Max-Product Particle Belief Propagation

Rajkumar Kothapa
Department of Computer Science
Brown University
Providence, RI 02915
r0jkumar@cs.brown.edu

Jason Pacheco
Department of Computer Science
Brown University
Providence, RI 02915
pachecoj@cs.brown.edu

Erik B.Sudderth(Advisor)
Department of Computer Science
Brown University
Providence, RI 02915
sudderth@cs.brown.edu

Abstract

Particle belief propagation has been successfully applied to efficiently compute marginal distributions in graphical models defined over random variables with continuous domains [1]. We show the application of the max-product particle belief algorithm to compute the MAP-configuration. We apply the max-product particle belief propagation (MP-PBP) algorithm to the stereo vision problem and compare its performance with the discretized version of loopy belief propagation. We show that MP-PBP yields similar performance to the discrete version with fewer particles.

1 Introduction

Graphical models provide a framework to model the joint probability distribution of many random variables. Nodes are used to model variables in the joint probability distribution and the edges express probabilistic relationship between the variables. Message passing algorithms can then be used to compute quantities of interest like the marginal distributions of all or a subset of all the variables in the graph and the MAP-configuration, i.e, the setting of the variables in the graphical model that maximizes the joint probability distribution of all the variables in the model.

Belief propagation is a message passing algorithm used to draw inference on graphical models. The sum-product version of belief propagation computes the marginal distribution of each variable in the model and the max-product version computes the MAP-configuration. When variables are low-dimensional and the state space of the variables is discrete, loopy belief propagation can be used to efficiently compute quantities of interest like the marginal distributions all the variables. When the variables are continuous, it is necessary to either discretize the state space of the variables or approximate the messages in some form. Sudderth et al. [2] showed that each message can be represented as a collection of samples and such messages can be used to run inference algorithms on graphical models with continuous state spaces. Particle filtering on the other hand discretizes the

continuous state space of the variables using a set of particles but has only been applied to Markov chains [3].

When the variables in the graphical model are univariate, it is possible to compare the results of sum-product version of particle belief propagation(PBP) with the discretized version of belief propagation. Ihler et al. [1] showed that L1 errors between the marginal distributions computed by PBP and those of the discretized version decrease as the number of particles are increased. Sampling methods that were considered include sampling from the local observation potential, sampling from the computed discretized marginals and sampling from the current estimated belief at each node.

It is natural to consider the max-product version of PBP to determine the MAP-configuration. Max-Product loopy belief propagation (MP-LBP) has been successfully applied to graphs with loops for discrete variables [4]. In the case where the variables are low-dimensional and are continous or have a large state space, it becomes possible to discretize the state space using a finite set of particles. For such small problems, discretizing the state space enables us to compare the performance of MP-PBP with MP-LBP, vis-a-vis the MAP-configuration. Ensuring good peformance of MP-PBP in comparison to MP-LBP for low-dimensional problems allows us to generalize MP-PBP for graphical models with continuous state spaces but having large dimenions where MP-LBP computations become intractable.

In this paper we analyze the performance of sampling the particles at each node using Markov chain Monte Carlo methods like Metropolis-Hastings algorithm. We apply the Metropolis Hastings sampling method to MP-PBP to compute the MAP-configuration. We compare the MAP-configuration found by PBP with those found by MP-LBP. Finally, we compare the results of MP-PBP with Simulated Annealing, a probabilistic meta-heuristic used to find a good approximation to a globally optimum MAP-configuration [5, 6].

2 Background

2.1 Notation

Let G be a graphical model with vertices V and edges E. Let $\Gamma(v)$ be the neighbors of vertex v in the graph. Let x_v be the random variable associated with the node v with continuous state space $\chi_{\mathbf{v}}$. For simplicity sake, we assume the joint probability distribution of all variables in the graph factorizes as a pairwise MRF :

$$p(\mathbf{X}) = \prod_{v \in V} \varphi_v(x_v) \prod_{\{u,v\} \in E} \varphi_{u,v}(x_u, x_v). \tag{1}$$

In equation (1), $\varphi_v(x_v)$ represents the node specific local observation potential and $\varphi_{u,v}(x_u, x_v)$ represents the pairwise potential function for edge $\{u, v\}$. A sample pairwise MRF is shown in Figure 1.

2.2 Sum-Product Particle Belief Propagation (PBP)

For graphical models with small, discrete state spaces, sum-product algorithm can be used to efficiently compute marginal distributions of the variables in the model. The sum-product messages in such graphical models are given by the equation :

$$m_{u \to v}^{(t)}(x_v) = \sum_{x_u \in \chi_{\mathbf{u}}} \left[\varphi_u(x_u) \varphi_{u,v}(x_u, x_v) \Big(\prod_{w \in \Gamma(u) \setminus v} m_{w \to u}^{(t-1)}(x_u) \Big) \right] \tag{2}$$

When the variables are continuous, one option is to discretize the continuous state space of each variable using a finite set of particle locations as in PBP [1]. In PBP, a finite set of particles for each node x_u are sampled from a proposed distribution $W_u(x_u)$ for that node . The particles thus sampled at each node are used to approximate the sum-product message in equation (2) as follows :

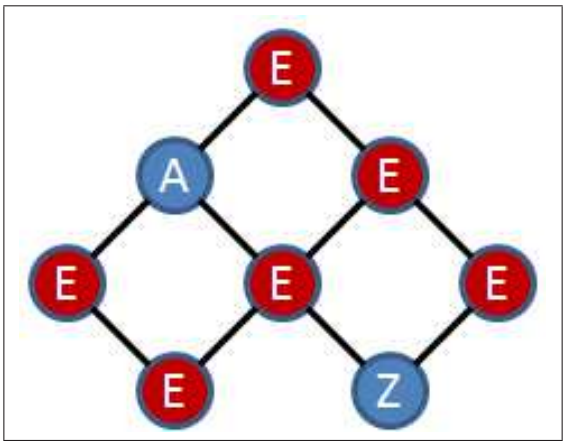

Figure 1: A pairwise MRF where any two non-adjacent random variables are conditionally independent given all other variables in the model. For example, the random variable A is conditionally independent of Z given all the other nodes. [https://facwiki.cs.byu.edu/cs677sp10/index.php]

$$m_{u\to v}^{(t)}(x_v) = \frac{1}{M}\sum_{i=1}^{M}\left[\frac{\varphi_u(x_u^{(i)})}{W_u(x_u^{(i)})}\varphi_{u,v}(x_u^{(i)}, x_v)\Big(\prod_{w\in\Gamma(u)\setminus v} m_{w\to u}^{(t-1)}(x_u^{(i)})\Big)\right] \tag{3}$$

Given the messages in equation (3), we can compute the current estimated belief at node x_v as follows [1] :

$$Q_v(x_v) \propto \varphi_u(x_v)\Big(\prod_{w\in\Gamma(v)} m_{w\to v}^{(t)}(x_v)\Big) \tag{4}$$

2.3 Max-Product Belief Propagation

Given the potential functions in equation (1), messages can be passed along the edges of the graph to compute the MAP-configuration. Let $m_{u\to v}^{(t)}(x_v)$ be the max-product message from graph node u to node v at timestep t as a function of x_v. Then the max-product message update equation is given by:

$$m_{u\to v}^{(t)}(x_v) = \max_{x_u\in\chi_{\mathbf{u}}}\left[\varphi_u(x_u)\varphi_{u,v}(x_u, x_v)\Big(\prod_{w\in\Gamma(u)\setminus v} m_{w\to u}^{(t-1)}(x_u)\Big)\right] \tag{5}$$

The message above is a continuous function of the state space of the random variable x_v and can be evaluated at any particular particle location x_v of $\chi_{\mathbf{u}}$. Let $\mu_v(x_v)$ be the max-marginal of x_v defined as :

$$\mu_v(x_v) = \max_{x'|x'_v=x_v} p(x'_1, x'_2, ...x'_N) \tag{6}$$

Messages computed in equation(5) can then be used to compute the above max-marginal using the equation :

$$\mu_v(x_v) \propto \varphi_v(x_v)\Big(\prod_{w\in\Gamma(v)} m_{w\to v}^{(t)}(x_v)\Big) \tag{7}$$

For a graph that is tree (cycle-free), the above max-marginals are exact [7]. For a graph with cycles, the max-marginals computed are approximate. Given these max-marginals, a backtracking procedure can be used to compute the MAP-configuration $\hat{x}$ such that $\hat{x} \in \arg\max_{x'} p(x'_1, x'_2, ...x'_N)$ of the distribution [8].

