

Particle-Based Belief Propagation for Structure from Motion and Dense Stereo Vision with Unknown Camera Constraints

Hoang Trinh and David McAllester

Toyota Technological Institute at Chicago
1427 E. 60th Street,
Chicago Illinois 60637
{ntrinh, mcallester}@tti-c.org
<http://www.tti-c.org>

Abstract. In this paper, we present a specific use of the Particle-based Belief Propagation (PBP) algorithm as an approximation scheme for the joint distribution over many random variables with very large or continuous domains. After formulating the problem to be solved as a probabilistic graphical model, we show that by running loopy Belief Propagation on the whole graph, in combination with an MCMC method such as Metropolis-Hastings sampling at each node, we can approximately estimate the posterior distribution of each random variable over the state space. We describe in details the application of PBP algorithm to the problem of sparse Structure from Motion and the dense Stereo Vision with unknown camera constraints. Experimental results from both cases are demonstrated. An experiment with a synthetic structure from motion arrangement shows that its accuracy is comparable with the state-of-the-art while allowing estimates of state uncertainty in the form of an approximate posterior density function.

Keywords: Belief Propagation, Particle filter, Structure from Motion, Dense Stereo Vision.

1 Introduction

Graphical models are considered a powerful tool to represent structures in distributions over many random variables. Such a structure can then be used to efficiently compute or approximate many quantities of interest such as the posterior modes, means, or marginals of the distribution, often using “message-passing” algorithms such as belief propagation [1]. Traditionally, most such work has focused on systems of many variables, each of which has a relatively small state space (number of possible values), or particularly nice parametric forms (such as jointly Gaussian distributions). For systems with continuous-valued variables, or discrete-valued variables with very large domains, one possibility is to reduce the effective state space through gating, or discarding low-probability states [2,3,4], or through random sampling [5,6,7,8,9]. The best-known example of the latter technique is *particle filtering*, defined on Markov chains, in which each distribution is represented using a finite collection of samples, or particles. It is therefore natural to consider generalizations of particle filtering applicable

to more general graphs (“particle” belief propagation); several variations have thus far been proposed, corresponding to different choices for certain fundamental questions.

As an example, consider the question of how to represent the messages computed during inference using particles. Broadly speaking, one might consider two possible approaches: to draw a set of particles for each message in the graph [5,7,8], or to create a set of particles for each variable, e.g., representing samples from the posterior marginal [6]. This decision is closely related to the choice of proposal distribution in particle filtering; indeed, choosing better proposal distributions from which to draw the samples, or moving the samples via Markov chain Monte Carlo (MCMC) to match subsequent observations, comprises a large part of modern work on particle filtering [10,11,12,13,14].

Either method can be made asymptotically consistent, i.e., will produce the correct answer in the limit as the number of samples becomes infinite. However, consistency is a weak condition—fundamentally, we are interested in the behavior of particle belief propagation for relatively small numbers of particles, ensuring computational efficiency. So far, little theory describes the finite sample behavior of these algorithms.

In this work, we demonstrate the application of the Particle-based Belief Propagation algorithm, most closely related to that described in [6], for the problem of Structure from Motion and Dense Stereo Vision with unknown camera constraints. Experimental results show proofs of the convergence for the accuracy of such a generic PBP algorithm with finite samples. In addition to accuracy, the PBP algorithm also allows us to estimate various properties of the distribution, thereby allows us to represent state uncertainty, although at some computational cost.

2 Review of Belief Propagation

Let G be an undirected graph consisting of nodes $V = \{1, \dots, k\}$ and edges E , and let Γ_s denote the set of neighbors of node s in G , i.e., the set of nodes t such that $\{s, t\}$ is an edge of G . In a probabilistic graphical model, each node $s \in V$ is associated with a random variable X_s taking on values in some domain, \mathcal{X}_s , and the structure of the graph is used to represent the Markov independence structure of the joint distribution $p(X_1, \dots, X_n)$. Specifically, the Hammersley-Clifford theorem [15] tells us that if $p(\cdot) > 0$, the distribution factors into a product of potential functions $\psi_c(\cdot)$ defined on the cliques (fully connected subsets) of G . For simplicity of notation, we further assume that these clique functions are pairwise. Now we can assume that each node s and edge $\{s, t\}$ are associated with potential functions Ψ_s and $\Psi_{s,t}$ respectively, and given these potential functions we define a probability distribution on assignments of values to nodes as

$$P(\vec{x}) = \frac{1}{Z} \left(\prod_s \Psi_s(\vec{x}_s) \right) \left(\prod_{\{s,t\} \in E} \Psi_{s,t}(\vec{x}_s, \vec{x}_t) \right). \quad (1)$$

