

PMBP: PatchMatch Belief Propagation for Correspondence Field Estimation

Frederic Besse · Carsten Rother ·
Andrew Fitzgibbon · Jan Kautz

Received: 31 January 2013 / Accepted: 5 August 2013 / Published online: 20 August 2013
© Springer Science+Business Media New York 2013

Abstract PatchMatch (PM) is a simple, yet very powerful and successful method for optimizing continuous labelling problems. The algorithm has two main ingredients: the update of the solution space by sampling and the use of the spatial neighbourhood to propagate samples. We show how these ingredients are related to steps in a specific form of belief propagation (BP) in the continuous space, called max-product particle BP (MP-PBP). However, MP-PBP has thus far been too slow to allow complex state spaces. In the case where all nodes share a common state space and the smoothness prior favours equal values, we show that unifying the two approaches yields a new algorithm, PMBP, which is more accurate than PM and orders of magnitude faster than MP-PBP. To illustrate the benefits of our PMBP method we have built a new stereo matching algorithm with unary terms which are borrowed from the recent PM Stereo work and novel realistic pairwise terms that provide smoothness. We have experimentally verified that our method is an improvement over state-of-the-art techniques at sub-pixel accuracy level.

Keywords Correspondence fields · Belief propagation · PatchMatch

1 Introduction

This paper draws a new connection between two existing algorithms for estimation of correspondence fields between images: belief propagation (BP; Pearl 1988; Yedidia et al. 2005) and PatchMatch (PM; Barnes et al. 2009, 2010). Cor-

respondence fields arise in problems such as dense stereo reconstruction, optical flow estimation, and a variety of computational photography applications such as recoloring, deblurring, high dynamic range imaging, and inpainting. By analysing the connection between the methods, we obtain a new algorithm which has performance superior to both its antecedents, and in the case of stereo matching, represents the current state-of-the-art on the Middlebury benchmark at sub-pixel accuracy. The first contribution of our work is a detailed description of PM and BP in terms that allow the connection between the two to be clearly described. This analysis is largely self-contained, and comprises the first major section of the paper. Our second contribution is in the use of this analysis to define a new algorithm: PMBP which, despite its relative simplicity, is more accurate than PM and orders of magnitude faster than max-product particle BP (MP-PBP).

BP is a venerable approach to the analysis of correspondence problems. The correspondence field is parametrized by a vector grid $\{\mathbf{u}_s\}_{s=1}^n$, where s indexes *nodes*, typically corresponding to image pixels, and $\mathbf{u}_s \in \mathbb{R}^d$ parametrizes the correspondence vector at node s . We shall consider a special case of BP, viewed as an energy minimization algorithm where the energy combines *unary* and *pairwise* terms

$$E(\mathbf{u}_1, \dots, \mathbf{u}_n) = \sum_{s=1}^n \psi_s(\mathbf{u}_s) + \sum_{s=1}^n \left[\sum_{t \in N(s)} \psi_{st}(\mathbf{u}_s, \mathbf{u}_t) \right], \quad (1)$$

with $N(s)$ being the set of *pairwise neighbours* of node s . The unary energy $\psi_s(\mathbf{u}_s)$, also called the *data term*, computes the local evidence for the correspondence \mathbf{u}_s . For example, if $\mathbf{u}_s = (u_s, v_s)$ is a parametrization of a two-dimensional (2D) flow field between images I_1 and I_2 , then one might define a weighted patch data term (where (x_s, y_s) are the image coordinates of pixel s)

F. Besse (✉) · C. Rother · A. Fitzgibbon · J. Kautz
University College London, London, UK
e-mail: f.besse@cs.ucl.ac.uk

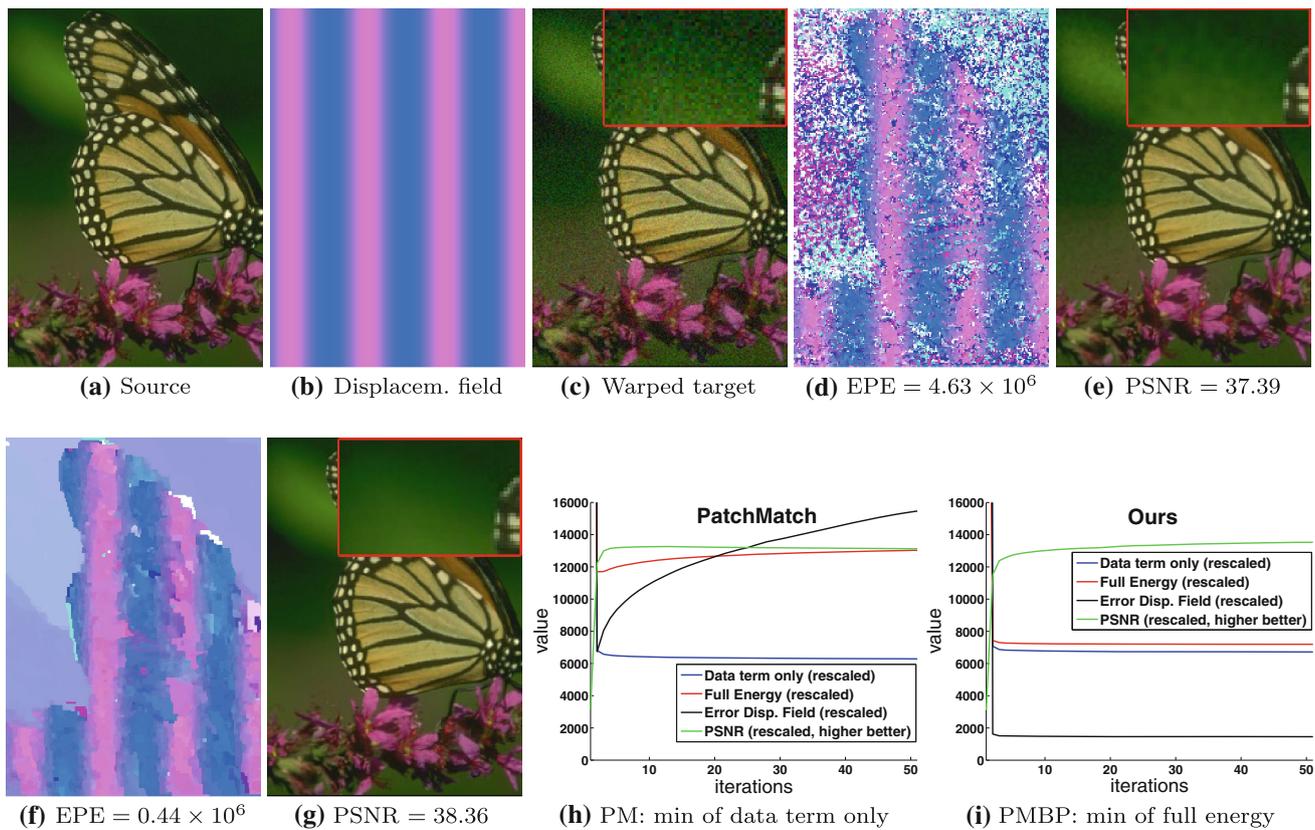


Fig. 1 Example: denoising with a reference image. **a** Source image. **b** Synthetic displacement field $\mathbf{u}_s^{\text{gt}} := [\sin x_s, 0]^\top$. **c** Warped target image with 10% Gaussian noise added (note, *red rectangle* is a zoom of the *top left corner*). All images can be found in supplementary material). Estimated displacement field using PatchMatch (**d**) and our method (**f**), with total end-point error $\text{EPE} = \sum_s \|\mathbf{u}_s - \mathbf{u}_s^{\text{gt}}\|^2$. Reconstructed target image using PatchMatch (**e**) and our method (**g**), with peak signal-to-noise ratio (PSNR). Our method is considerably better for both

error measures. The difference between **e** and **g** is especially noticeable in the smooth, *green background* where PatchMatch suffers from the ambiguous data term. **h, i** Plots error and energy for PatchMatch and our method. It is noticeable that the full energy with pairwise terms is a much better fit for the task, since in **i** both error measures are well correlated with the regularized energy, in contrast to **h**, where the *error curves* increase as the PatchMatch iterations decrease the unary-only energy (Color figure online)

$$\psi_s^{\text{wpcf}} \left(\begin{bmatrix} u_s \\ v_s \end{bmatrix} \right) = \sum_{i=-h}^h \sum_{j=-h}^h w_{sij} \left\| I_1(x_s + i, y_s + j) - I_2(x_s + i + u_s, y_s + j + v_s) \right\|. \quad (2)$$

