Driven Markov Chain Monte Carlo”, Proc. of Int'l Conf. on Computer Vision and Pattern Recognition, Hilton Head, SC, 2000.