Here \vec{x} is an assignment of values to all k variables, \vec{x}_s is the value assigned to X_s by \vec{x} , and Z is a scalar chosen to normalize the distribution P (also called the partition function). We consider the problem of computing marginal probabilities, defined by

$$P_s(x_s) = \sum_{\vec{x}: \vec{x}_s = x_s} P(\vec{x}). \quad (2)$$

In the case where G is a tree and the sets \mathcal{X}_s are small, the marginal probabilities can be computed efficiently by belief propagation [1]. This is done by computing messages $m_{t \rightarrow s}$ each of which is a function on the state space of the target node, \mathcal{X}_s . These messages can be defined recursively as

$$m_{t \rightarrow s}(x_s) = \sum_{x_t \in \mathcal{X}_t} \Psi_{t,s}(x_t, x_s) \Psi_t(x_t) \prod_{u \in \Gamma_t \setminus s} m_{u \rightarrow t}(x_t). \quad (3)$$

When G is a tree this recursion is well founded (loop-free) and the above equation uniquely determines the messages. We will use an unnormalized belief function defined as follows.

$$B_s(x_s) = \Psi_s(x_s) \prod_{t \in \Gamma_s} m_{t \rightarrow s}(x_s). \quad (4)$$

When G is a tree the belief function is proportional to the marginal probability P_s defined by (2). It is sometimes useful to define the “pre-message” $M_{t \rightarrow s}$ as follows for $x_t \in \mathcal{X}_t$.

$$M_{t \rightarrow s}(x_t) = \Psi_t(x_t) \prod_{u \in \Gamma_t \setminus s} m_{u \rightarrow t}(x_t). \quad (5)$$

Note that the pre-message $M_{t \rightarrow s}$ defines a weighting on the state space of the source node \mathcal{X}_t , while the message $m_{t \rightarrow s}$ defines a weighting on the state space of the destination, \mathcal{X}_s . We can then re-express (3)–(4) as

$$m_{t \rightarrow s}(x_s) = \sum_{x_t \in \mathcal{X}_t} \Psi_{t,s}(x_t, x_s) M_{t \rightarrow s}(x_t) \quad B_t(x_t) = M_{t \rightarrow s}(x_t) m_{s \rightarrow t}(x_t).$$

Although we develop our results for tree-structured graphs, it is common to apply belief propagation to graphs with cycles as well (“loopy” belief propagation). We note connections to and differences for loopy BP in the text where appropriate.

Loopy belief propagation maintains a state which stores a numerical value for each message value $m_{t \rightarrow s}(x_s)$. In loopy BP one repeatedly updates one message at a time. More specifically, one selects a directed edge $t \rightarrow s$ and updates all values $m_{t \rightarrow s}(x_s)$ using equation (3). Loopy BP often involves a large number of message updates. As the number of updates increases message values typically diverge — usually tending toward zero but possibly toward infinity if the potentials Ψ_s and $\Psi_{t,s}$ are large. This typically results in floating point underflow or overflow. For reasons of numerical stability, it is common to normalize each message $m_{t \rightarrow s}$ so that it has unit sum. However, such normalization of messages has no other effect on the (normalized) belief functions (4). We can normalize an entire state by normalizing each message. It is important to note that normalization commutes with update. More specifically, normalizing a message, updating, and then normalizing the resulting state, is the same as updating (without normalizing) and then normalizing the resulting state. This implies that any normalized state of the system can be viewed as the result of running unnormalized updates and

then normalizing the resulting state only at the end. Thus for conceptual simplicity in developing and analyzing particle belief propagation we avoid any explicit normalization of the messages; such normalization can be included in the algorithms in practice.

Additionally, for reasons of computational efficiency it is helpful to use the alternative expression $m_{t \rightarrow s}(x_s) = \sum \Psi_{t,s}(x_t, x_s) B_t(x_t) / m_{s \rightarrow t}(x_t)$ when computing the messages. By storing and updating the belief values $B_t(x_t)$ incrementally as incoming messages are re-computed, one can significantly reduce the number of operations required. Although our development of particle belief propagation uses the update form (3), this alternative formulation can be applied to improve its efficiency as well.

3 Particle Belief Propagation

We now consider the case where $|\mathcal{X}_s|$ is too large to enumerate in practice and define a generic particle (sample) based algorithm. This algorithm essentially corresponds to a non-iterative version of the method described in [6].