Here, the weights w_{sij} are precomputed based on the intensity values surrounding pixel s , and the norm $\|\cdot\|$ represents magnitude of difference in an appropriate colour space. For stereo correspondence, with $\mathbf{u}_s = [\Delta_s]$ being the single scalar disparity, the equivalent data term is $\psi_s^{\text{wps}}([\Delta_s]) = \psi_s^{\text{wpcf}}([\Delta_s, 0]^\top)$. The problem with such a data term is that it implicitly assumes a constant correspondence field in the $(2h + 1) \times (2h + 1)$ patch surrounding every pixel. For large h , this over smooths the solution, even with clever choices of w_{sij} . The oversmoothing can be addressed by using more complex parametrizations of the field within the patch (see ψ_s^{pms} below), but within traditional BP frameworks, this comes at intractable computational cost. Alternatively, h may be reduced, but as h decreases, the data term becomes increas-

ingly ambiguous. This ambiguity is addressed by the introduction of pairwise terms, typically encouraging piecewise smoothness of the correspondence field, by assigning low energy to neighbouring nodes with similar parameter vectors, for example $\psi_{st}(\mathbf{u}_s, \mathbf{u}_t) = \min(\tau_{st}, \omega_{st} \|\mathbf{u}_s - \mathbf{u}_t\|^2)$ for image-derived constants τ_{st}, ω_{st} . It is generally understood that the presence of such pairwise term makes energy minimization difficult. For discrete problems, where the \mathbf{u} live in a finite set of size D , this is clearly true in principle: without pairwise terms, minimization can be computed in $O(nD)$ time, while with pairwise terms, the worst-case complexity becomes $O(D^n)$. In practice, although BP offers no strong guarantees, it often finds good minimizers in time far below this worst case prediction. For correspondence problems, however, the \mathbf{u} live in an effectively continuous space, so D must be very large (say 10^2 – 10^5), meaning that even the $O(nD)$ complexity of tabulating the unary costs is extremely high. Some algorithms have been proposed to address this complexity (Ihler and McAllester 2009;

Noorshams and Wainwright 2011; Pal et al. 2006; Sudderth et al. 2010), and it is on this class of methods that we improve in this paper. First, however, let us consider another school of related work.

The PM algorithm (Barnes et al. 2009) was initially introduced as a computationally efficient way to compute a *nearest neighbour field* (NNF) between two images. The NNF is then used for image editing operations such as denoising, inpainting, deblurring, as illustrated in Fig. 1. In terms of energy minimization, the NNF is the global minimizer of an energy comprising unary terms only ($\psi_{st} = 0$). The PM algorithm computes good minima while evaluating the unary term many fewer than D times per node. With such a powerful optimizer, more complex unary terms can be defined, yielding another class of state-of-the-art correspondence finders, exemplified by the recent introduction of PM Stereo (Bleyer et al. 2011). There, disparity is *overparametrized* by a 3D vector at each node $\mathbf{u}_s = [a_s, b_s, c_s]^\top$, parametrizing a planar disparity surface $\Delta_s(x, y) = a_s(x - x_s) + b_s(y - y_s) + c_s$, giving a unary cost whose essential form is:

$$\psi_s^{\text{pms}}([a_s, b_s, c_s]^\top) = \sum_{i=-h}^h \sum_{j=-h}^h w_{sij} \left\| I_1(x_s + i, y_s + j) - I_2(x_s + i + (a_s i + b_s j + c_s), y_s + j) \right\|. \quad (3)$$

Without PM, optimization of an energy containing such a data term, even without pairwise terms, would be computationally demanding, requiring millions of operations per pixel. Intriguingly, the key operations to which PM owes its efficiency are very close to those used in continuous BP, and in particular to the message-passing that is central to optimization in the presence of pairwise terms. Conversely, a key deficiency of PM is that it lacks an explicit smoothness control on the output field. Indeed, recent developments of PatchMatch have noted that PM “has difficulty finding reliable correspondences in very large smooth regions” (HaCohen et al. 2011). He et al. (2011) require a smooth field when applying PM to an alpha matting problem, but impose smoothness as a post-process, by solving the matting Laplacian. Boltz and Nielsen (Boltz and Nielsen (2010)) achieve smoothness by dividing the images into superpixels and running PM on these, meaning that a failure of superpixelization cannot be recovered from. A related deficiency is the tendency of PatchMatch to require a form of “early stopping”: the global optimum of the unary energy is not necessarily the best solution in terms of image error, as we show in Fig. 1h, and as can be seen in Fig. 9 of Mansfield et al. (Mansfield et al. (2011)). These difficulties are exacerbated by more powerful PM algorithms (Barnes et al. 2010; Korman and Avidan 2011) which, although getting closer to the globally optimal NNF, lose the implicit smoothness that early stopping provides. We characterize this tradeoff by looking at *error* versus *energy*: the

correlation between ground-truth errors (e.g. peak signal-to-noise ratio (PSNR) for denoising problems, end-point error (EPE) for 2D correspondence fields, or disparity error for stereo) and the values of the energy functions the algorithms implicitly or explicitly minimize.

The contribution of this paper is to define a new family of algorithms, called PMBP, which combine the best features of both existing approaches, and which includes the existing methods as special cases. We first describe both existing algorithms using a unified notation, showing the close relations between the two (also illustrated as an “algorithm by numbers” in Table 1). We then investigate the combination in various experimental settings, in order to explore the key terms which contribute to the combined algorithm’s performance. The paper closes with a discussion of future directions.

Notation To simplify the descriptions below, the following notation will be helpful. Define the application of a function f to a set S by $f(S) := \{f(s) | s \in S\}$. Define the function $\text{fargmin}_K(S, f)$ as the function that returns the K elements of S which minimize f :

$$\begin{aligned} \text{fargmin}_K(S, f) &:= S_K \subset S \quad \text{s.t. } |S_K| = \min(K, |S|) \\ &\quad \text{and } \max f(S_K) \leq \min f(S \setminus S_K). \end{aligned} \quad (4)$$

1.1 PM with Particles

In this section we describe generalized PM (GPM; Barnes et al. 2010) in terms that will allow easy unification with standard descriptions of continuous-domain BP. With each node s , we associate a set of K *particles* $P_s \subset \mathbb{R}^d$, where each particle $p \in P_s$ is a candidate solution for the minimizing correspondence parameters \mathbf{u}_s^* . Initializing these sets uniformly at random gives good performance, which may be improved slightly by using some more data-driven strategy, as discussed in Sect. 4.1.

