MIT Laboratory for Information and Decision Systems
Technical Report P-2551 (October 15, 2002)

Nonparametric Belief Propagation

Erik B. Sudderth, Alexander T. Ihler,
William T. Freeman, and Alan S. Willsky
Department of Electrical Engineering and Computer Science
Massachusetts Institute of Technology
Cambridge, MA 02139
esuddert@mit.edu, ihler@mit.edu, wtf@ai.mit.edu, willsky@mit.edu

Abstract

In applications of graphical models arising in fields such as computer vision, the hidden variables of interest are most naturally specified by continuous, non–Gaussian distributions. However, due to the limitations of existing inference algorithms, it is often necessary to form coarse, discrete approximations to such models. In this paper, we develop a *nonparametric belief propagation* (NBP) algorithm, which uses stochastic methods to propagate kernel–based approximations to the true continuous messages. Each NBP message update is based on an efficient sampling procedure which can accomodate an extremely broad class of potential functions, allowing easy adaptation to new application areas. We validate our method using comparisons to continuous BP for Gaussian networks, and an application to the stereo vision problem.

1 Introduction

Graphical models provide an efficient, powerful framework for modeling probabilistic relationships. Most existing applications of graphical models employ hidden variables which take values from a discrete, finite set. In some fields, such as error correcting codes [1], these discrete models are entirely appropriate. However, in areas such as computer vision [2–5], they typically arise from quantizations of continuous variables. Computional costs invariably limit the accuracy of such quantizations, leading to biases and artifacts in the resulting estimates.

The popularity of discrete graphical models can be traced to the existence of efficient algorithms for solving learning and inference problems [6]. For discrete models, the difficulties associated with inference are purely *computational*, and involve finding ways to efficiently decompose and/or approximate sums of exponentially many terms. Although exact inference in general discrete graphs is NP hard [7], approximate inference algorithms such as loopy belief propagation (BP) [1, 8, 9] have been shown to produce good results for a wide range of interesting models [2, 5].

For graphical models containing continuous hidden variables, there is an additional *representational* challenge. The conditional distributions sought by inference algorithms are not finite vectors but continuous functions. Unfortunately, even if the potential functions specifying the graphical model have simple parametric forms, the integral equations used to propagate information across the graph rarely have tractable analytic solutions [10]. A notable exception occurs when all variables are jointly Gaussian, so that distributions may be parameterized by their mean and covariance [9]. However, many real–world phenomena exhibit significant outliers, bimodalities, and other statistical features which are far from Gaussian.

Since the conditional densities arising in non–Gaussian graphical models lack tractable parametric forms, it is natural to consider nonparametric approximations. For temporal inference problems defined on Markov chains, a variety of density representations have been explored. Gaussian sum filters [11] use local linearizations to deterministically propagate Gaussian mixture models. In contrast, particle filters [12] use Monte Carlo methods to stochastically update a set of weighted point samples. The stability and robustness of particle filters can often be improved by regularization methods [12, Chapter 12] in which smoothing kernels [13, 14] explicitly represent the uncertainty associated with each point sample.

For more general graphs, the junction tree representation [15] has been used to develop structured approximate inference techniques. A wide variety of algorithms can be specified by combining an approximate clique variable representation with local methods for updating these approximations [16, 17]. For example, distributions over large cliques of discrete variables can be approximated by a set of weighted point samples, and then related to neighboring nodes using standard message–passing recursions [18, 19]. Koller et al. [17] propose a more sophisticated 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 extends this earlier work 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. In addition, we provide a message update algorithm specifically adapted to graphs containing continuous, non–Gaussian potentials. Each message is represented by a kernel–based nonparametric density estimate, and message products are found by an efficient *local* Gibbs sampling algorithm. We validate the NBP algorithm on a small Gaussian network, and present stereo vision results demonstrating its effectiveness.