3 Max-Product PBP

We describe here the max-product version of particle belief propagation. MP-PBP maintains a set of particles $\{x_u^{(1)}, x_u^{(2)} \dots x_u^{(M)}\}$ at every node x_u of the graph. Equation (5) can be approximated using the current discrete set of particle locations of node x_u as follows :

$$m_{u\to v}^{(t)}(x_v) = \max_{\{x_u^{(j)}: j=1,2....M\}} \left[\varphi_u(x_u^{(j)}) \varphi_{u,v}(x_u^{(j)}, x_v) \Big(\prod_{w \in \Gamma(u)\setminus v} m_{w\to u}^{(t-1)}(x_u^{(j)}) \Big) \right] \tag{8}$$

Note the both the max-product message from node x_u to node x_v given by equation (5) and the max-marginal at node x_v given by equation (7) are both continuous functions of the state space of node x_v and hence can be evaluated for the current discrete set of particle locations at x_v. The max-marginal at node x_v for a particle location $x_v^{(k)}$ is given by the equation :

$$\mu_v(x_v^{(k)}) \propto \varphi_v(x_v^{(k)}) \Big(\prod_{w \in \Gamma(v)} m_{w\to v}^{(t)}(x_v^{(k)}) \Big) \tag{9}$$

At the end of every iteration, the current set of particle locations $\{x_u^{(1)}, x_u^{(2)} \dots x_u^{(M)}\}$ at every node x_u are updated based on the sampling distributions used. We say more about this in Section 3.1.

Note that the key difference between equation (3) and equation (8) is that the message update in (3) requires that we reweight the message by the corresponding proposal distribution at node x_u. This reweighting is necessary since we need to correct for the use of a biased proposal distribution for PBP. This reweighting is removed in equation (8) because of the application of the max-operator in the message update equation. This is further illustrated in Figure 2.

3.1 Sampling Distributions

An important thing to consider for MP-PBP is the sampling distribution for each node x_u. Koller et al. [9] suggested that a good choice would be to use the true sum-product marginal at node x_u. Since we do not have the true sum-product marginal to sample from, we can instead sample from the current estimated marginal given by equation (4). For MP-PBP, we need to sample from the current estimated max-marginals and we consider 2 choices : 1) Sample from the discretized current estimated max-marginals obtained at each node on a regular dense grid by keeping the particles at its neighbors fixed. Note that this is only possible when the variable x_u is low-dimensional. 2) Run a short MCMC simulated algorithm at each node at every iteration of MP-PBP similar to the approach used in particle filters [10]. We describe the MCMC simulation in more detail in Section 3.2.

For illustration purposes, we also consider sampling from the local observation potential and the discretized true max-marginals obtained from LBP with no resampling. Note that sampling from the discretized true max-marginals is only possible in the univariate case since LBP can be used to compute the true max-marginals on a regular dense grid.

3.2 Sampling using Metropolis-Hastings for MP-PBP

We consider the affect of sampling particles for PBP using Metropolis-Hastings algorithm. The marginal distribution of each node x_v given by equation (7) is difficult to sample from but easy to evaluate up to a normalization constant at any given particle location. We maintain a set of particle locations $\{x_u^{(1)}, x_u^{(2)}, \dots, x_u^{(M)}\}$ at each node x_u of the graph at every iteration of PBP. At each iteration, we perturb each particle location by running a short MCMC simulation using the Metropolis-Hastings algorithm. There are two important things to consider while running Metropolis-Hastings.

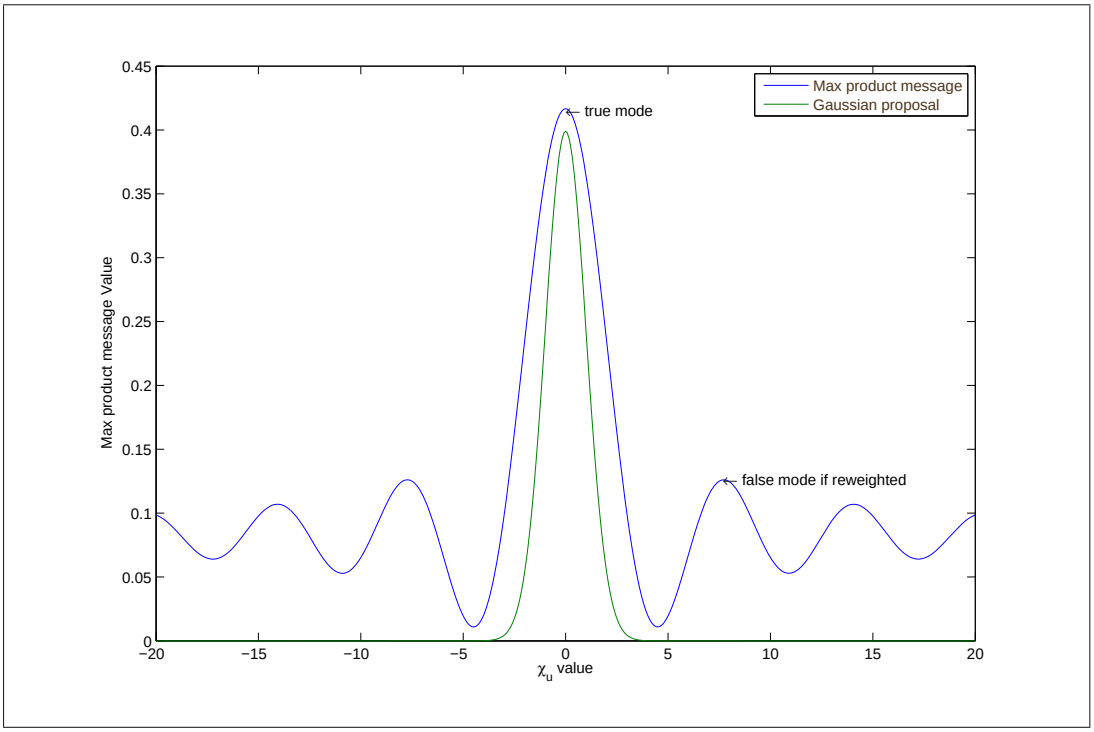

Figure 2: Plot of a sample true max-product message from node x_u to node x_v as a function of the state space of node x_u for a fixed value of node x_v along with the Gaussian proposal distribution from where the particles of node x_u are sampled. The max-product message from node x_u to node x_v is evaluated at each particle location of node x_u sampled from the Gaussian proposal distribution. If the max-product were reweighted by the Gaussian proposal, then the max-product message would incorrectly place more mass in particle locations that are in the tails of the Guassian distribution as shown in the above figure and vice versa.

First is the choice of the proposal distribution to use for the random walk. We propose a random walk for each particle location at each node using a Gaussian distribution that is centered at that particle location. Second is the choice of the bandwidth parameter(σ) for the Gaussian proposal distribution. The bandwidth parameter is set so that acceptance rate for Metropolis-Hastings perturbations is neither too high or too long. We say more about this in Section 4.1.

The issues considered while applying Metropolis-Hastings sampling method to MP-PBP are similar to those for the sum product version of particle belief propagation. We note the two main differences between using Metropolis-Hastings sampling for PBP and MP-PBP : the target distribution for MP-PBP is the current estimated max-marginals given by equation (7) as opposed to the current estimated belief given by equation (4) for PBP; and the MCMC simulation for MP-PBP needs fewer proposal distributions in comparison to PBP because of the absence of re-weighting in the message update equation (5) and as illustrated in Figure 2.

The details of the implementation for the Metropolis-Hastings sampling method are given in Figure 3.

Let N be the total number of nodes in the graph and let M be the number of particles used at every node. Let T be the number of PBP iterations to use.
Let P be the number of proposal distributions of Metropolis to use in each iteration of PBP.
Let σ be the standard deviation of the proposal distribution of Metropolis.
Let $m_{u\to v}^{(t)}(x_v)$ be the message from graph node u to node v at timestep t as a function of x_v.
Let $Q^{(t)}(x_u)$ be the node belief at node x_u at timestep t.