One PM iteration then comprises a linear sweep through all nodes. The order in which nodes are visited is defined by a *schedule function* $\phi(s)$, so that s is visited before s' if $\phi(s) < \phi(s')$. We also define the *predecessor set* $\Phi s = \{s' | \phi(s') < \phi(s)\}$. On odd-numbered iterations, the typical choice of scheduling function $\phi(\cdot)$ defines a top-left to bottom-right ordering, while even-numbered iterations reverse the ordering, from bottom-right to top-left. If *iter* is an iteration counter, we write $\phi_{\text{iter}}(\cdot)$ to select the appropriate schedule. At node s , two update steps are performed: *propagation* and *resampling*:

- In the *propagation* step, the particle set is updated to contain the best K particles from the union of the current

set and the set C_s of already-visited neighbour candidates

$$C_s = \bigcup \{P_t \mid t \in N(s) \cap \Phi_s\}, \tag{5}$$

where “best” is defined as minimizing the unary cost $\psi_s(\cdot)$:

$$P_s \leftarrow \text{fargmin}_K (P_s \cup C_s, \psi_s). \tag{6}$$

- The local *resampling* step (called “random search” in Barnes et al. 2010) perturbs the particles locally according to a proposal distribution which we model as a Gaussian $\mathcal{N}(0, \sigma)$. This mechanism mimics the random search of the original PM algorithm which uses concentric circles of decreasing radii instead. The second step of the PM iteration updates P_s with any improved estimates from the local resampling set, for m resampling steps:

$$R_s = \{p + \mathcal{N}(0, \sigma) \mid p \in P_s\} \tag{7}$$

$$P_s \leftarrow \text{fargmin}_K (P_s \cup R_s, \psi_s). \tag{8}$$

After several alternating sweeps, the best particle in each set typically represents a good optimum of the unary-only energy. At first sight, it may appear surprising that such a simple algorithm can effectively minimize complex energies such as $\sum_s \psi_s^{\text{pms}}$, but as the analysis in Barnes et al. (2009) shows, the piecewise smoothness in typical image flow fields¹ effectively shares the optimization burden among neighbouring pixels in the same smooth segment, without any need to identify those segments in advance.

1.2 Max Product Particle BP (MP-PBP)

As mentioned above, our view of BP is as a minimizer of the energy (1). Thus we present a rather spartan description of max-product BP, sufficient to derive our new algorithm. BP is a *message-passing* algorithm, where messages are defined as functions from nodes to their neighbours, so that the message $M_{t \rightarrow s}(\mathbf{u}_s)$ represents, in words, “node t ’s opinion of the [negative log of the] likelihood that node s has value \mathbf{u}_s ”. Before defining the messages, which are themselves defined recursively, it is useful to define the *log disbelief*² at node s as

$$B_s(\mathbf{u}_s) := \psi_s(\mathbf{u}_s) + \sum_{t \in N(s)} M_{t \rightarrow s}(\mathbf{u}_s), \tag{9}$$

¹ Note that “flow field” is intentionally left imprecise here. The key is that the globally optimum NNF is *not* smooth, but the approximate NNF found by PM tends to be, due to the smoothness of the underlying real-world physical process which generates the image correspondences.

² This energy-based formulation can be converted to a probabilistic form using the conversions: belief $b_s(\mathbf{u}_s) := \exp(-B_s(\mathbf{u}_s))$ and message $m_{t \rightarrow s}(\mathbf{u}_s) = \exp(-M_{t \rightarrow s}(\mathbf{u}_s))$.

in terms of which the messages are defined as

$$M_{t \rightarrow s}(\mathbf{u}_s) := \min_{\mathbf{u}_t} \psi_{st}(\mathbf{u}_s, \mathbf{u}_t) + B_t(\mathbf{u}_t) - M_{s \rightarrow t}(\mathbf{u}_s), \tag{10}$$

or, in words: “the belief at t , modified by the pairwise term, and neglecting s ’s contribution to t ’s belief”. When implemented as an iterative algorithm, messages are updated according to a schedule, like PM, and messages on the right-hand side of (10) are those of the previous iteration, or those computed earlier in the current iteration. Messages are typically initialized to all-zero. At convergence, $\hat{\mathbf{u}}_s := \text{argmin}_{\mathbf{u}} B_s(\mathbf{u})$ is the estimate of the minimizer.

The key to implementing BP for continuous state variables \mathbf{u} is in the representation chosen for the messages and beliefs. Isard et al. (2008) propose a solution by discretizing the space in a way that minimises a Kullback–Leibler divergence measure. Noorshams and Wainwright (2011) work on large discrete spaces, and use a randomisation step to incrementally and stochastically update partial messages, reducing the complexity from quadratic to linear. Pal et al. (2006) also operate on large discrete spaces, and maintain sparse local marginals by using Kronecker delta functions, keeping only labels carrying the highest probability mass. Sudderth et al. (2010) extend particle filters to loopy BP, and use a regularisation kernel to ensure that message products are well defined. Particle convex BP (Peng et al. 2011) uses a local resampling step like MP-PBP, but instead of keeping the K best particles per node, or drawing from a distribution, it keeps the one particle which optimizes a discrete MRF with K candidate particles per node. Very recently, Yamaguchi et al. (2012) apply it to dense stereo estimation, combining the plane parameterization from (3) with a discrete line process. However, to allow tractable inference, they use a superpixelization into 1,200 regions, meaning the results are strongly dependent on an accurate segmentation.

In our case, a natural representation already presents itself, closely related to the MP-PBP of Kothapa et al. (2011), based in turn on Ihler and McAllester (2009). As above, we associate with each node s a particle set P_s . Then all messages and beliefs evaluated at any node η are in terms of the particles P_η , so the message definition becomes

$$M_{t \rightarrow s}(\mathbf{u}_s) := \min_{\mathbf{u}_t \in P_t} \psi_{st}(\mathbf{u}_s, \mathbf{u}_t) + B_t(\mathbf{u}_t) - M_{s \rightarrow t}(\mathbf{u}_t). \tag{11}$$

We note that this definition is still in terms of a continuous \mathbf{u}_s , not restricted to the current particle set P_s , but the continuous minimization over \mathbf{u}_t in (10) is replaced by a discrete minimization over the particles P_t .

The final step of each iteration at node s is to choose a new set of particles P_s to represent the belief at s . The ideal set of particles would be a draw (including the mode, as our goal is to minimize the energy) from the true belief $b_s^*(\cdot)$, which is of course unavailable. As an alternative, Kothapa et al. (2011) propose MCMC sampling from the current belief

Table 1 Pseudo-code for different algorithms

Let P_s be the set of particles at node s , and K the desired number of particles. Let N be the number of iterations, and m the number of randomization iterations. Let \mathcal{I} be the initialization distribution: uniform or local potentials ψ .			
PM	MPPBP	PMBP	Steps
•	•	•	1 for all nodes $s \in \{1..n\}$, repeat K times: // initialization
•	•	•	2 Draw $p \sim \mathcal{I}$, add to P_s
•	•	•	3 for $i = 1$ to N : // main loop
•	•	•	4 for all nodes $s \in \{1..n\}$ orderby ϕ_i : // PatchMatch schedule
•	•	•	5 for all proposal sets R_s in:
•	-	•	6 $R_s = \bigcup \{P_t \mid t \in N(s) \cap \Phi_i(s)\}$ // resampling using neighbours
•	•	•	7 $R_s = P_s$ // local resampling
•	•	•	8 do
•	•	•	9 for all particles $p \in R_s$:
•	•	•	10 repeat m times // Possibly different m for each R_s
•	•	•	11 $p' = p + \mathcal{N}(0, \sigma)$
•	-	-	12 Compute $B_s(p') = \psi_s(p')$
-	•	•	13 Compute $B_s(p') = \psi_s(p') + \sum_{t \in N(s)} M_{t \rightarrow s}(p')$
•	-	•	14 $P_s = \text{fargmin}_K(P_s \cup \{p'\}, B_s)$ // Update best K in P_s .
-	•	-	15 if $B_s(p') < B_s(p) - \log(\text{rand})$: $p \leftarrow p'$ // MCMC sampling
-	•	-	16 Replace p with p' in P_s // Only after MCMC
•	•	•	17 for all nodes $s \in 1..n$: // read out the final solution
•	-	-	18 return $\text{fargmin}(P_s, B_s)$ where $B_s(p) = \psi_s(p)$
-	•	•	19 return $\text{fargmin}(P_s, B_s)$ where $B_s(p) = \psi_s(p) + \sum_{t \in N(s)} M_{t \rightarrow s}(p)$

Note that whenever B_s is computed, for MP-PBP and PMBP, we have to also recompute the minimizations in the messages $M_{t \rightarrow s}$
 PM PatchMatch, MP-PBP max product particle BP, PMBP PatchMatch BP

estimate with a Gaussian proposal distribution. We show that other alternatives can be valuable.