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 $\Gamma(s) \triangleq \{t | (s,t) \in \mathcal{E}\}$, the set of all nodes which are directly connected to s. 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 $x = \{x_s\}_{s \in \mathcal{V}}$ and $y = \{y_s\}_{s \in \mathcal{V}}$ denote the sets of all hidden and observed variables, respectively. To simplify the presentation, we consider models with pairwise potential functions, for which $p(x,y)$ factorizes as

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

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

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 distributions may be used to calculate the best estimates of the hidden variables x_s relative to a wide range of criteria. For example, the conditional mean $E[x_s \mid y]$ provides the Bayes' least squares estimate. More generally, conditional distributions are useful because they provide information about the degree of uncertainty in the estimate of each hidden node.

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) [1,6]. 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 $s \in \Gamma(t)$:

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

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

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

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. Conceptually, the steady state BP message $m_{ts}(x_s)$ is a sufficient statistic of the observations in the subgraph which node t separates from node s.

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 properly accounted for, and the sequence of beliefs $\hat{p}^n(x_s \mid y)$ will *not* converge to the true marginal distributions. In many applications, however, the resulting loopy BP algorithm exhibits excellent empirical performance [2,5]. Recently, several theoretical studies have provided insight into the approximations made by loopy BP, partially justifying its application to graphs with cycles [1,8,9].

2.2 Nonparametric Representations

Exact evaluation of the BP update equation (2) involves an integration which, as discussed in the Introduction, is not analytically tractable for most continuous hidden variables. An interesting alternative is to represent the resulting message $m_{ts}(x_s)$ nonparametrically as a kernel–based density estimate [13,14]. Let $\mathcal{N}(x; \mu, \Lambda)$ denote the value of a Gaussian density of mean μ and covariance Λ at the point x. We may then approximate $m_{ts}(x_s)$ by a mixture of M Gaussian kernels as

$$m_{ts}(x_s) = \sum_{i=1}^{M} w_s^i \mathcal{N}(x_s; x_s^i, \Lambda_s) \tag{4}$$

where w_s^i is the weight associated with the i^{th} kernel mean x_s^i, and Λ_s is a bandwidth or smoothing parameter. Other choices of kernel functions are possible [14], but in this paper we restrict our attention to mixtures of diagonal–covariance Gaussians.

In the following section, we describe stochastic methods for determining the kernel centers x_s^i and associated weights w_s^i. The resulting nonparametric representations are only meaningful when the messages $m_{ts}(x_s)$ are finitely integrable.[1] To guarantee this, it is sufficient to assume that all potentials satisfy the following constraints:

$$\int_{x_s} \psi_{s,t}(x_s, x_t = \bar{x})\, dx_s < \infty \qquad\qquad \int_{x_s} \psi_s(x_s, y_s = \bar{y})\, dx_s < \infty \tag{5}$$

Under these assumptions, a simple induction argument will show that all messages are normalizable. Heuristically, equation (5) requires all potentials to be "informative," so that fixing the value of one variable constrains the likely locations of the other. In most application domains, this can be trivially achieved by assuming that all hidden variables take values in a large, but bounded, range.

3 Nonparametric Message Updates

Conceptually, the BP update equation (2) naturally decomposes into two stages. First, the message product $\psi_t(x_t, y_t) \prod_u m_{ut}^{n-1}(x_t)$ combines information from neighboring nodes with the local evidence y_t to produce a likelihood function for x_t. Second, this likelihood function is combined with the compatibility potential $\psi_{s,t}(x_s, x_t)$, and then integrated to produce likelihoods for x_s. The nonparametric belief propagation (NBP) algorithm stochastically approximates these two stages, producing consistent nonparametric representations of the messages $m_{ts}(x_s)$. Approximate marginals $\hat{p}(x_s \mid y)$ may then be determined from these messages by applying the following section's stochastic product algorithm to equation (3).