The procedure samples a set of particles $x_s^{(1)}, \dots, x_s^{(n)}$ with $x_s^{(i)} \in \mathcal{X}_s$ at each node s of the network¹, drawn from a sampling distribution (or weighting) $W_s(x_s) > 0$ (corresponding to the proposal distribution in particle filtering). First we note that (3) can be written as the following importance-sampling corrected expectation.

$$m_{t \rightarrow s}(x_s) = \mathbb{E}_{x_t \sim W_t} \left[\frac{\Psi_{s,t}(x_s, x_t) \Psi_t(x_t) \prod_{u \in \Gamma_t \setminus s} m_{u \rightarrow t}(x_t)}{W_t(x_t)} \right]. \quad (6)$$

Given a sample $x_t^{(1)}, \dots, x_t^{(n)}$ of points drawn from W_t we can estimate $m_{t \rightarrow s}(x_s^{(i)})$ as follows where $w_s^{(i)} = W_s(x_s^{(i)})$.

$$\hat{m}_{t \rightarrow s}^{(i)} = \frac{1}{n} \sum_{j=1}^n \frac{\Psi_{t,s}(x_t^{(j)}, x_s^{(i)}) \Psi_t(x_t^{(j)}) \prod_{u \in \Gamma_t \setminus s} \hat{m}_{u \rightarrow t}^{(j)}}{w_t^{(j)}}. \quad (7)$$

Equation (7) represents a finite sample estimate for (6). Alternatively, (7) defines a belief propagation algorithm where messages are defined on particles rather than the entire set \mathcal{X}_s . As in classical belief propagation, for tree structured graphs and fixed particle locations there is a unique set of messages satisfying (7). Equation (7) can also be applied for loopy graphs (again observing that message normalization can be conceptually ignored). In this simple version, the sample values $x_s^{(i)}$ and weights $w_s^{(i)}$ remain unchanged as messages are updated.

We now show that equation (7) is consistent—it agrees with (3) in the limit as $n \rightarrow \infty$. For any finite sample, define the particle domain $\hat{\mathcal{X}}_s$ and the count $c_i(x)$ for $x \in \hat{\mathcal{X}}_s$ as

¹ It is also possible to sample a set of particles $\{x_{st}^{(i)}\}$ for each message $m_{s \rightarrow t}$ in the network from potentially different distributions $W_{st}(x_s)$, for which our analysis remains essentially unchanged. However, for notational simplicity and to be able to apply the more computationally efficient message expression described in Section 2, we use a single distribution and sample set for each node.

$$\widehat{\mathcal{X}}_s = \{x_s \in \mathcal{X}_s : \exists k x_s^{(i)} = x_s\} \quad c_s(x_s) = |\{k : x_s^{(i)} = x_s\}|.$$

Equation (7) has the property that if $x_s^{(i)} = x_s^{(i')}$ then $m_{t \rightarrow s}^{(i)} = m_{t \rightarrow s}^{(i')}$; thus we can rewrite (7) as

$$\widehat{m}_{t \rightarrow s}(x_s) = \frac{1}{n} \sum_{x_t \in \widehat{\mathcal{X}}_t} \frac{c_t(x_t)}{W_t(x_t)} \Psi_{t,s}(x_t, x_s) \Psi_t(x_t) \prod_{u \in \Gamma_t \setminus s} \widehat{m}_{u \rightarrow t}(x_t) \quad x_s \in \widehat{\mathcal{X}}_s. \quad (8)$$

Since we have assumed $W_s(x_s) > 0$, in the limit of an infinite sample $\widehat{\mathcal{X}}_t$ becomes all of \mathcal{X}_t and the ratio $(c_t(x_t)/n)$ converges to $W_t(x_t)$. So for sufficiently large samples (8) approaches the true message (3). Fundamentally, we are interested in particle-based approximations to belief propagation for their finite-sample behavior, i.e., we hope that a relatively small collection of samples will give rise to an accurate estimate of the beliefs - the true marginal $P_t(x_t)$. At any stage of BP we can use our current marginal estimate to construct a new sampling distribution for node t and draw a new set of particles $\{x_t^{(i)}\}$. This leads to an iterative algorithm which continues to improve its estimates as the sampling distributions become more accurately targeted, although such iterative resampling processes often require more work to analyze; see e.g. [9].

