Belief-Propagation on Edge Images for Stereo Analysis of Image Sequences

Shushi Guan and Reinhard Klette

Computer Science Department, CITR, The University of Auckland
Private Bag 92019, Auckland 1142, New Zealand

Abstract. The history of stereo analysis of images dates back more than one hundred years, but stereo analysis of image sequences is a fairly recent subject. Sequences allow time-propagation of results, but also come with particular characteristics such as being of lower resolution, or with less contrast. This article discusses the application of belief propagation (BP), which is widely used for solving various low-level vision problems, for the stereo analysis of night-vision stereo sequences. For this application it appears that BP often fails on the original frames for objects with blurry borders (trees, clouds, . . .). In this paper, we show that BP leads to more accurate stereo correspondence results if it is applied on edge images, where we have decided for the Sobel edge operator, due to its time efficiency. We present the applied algorithm and illustrate results (without, or with prior edge processing) on seven, geometrically rectified night-vision stereo sequences (provided by Daimler AG, Germany).

Keywords: Stereo analysis, belief propagation, Sobel operator, image sequence analysis.

1 Introduction

Stereo algorithms often use a Markov Random Field (MRF) model for describing disparity images. The basic task of an MRF algorithm is then to find the most likely setting of each node in an MRF by applying an inference algorithm. Belief propagation (BP) is one of the possible inference algorithms; it can be applied for calculating stereo disparities, or for other labeling processes defined by finite sets of labels. [5] shows that BP is such an algorithmic strategy for solving various problems. BP is recommended for finding minima over large neighborhoods of pixel, and it produces promising results in practice (see, e.g., evaluations at website *http://vision.middlebury.edu/*). Performance analysis is an important issue in computer vision [6].

A belief propagation algorithm applies a sum-product or max-product procedure, and in this paper we choose the max-product option. The max-product procedure computes the Maximum A-Posteriori (MAP) estimate over a given MRF [11].

[7] reports about the general task of evaluating stereo and motion analysis algorithms on a given test set of seven rectified night-vision image sequences

(provided by Daimler AG, Germany [4]). In this article we consider the application of BP algorithms on these seven sequences, each defined between 250 and 300 pairs of stereo frames. We show that a straight application of BP fails, but it leads to promising results after prior application of an edge operator.

The article is structured as follows. In Section 2, we briefly introduce the BP algorithm, which includes the definition of an energy function, max-product, message passing, Potts model, and also of some techniques to speed up the standard BP algorithm, following [3]. In Section 3 we calculate Sobel edge images, and show that subsequent BP analysis leads to improvements compared to results on the original sequences, verified by results for those seven test sequences mentioned above. Some conclusions are presented in Section 4.

2 BP Algorithm

Solving the stereo analysis problem is basically achieved by pixel labeling: The input is a set P of pixel (of an image) and a set L of labels. We need to find a labeling

$$f : P \rightarrow L$$

(possibly only for a subset of P). Labels are, or correspond to disparities which we want to calculate at pixel positions. It is general assumption that labels should vary only smoothly within an image, except at some region borders. A standard form of an energy function, used for characterizing the labeling function f, is (see [1]) as follows:

$$E(f) = \sum_{p \in P} D_p(f_p) + \sum_{p,q \in A} V_{p,q}(f_p, f_q)$$

Since we aim at minimizing the energy, this approach corresponds to the Maximum A-Posteriori (MAP) estimation problem.

$D_p(f_p)$ is the *cost of assigning a label* f_p to pixel p. We use the differences in intensities between corresponding pixel (i.e., defined to be corresponding when applying disparity f_p). To be precise, in our project we use absolute differences. A is an assumed symmetric and irreflexive adjacency relation on P.

Each pixel p (say, in the left image at time t) may take one disparity at that position, out of a final subset of L (e.g., defined by a maximum disparity). The corresponding pixel is then in the right image at time t. Because the given image sequences are rectified, we can simply search in identical image rows. The given gray-level (or intensity) images allow that differences in gray-levels define the cost $D_p(f_p)$. The smaller an intensity difference, the higher the compatibility.