3.1 Message Products

For the moment, assume that the local observation potentials $\psi_t(x_t, y_t)$ are represented by weighted Gaussian mixtures (such potentials arise naturally from learning–based approaches to model identification [2]). The product of d Gaussian densities is itself Gaussian, with mean and covariance given by

$$\prod_{i=1}^{d} \mathcal{N}(x; \mu_i, \Lambda_i) \propto \mathcal{N}\big(x; \bar{\mu}, \bar{\Lambda}\big) \qquad \bar{\Lambda}^{-1} = \sum_{i=1}^{d} \Lambda_i^{-1} \qquad \bar{\Lambda}^{-1}\bar{\mu} = \sum_{i=1}^{d} \Lambda_i^{-1}\mu_i \tag{6}$$

Thus, a BP update operation which multiplies d Gaussian mixtures, each containing M components, will produce another Gaussian mixture with M^d components. The weight $\bar{w}$ associated with product mixture component $\mathcal{N}\big(x; \bar{\mu}, \bar{\Lambda}\big)$ is given by

$$\bar{w} \propto \frac{\prod_{i=1}^{d} w_i \mathcal{N}(x; \mu_i, \Lambda_i)}{\mathcal{N}\big(x; \bar{\mu}, \bar{\Lambda}\big)} \tag{7}$$

where $\{w_i\}_{i=1}^d$ are the weights associated with the input Gaussians. Since integration of Gaussian mixtures is straightforward, in principle the BP message updates could be performed exactly by repeated use of equations (6,7). In practice, however, the exponential growth of the number of mixture components forces approximations to be made. Given d input mixtures of M Gaussian, the NBP algorithm approximates their M^d–component product mixture by drawing M independent samples.

Direct sampling from this product, achieved by explicitly calculating each of the product component weights (7), would require $\mathcal{O}(M^d)$ operations. The complexity

[1] Probabilistically, BP messages are likelihood functions $m_{ts}(x_s) \propto p(y = \bar{y} \mid x_s)$, not densities, and are *not* necessarily integrable (e.g., when x_s and y are independent).

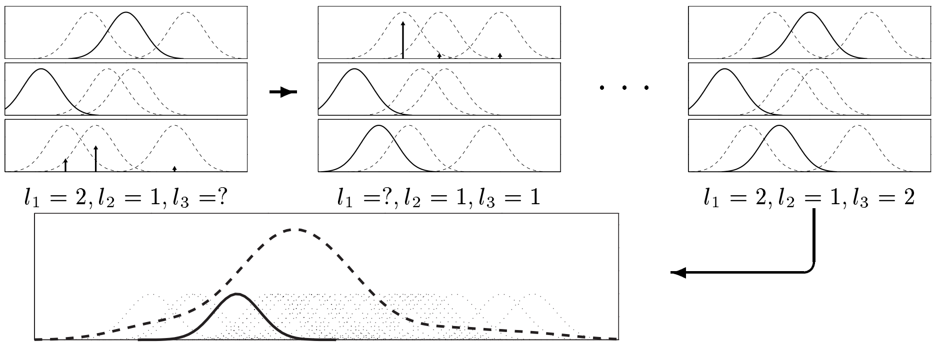


Figure 1: Gibbs sampler for a product of 3 Gaussian mixtures, with 3 kernels each. New indices are sampled according to weights (arrows) determined by the two fixed components (solid). The final labels identify one of the 3^3 components of the product density (dashed).

associated with this sampling is combinatorial: each product component is defined by d labels $\{l_i\}_{i=1}^d$, where l_i identifies a kernel in the i^{th} input mixture. Although the joint distribution of the d labels is complex, the conditional distribution of any individual label l_j is simple. In particular, assuming fixed values for $\{l_i\}_{i \neq j}$, equation (7) can be used to sample from the conditional distribution of l_j in $\mathcal{O}(M)$ operations.

Since the mixture label conditional distributions are tractable, we may use a Gibbs sampler [3] to draw asymptotically unbiased samples from the product distribution. At each iteration, the labels $\{l_i\}_{i \neq j}$ for $d-1$ of the input mixtures are fixed, and a new value for the j^{th} label is chosen according to equation (7). At the following iteration, the newly chosen l_j is fixed, and another label is updated (see Figure 1). Following the final iteration, the mean and covariance of the selected mixture component is found using equation (6), and a sample point is drawn.

Assuming the number of iterations for the Gibbs sampler to equilibrate is independent of M, we may draw M samples from the product mixture in $\mathcal{O}(dM^2)$ operations. Although formal verification of the Gibbs sampler's convergence is difficult, in our experiments we have observed good performance using far fewer computations than required by direct sampling. Note that in the NBP algorithm, the Gibbs sampler only involves a few hidden variables, in contrast to the very large state spaces of classic simulated annealing [3].