In [6], the sampling distributions were constructed using a density estimation step (fitting mixtures of Gaussians). However, the fact that the belief estimate $\widehat{M}_t(x_t)$ can be computed at any value of x_t allows us to use another approach, which has also been applied to particle filters [13,14] with success. By running a short MCMC simulation such as the Metropolis-Hastings algorithm, one can attempt to draw samples directly from the weighting \widehat{M}_t . This manages to avoid any distributional assumptions inherent in density estimation methods, but has the disadvantage that it can be difficult to assess convergence during MCMC sampling.

4 Experimental Results

In this section, we describe the application of Particle Belief Propagation (PBP) to the problem of Structure from Motion (SfM) and the problem of Dense Stereo Vision with unknown camera constraints.

4.1 Particle Belief Propagation for SfM

In SfM, given a set of 2D images of the same scene, the objective is to simultaneously recover both the camera trajectories and the 3D structure of the scene. For this problem, it is commonly accepted that Bundle Adjustment (BA) [16] is the Gold standard. There has been a lot of work on this problem [17,18,19], mostly using geometric based methods. All of this work demonstrated that their result using BA as the last step is much better than without BA. With each camera pose and each map object 3D location being represented as a finite set of particles, we show that PBP can successfully estimate their true states by estimating their posterior distributions over the state space given the image observations. The algorithm was tested on both real and synthetic data. An

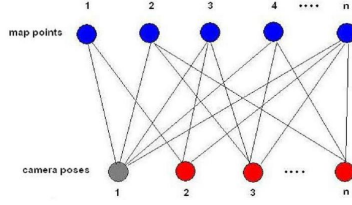


Fig. 1. The representation of the Structure from Motion problem as a bipartite graphical model

experiment on synthetic data, when the ground truth is available, allows us to compare the performance of our method with BA.

First, we represent the problem of SfM in the context of PBP. For each observed image, we detect a sparse set of special image keypoints. These points should be high-level image features, obtained by a feature detector (such as corner detector or SIFT detector [20]). The correspondences of image points between images are automatically obtained using a feature matching method, such as an efficient nearest neighbour algorithm [20]. The obtained matching result also defines the correspondences between the image points and the map points. The 3D scene can now be represented as a sparse set of 3D points. Each camera pose is assigned to a 3D location, combined with 3 angles of rotation, which define the rotation of the camera about 3 axes. Our method is based on the assumption that given a set of image observations and the estimate of all map points (resp. camera poses), there exists a probability distribution for each camera pose (resp. map points) over its state space. The true state of map points and camera poses are hidden variables that we want to estimate. Let P_i denote the random variable for the i^{th} camera pose. Let Y_j denote the random variable for the j^{th} map point. The observed data are image points and their correspondences. We denote x_{ij} the image point variable associated with map point Y_j as seen from camera P_i .

Our graphical model $G(V, E)$ consists of two types of nodes. Each camera node i is associated with a camera variable P_i , each map node j is associated with a map variable Y_j . For simplicity of notation we name each node after its variable. We add to E every edge connecting each camera node and a map point node it observes. No edge connects any two camera nodes, or any two map nodes. If each camera has an entire view of the whole set of map points, we have a complete bipartite graph. Each edge in the graph is associated with a binary potential function, denoted $\Psi_{i,j}(p_i, y_j)$. More specifically, for an edge connecting camera node P_i and map node Y_j , we have:

$$\Psi_{i,j}(p_i, y_j) = e^{-\frac{\|Q(p_i, y_j) - x_{ij}\|^2}{2\sigma^2}} . \tag{9}$$

where $Q(p_i, y_j)$ is the reprojection function that takes a camera pose and a map point and returns the reprojected image point, x_{ij} is the image observation of map point y_j as seen from camera p_i , σ^2 is the variance of the Gaussian image noise. This is equivalent to assuming a normal distribution of the true observation around the given observation. This allows us to represent the uncertainty of observation.

At this point the message function and belief function on G are well defined using (3) and (4). However, the state space of each hidden variable in G is too large to do inference with BP. In order to use PBP, we discretize the state space of each variable into a relatively small number of states, each of which is represented by a particle. These states can be sampled from a normal distribution with some initial mean and variance. For example, we can initialize the state of each 3D map point by sampling from a 3D line going through the first camera centre point and the corresponding image point. As we assume that the camera motion is smooth, the initial states of the next camera pose can be sampled from a region nearby the previous one.