1. For each $t = 1 : T$, do the following :
   (a) Compute max-product message on every edge based on the schedule as follows :
   $m_{u\to v}^{(t)}(x_v) =$
   $max_{\{x_u^{(j)}:j=1,2....M\}}\left[\varphi_u(x_u^{(j)})\varphi_{u,v}(x_u^{(j)},x_v)\Big(\prod_{w\in\Gamma(u)/v} m_{w\to u}^{(t-1)}(x_u^{(j)})\Big)\right]$
   (b) For $p = 1 : P$, do the following :
      i. For each node x_u in the graph :
         A. Perturb each particle location $\bar{x}_u^{(j)} \sim \mathcal{N}\Big(x|x_u^{(j)},\sigma\Big)$.
         B. Compute the node belief $Q^t(x_u) \propto \varphi_u(x_u)\Big(\prod_{w\in\Gamma(u)} m_{w\to u}^{(t)}(x_u)\Big)$.
         C. Evaluate the node belief in Part (b) at particle locations x_u and $\bar{x}_u$.
         D. Compute the acceptance probability for metropolis as $\max(1, \frac{Q^{(t)}(\bar{x}_u^{(j)})}{Q^{(t)}(x_u^{(j)})})$.

Figure 3: Pseudo-code showing the implementation of MP-PBP for Metropolis-Hastings sampling method.

4 Experimental Results

We evaluate the performance of PBP and MP-PBP on the stereo-vision problem, i.e the problem of estimating the most likely depth at each pixel in an image. To illustrate the performance of particle belief propagation on different kinds of images, we consider one image with largely uniform disparities and another with a more variation in disparities. These images are shown in Figures 4 and 5 respectively.

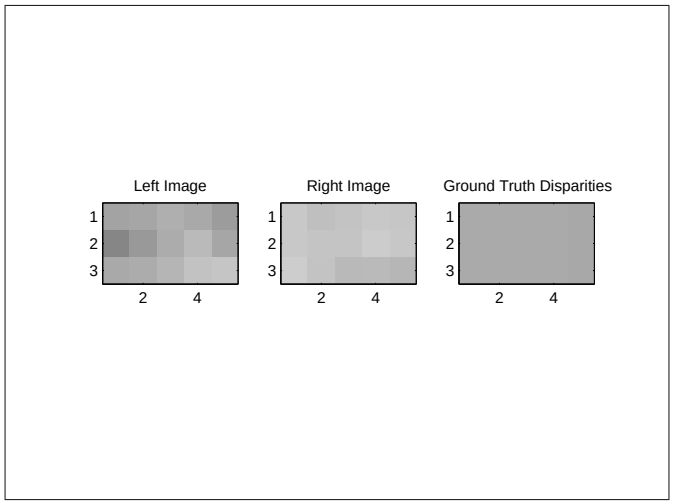

Figure 4: The left and right stereo vision images along with the ground truth dispartities for a small 3 by 5 image. This image has largely uniform disparities.

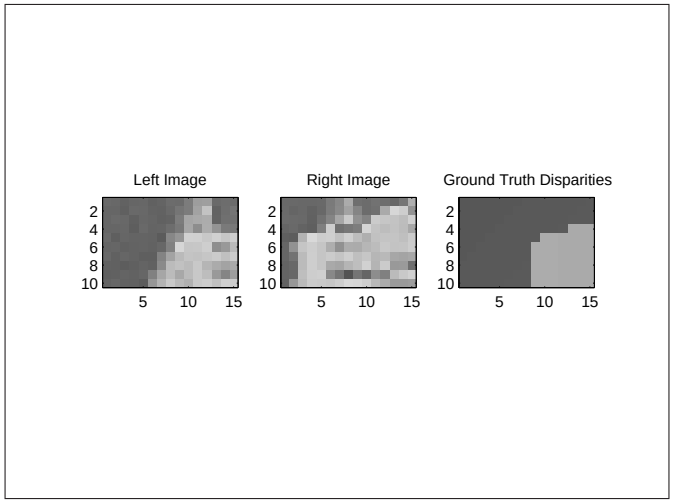

Figure 5: The left and right stereo vision images along with the ground truth dispartities for a larger 10 by 15 image. This image has more variation in disparities.

4.1 PBP with Metropolis-Hastings sampling

We evaluate the performance of PBP on the image shown in Figure 4 when sampling using the Metropolis-Hastings sampling method. We note the following about the Metropolis-Hastings sampling method for PBP : 200 proposal distributions were considered for each particle location at each node at every iteration of PBP. This is necessary in order to allow the MCMC simulation to converge and compute the correct weights $W_u(x_u)$ in equation (3); bandwidth parameter σ for the Gaussian proposal distribution was adjusted so that $25 - 50\%$ of the particle perturbations were accepted.

The L1 error plot is shown in Figure 6 for the image in Figure 4.

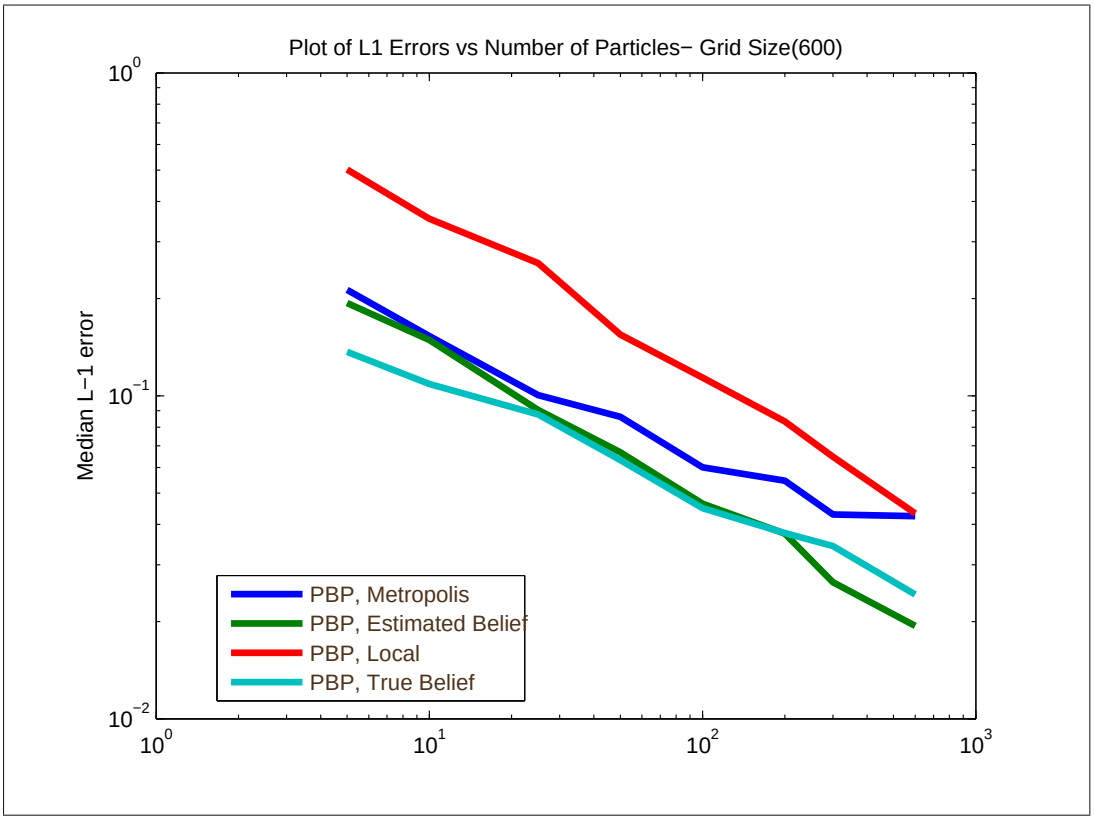

Figure 6: The above figure is a log-log plot of the L1 errors between the marginals computed by LBP on a dense regular grid and those computed by PBP smooted on the same grid for the image in Figure 4. This plot displays L1 errors averaged over 25 different runs of PBP. Notice that for all sampling methods, the L1 errors decrease with an increase in the number of particles. Also note that the L1 error plot when using the Metropolis-Hastings sampling method closely shadows the L1 error plot when sampling from the current estimated marginals.

4.2 MP-PBP

We evaluate the performance of MP-PBP on the images shown in Figures 4 and 5 with the following sampling methods :

- Sampling using the local observation potential once with no resampling.
- Sampling once from the true max-marginals computed by running MP-LBP with no resampling.
- Sampling from the current estimated max-marginals smooted on a regular dense grid.
- Sampling using Metropolis-Hastings method.