In some applications, the observation potentials $\psi_t(x_t, y_t)$ are most naturally specified by analytic functions. The previously proposed Gibbs sampler may be easily adapted to this case using *importance sampling* [12]. At each iteration, the weights produced by equation (7) are rescaled by $\psi_t(\bar{\mu}, y_t)$, the observation likelihood at that kernel's center. Then, the final sample x_s^i is assigned weight $w_s^i = \psi_t\left(x_s^i, y_t\right) / \psi_t(\bar{\mu}, y_t)$ to account for variations of the analytic potential over the kernel's support. This procedure will be most effective when $\psi_t(x_t, y_t)$ varies slowly relative to the typical kernel bandwidth.

3.2 Message Propagation

In the second stage of the NBP algorithm, the information contained in the incoming message product is propagated by stochastically approximating the belief update integral (2). To perform this stochastic integration, the pairwise potential $\psi_{s,t}(x_s, x_t)$ must be decomposed to separate its *marginal* influence on x_t from the *conditional* relationship it defines between x_s and x_t.

The marginal influence function $\zeta(x_t)$ is determined by the relative weight assigned to *all* x_s values for each x_t:

$$\zeta(x_t) = \int_{x_s} \psi_{s,t}\left(x_s, x_t\right)\, dx_s \tag{8}$$

The NBP algorithm accounts for the marginal influence of $\psi_{s,t}\left(x_s, x_t\right)$ by incorporating $\zeta(x_t)$ into the Gibbs sampler. If $\psi_{s,t}\left(x_s, x_t\right)$ is a Gaussian mixture, extraction of $\zeta(x_t)$ is trivial. Alternately, if $\zeta(x_t)$ can be evaluated (or approximated) pointwise, analytic pairwise potentials may be dealt with using importance sampling. In the common case where pairwise potentials depend only on the difference between their arguments ($\psi_{s,t}\left(x, \bar{x}\right) = \psi_{s,t}\left(x - \bar{x}\right)$), $\zeta(x_t)$ is constant and can be neglected.

To complete the stochastic integration, each particle x_t^j produced by the Gibbs sampler is propagated to node s by sampling $x_s^j \sim \alpha\psi_{s,t}(x_s, x_t^j)$. Note that the assumptions of section 2.2 ensure that $\psi_{s,t}(x_s, x_t^j)$ is normalizable for any x_t^j. The method by which this sampling step is performed will depend on the specific functional form of $\psi_{s,t}\left(x_s, x_t\right)$, and may involve importance sampling or MCMC techniques.

Having produced a set of independent samples from the desired output message $m_{ts}\left(x_s\right)$, NBP must choose a kernel bandwidth to complete the nonparametric density estimate. There are many ways to make this choice; for the results in this paper, we used leave–one–out likelihood cross–validation [14].

4 Gaussian Graphical Models

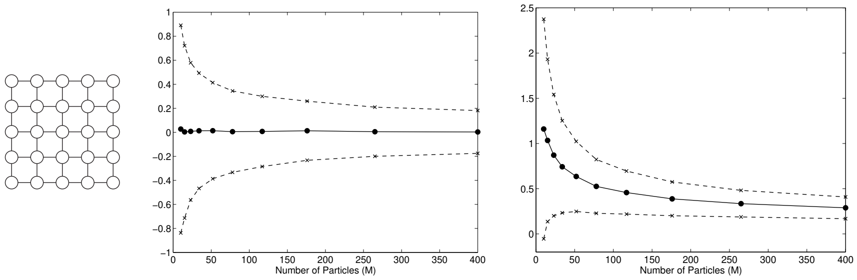
Gaussian graphical models provide one of the few continuous distributions for which the BP algorithm may be implemented exactly [9]. For this reason, Gaussian models may be used to test the accuracy of the nonparametric approximations made by NBP. Note that we cannot hope for NBP to outperform algorithms (like Gaussian BP) designed to take advantage of the linear structure underlying Gaussian problems. Instead, our goal is to verify NBP's performance in a situation where exact comparisons are possible.