Now we have at each camera node P_i , a set of M particles: $p_i^{(1)}, \dots, p_i^{(M)}$, at each map node Y_j , a set of N particles: $y_j^{(1)}, \dots, y_j^{(N)}$. If we assume no unary potential at each node, we can define the message going from camera node i to map node j in G by (7) as follows:

$$\hat{m}_{i \rightarrow j}^{(k)} = \frac{1}{M} \sum_{h=1}^M w_i^{(h)} \Psi_{i,j}(p_i^{(h)}, y_j^{(k)}) \prod_{u \in \Gamma_i \setminus j} \hat{m}_{u \rightarrow i}^{(h)}. \quad (10)$$

At the beginning, all particles in a node are assigned uniform weights and no message has been computed, equation (10) becomes:

$$\hat{m}_{i \rightarrow j}^{(k)} = \frac{1}{M} \sum_{h=1}^M \Psi_{i,j}(p_i^{(h)}, y_j^{(k)}). \quad (11)$$

This can be interpreted as the marginal probability $\sum_{h=1}^M P(Y_j = y_j^{(k)} | x_{ij}, P_i = p_i^{(h)})$. It follows that the belief of a map node becomes the marginal probability over all assignments of its neighbouring camera nodes, conditioned on relevant observations:

$$\hat{B}_j(y_j^{(k)}) = \prod_{i \in \Gamma_j} \sum_{h=1}^M P(Y_j = y_j^{(k)} | x, P_i = p_i^{(h)}) = \sum_{\mathbf{p}} P(Y_j = y_j^{(k)} | x, \mathbf{p}). \quad (12)$$

where \mathbf{p} is an assignment of values to all neighbouring pose nodes. The message from a map node to a pose node and the belief of a pose node can be interpreted similarly. As PBP proceeds, the information from one node is sent to all other nodes in the graph. This allows (2) to be approximated by (12). As shown in section 3, the posterior marginal estimated by PBP is almost guaranteed to converge to the true posterior marginal distribution estimated by BP. In addition to PBP, the samples at each node are iteratively updated by Metropolis-Hastings sampling from the estimated posterior marginal distribution at that node. Metropolis-Hastings sampling allows particles to freely explore the state space and thus compensates the inadequacy of representing a large state space by a small number of samples.

4.2 Comparison with Bundle Adjustment

The objective of this experiment is to compare PBP with the well known method BA in terms of reconstruction accuracy, by measuring the deviation of their reconstruction

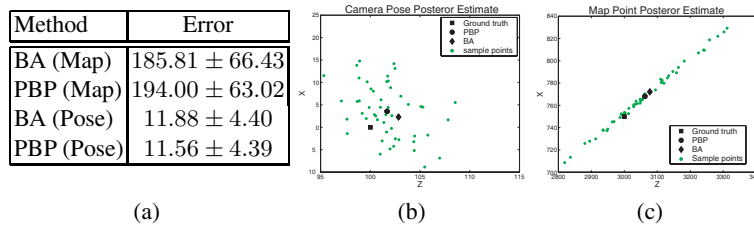


Fig. 2. Comparing bundle adjustment to PBP in structure from motion. (a) Estimated mean and standard deviation of reconstruction errors for camera pose and map positions; (b) example posterior for one camera pose and (c) for one map point using PBP.

estimates from the ground truth. The model for bundle adjustment is a pairwise graphical model over poses p_i and object positions y_j given observed locations x_{ij} in the images:

$$P(\{p_i\}, \{y_j\}) = \prod \psi_{i,j}(p_i, y_j) \quad \Psi_{i,j}(p_i, y_j) = e^{-\frac{\|Q(p_i, y_j) - x_{ij}\|^2}{2\sigma^2}}.$$

Bundle adjustment then searches for a local maximum (i.e., a mode) of the posterior distribution over the $\{p_i, y_j\}$. However, if the intent is to minimize the expected squared error loss, we should prefer the mean estimate of the posterior distribution instead of its mode. Note that these two quantities can be very different in problems where the posterior distribution is very skewed or multi-modal; in such cases, it may be advantageous to estimate the posterior distribution and its mean using methods such as belief propagation. Additionally, an explicit representation of uncertainty can be used to assess confidence in the estimate.

We show a comparison between estimates given by BA² and PBP on synthetic structure from motion data in Figure 2. For the data, we generate a fixed set of map points and a few camera poses. Each point’s observations are obtained by projecting the point on each camera’s image planes and adding Gaussian noise. We assume that the camera calibration parameters and the image correspondences are known, and initialize the estimates for both algorithms to ground truth, then run each to convergence and compare the resulting error.