$V_{p,q}(f_p, f_q)$ is the cost of assigning labels f_p and f_q to two adjacent pixel p and q, respectively. It represents a *discontinuity cost*. The cost V in stereo analysis is typically defined by the difference between labels; each label is a non-negative disparity. Thus, it is common to use the formula

$$V_{p,q}(f_p, f_q) = V(f_p - f_q)$$

The resulting energy is (see [3]) as follows:

$$E(f) = \sum_{p \in P} D_p(f_p) + \sum_{p,q \in A} V_{p,q}(f_p - f_q)$$

The task is a minimization of $E(f)$.

2.1 Max-product

A max-product algorithm is used to approximate the MAP solution to MRF problems. The max-product algorithm is guaranteed to converge, and also guaranteed to give the optimal assignment values to the MAP solution, based on the message at time of convergence [10]. The max-product BP algorithm is defined by iterations of passing a message (the *belief*) around in the image grid. Typically, (and also in our tests) 4-adjacency is assumed. The message update schedule is in iterations, and messages pass from node to node in parallel.

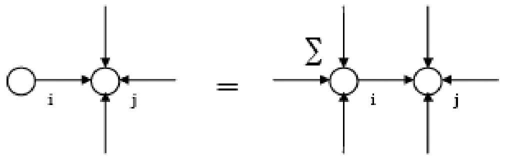

Fig. 1. Illustration for a messages update in the graph: the circles represent pixel in an image, and the arrows indicate directions of message passing.

Figure 1 illustrates message passing in the adjacency graph of the given pixel. If node i is left of node j then node i sends a message to node j at each iteration. The message from node i contains the messages already received from its neighbors. In parallel, each node of the adjacency graph computes its message, and then those messages will be sent to adjacent nodes in parallel. Based on these received messages, we compute the next iteration of messages. In other words, for each iteration, each node uses the previous iteration's messages from adjacent nodes, in order to compute its messages send to those neighbors next. Meanwhile, the larger $D_p(f_p)$ is, the more difficult it is to pass a message to an adjacent node. That means, the influence of an adjacent node decreases when the cost at this node increases.

Each message is represented as an array; its size is determined by the maximum disparity (assuming that disparities start at zero, and are subsequent integers), denoted by K.

Assume that $m^t_{p \to q}$ is the message, send from node p to adjacent node q at iteration t. For each iteration, the new message is now given by (see [3]) the following:

$$m^t_{p\to q}(f_p) = \min_{f_p}\left(V_{p,q}(f_p - f_q) + D_p(f_p) + \sum_{s\in A(p)\backslash q} m^{t-1}_{s\to q}(f_p)\right)$$

where $A(p)\backslash q$ is the adjacency set of p except node q. The message array contains at its nodes the following

$$b_q(f_q) = D_q(f_q) + \sum_{p\in N(q)} m^t_{p\to q}(f_q)$$

after T iterations; see [3]. Each iteration computes $\mathcal{O}(n)$ messages, where n is the cardinality of set P.

2.2 Potts model

The Potts model[1] is a method for minimizing energy; see, for example, [2]. In this model, discontinuities between labels are penalized equally, and we only consider two states of nodes: equality and inequality. We measure differences between two labels, the cost is 0 if the labels are the same; the cost is a constant otherwise. Let the cost function between labels be $V(x_i, x_j)$. Then, (see [9]) we have that

$$V(x_i, x_j) = \begin{cases} 0 & \text{if } x_i = x_j \\ d & \text{otherwise} \end{cases}$$

The Potts model is useful when labels are "piecewise constant", with discontinuities at borders. It was suggested to apply this cost function to the message update formula. The formula is now rewritten (see [3]) as follows

$$m^t_{p\to q}(f_p) = \min_{f_p}\Big(V_{p,q}(f_p - f_q) + h(f_p)\Big)$$

where

$$h(f_p) = D_p(f_q) + \sum_{s\in A(p)\backslash q} m^{t-1}_{s\to p}(f_p)$$

This form is very similar to that of a minimum convolution. After applying the cost function, the minimization over f_p yields a new equation in the following way (see [3]):

$$m^t_{p\to q}(f_p) = \min\left(h(f_p), \min_{f_p} h(f_p) + d\right)$$

Except the f_p minimization, the message computation time reduce to $\mathcal{O}(K)$, see [3]. At first we compute $\min_{f_p} h(f_p)$, then we use the result to compute the message in constant time.

[1] Described in the 1952 Ph.D. by R. B. Potts.