We have tested the NBP algorithm on Gaussian models with a range of graphical structures, including chains, trees, and grids. Similar results were observed in all cases, so here we only present data for a single typical 5×5 nearest–neighbor grid, with randomly selected inhomogeneous potential functions. For each node $s \in \mathcal{V}$, Gaussian BP converges to a steady–state estimate of the marginal mean μ_s and variance σ_s^2 after about 15 iterations. To evaluate NBP, we performed 15 iterations of the NBP message updates using several different particle set sizes $M \in [10, 400]$. We then found the marginal mean $\hat{\mu}_s$ and variance $\hat{\sigma}_s^2$ estimates implied by the final NBP density estimates. For each tested particle set size, the NBP comparison was repeated 100 times.

Using the data from each NBP trial, we computed the error in the mean and variance estimates, normalized so each node behaved like a unit–variance Gaussian:

$$\tilde{\mu}_s = \frac{\hat{\mu}_s - \mu_s}{\sigma_s} \qquad\qquad \tilde{\sigma}_s^2 = \frac{\hat{\sigma}_s^2 - \sigma_s^2}{\sqrt{2}\sigma_s^2} \tag{9}$$

Figure 2 shows the mean and variance of these error statistics, across all nodes and trials, for different particle set sizes M. The NBP algorithm always provides unbiased estimates of the conditional mean, but overly large variance estimates. This



Mean ($\tilde{\mu}_s$) Variance ($\tilde{\sigma}_s^2$)

Figure 2: NBP performance for a 5×5 grid with Gaussian potentials and observations. Plots show the mean (solid line) and standard deviation (dashed line) of the normalized error measures of equation (9), as a function of particle set size M.

bias, which decreases as more particles are used, is due to the smoothing inherent in kernel–based density estimates. As expected for samples drawn from Gaussian distributions, the standard deviation of both error measures falls as $M^{-1/2}$.

5 Stereo Vision

Stereo vision algorithms infer point correspondences between horizontally aligned image pairs of the same three–dimensional scene [4, 20]. The distance, or disparity, between matching pixels is inversely proportional to the corresponding object's distance from the camera. Estimation of accurate, dense disparity maps is complicated by textureless regions, pixelization effects, and occlusions. For these reasons, prior knowledge about disparity variations plays a critical role in stereo computation.

The prior assumptions underlying stereo vision algorithms are often represented by grid–structured graphical models in which pairwise potentials correlate neighboring disparity values, while observation potentials measure similarities between local image features [5, 20]. In [4], three types of potential functions are suggested, each encoding different assumptions about the underlying disparity "worlds":

World I Pairwise potentials assume disparity differences are Gaussian. Similarly, observation potentials assign Gaussian distributions to image feature differences. Note that the resulting disparity likelihoods are *not* Gaussian.

World II The Gaussian potentials of World I are augmented by binary outlier processes to model occlusions and depth discontinuities. We instead use Gaussians contaminated by uniform noise, as suggested by [21]. A similar model has been used with the discrete BP algorithm [5].

World III Observation potentials are as in World II, but the hidden variable at each node is augmented to model horizontal and vertical disparity gradients. Uniform–contaminated Gaussian potentials model differences in adjacent gradients.

Our goal is not to present state–of–the–art stereo results, but to explore the characteristics of the NBP algorithm using the unique features of each world.

5.1 Worlds I & II

Figure 3 compares the NBP algorithm's performance to discretized BP on two 50×50 patches containing interesting disparity features. The patches were extracted from the standard "Venus" and "Sawtooth" stereo test images [20]. Discrete BP