Figure 2(a) shows the estimated mean and standard deviation of reconstruction errors of BA and PBP from 200 different runs (units are synthetic coordinates). This shows that, at the same level of image noise and from the same initial conditions, PBP produces essentially the same accuracy as BA for both camera and map points. However, we expect that in less idealized cases (including, for example, incorrect feature associations or outlier measurements), PBP may perform much better [22]. Moreover, PBP does not require an initialization step to provide a good initial state values, as we allow the set of initial states to be chosen randomly. In addition, PBP provides an estimate of state uncertainty (although at some computational cost). Figure 2(b)–(c) shows the estimated posterior distributions given by PBP for a camera pose and map point, respectively,

² We use the *sba* package in [21].

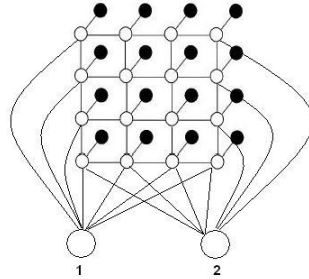


Fig. 3. The graphical representation of the Dense Stereo Vision problem

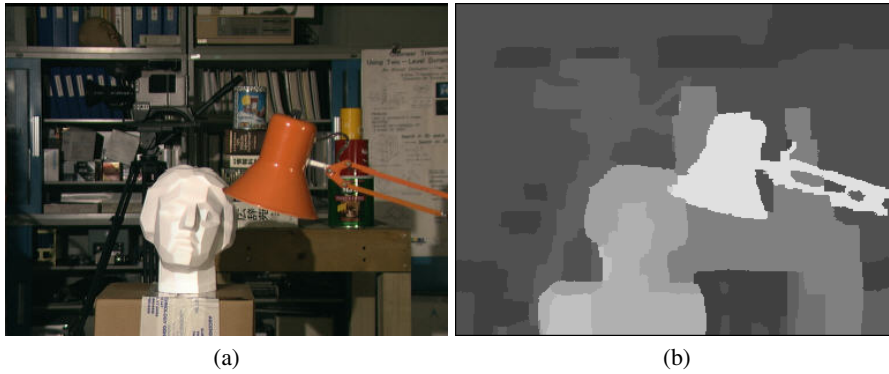


Fig. 4. (a) left image of the stereo pair; (b) estimated depth map at convergence of PBP

along with the mean found via PBP (circle), mode found via BA (diamond), and true position (square).

4.3 Particle Belief Propagation for Dense Stereo with Unknown Camera Constraints

Classical dense two-frame Stereo algorithms compute a dense disparity map for each pixel given a pair of images under known camera constraints (i.e. the configuration of the 2 cameras and the distance between them are known) [23]. Here, given a pair of stereo images with unknown camera constraints, we use PBP to simultaneously compute the dense depth map of the first image, and the configuration of the second camera relative to the first. Belief propagation [24,25] has been successfully applied to standard binocular stereo. To obtain computational efficiency, we take advantage of the algorithmic techniques in [25] which considerably speed up the the standard BP algorithm.

The formulation of the graphical model in this case is quite similar to the previous problem. There are 2 camera nodes, and the number of map nodes equals the number of pixels in the first image. An edge is added between every pixel node and every camera node. (see Figure 3). A new energy function, which evaluates the quality of the labeling

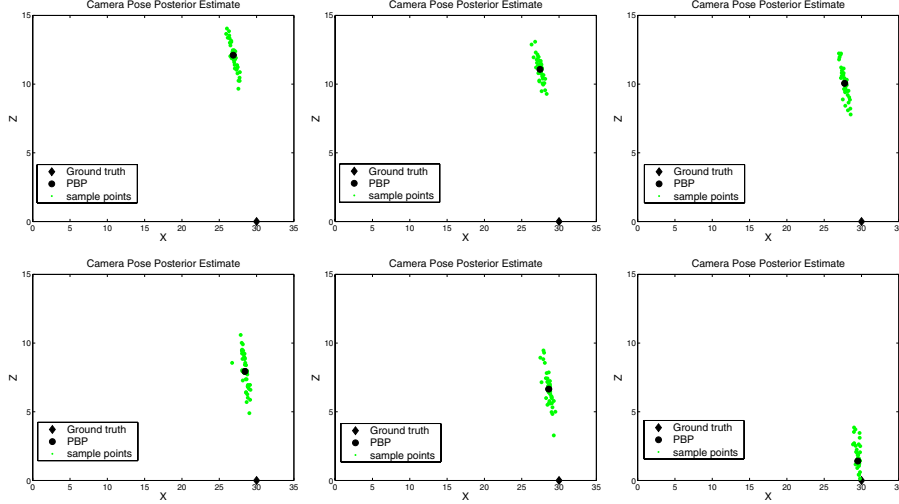


Fig. 5. Estimated posterior distribution of the camera pose over time

of both the camera poses and the 3D depth of each image point, has the following similar form,