In order to illustrate the performance of MP-PBP against MP-LBP, we make the following plots:

- A plot showing the L1 error difference between the max-marginals computed by MP-PBP and MP-LBP.
- A log probability difference plot of the modes found by MP-PBP and MP-LBP.
- Plot showing the max-marginals computed by PBP versus the max-marginals computed by LBP for 200 particles.

L1 error plot and the log probability plot for the images in Figures 4 and 5 are shown in Figures 7, 8 and 9. The max-marginal plots for the images in Figures 4 and 5 are shown in Figures 10 and 11 respectively.

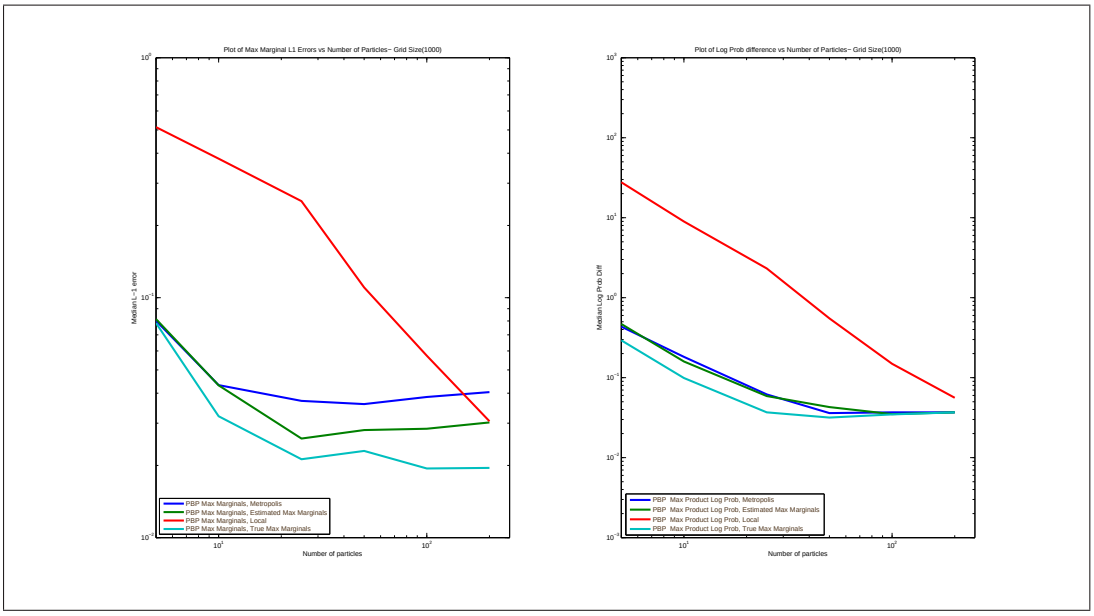

Figure 7: The figure on the left shows the log-log plot of the median L1 error between the max-marginals computed by MP-PBP and the max-marginals computed by LBP versus the number of particles used for different sampling methods for the image in Figure 4. LBP uses a regular grid of size 1000. The figure on the right shows the log-log plot of the log probability difference of the modes found by MP-PBP and MP-LBP for the same image. Note that the L1 errors for some sampling methods increases with an increase in the number of particles though the log probability difference decreases since MP-PBP is able to find a better mode when compared to MP-LBP.

4.3 Comparison with Simulated Annealing

We compare the performance of MP-PBP with that of Simulated Annealing with Metropolis-Hastings for different particle sizes. For a fair comparison, we use the same Gaussian proposal distribution as the one used for MP-PBP. The target probability distribution for Metropolis-Hastings is the joint probability distribution of all the variables raised to a temperature and is given by :

$$p(\mathbf{X}) \propto \prod_{v \in V} \Big(\varphi_v(x_v)\Big)^{\frac{1}{T}} \prod_{\{u,v\} \in E} \Big(\varphi_{u,v}(x_u, x_v)\Big)^{\frac{1}{T}}. \tag{10}$$

The details of the implementation are given in Figure 12.

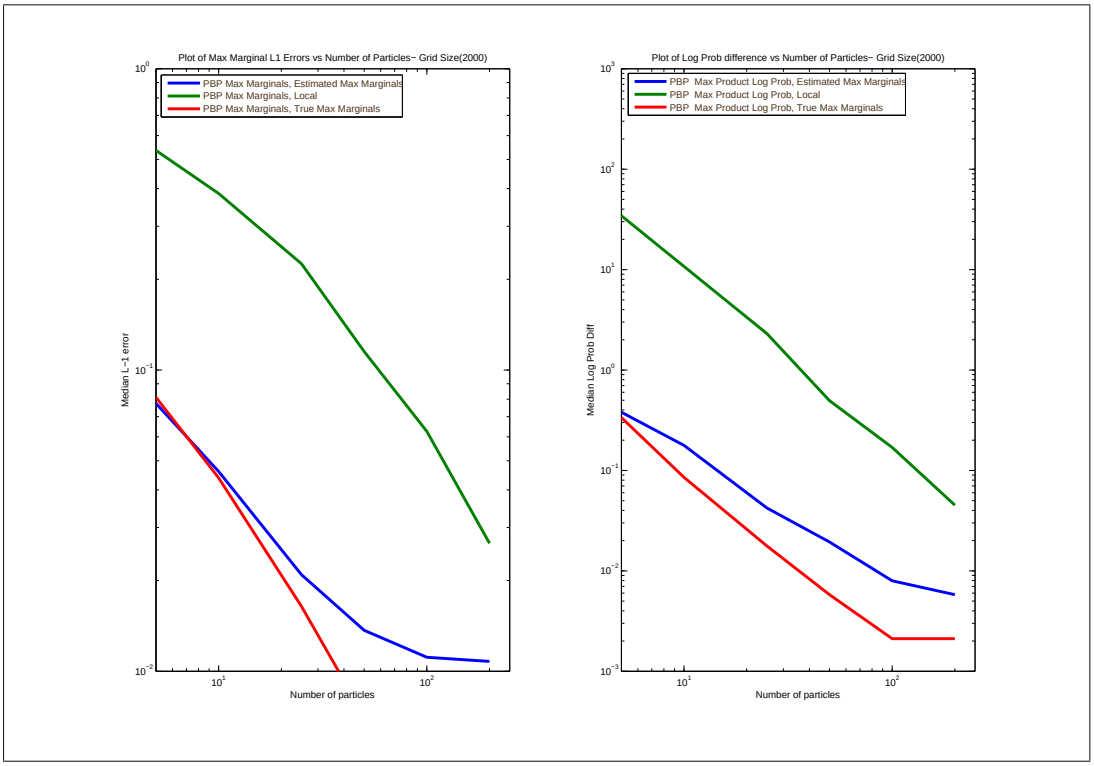

Figure 8: The figure on the left shows the log-log plot of the median L1 error between the max-marginals computed by MP-PBP and the max-marginals computed by LBP versus the number of particles used for different sampling methods for the image in Figure 4. LBP uses a regular grid of size 2000. The figure on the right shows the log-log plot of the log probability difference of the modes found by MP-PBP and MP-LBP for the same image. Note that the L1 errors for all sampling methods decreases with an increase in the number of particles and the log probabilities decrease since MP-PBP is able to find a better mode when compared to MP-LBP.