2 PatchMatch BP

We are now in a position to make the second of our contributions, combining the PM and MP-PBP algorithms. We shall consider MP-PBP our base, as the goal is to minimize a more realistic energy than PM, that is to say, an energy with pairwise terms encouraging piecewise smoothness. Referring to Table 1, two key differences between PM and MP-PBP are evident.

First, PM resamples P_s from the neighbours of node s , while MP-PBPs resampling is only via MCMC from the elements of P_s . As illustrated in Fig. 2, this may be viewed as sampling from the continuous incoming messages at s , with the property that important modes of the belief may be uncovered, even when P_s lacks particles at those modes. It should be clarified that the samples are evaluated using B_s , so this is a resampling of the particle set under the current belief, as proposed in MP-PBP, but with a quite different source of particle proposals. Thus PMBP augments MP-PBP with samples from the neighbours (or, as argued in Fig. 2, samples from the incoming messages). This can also viewed as a return to the sampling strategies of non-parametric BP (Sudderth et al. 2010), but with a much simpler message representa-

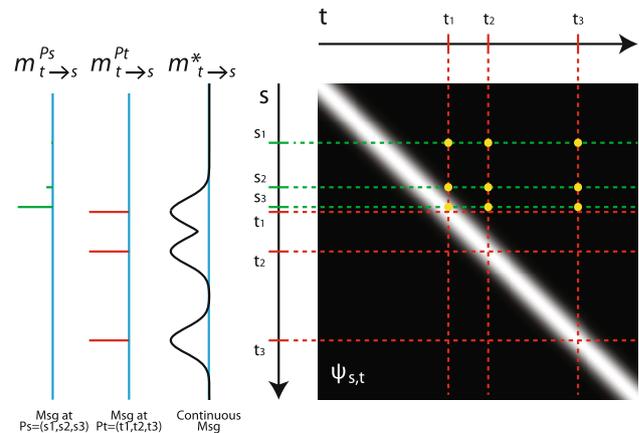


Fig. 2 Message calculation. Green bars represent the set of particles at s , $P_s = (s_1, s_2, s_3)$ and the red bars represent $P_t = (t_1, t_2, t_3)$. In MP-PBP (Ihler and McAllester 2009; Kothapa et al. 2011), the continuous message $m_{t \rightarrow s}^* \mathbf{u}_s$ is evaluated only at particles in P_s , and minimized only over P_t , evaluated at the yellow dots. When P_s and P_t differ, much of the message may be uninformative (represented by the green particles $m_{t \rightarrow s}^{P_s}$). If the pairwise potential favours smoothness, including particles from P_t increases the likelihood that high probability parts of the message are included (Color figure online)

tion. One way to look at this contribution is simply to say we are running some form of NBP but with algorithm settings (number of particles, number of samples) that would never make sense for NBP, and that this in itself is a useful contribution. Note that taking directly particles from the

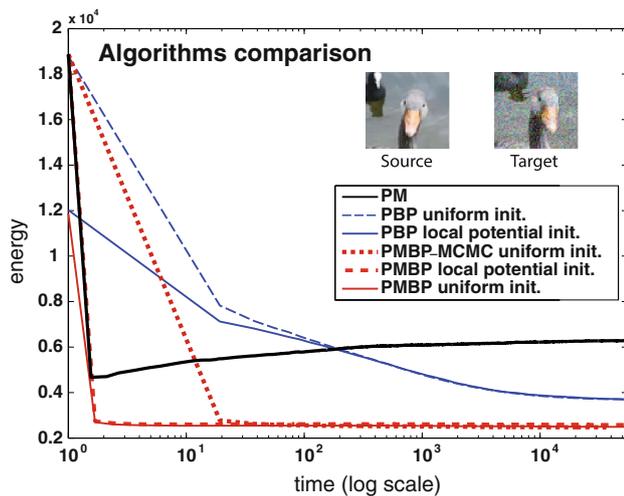


Fig. 3 Comparison of the energies produced by the different algorithms on a denoising experiment. Notice that MP-PBP cannot reach the energy of PMBP even if allowed four orders of magnitude longer, supporting our claim that previous BP implementations were intractable

neighbouring node only works because our pairwise term is a smoothing term, i.e. has the lowest value when both entries are the same. Hence for arbitrary pairwise terms this strategy has to be modified.

Second, MP-PBP uses an MCMC framework where particles are replaced in P_s with probability given by the Metropolis acceptance ratio, while PM accepts only particles with higher belief than those already in P_s . We have found that this non-Metropolis replacement strategy further accelerates convergence, so it is included in PMBP.

Making these two modifications yields “PatchMatch BP”, a powerful new optimization algorithm for energies with pairwise smoothness terms. In the case of a zero pairwise term $\psi_{st} = 0$, PMBP exactly yields GPM. Conversely, running PMBP with a non-zero pairwise term is a strict generalization of GPM, allowing the incorporation of an explicit smoothness control which directly addresses the deficiencies of PM while retaining its speed.

Note that we can also use any external information to get reasonable candidate particles, such as matching nodes between image pairs in the stereo matching case, similarly to Bleyer et al. (2011).

3 Implementation Details

While the algorithm is described in Table 1, there are some implementation details that are worth describing.

3.1 Caching

First, PMBP and in general all loopy BP algorithms are defined in a recursive manner. Equations 10 and 11 illustrate

this, as the computation of a message at one node depends on the messages at other nodes. Therefore, the usual method is to proceed in an iterative fashion, where the new messages are computed from the messages calculated in a previous iteration. This implies that the messages need to be stored for later use. To illustrate this, let us explicitly label messages and particle sets with the iteration number at which they are computed. We consider the message from node t to s at iteration k , $M_{t \rightarrow s}^k$. We call P_s^k the particle set at node s at iteration k . We also call B_s^k the disbelief at node s at iteration k , and it is computed as follows:

$$B_s^k(\mathbf{u}_s) := \psi_s(\mathbf{u}_s) + \sum_{t \in N(s)} M_{t \rightarrow s}^{k-1}(\mathbf{u}_s), \tag{12}$$

which is equivalent to Eq. 9, except that we explicitly indicate which messages need to be used.

Having introduced these notations, we can now see that there is a conceptual problem with respect to storing previously computed messages, as messages are functions of the receiver state. In a nutshell, this is due to the fact that the particle set at each node changes continuously due to the resampling procedure, which invalidates previously stored messages as they can only be evaluated on the set of particles that was present at the time they were stored. This can be easily seen in Eq. 12: the belief needs to evaluate several messages at particle position \mathbf{u}_s , which might not have been part of the previous particle set P_{k-1} ; in other words the values of the message at \mathbf{u}_s might simply not exist.