2.3 Speed-up techniques

In this section, we recall some techniques that may be used to speed up a BP analysis. We start with a technique called multi-grid method in [3]. This technique allows to obtain good results with just a small number of message passing iterations.

[8] shows that message propagation over long distances takes more message update iterations. To circumvent this problem, a common data pyramid was used. (All nodes in one 2×2 array in one layer are adjacent to one node in the next layer; all $2^n \times 2^n$ pixel (nodes) at the bottom layer zero are connected this way with a single node at layer n). Using such a pyramid, long distances between pixels are shortened, what makes message propagation more efficient. We do not reduce the image resolution, but aggregate data over connections defined in this pyramid. Such a coarse to fine approach allows that a small number of iterations (dependent on the level of the pyramid) is sufficient for doing a BP analysis.

The red-black algorithm provides a second method for speeding up BP; see [3]. The message passing scheme adopts a red-black algorithm which allows that only half of all messages are updated at a time. Such a red-black technique is also used for Gauss-Seidel relaxations. A Gauss-Seidel relaxation attempts to increase the convergence rate by using values computed for the kth iteration in subsequent computations within the kth iteration.

We can think of the image as being a checkerboard, so each pixel is differently "colored" compared to its adjacent pixel. The basic idea is that we just update the message sent from a "red" pixel to a "black" pixel at iteration t; in the next iteration $t+1$, we only update the message sent from a "black" pixel to a "red" pixel.

We recall this message updating scheme at a more formal level: assume that B and C represent nodes of both defining classes in a bipartite graph, at iteration

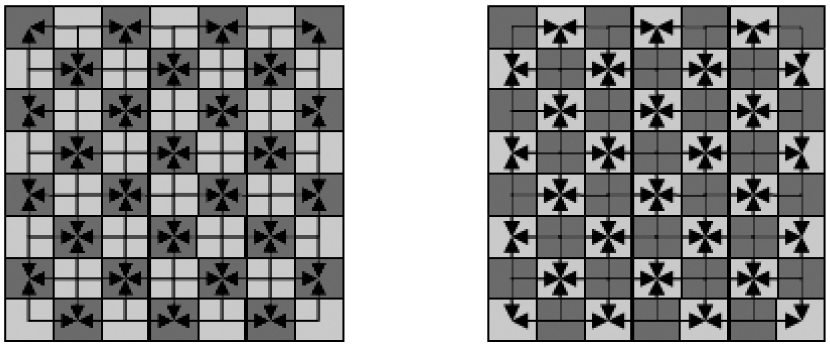

Fig. 2. These two images illustrate the message passing scheme under a red-black algorithm; the left image shows messages only send from a "black" (dark gray) to a "red" (light gray) pixel at iteration t; at iteration $t+1$, "red" sends messages back to "black" in the right image.

t, we know message M_1 which is sent from nodes in B to those in C; based on message M_1, we can compute message M_2 sent from nodes in C to those in B at iteration $t+1$. That means, we can compute message M_3 from nodes in B at iteration $t+2$ without knowing the message send from nodes in B at iteration $t+1$. Thus, we only compute half the messages at each iteration. See Figure 2 for further illustration. This alternating message updating algorithm is described as follows:

$$\overline{m} = \begin{cases} m^t_{p\to q} & \text{if } p \in B \\ m^{t-1}_{p\to q} & \text{otherwise} \end{cases}$$

This concludes the specification of the BP algorithm as implemented for our experiments. We aimed at using a standard approach, but with particular attention of ensuring time efficiency.

3 Experiments

The seven image sequences used for stereo analysis are as described in [7], provided by Daimler AG, Germany (for a download of those sequences, see *citr.auckland.ac.nz/6D*). These night vision stereo sequences are geometrically rectified. The following figures address each exactly one of those seven sequences, and each figure show one unprocessed representative frame of the sequence at its upper left. (We provide frame and sequence number.)

These seven sequences are already geometrically rectified, and stereo matching is reduced to a search for corresponding pixels along the same horizontal line in both images. For explaining the following test results, the used test environment was as follows: AMD 64Bit 4600 2.4GHz, 2 Gigabyte memory, NVIDIA Geforce 7900 video card, WinXP operation system.