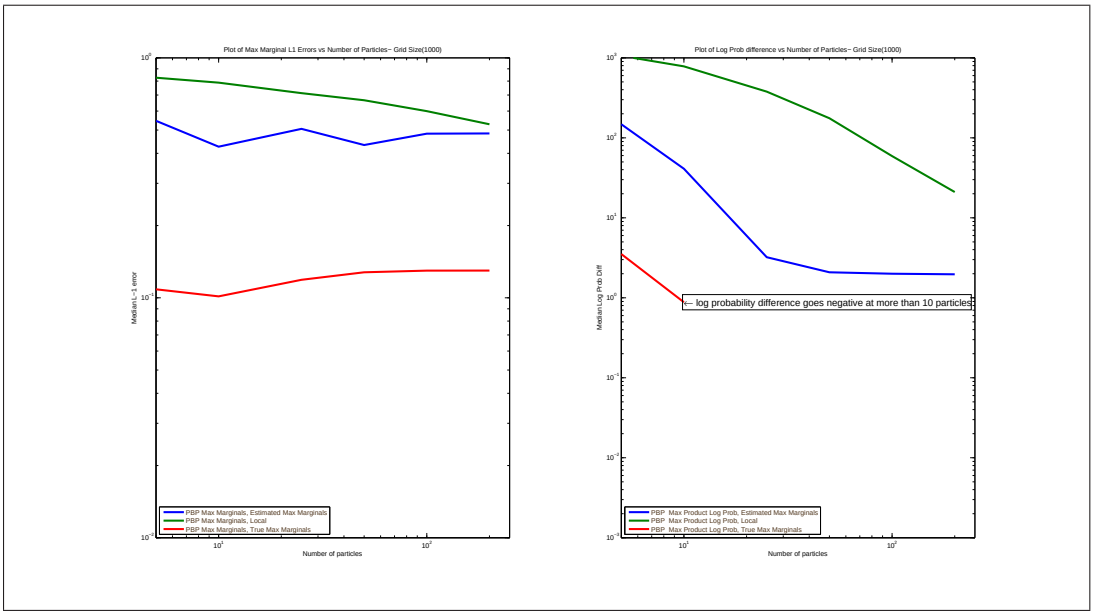

Figure 9: The figure on the left shows the log-log plot of the median L1 error between the max-marginals computed by MP-PBP and the max-marginals computed by LBP versus the number of particles used for different sampling methods for the image in Figure 5. LBP uses a regular grid of size 1000. The figure on the right shows the log-log plot of the log probability difference of the modes found by MP-PBP and MP-LBP for the same image. Note that the L1 errors for some sampling methods increases with an increase in the number of particles though the log probability difference decreases since MP-PBP is able to find a better mode when compared to MP-LBP. Also note that the log probability difference goes negative at more than 10 particles when sampling from the true max-marginals for the same reason.

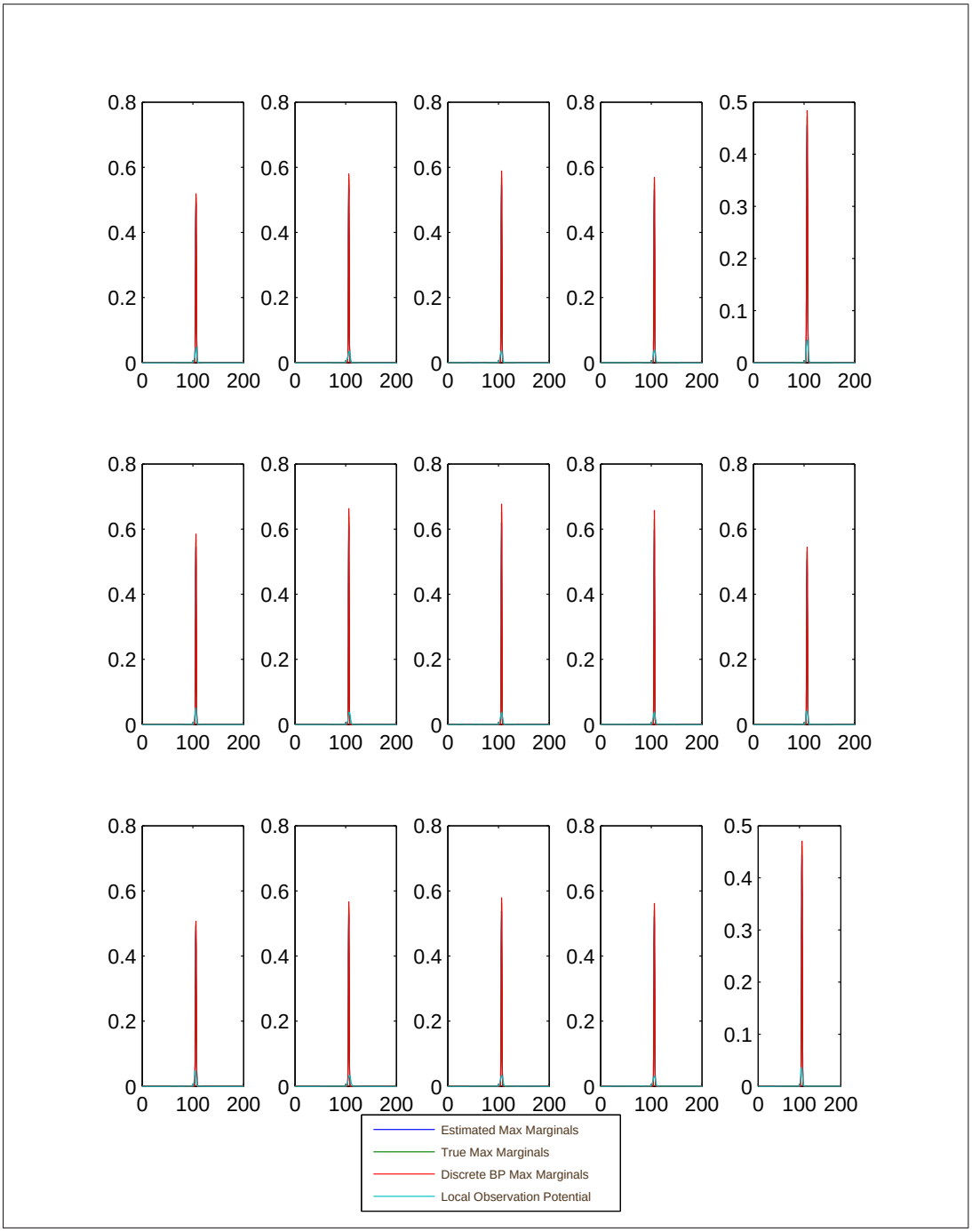

Figure 10: Plot comparing max-marginals computed by MP-PBP with max-marginals computed by LBP on the image in Figure 4. MP-LBP max-marginals are computed on regular grid of size 1000 and MP-PBP uses 200 particles and a regular grid of size 1000 for the smoothing step. We also show the normalized local observation potential in the same plot for illustration purpose. Note that the max-marginals computed by MP-PBP are almost the same as the ones computed by MP-LBP and both the max-marginals are consistent with the local observation potential at all nodes. The key thing to note is that MP-PBP finds the max-marginals consistent with MP-LBP with fewer particles.

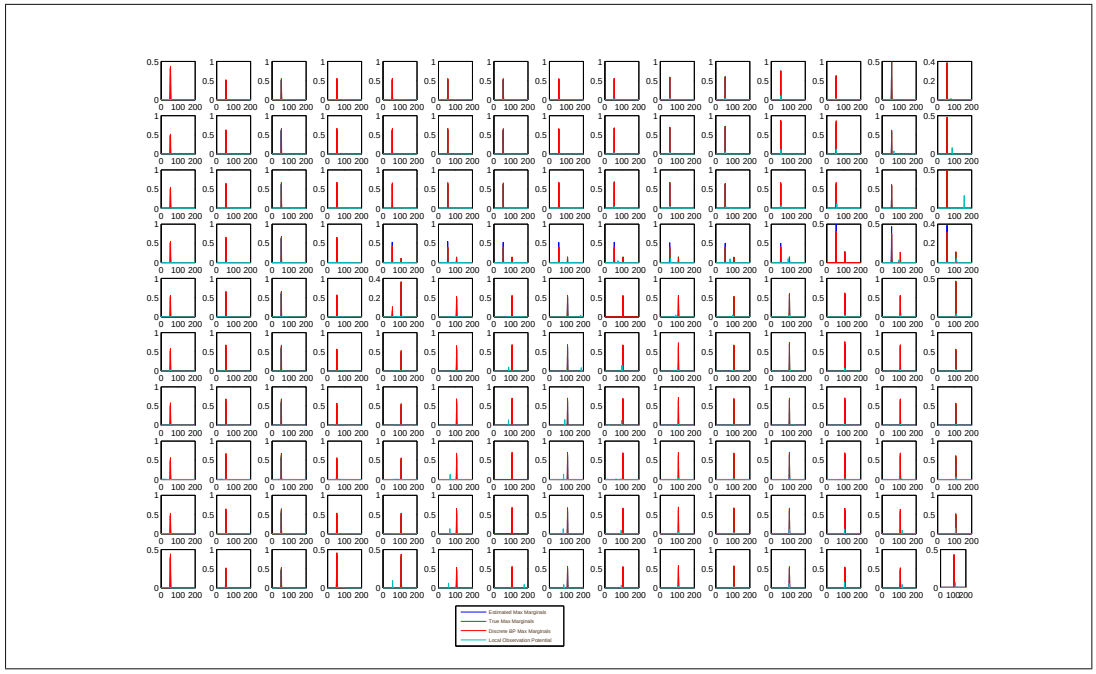

Figure 11: Plot comparing max-marginals computed by PBP with max-marginals computed by LBP on the image in Figure 5. MP-LBP max-marginals are computed on regular grid of size 1000 and MP-PBP uses 200 particles and a regular grid of size 1000 for the smoothing step. We also show the normalized local observation potential in the same plot for illustration purpose. The key difference between this figure and Figure 10 is that the max-marginals computed by MP-LBP is not necessarily consistent with the local observation potential at all nodes. This is because the edge compatibility function that forces adjacent nodes to have to the same depth overrides the local evidence at that node. Note that the max-marginals computed by PBP are consistent with the max-marginals computed by MP-LBP at all nodes. The key thing to note is that MP-PBP finds the max-marginals consistent with MP-LBP with fewer particles even for an image with variation in disparities.