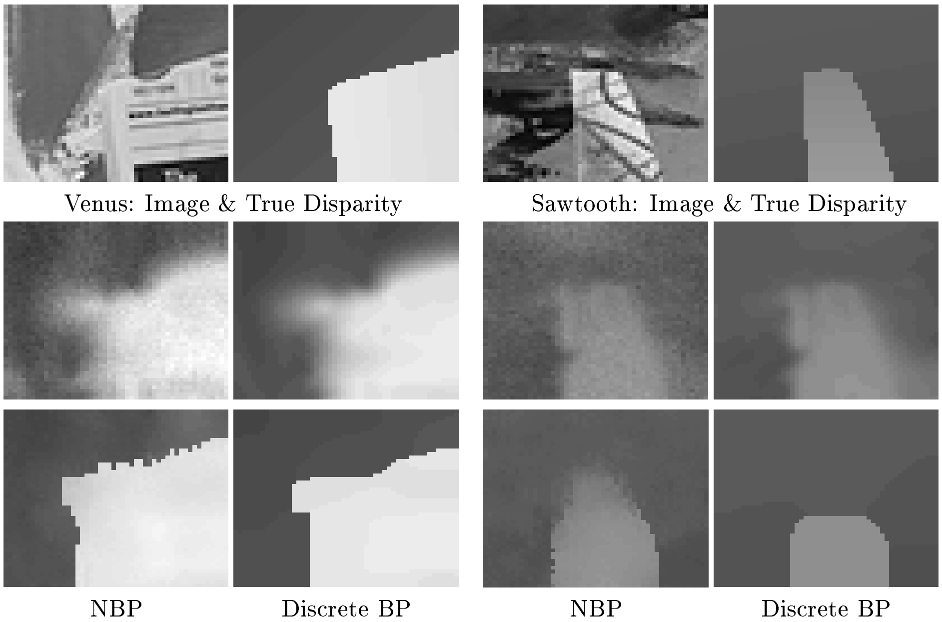


Figure 3: NBP and discrete BP stereo results for the "Venus" and "Sawtooth" patches. Top row: reference image and true disparities. Second row: World I. Third row: World II.

quantized the one–dimensional disparity space to 100 levels, while NBP used 100 particles sampled with 100 MCMC iterations. Because of the limited range of possible disparities, discrete BP's results approximate the true continuous BP results extremely accurately. We see that NBP accurately captures the qualitative features of both worlds, matching the discrete BP results extremely well.

5.2 World III

All of the objects in the Venus and Sawtooth images have slopes which are nearly parallel to the image plane. For this reason, the gradient information used in World III's sophisticated prior does not significantly modify the World II results. We have constructed a synthetic stereo problem, shown in Figure 4, consisting of a single constant slope scan line. Both ends of the scan line contain good observations, but the central section contains no disparity information, as would be caused by a textureless image region. Figures 4(b,c) show that World II's bias towards horizontal surfaces is inappropriate for this data set. Figure 4(d) shows performance attained by discrete BP with slopes chosen to exactly match the true surface. However, in practical situations it is not possible to discretize the three–dimensional World III state space finely enough to achieve this. Figure 4(e) shows the heavy quantization artifacts produced by discrete BP using 1000 ($40 \times 5 \times 5$) points. In contrast, using only 100 points, NBP produces the high–resolution density estimates of Figure 4(f).

6 Discussion

We have developed a nonparametric sampling–based variant of the belief propagation algorithm for graphical models with continuous, non–Gaussian random variables. Our stereo vision results suggest that NBP achieves performance comparable

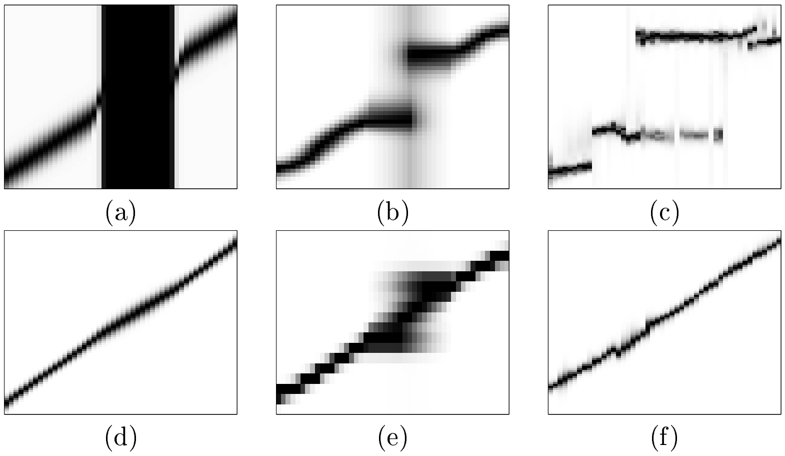


Figure 4: Stereo results for a synthetic scan line. Each plot is a columnwise representation of the nodes' marginal distributions. (a) Observation likelihoods. (b) World II discrete BP. (c) World II NBP. (d) World III ideal. (e) World III discrete BP. (f) World III NBP.