The following figures use a uniform scheme of presentation, and we start with explaining for Figure 3. A sample of the original, unprocessed sequence is shown in the upper left. The described BP stereo analysis algorithm was applied, and the resulting depth map is shown in the upper right. The maximum disparity for the illustrated pair of images (being "kind of representative" for the given stereo sequence) is 70, and 7 message iterations have been used. Table 1 shows these values, also the size of the used area in the given 640×480 frames (called "image size" in this table), which were the parameters used in our test, and finally the run time rounded to seconds. Note that no particular efforts have been made for run-time optimization besides those mentioned in the previous section.

The experiments indicated (quickly) two common problems in stereo matching, namely bad matching due to lack of texture (such as at the middle of the road), and mismatching due to "fuzzy depth discontinuities" (such as in sky or in trees).

For the other six image sequences, see the original frame of the stereo pair, reported further in Table 1, ate the upper left of Figures 4 to 9. The depth maps shown at the upper right (BP analysis as described, for the original sequence) all shows similar problems.

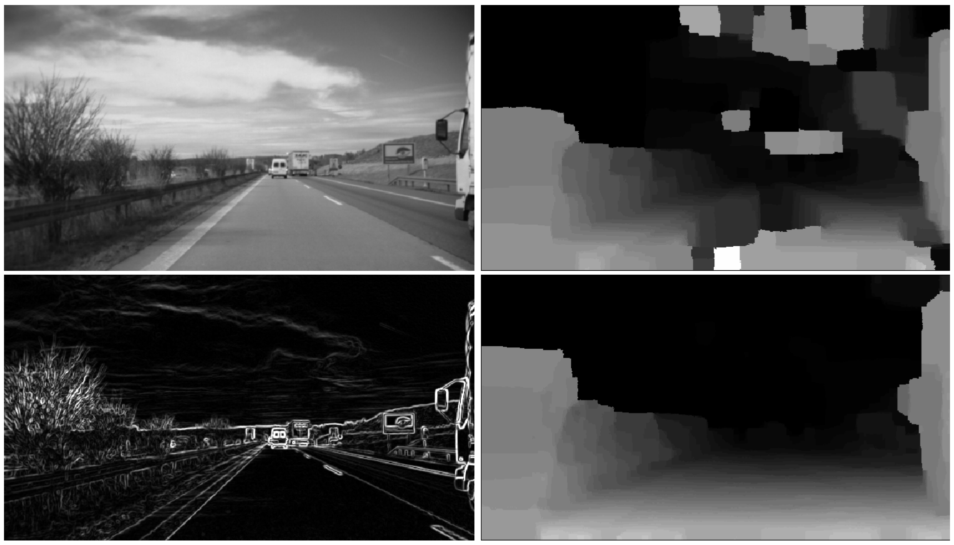

Fig. 3. Image 0001_c0 of sequence 1 (upper left) and its associated BP result (upper right). The Sobel edge image (lower left) and the corresponding BP result (lower right).

As a result of our analysis of those problems, we expected that the use of some edge enhancement (contrast improvement) could support the message passing mechanism. Surprisingly, we can already recommend the use of the simple Sobel edge operator. The resulting Sobel edge image are certainly "noisy", but provide borders or details of the original images which allow the message passing mechanism to proceed more in accordance to the actual data.

See the Sobel edge images (for the selected "key frames" of those seven sequences) at the lower left of Figures 3 to 9.

The lower right images in Figures 3 to 9 show the subsequent result of the specified BP analysis algorithm on Sobel edge images. Obviously, the new result is much better than our preliminary result (upper right images). In general

Figure	Max-disparity	Iterations	Image size	Running time
3	70 *pixel*	7	633 × 357 *pixel*	9 *s*
4	55 *pixel*	7	640 × 353 *pixel*	7 *s*
5	40 *pixel*	5	640 × 355 *pixel*	4 *s*
6	60 *pixel*	7	640 × 370 *pixel*	8 *s*
7	30 *pixel*	5	631 × 360 *pixel*	3 *s*
8	35 *pixel*	6	636 × 356 *pixel*	4 *s*
9	40 *pixel*	5	636 × 355 *pixel*	4 *s*

Table 1. Table of parameter used for BP algorithm and program running time.