Let T_{max} be the number of simulated annealing iterations to run.
Let P be the number of proposal distributions of Metropolis to use in each iteration of PBP.
Let σ be the standard deviation of the proposal distributions of Metropolis.

1. For each $i = 1 : T_{\text{max}}$, do the following :
   (a) Set the current temperature T as follows :
   $T = T_{\text{max}} - i + 1$
   (b) For $p = 1 : P$, do the following :
      i. For each node x_u in the graph :
         A. Perturb the particle location $\bar{x}_u^{(j)} \sim \mathcal{N}\Big(x | x_u^{(j)}, \sigma\Big)$.
         B. Compute the target probability distribution as a function of $\chi_{\mathbf{u}}$ using equation $p(\mathbf{X}) \propto \prod_{v \in V} \Big(\varphi_v(x_v)\Big)^{\frac{1}{T}} \prod_{\{u,v\} \in E} \Big(\varphi_{u,v}(x_u, x_v)\Big)^{\frac{1}{T}}$ keeping the particle locations for all other nodes fixed.
         C. Evaluate the target probability distribution in Part (B) at particle locations x_u and $\bar{x}_u$.
         D. Compute the acceptance probability for metropolis as $\max(1, \frac{P(\bar{x}_u^{(j)})}{P(x_u^{(j)})})$.

Figure 12: Pesudo-code for Simulated Annealing implementation.

5 Summary and Conclusions

In this paper we discussed the implementation of particle belief propagation with the Metropolis-Hastings sampling method. We compared its performance with other sampling techniques and show that it performs just as well as sampling from the discretized estimated marginals in the case when the variables are univariate. Further, we discussed the max-product version of particle belief propagation and its main differences with the sum-product version of belief propagation. We illustrated the performance of MP-PBP on the stero vision problem with various sampling methods including Metropolis-Hastings sampling method. We showed that for the univariate case, MP-PBP finds a mode with fewer number of particles compared to MP-LBP. We further showed that MP-PBP finds a better mode when compared to Simulated Annealing in signficantly fewer number of iterations

Running a MCMC simulation at every node at every iteration of MP-PBP is computionally intensive and often unsatisfactory for large models. Adaptive Metropolis-Hastings samplers like those described in [11] can be used to improve the performance of MCMC described in this paper. Further work in MP-PBP would include testing the performance of PBP Metropolis-Hastings sampling method when the variables in the model are multivariate.

Acknowledgments

I'd like to thank my advisor, Erik Sudderth, for his valuable insights and ideas. Erik encouraged me to think on my own and gain intuition for machine learning and statistics. I wish to thank Jason Pacheco for his valuable suggestions whenever i got stuck and his helpful comments while preparing this report.

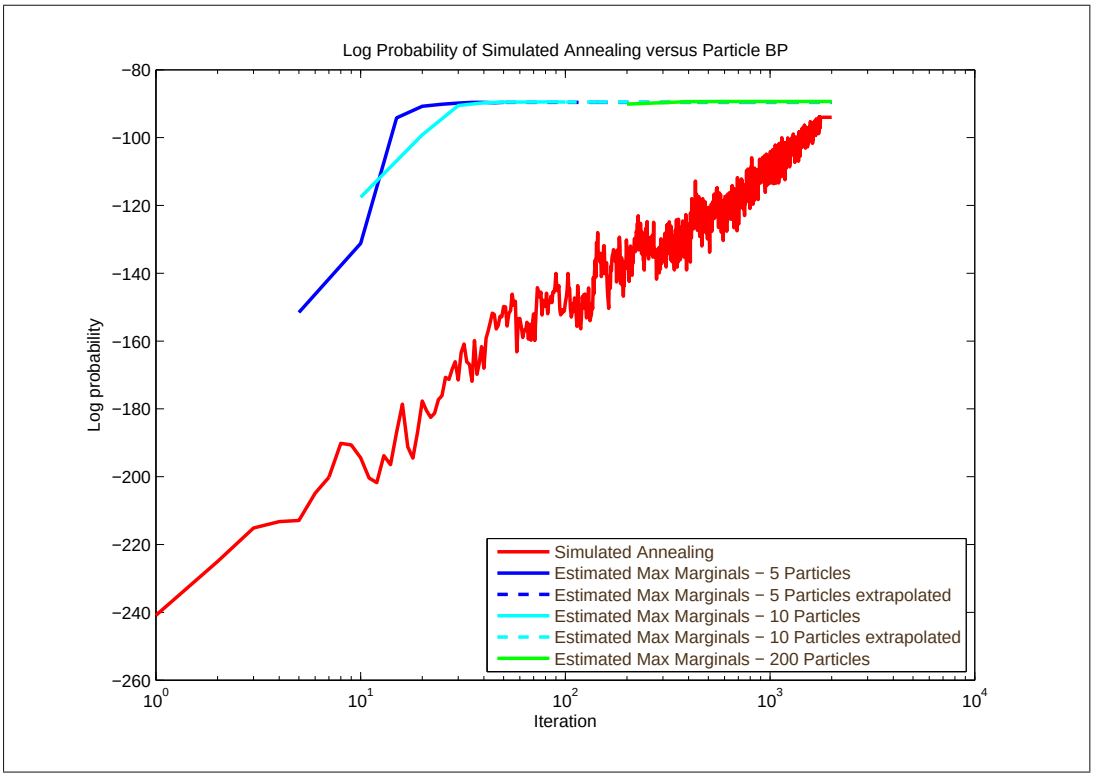

Figure 13: Plot of the log probabilities of the modes found by MP-PBP and Simulated Annealing versus the iteration number for the same proposal distributions and identical number of perturbations in each iteration for the image in Figure 4. Both MP-PBP and simulated annealing plots are averaged over 20 trials and MP-PBP uses a regular grid of size 1000 for smooting. Note that simulated annealing takes a significantly larger number of iterations to find a mode as good as the one found by MP-PBP.