$$E(z, c) = \sum_{(p,q) \in N} V(z_p, z_q) + \sum_{p \in P} D(z_p, c). \quad (13)$$

where P is the set of all pixels in the first image, N contains all edges connecting 2 pixel nodes, the labeling z is a function which assigns a depth value z_p to each pixel $p \in P$, c is the assigned label for the 3D configuration of the second camera, namely a 6-vector $(x, y, z, \alpha, \beta, \gamma)$. $V(z_p, z_q)$ denotes the smoothness term, which is the cost of assigning labels z_p and z_q to two adjacent pixels. In this case we use the following $V(z_p, z_q) = \min(\|z_p - z_q\|, \theta)$, which captures the assumption that the scene has smooth depth almost everywhere, except at certain locations (such as boundaries). Note that this equation also defines the binary potential function associated with edges connecting two pixel nodes. The data term $D(z_p, c)$ computes the cost of simultaneously assigning label z_p to pixel p , and assigning configuration c to the second camera. We use the following data term,

$$D(z_p, c) = \min(\|I_2(Q(p, z_p, c)) - I_1(p)\|, \tau). \quad (14)$$

where the function $Q(p, z_p, c)$ takes the pixel p on the first image, its 3D depth z_p , the configuration c of the second camera, and returns the corresponding pixel on the second image. Note that we also use a truncation value τ , which allows abrupt changes in image intensity, and makes the data cost robust to occlusion and specularities. (14) also defines the binary potential associated with edges connecting a pixel node and a camera node.

Minimizing the energy function $E(z, c)$ is now equivalent to finding the maximum posterior probability (MAP) estimate for the defined graphical model. For an energy

Method	nonocc	all	disc
Adapting BP	1.11	1.37	5.79
Graph cuts with occ	1.19	2.01	6.24
Our method	2.11	4.22	11.1
SSD + min-filter	5.23	7.07	24.1
Phase difference alg	4.89	7.11	16.3
Infection alg	7.95	9.54	28.9

Fig. 6. Comparative performance on the Tsukuba data with some other well-known stereo algorithms, using the three performance measures, *nonocc* (ratio of bad pixels at nonoccluded regions), *all* (ratio of bad pixels at all regions) and *disc* (ratio of bad pixels at regions near depth discontinuities)

function of type (13), the max-product algorithm [26] can be used as the message updating method, which becomes min-sum when performed in negative log.

As shown in Figure 5, we plot the estimated posterior distribution of the second camera pose at each iteration, thus show that the distribution gradually approaches the true state over time. Figure 4 shows the estimated depth map result on frame 3 of the Tsukuba image sequence at convergence.

Finally, Figure 6 demonstrates the comparative performance on the Tsukuba data with some other well-known stereo algorithms. As the labels we assign for pixels are their true 3D depths instead of their disparities between the two images, in order to do this, we quantize the output depth into a number of depth range equivalent to the number of disparity levels used in the evaluation routine, as shown in [23] and the Middlebury College Stereo Evaluation webpage (<http://www.middlebury.edu/stereo>).

5 Summary and Conclusions

In this paper we have demonstrated the application of the Particle-based Belief Propagation algorithm to the problems of Structure from Motion, and the problem of Dense Stereo with unknown camera constraints. To handle the cases with continuous-valued variables, or discrete-valued variables with very large domains, our approach creates a set of particles for each variable, representing samples from the posterior marginal. The algorithm then continues to improve the current marginal estimate by constructing a new sampling distribution and draw new sets of particles. It's shown by experiments that the algorithm is consistent, i.e. approaches the true values of the message and belief functions with finite samples. Besides accuracy, PBP algorithm also provides good estimate for properties of the distribution, and a representation of state uncertainty.

Although an “adaptive” choice for the sampling distribution, and such iterative re-sampling processes require further work to analyze, our results seem to support the notion of sampling from the current marginal estimates themselves, whether from fitted distributions [6] or via a series of MCMC steps.

Acknowledgments. The authors would like to express their thanks to Dr. Alexander Ihler of the Toyota Technological Institute at Chicago for his constructive comments and suggestions.