To resolve this issue, let us consider how the messages are computed. We can rewrite Eq. 11 as follows:

$$M_{t \rightarrow s}(\mathbf{u}_s) := \min_{\mathbf{u}_t \in P_t} \psi_{st}(\mathbf{u}_s, \mathbf{u}_t) + m_{t \rightarrow s}(\mathbf{u}_t), \tag{13}$$

with

$$m_{t \rightarrow s}(\mathbf{u}_t) = B_t(\mathbf{u}_t) - M_{s \rightarrow t}(\mathbf{u}_t). \tag{14}$$

We call the object $m_{t \rightarrow s}$ the “pre-message” or “message foundation” from node t to s . This object has an interesting property: it is a function of the sender node and not of the receiver. It contains information about t and is expressed in terms of particles of t . It is converted through Eq. 13 into $M_{t \rightarrow s}(\mathbf{u}_s)$, i.e., information about s and is expressed in terms of particles of s . As the pairwise term ψ_{st} is known, our caching strategy can be reduced to caching the message foundations only. As they are not tied to the particle states of the receiver, and depend *only* on the state of the sender, it is suitable for our framework. In practice, after having updated the particle set of node t at a given iteration, we compute all the message foundations outgoing from t and store them for later use, before continuing to the next node according to the schedule.

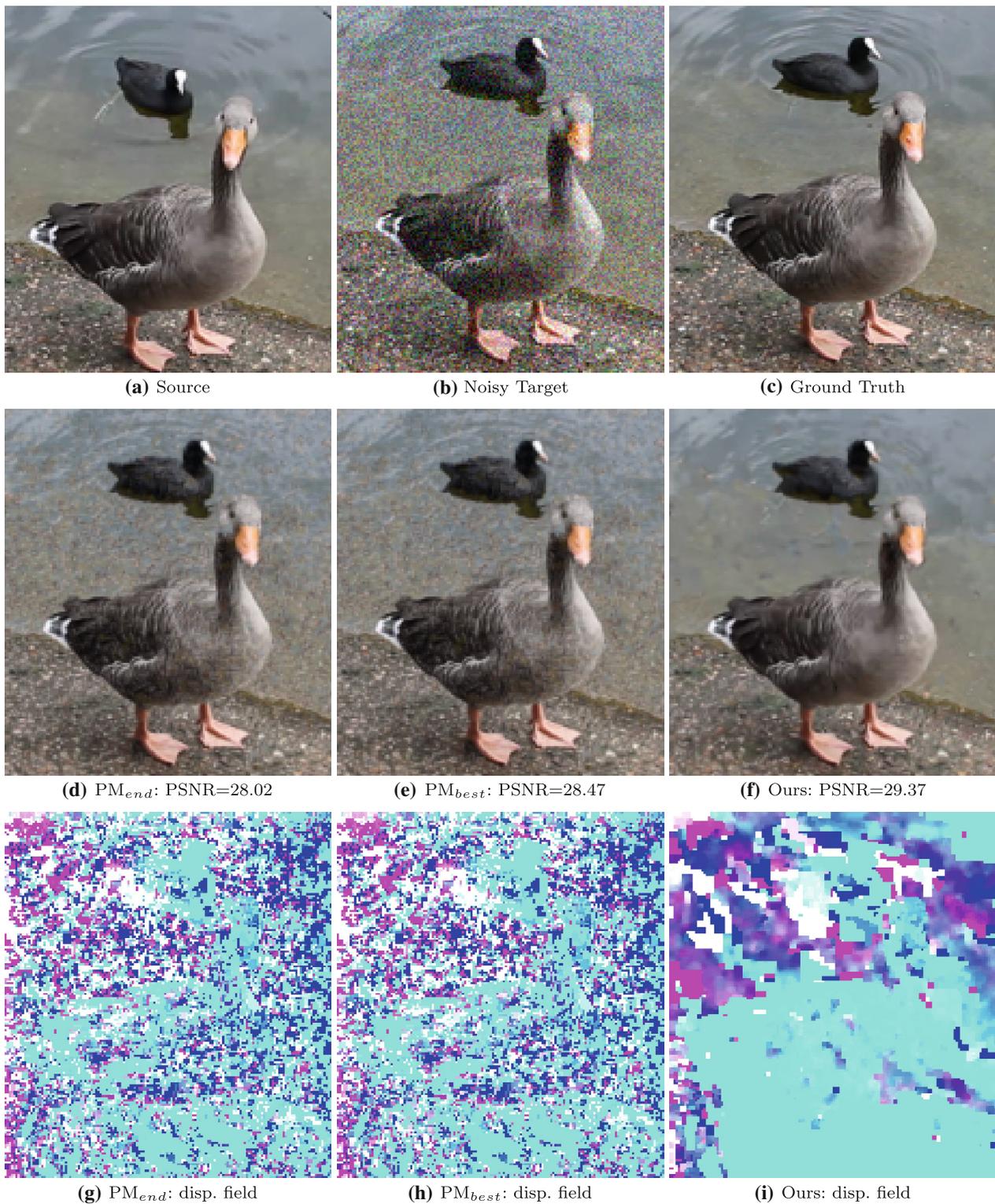


Fig. 4 A comparison of the results of PM and our algorithm on a pair of images taken from a video, one of which having been degraded by adding noise. We run the algorithms for 500 iterations, using 3×3 patches and we allow for subpixel translations only. We can see that our algorithm, using a smooth Gaussian pairwise term, manages to recon-

struct an image having a higher PSNR than the result output by PM. Furthermore, we once again see that the end iteration of PM yields worse results than its result at an early iteration (iteration 19), from which the PSNR starts decreasing

3.2 Existing Particles Update

Another implementation detail that is important to consider is the update of the disbelief at the previously existing particles of a set. When following the schedule and reaching a node s , the particles of the current set P_s are associated with existing disbelief values that were computed at the previous iterations. These values were computed using the messages that existed at the previous iteration, which might now have changed. Therefore, the first step that is run when reaching a node is to *recompute* the disbelief values for the existing set of particles, so that they reflect the new changes of the incoming messages.

3.3 Normalisation

As mentioned in [Nowozin and Lampert \(2011\)](#), using a max-product formulation creates a numerical instability as large numbers are formed from accumulation of the messages. This happens very quickly, as the growth is exponential. To keep the algorithm stable, a normalisation is performed on the message foundations whenever they are recomputed, which consists in shifting the values so that they all sum to zero. That is, assuming that $\bar{m}_{t \rightarrow s}$ is the non-normalised message foundation from t to s , we have:

$$\delta = \frac{1}{|P_t|} \sum_{\mathbf{u}_t \in P_t} \bar{m}_{t \rightarrow s}(\mathbf{u}_t), \quad (15)$$

$$m_{t \rightarrow s}(\mathbf{u}_t) = \bar{m}_{t \rightarrow s}(\mathbf{u}_t) - \delta. \quad (16)$$

Therefore, after having finished caching the message foundations at a node, we run this normalization process to ensure stability of the algorithm.

4 Experiments

Experiments were performed to quantify the effects of the various algorithm components, as well as real-world performance on a stereo benchmark.