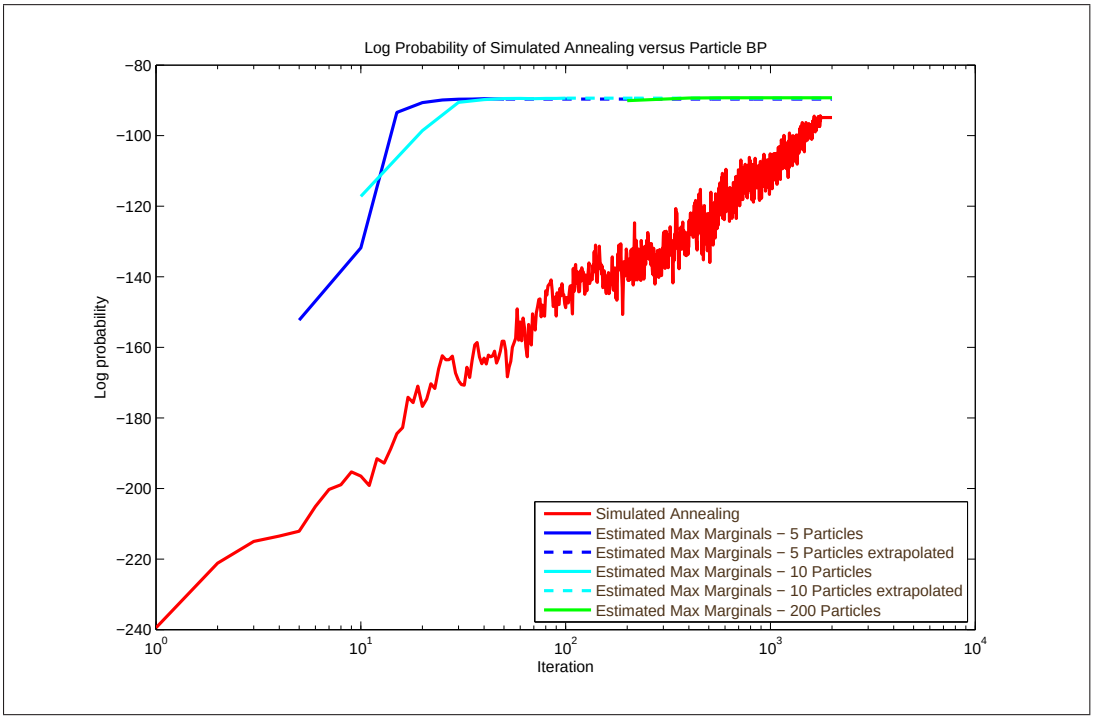

Figure 14: Plot of the log probabilities of the modes found by MP-PBP and Simulated Annealing versus the iteration number for the same proposal distributions and identical number of perturbations in each iteration for the image in Figure 4. Both MP-PBP and simulated annealing plots are averaged over 20 trials and MP-PBP uses a regular grid of size 2000 for smooting. Note that simulated annealing takes a significantly larger number of iterations to find a mode as good as the one found by MP-PBP.

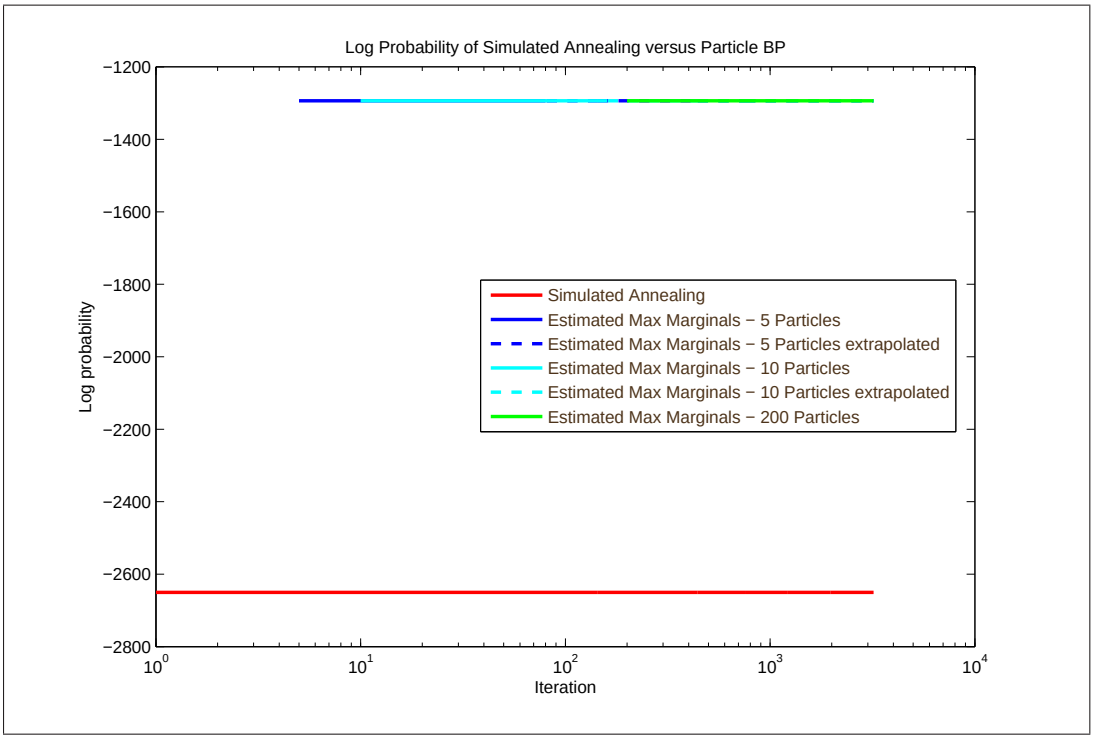

Figure 15: Plot of the log probabilities of the modes found by MP-PBP and Simulated Annealing versus the iteration number for the same proposal distributions and identical number of perturbations in each iteration for the image in Figure 5. Both MP-PBP and simulated annealing plots are averaged over 20 trials and MP-PBP uses a regular grid of size 1000 for smooting. Note that simulated annealing does significantly worse than MP-PBP in terms of find a mode and cannot find a better MAP-configuration with an increase in iterations for the image in Figure 5.

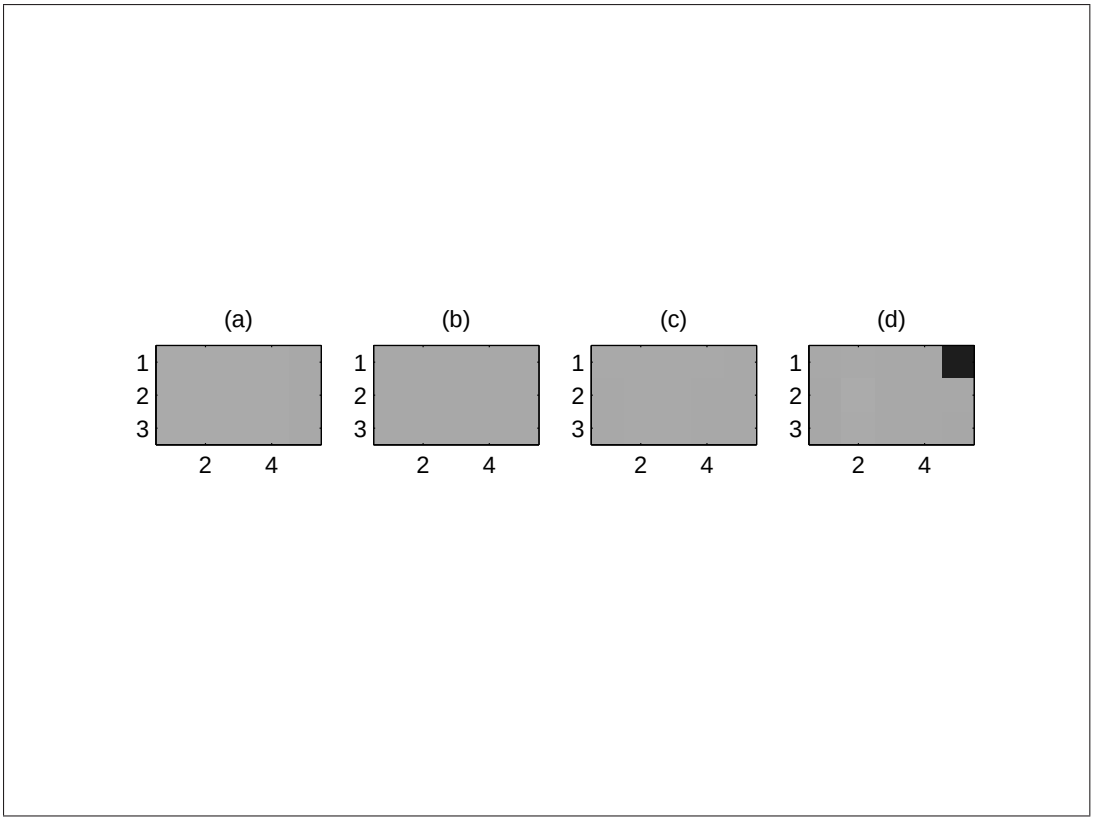

Figure 16: Images reconstructed using the MAP-configuration found by various belief propagation methods in one trial along with the ground truth disparities for the image in Figure 4. (a) Ground truth disparities. (b) MAP-configuration found by MP-LBP using 1000 states. (c) MAP-configuration found by MP-PBP using 200 particles at each node and a regular grid of size 1000 in the smoothing step. (d) MAP-configuration found by Simulated Annealing after 2000 iterations.