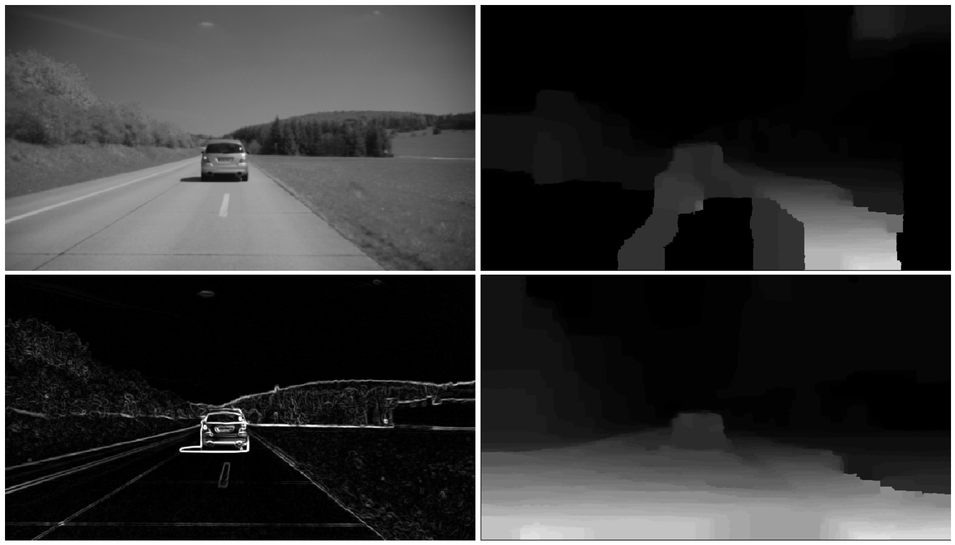

Fig. 4. Image 0001_c0 of sequence 5 (upper left) and its associated BP result (upper right). The Sobel edge image (lower left) and the corresponding BP result (lower right).

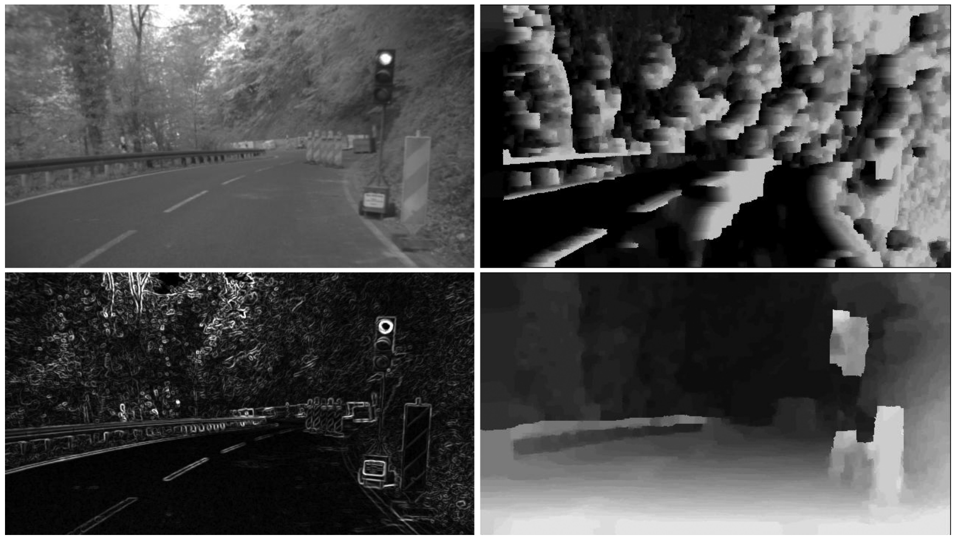

Fig. 5. Image 0001_c0 of sequence 6 (upper left) and its associated BP result (upper right). The Sobel edge image (lower left) and the corresponding BP result (lower right).