4.1 Initialization

As mentioned above, there are two ways of initializing the particles: using a uniform distribution, or using the local potentials, as suggested in [Ihler and McAllester \(2009\)](#). However, sampling from the local potentials is not an easy task, as they are defined on a continuous, high dimensional space. The original PM algorithm, optimizing only the unary energy, can be used to find an approximation of these local potentials. A benchmark can be seen in [Fig. 3](#), which shows that PMBP outperforms MP-PBP, with both types of initializations, and that convergence is orders of magnitude faster. Furthermore,

we show that resampling using the neighbours is the key step of our algorithm. To do so, we run PMBP with MCMC instead of using the PM randomisation mechanism, which in effect replicates MP-PBP, the only difference now being the use of the neighbours for resampling, and we see that although much slower than PMBP, it converges to the same energy.

4.2 Denoising

Denoising an image is an application where PM is expected to perform poorly, as it is only optimizing the unary term and thus will ultimately match the noise which is a behaviour observed in [Barnes et al. \(2010\)](#). We use this example application to illustrate the benefits of being able to optimize an energy comprising a pairwise term. In this experiment we aim at reconstructing a noisy target image from a noise-free, but slightly different, source image. The results can be seen in [Fig. 4](#). PM reaches its minimum energy after a few iterations (19 in our case) before starting producing worse results. We can see that final image still seems to contain noise. On the other hand, PMBP manages to produce a smoother solution (this is visible on both the reconstructed image and the displacement field) while reaching a higher PSNR.

We also use this application to show the effect of using more particles in PMBP. Results on a cropped region of the Goose example can be found in [Fig. 5](#). We see that, at an early stage, fewer particles yield a lower energy than using more particles. After enough processing time more particles yield a solution with slightly lower energy. However, the differences in energy between the solutions after 500 s is relatively small.

4.3 Stereo

In the following we demonstrate the benefits of introducing smoothness for the stereo matching case, and by doing so we are able to achieve state-of-the art results.

For the data term we use the same energy as in PM Stereo ([Bleyer et al. 2011](#)). The weight w_{sij} is defined as

$$w_{sij} = \exp(-\|I(x_s, y_s) - I(x_s + i, y_s + j)\| / \omega). \quad (17)$$

In this equation ω is a user-defined parameter and $\|I_s - I_t\|$ is the L_1 distance between s and t in RGB space. The image difference is adapted to include an image gradient term, so that $\|I_1(x, y) - I_2(x', y')\|$ in (3) is replaced by

$$(1 - \alpha) \min(\|I_1(x, y) - I_2(x', y')\|, \tau_{\text{col}}) + \alpha \min(\|\nabla I_1(x, y) - \nabla I_2(x', y')\|, \tau_{\text{grad}}), \quad (18)$$

where $\|\nabla I - \nabla I'\|$ is the L_1 -distance between the grey-level gradient, and α is a parameter controlling the influence of the colour and the gradient terms. τ_{col} and τ_{grad} are the truncated costs used to add robustness.

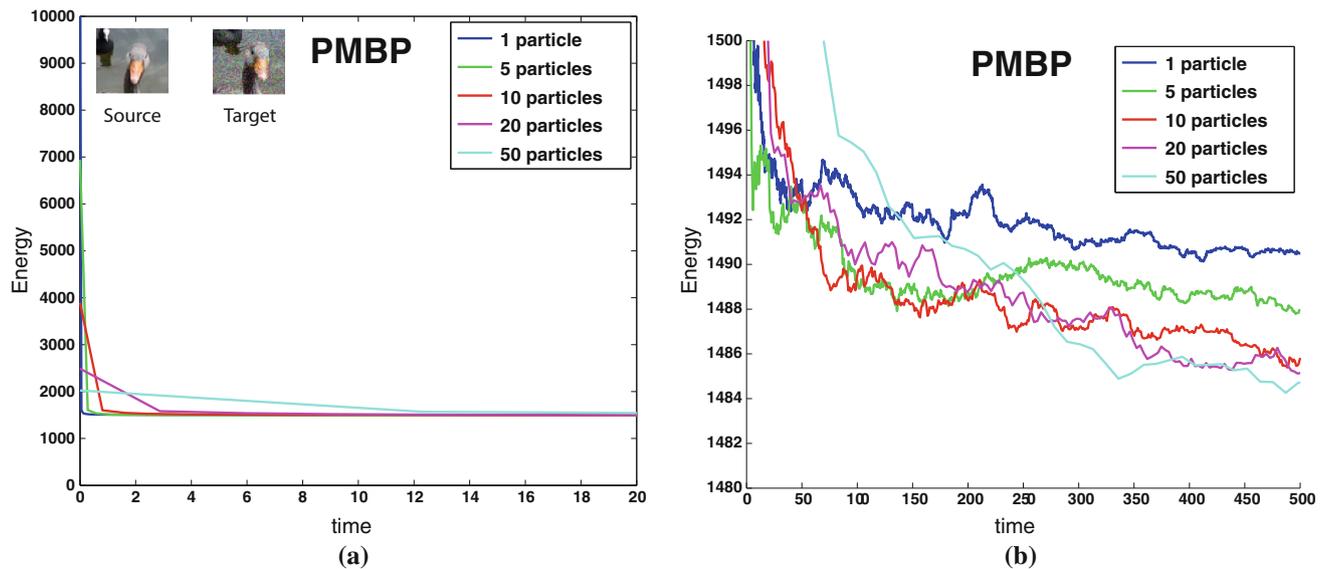


Fig. 5 Effect of using more particles. **a** First 20 s. **b** Close-up of the energy range 1,480–1,500 for the whole time range (0–500 s)

Table 2 Results on the Middlebury dataset with subpixel threshold ($t = 0.5$) as of 2012-05-10

	Tsukuba			Venus			Teddy			Cones		
	NOCC	All	Disc	NOCC	All	Disc	NOCC	All	Disc	NOCC	All	Disc
PM Stereo	15.0 (45)	15.4 (44)	20.3 (56)	1.00 (6)	1.34 (6)	7.75 (9)	5.66 (2)	11.8 (2)	16.5 (2)	3.80 (2)	10.2 (2)	10.2 (2)
Ours	11.9 (27)	12.3 (24)	17.8 (29)	0.85 (5)	1.10 (3)	6.45 (6)	5.60 (1)	12.0 (3)	15.5 (1)	3.48 (1)	8.88 (1)	9.41 (1)

Bold entries indicates where our algorithm is ranked first. Our method has the first rank, with an average rank of 8.5, in contrast to 14.9 for PatchMatch Stereo

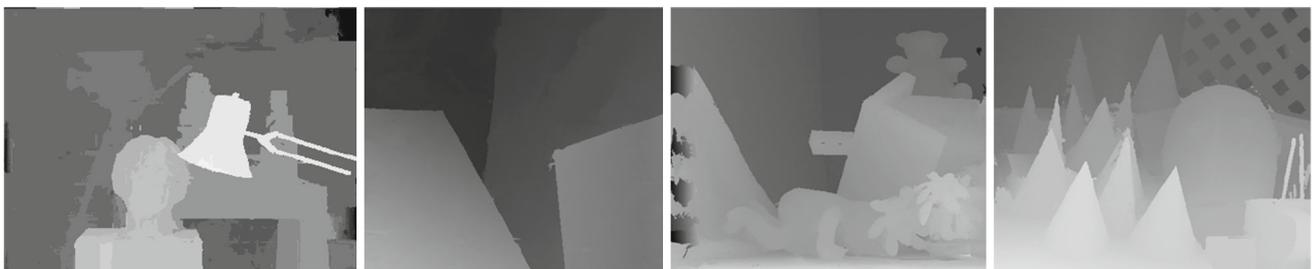


Fig. 6 Qualitative results of PMBP on the Middlebury dataset

The pairwise term captures the deviation between the two local planes in $(x, y, disparity)$ space. Let the plane normal at node s be $\mathbf{n}_s = \text{orth}([a_s, b_s, -1]^T)$, where $\text{orth}(v) := v/\|v\|$, and let $\mathbf{x}_s = [x_s, y_s, c_s]^T$ be a point on the plane. Then the pairwise energy is given by

$$\psi_{st}(\mathbf{u}_s, \mathbf{u}_t) = \beta w_{st} (|\mathbf{n}_s \cdot (\mathbf{x}_t - \mathbf{x}_s)| + |\mathbf{n}_t \cdot (\mathbf{x}_s - \mathbf{x}_t)|). \quad (19)$$

The data-dependent term w_{st} is defined as in Eq. 17 with $i = x_t - x_s$ and $j = y_t - y_s$. The weight β is a constant

weighting of the pairwise term with respect to the unary term. Note, for $\beta = 0$ we obtain PM Stereo.