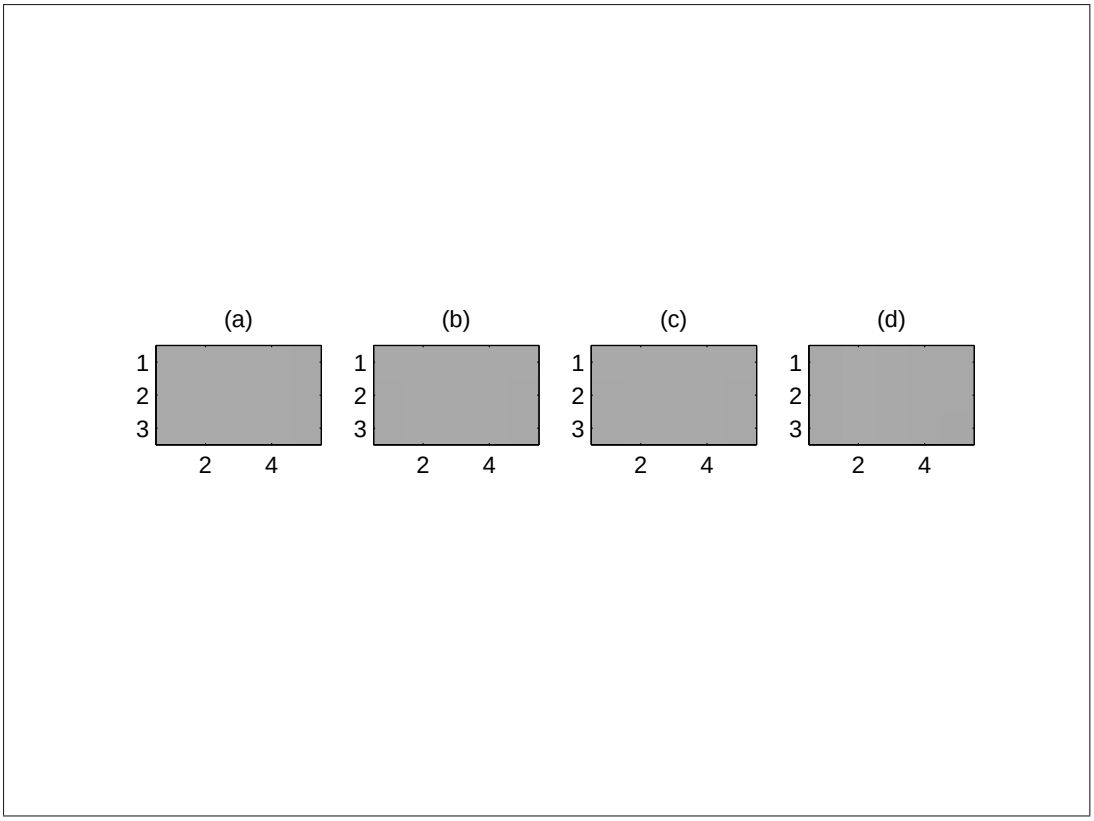

Figure 17: Images reconstructed using the MAP-configuration found by various belief propagation methods in one trial along with the ground truth disparities for the image in Figure 4. (a) Ground truth disparities. (b) MAP-configuration found by MP-LBP using 2000 states. (c) MAP-configuration found by MP-PBP using 200 particles at each node and a regular grid of size 2000 in the smoothing step. (d) MAP-configuration found by Simulated Annealing after 2000 iterations.

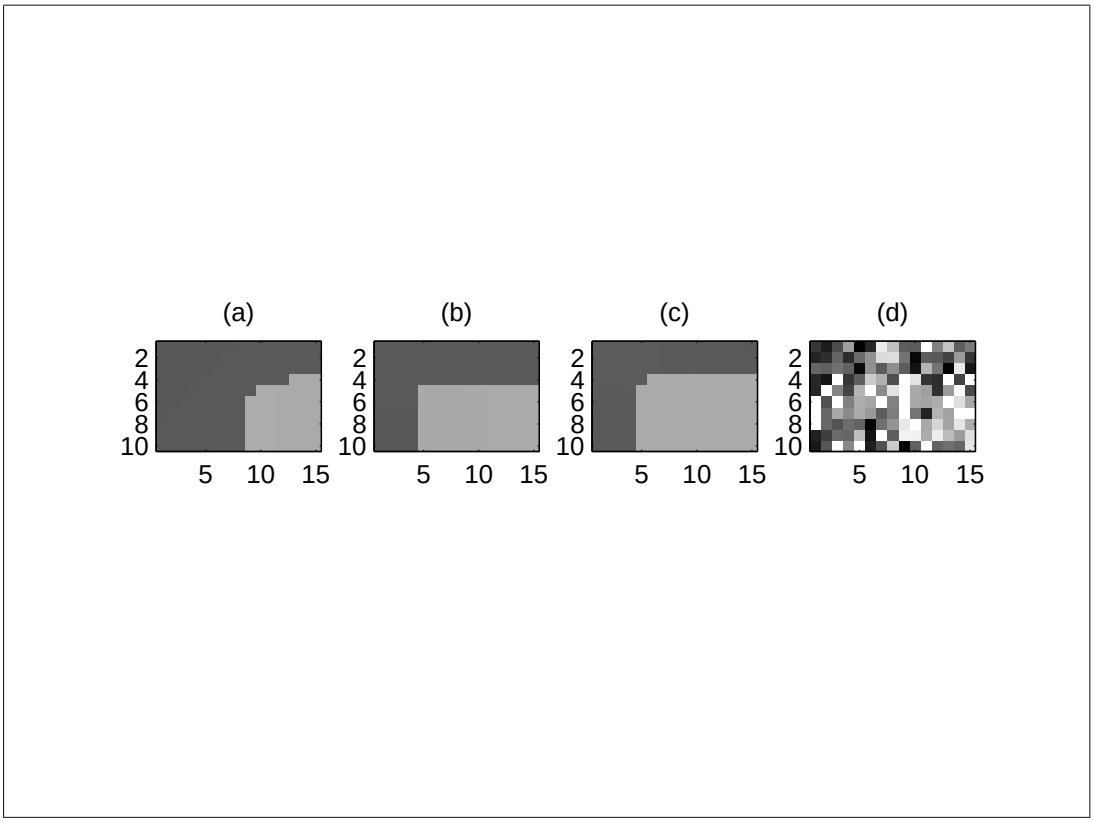

Figure 18: Images reconstructed using the MAP-configuration found by various belief propagation methods in one trial along with the ground truth disparities for the image in Figure 5. (a) Ground truth disparities. (b) MAP-configuration found by MP-LBP using 1000 states. (c) MAP-configuration found by MP-PBP using 200 particles at each node and a regular grid of size 1000 in the smoothing step. (d) MAP-configuration found by Simulated Annealing after 3200 iterations.

References

[1] Alexander Ihler & David McAllester (2009) Particle Belief Propagation *Proceedings of the Twelfth International Conference on Artificial Intelligence and Statistics (AISTATS).* , pp. 256-263. Clearwater Beach, Florida.

[2] E. Sudderth, A. Ihler, W. Freeman, and A. Willsky (2003) Nonparametric belief propagation. *In CVPR, 2003.*

[3] M. S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp (2002) A tutorial on particle filters for online nonlinear/nonGaussian Bayesian tracking. *IEEE Trans. SP, 50(2): 174-188.*

[4] McEliece, R.J., MacKay, D.J.C. & Jung-Fu Cheng (1998) Turbo decoding as an instance of Pearl's "Belief Propagation" Algorithm. *IEEE Journal on Selected Areas in Communications, 16 (2)*, pp. 140-152.

[5] Olivier Catoni (1996) Metropolis, Simulated Annealing, and Iterated Energy Transformation Algorithms: Theory and Experiments. *Journal of Complexity* **12**(4):595-623.

[6] Geman, S. and D. Geman (1984) Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE Trans. Pattern Analysis and Machine Intelligence 6: 721-741.*

[7] Judea Pearl (1988) Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference. *Morgan-Kaufmann Publishers, INC.*

[8] Martin J. Wainwright & Michael I. Jordan (2008) Graphical Models, Exponential Families, and Variational Inference. *Foundations and Trends in Machine Learning: Vol.1: No 1-2* , pp. 1-305.

[9] D. Koller, U. Lerner, & D. Anguelov (1999) A General Algorithm for Approximate Inference and Its Application to Hybrid Bayes Nets. *Proc. Fourteenth Annual Conference on Uncertainty in AI (UAI)*, pp. 324-333.

[10] Doucet, A., De Freitas, N. & Gordon (2001) Sequential Monte Carlo Methods in Practice. *Springer*.

[11] Stephen Kh Yeung & Darren J Wilkinson (2002) Adaptive Metropolis-Hastings samplers for the Bayesian analysis of large linear Gaussian systems. *Comp. Sci. Statist*,