Although particle filters are often very effective, they are specialized to temporal problems whose corresponding graphs are simple Markov chains (see Figure 1). Many vision problems, however, are characterized by non–causal (e.g., spatial or model–induced) structure which is better represented by a more complex graph. Because particle filters cannot be applied to arbitrary graphs, graphical models containing high–dimensional variables may pose severe problems for existing inference algorithms. Even for tracking problems, there is often structure within each time instant (for example, associated with an articulated model) which is ignored by standard particle filters.

Some authors have used junction tree representations [12] to develop structured approximate inference techniques for general graphs. These algorithms begin by clustering nodes into cliques chosen to break the original graph's cycles. A wide variety of algorithms can then be specified by combining an approximate clique variable representation with local methods for updating these approximations [3, 11]. For example, Koller et al. [11] propose a framework in which the current clique potential estimate is used to guide message computations, allowing approximations to be gradually refined over successive iterations. However, the sample algorithm they provide is limited to networks containing mixtures of discrete and Gaussian variables. In addition, for many graphs (e.g. nearest–neighbor grids) the size of the junction tree's largest cliques grows exponentially with problem size, requiring the estimation of extremely high–dimensional distributions.

The *nonparametric belief propagation* (NBP) algorithm we develop in this paper differs from previous nonparametric approaches in two key ways. First, for graphs with cycles we do *not* form a junction tree, but instead iterate our local message updates until convergence as in loopy BP. This has the advantage of greatly reducing the dimensionality of the spaces over which we must infer distributions. Second, we provide a message update algorithm specifically adapted to graphs containing continuous, non–Gaussian potentials. The primary difficulty in extending particle filters to general graphs is in determining efficient methods for combining the information provided by several neighboring nodes. Representationally, we address this problem by associating a regularizing kernel with each particle, a step which is necessary to make message products well defined. Computationally, we show that message products may be computed using an efficient *local* Gibbs sampling procedure. The NBP algorithm may be applied to arbitrarily structured graphs containing a broad range of potential functions, effectively extending particle filtering methods to a much broader range of vision problems. For a related generalization of particle filters to graphical models, see [9].

Following our presentation of the NBP algorithm, we validate its performance on a small Gaussian network. We then show how NBP may be combined with parts–based local appearance models [4, 6, 14, 22] to locate and reconstruct occluded facial features.

## 2. Undirected Graphical Models

An undirected graph $\mathcal{G}$ is defined by a set of nodes $\mathcal{V}$, and a corresponding set of edges $\mathcal{E}$. The *neighborhood* of a node $s \in \mathcal{V}$ is defined as $\Gamma(s) \triangleq \{t \mid (s,t) \in \mathcal{E}\}$. Graphical models associate each node $s \in \mathcal{V}$ with an unobserved, or hidden, random variable $x_s$, as well as a noisy local observation $y_s$. Let $x = \{x_s \mid s \in \mathcal{V}\}$ and $y = \{y_s \mid s \in \mathcal{V}\}$ denote the sets of all hidden and observed variables, respectively. For simplicity, we consider models with pairwise potential functions, for which $p(x, y)$ factorizes as

$$p(x,y) = \frac{1}{Z} \prod_{(s,t)\in\mathcal{E}} \psi_{s,t}(x_s, x_t) \prod_{s\in\mathcal{V}} \psi_s(x_s, y_s) \qquad (1)$$

However, the nonparametric updates we present may be directly extended to models with higher–order potentials.

In this paper, we focus on the calculation of the conditional marginal distributions $p(x_s \mid y)$ for all nodes $s \in \mathcal{V}$. These densities provide not only estimates of $x_s$, but also corresponding measures of uncertainty.

### 2.1. Belief Propagation

For graphs which are acyclic or tree–structured, the desired conditional distributions $p(x_s \mid y)$ can be directly calculated by a local message–passing algorithm known as *belief propagation* (BP) [16, 24]. At iteration $n$ of the BP algorithm, each node $t \in \mathcal{V}$ calculates a message $m_{ts}^n(x_s)$ to be sent to each neighboring node $s \in \Gamma(t)$:

$$m_{ts}^n(x_s) = \alpha \int_{x_t} \psi_{s,t}(x_s, x_t)\, \psi_t(x_t, y_t) \times \prod_{u\in\Gamma(t)\setminus s} m_{ut}^{n-1}(x_t)\, dx_t \qquad (2)$$

Here, $\alpha$ denotes an arbitrary proportionality constant. At any iteration, each node can produce an approximation $\hat{p}^n(x_s \mid y)$ to the marginal distribution $p(x_s \mid y)$ by combining the incoming messages with the local observation:

$$\hat{p}^n(x_s \mid y) = \alpha \psi_s(x_s, y_s) \prod_{t\in\Gamma(s)} m_{ts}^n(x_s) \qquad (3)$$

For tree–structured graphs, the approximate marginals, or beliefs, $\hat{p}^n(x_s \mid y)$ will converge to the true marginals $p(x_s \mid y)$ once the messages from each node have propagated to every other node in the graph.

Because each iteration of the BP algorithm involves only local message updates, it can be applied even to graphs with cycles. For such graphs, the statistical dependencies between BP messages are not accounted for, and the sequence of beliefs $\hat{p}^n(x_s \mid y)$ will *not* converge to the true marginals. In many applications, however, the resulting loopy BP algorithm exhibits excellent empirical performance [7, 20]. Recently, several theoretical studies have provided insight into