The energy ψ^{pms} is augmented to symmetrize left and right views, and we label the left and right images in two consecutive steps. To be precise, the main loop at line 3 in Table 1 is first executed for the left view and then for the right view. Furthermore, as in the PM Stereo algorithm, we have implemented the concept of “view propagation”. The idea is that a good particle for a pixel s in the left view, may be in the particle set P_t of the corresponding (warped) pixel t in the right view, and vice versa. In terms of code, lines

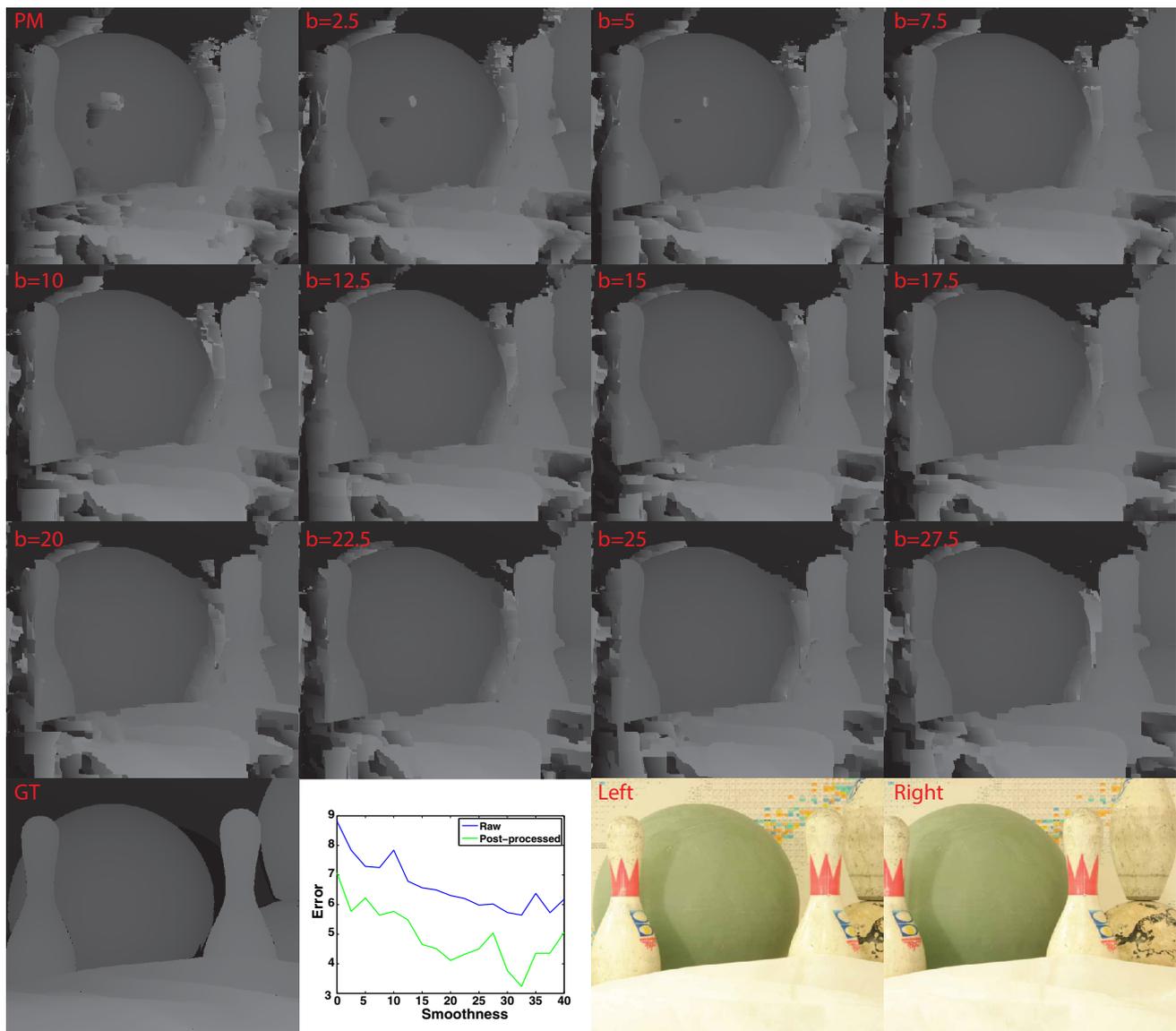


Fig. 7 Raw results on the Bowling1 dataset. b is a pairwise weighting coefficient, controlling the amount of smoothness

5–8 in Table 1 are duplicated, with the change that in line 5, the neighbourhood $N(s)$ is $t = (x_s + c_s, y_s)$. Finally, after optimizing the energy, there is a post-processing which is the same left-right consistency check as in Bleyer et al. (2011) in order to fill-in occluded pixels.

We use the same parameters as Bleyer et al. (2011), which are $\{\omega, \alpha, \tau_{col}, \tau_{grad}\} = \{10, 0.9, 10, 2\}$, with a larger patch size of 40×40 pixels. The weighting of the pairwise terms is set to $\beta = 7.5$.

We tested our algorithm on stereo pairs of the Middlebury dataset. We run our PMBP on the full energy and compare it to PM Stereo with no smoothness cost, i.e. $\beta = 0$. In both cases we use the same number of particles $K = 5$. The results are summarized in Table 2 and Fig. 6. We observe that we are superior to PM Stereo in all cases. For the sub-pixel

accuracy level, we are overall rank 1 of all methods. Note that we perform particularly well on the challenging datasets “Teddy” and “Cones”.

Figures 7 and 8 illustrate again the importance of the smoothness term. As expected, PM Stereo struggles in areas of low textures (e.g. middle of the bowling ball [top row], and white pages of the book [bottom row]). By increasing the weight β of the pairwise term, the output becomes increasingly smoother. Naturally, overshooting occurs after a certain point, which can be seen in Figs. 7 and 8 for large values of β .

A comment on the performance, with our settings and with five particles PMBP has a 20% overhead compared to PM, due to the message computations being more expensive.