to discretization for low–dimensional hidden variables, and may offer significant advantages in higher dimensional spaces. In the future, the application of more sophisticated Monte Carlo and density estimation techniques could improve the statistical accuracy and computational efficiency of the message updates. We hope that the nonparametric approach will allow richer, more realistic models to be designed and used in applications in which continuous variables naturally arise.

Acknowledgments

This research was supported in part by ONR under Grant N00014-00-1-0089, by AFOSR under Grant F49620-00-1-0362, and by an ODDR&E MURI funded through ARO Grant DAAD19-00-1-0466. E.S. supported by an NDSEG fellowship.

References

[1] J. S. Yedidia, W. T. Freeman, and Y. Weiss. Understanding belief propagation and its generalizations. In *IJCAI*, August 2001.

[2] W. T. Freeman, E. C. Pasztor, and O. T. Carmichael. Learning low–level vision. *IJCV*, 40(1):25–47, 2000.

[3] S. Geman and D. Geman. Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE Trans. PAMI*, 6(6):721–741, November 1984.

[4] P. N. Belhumeur. A Bayesian approach to binocular stereopsis. *IJCV*, 19(3):237–262, 1996.

[5] J. Sun, H. Shum, and N. Zheng. Stereo matching using belief propagation. In *ECCV*, pages 510–524, 2002.

[6] J. Pearl. *Probabilistic Reasoning in Intelligent Systems*. Morgan Kaufman, San Mateo, 1988.

[7] G. Cooper. The computational complexity of probabilistic inference using Bayesian belief networks. *Artificial Intelligence*, 42:393–405, 1990.

[8] M. J. Wainwright, T. Jaakkola, and A. S. Willsky. Tree–based reparameterization for approximate inference on loopy graphs. In *NIPS 14*. MIT Press, 2002.

[9] Y. Weiss and W. T. Freeman. Correctness of belief propagation in Gaussian graphical models of arbitrary topology. *Neural Comp.*, 13:2173–2200, 2001.

[10] A. H. Jazwinski. *Stochastic Processes and Filtering Theory.* Academic Press, New York, 1970.

[11] D. L. Alspach and H. W. Sorenson. Nonlinear Bayesian estimation using Gaussian sum approximations. *IEEE Trans. AC*, 17(4):439–448, August 1972.

[12] A. Doucet, N. de Freitas, and N. Gordon, editors. *Sequential Monte Carlo Methods in Practice*. Springer-Verlag, New York, 2001.

[13] E. Parzen. On estimation of a probability density function and mode. *Ann. of Math Stats.*, 33:1065–1076, 1962.

[14] B. W. Silverman. *Density Estimation for Statistics and Data Analysis.* Chapman & Hall, London, 1986.

[15] S. L. Lauritzen. *Graphical Models.* Oxford University Press, Oxford, 1996.

[16] A. P. Dawid, U. Kjærulff, and S. L. Lauritzen. Hybrid propagation in junction trees. In *Advances in Intelligent Computing*, pages 87–97, 1995.

[17] D. Koller, U. Lerner, and D. Angelov. A general algorithm for approximate inference and its application to hybrid Bayes nets. In *UAI 15*, pages 324–333, 1999.

[18] U. Kjærulff. HUGS: Combining exact inference and Gibbs sampling in junction trees. In *UAI 11*, pages 368–375, 1995.

[19] L. D. Hernández and S. Moral. Mixing exact and importance sampling propagation algorithms in dependence graphs. *Int. J. Intelligent Systems*, 12:553–576, 1997.

[20] D. Scharstein and R. Szeliski. A taxonomy and evaluation of dense two–frame stereo correspondence algorithms. *IJCV*, 47(1):7–42, 2002.

[21] M. J. Black and A. Rangarajan. On the unification of line processes, outlier rejection, and robust statistics with applications in early vision. *IJCV*, 19(1):57–91, 1996.