References

1. Pearl, J.: Probabilistic Reasoning in Intelligent Systems. Morgan Kaufman, San Mateo (1988)
2. Freeman, W.T., Pasztor, E.C., Carmichael, O.T.: Learning low-level vision. *IJCV* 40(1), 25–47 (2000)
3. Coughlan, J.M., Ferreira, S.J.: Finding deformable shapes using loopy belief propagation. *ECCV* 7, 453–468 (2002)
4. Coughlan, J.M., Shen, H.: Shape matching with belief propagation: Using dynamic quantization to accommodate occlusion and clutter. In: *CVPR Workshop on Generative Model Based Vision* (2004)
5. Arulampalam, M.S., et al.: A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking. *IEEE Trans. SP* 50(2), 174–188 (2002)
6. Koller, D., Lerner, U., Angelov, D.: A general algorithm for approximate inference and its application to hybrid Bayes nets. *UAI* 15, 324–333 (1999)
7. Sudderth, E.B., et al.: Nonparametric belief propagation. In: *CVPR* (2003)
8. Isard, M.: PAMPAS: Real-valued graphical models for computer vision. In: *CVPR* (2003)
9. Neal, R.M., Beal, M.J., Roweis, S.T.: Inferring state sequences for non-linear systems with embedded hidden Markov models. In: *NIPS*, vol.16 (2003)
10. Thrun, S., Fox, D., Burgard, W.: Monte carlo localization with mixture proposal distribution. In: *AAAI*, pp. 859–865 (2000)
11. van der Merwe, R., et al.: The unscented particle filter. In: *NIPS*, vol. 13 (2001)
12. Doucet, A., de Freitas, N., Gordon, N. (eds.): *Sequential Monte Carlo Methods in Practice*. Springer, New York (2001)
13. Godsill, S., Clapp, T.: Improvement strategies for Monte Carlo particle filters. In: Doucet, J.F.G.D.F.A., Gordon, N.J. (eds.) *Sequential Monte Carlo Methods in Practice*, Springer, New York (2001)
14. Khan, Z., Balch, T., Dellaert, F.: MCMC-based particle filtering for tracking a variable number of interacting targets. *IEEE Trans. PAMI*, 1805–1918 (2005)
15. Clifford, P.: Markov random fields in statistics. In: Grimmett, G.R., Welsh, D.J.A. (eds.) *Disorder in Physical Systems*, pp. 19–32. Oxford University Press, Oxford (1990)
16. Triggs, B., et al.: Bundle adjustment - a modern synthesis. In: *Vision Algorithms* (1999)
17. Hartley, R.I., Zisserman, A.: *Multiple View Geometry in Computer Vision*, 2nd edn. Cambridge University Press, Cambridge (2004)
18. Kaucic, R., Hartley, R., Dano, N.: Plane-based projective reconstruction. In: *ICCV 2001. Proceedings of the Ninth IEEE International Conference on Computer Vision*. IEEE Computer Society Press, Los Alamitos (2001)
19. Snavely, N., Seitz, S.M., Szeliski, R.: Photo tourism: Exploring photo collections in 3D. *ACM Transactions on Graphics* 25(3), 835–846 (2006)
20. Lowe, D.G.: Object recognition from local scale-invariant features. In: *Proc. of the International Conference on Computer Vision, Corfu.*, pp. 1150–1157 (1999)
21. Lourakis, M., Argyros, A.: The design and implementation of a generic sparse bundle adjustment software package based on the levenberg-marquardt algorithm. In: *Technical Report 340, Institute of Computer Science - FORTH, Heraklion, Crete, Greece* (2004), <http://www.ics.forth.gr/~lourakis/sba>
22. Ihler, A.T., et al.: Nonparametric belief propagation for self-calibration in sensor networks. *IEEE Trans. Jsc.*, 809–819 (2005)
23. Scharstein, D., Szeliski, R., Zabih, R.: A taxonomy and evaluation of dense two-frame stereo correspondence algorithms (2001)

24. Sun, J., Zheng, N.N., Shum, H.Y.: Stereo matching using belief propagation. *IEEE Trans. Pattern Anal. Mach. Intell.* 25(7), 787–800 (2003)
25. Felzenszwalb, P.F., Huttenlocher, D.P.: Efficient belief propagation for early vision. *Int. J. Comput. Vision* 70(1), 41–54 (2006)
26. Weiss, F.: On the optimality of solutions of the max-product belief-propagation algorithm in arbitrary graphs. *IEEE Trans. on Information Theory* 47 (2001)