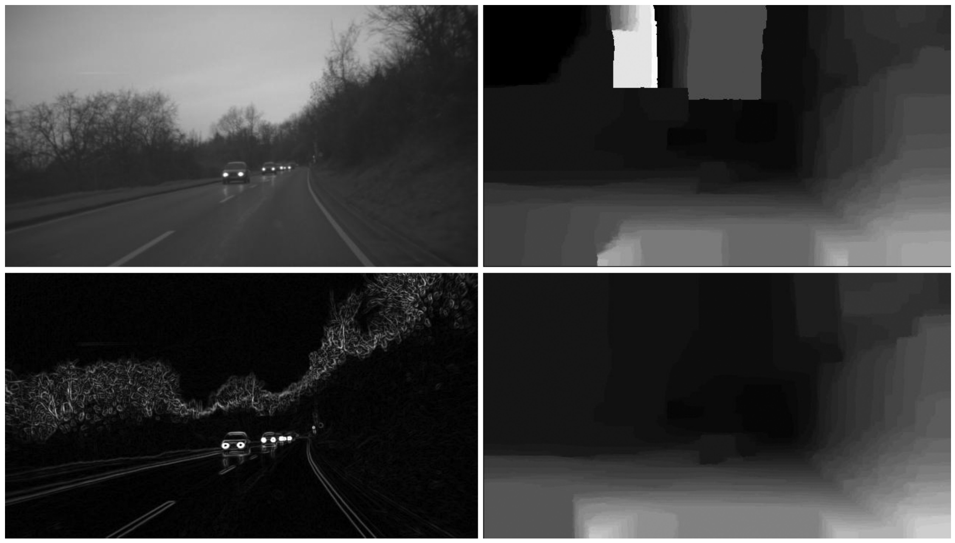

Fig. 6. Image 0001_c0 of sequence 3 (upper left) and its associated BP result (upper right). The Sobel edge image (lower left) and the corresponding BP result (lower right).

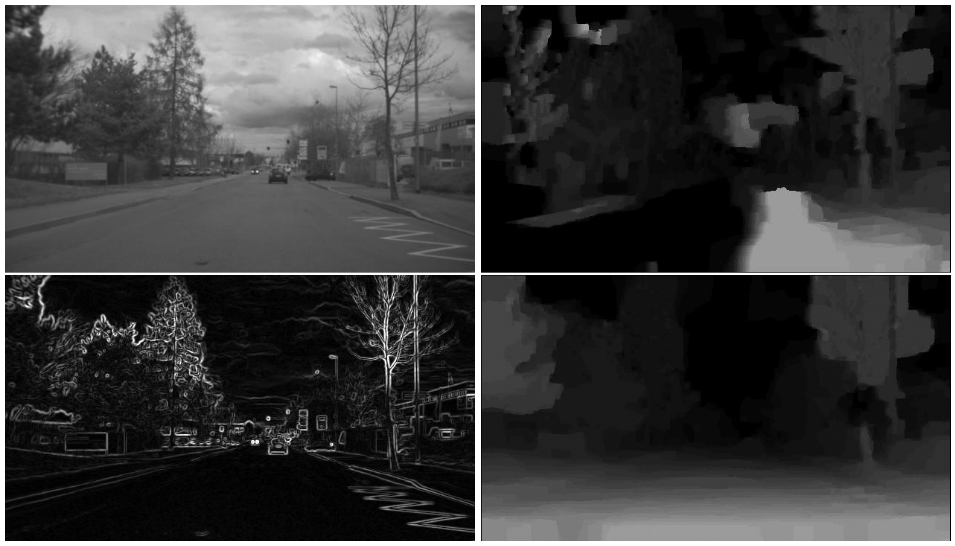

Fig. 7. Image 0001_c0 of sequence 2 (upper left) and its associated BP result (upper right). The Sobel edge image (lower left) and the corresponding BP result (lower right).