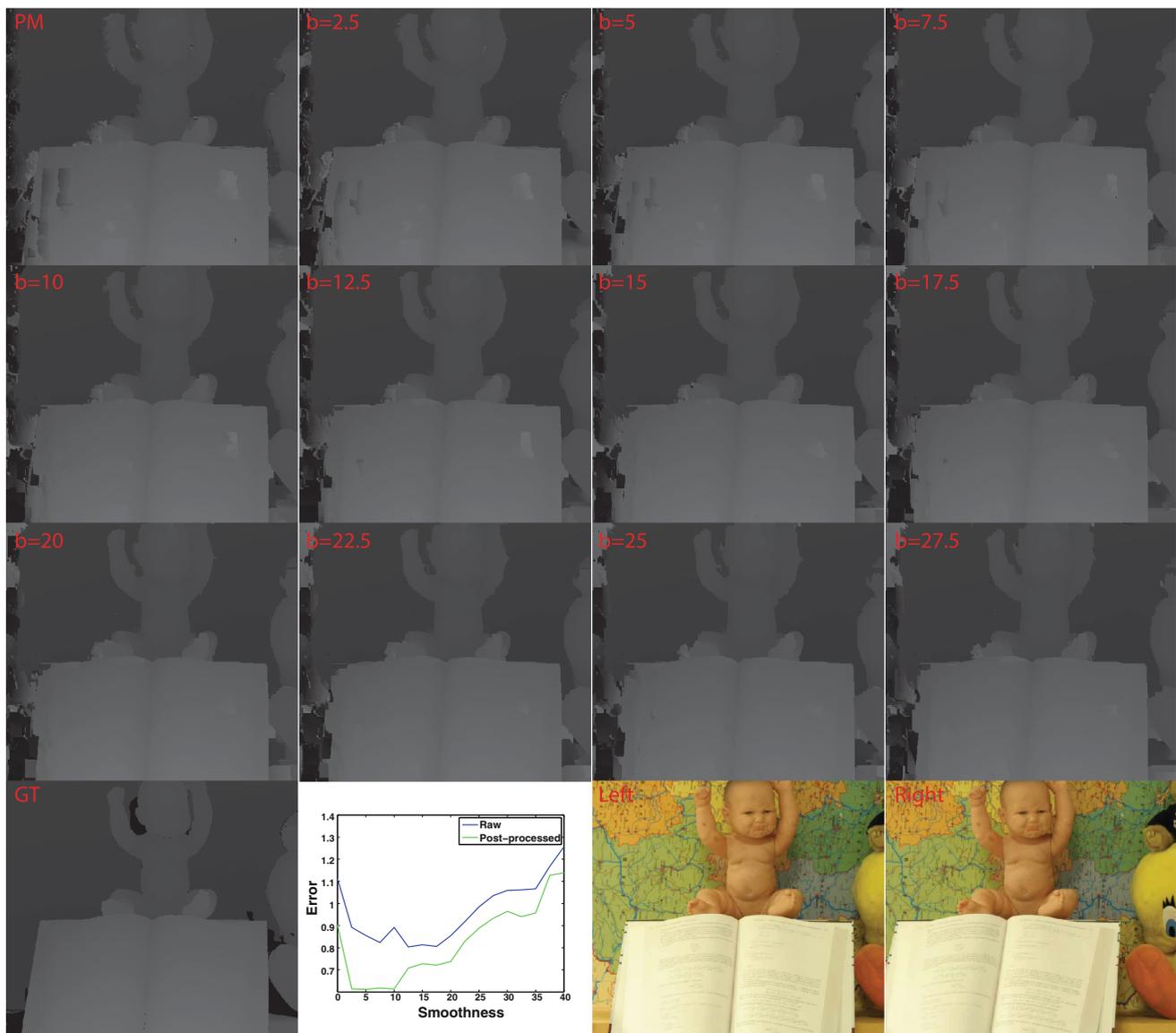


Fig. 8 Raw results on the Baby2 dataset. b is a pairwise weighting coefficient, controlling the amount of smoothness

5 Conclusion

In this work we have made the link between the popular PM method and the very well-known BP algorithm. By doing so, we were able to extend the PM algorithm by introducing additional pairwise terms. We validated experimentally that we achieve state-of-the-art results for stereo matching at sub-pixel accuracy level.

There are many exciting avenues for future work, both in terms of applications, such as optical flow, as well as algorithms, such as adapting PMBP to different forms of message passing e.g. tree-reweighted message passing (Kolmogorov 2006). Another interesting component to add to the algorithm would be a mechanism to ensure a certain amount of

diversity of the particles within each set, as in our current implementation it is possible to obtain several particles with the same, or similar states.

Acknowledgements We thank Christoph Rhemann and Michael Bleyer for their help with the PatchMatch Stereo code and also for fruitful discussions.

References

- Barnes, C., Shechtman, E., Finkelstein, A., & Goldman, D. B. (2009). PatchMatch: A randomized correspondence algorithm for structural image editing. *ACM Transactions on Graphics (Proceedings of SIGGRAPH)*, 28(3), 24.

- Barnes, C., Shechtman, E., Goldman, D. B., & Finkelstein, A. (2010). The generalized PatchMatch correspondence algorithm. In *Proceedings of ECCV*.
- Bleyer, M., Rhemann, C., & Rother, C. (2011). PatchMatch Stereo—Stereo matching with slanted support windows. In *Proceedings of BMVC*.
- Boltz, S., & Nielsen, F. (2010). Randomized motion estimation. In *Proceedings of ICIP* (pp. 781–784).
- HaCohen, Y., Shechtman, E., Goldman, D. B., & Lischinski, D. (2011). Non-rigid dense correspondence with applications for image enhancement. *ACM Transactions on Graphics (Proceedings of SIGGRAPH)*, 30(4), 70:1–70:9.
- He, K., Rhemann, C., Rother, C., Tang, X., & Sun, J. (2011). A global sampling method for alpha matting. In *Proceedings of CVPR* (pp. 2049–2056).
- Ihler, A., & McAllester, D. (2009). Particle belief propagation. In *Proceedings of AISTATS* (Vol. 5, pp. 256–263).
- Isard, M., MacCormick, J., & Achan, K. (2008). Continuously-adaptive discretization for message-passing algorithms. In *Proceedings of NIPS* (pp. 737–744).
- Kolmogorov, V. (2006). Convergent tree-reweighted message passing for energy minimization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(10), 1568–1583.
- Korman, S., & Avidan, S. (2011). Coherency sensitive hashing. In *Proceedings of ICCV* (pp. 1607–1614).
- Kothapa, R., Pachecho, J., & Sudderth, E. B. (2011). *Max-product particle belief propagation*. Master's Thesis, Brown University.
- Mansfield, A., Prasad, M., Rother, C., Sharp, T., Kohli, P., & Van Gool, L. (2011). Transforming image completion. In *Proceedings of BMVC*.
- Noorshams, N., & Wainwright, M. J. (2011). Stochastic belief propagation: Low-complexity message-passing with guarantees. In *Communication, Control, and Computing (Allerton), 2011 49th Annual Allerton Conference*. IEEE (pp. 269–276).
- Nowozin, S., & Lampert, C. (2011). *Structured learning and prediction in computer vision* (Vol. 6). Boston: Now publishers Inc.
- Pal, C., Sutton, C., & McCallum, A. (2006). Sparse forward-backward using minimum divergence beams for fast training of conditional random fields. In *Proceedings of ICASSP* (Vol. 5).
- Pearl, J. (1988). *Probabilistic reasoning in intelligent systems: Networks of plausible inference*. San Francisco: Morgan Kaufmann Publishers Inc.
- Peng, J., Hazan, T., McAllester, D. A., & Urtasun, R. (2011). Convex max-product algorithms for continuous MRFs with applications to protein folding. In *Proceedings of ICML*.
- Sudderth, E. B., Ihler, A. T., Isard, M., Freeman, W. T., & Willsky, A. S. (2010). Nonparametric belief propagation. *Communications of the ACM*, 53(10), 95–103.
- Yamaguchi, K., Hazan, T., McAllester, D., & Urtasun, R. (2012). Continuous Markov random fields for robust stereo estimation. In *Computer Vision—ECCV 2012* (pp. 45–58). Berlin, Heidelberg: Springer.
- Yedidia, J., Freeman, W., & Weiss, Y. (2005). Constructing free-energy approximations and generalized belief propagation algorithms. *IEEE Transactions on Information Theory*, 51(7), 2282–2312.