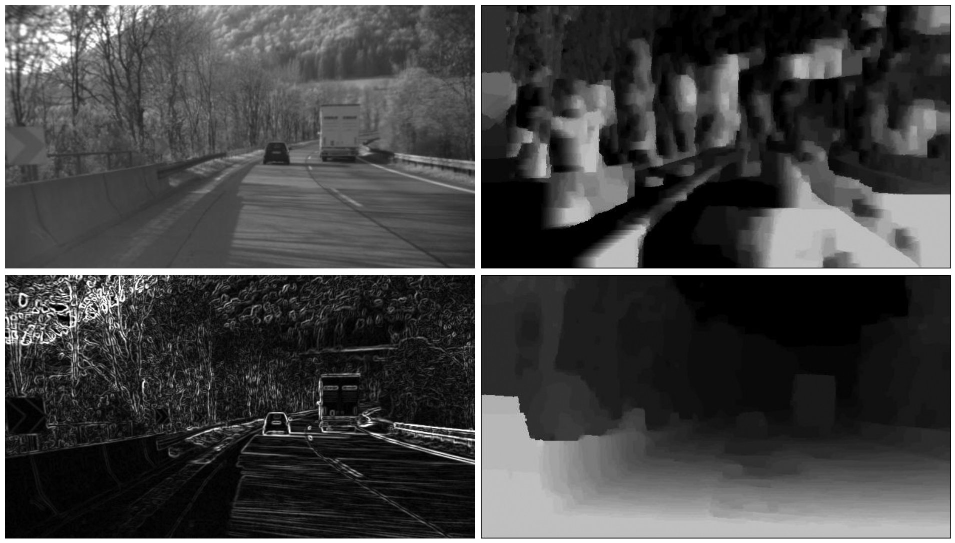

Fig. 8. Image 0227_c0 of sequence 4 (upper left) and its associated BP result (upper right). The Sobel edge image (lower left) and the corresponding BP result (lower right).

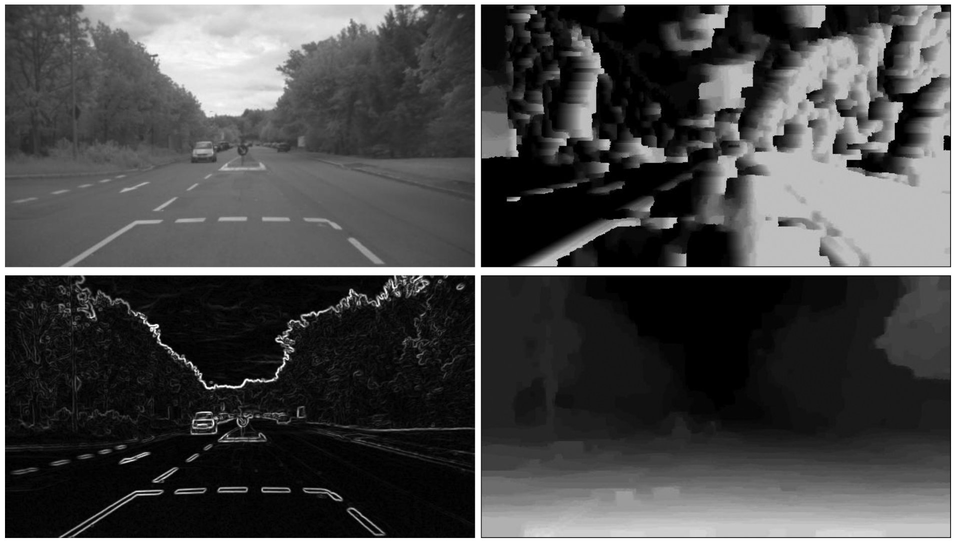

Fig. 9. Image 0184_c0 of sequence 7 (upper left) and its associated BP result (upper right). The Sobel edge image (lower left) and the corresponding BP result (lower right).

comparison with the BP analysis for the original image pairs of those seven sequences, major discontinuities are now often correctly detected.

For example, the visual border of a tree may be recovered despite of an obvious fuzziness of its intensity edge. Especially the road and the sky are now often accurately located. In most cases, a car is also detected if at a reasonable distance. But there are still remaining problems.

For example, in Figure 5, the traffic light is not matching correctly, we can see that there are two traffic lights in the depth map. The use of the ordering constraint could help in such circumstances.

In some images, we can not identify many depth details especially in images with lots of trees. See again Figure 5 for an example. Vertical edges disappeared in the depth map image. The reason might be that we have chosen a small discontinuity penalty only (see Section 2.2, the Potts model) to do these tests illustrated in the figures.

When using a higher discontinuity penalty in BP, this produces more edges or details in depth maps, but also more noise or matching errors.

Adaptation might be here a good subject, for identifying a balance point. In Figure 8, the road is in the depth map (lower right) not a smooth, even, leveled surface; this is caused by the shadows of the trees on the road which cause about horizontal stripes in the images. This means that the pixel's intensity in an epipolar line is about constant, what makes mismatching more easy.

4 Conclusions

In this paper, we proposed the use of a simple edge detector prior to using belief propagation for stereo analysis. The proposed method is intended for BP stereo correspondence analysis where borders in given scenes or images are fuzzy. We detailed the used BP algorithm by discussing the max-product algorithm of belief propagation, and how messages propagate in the graph, especially also under the circumstances of the used two run-time optimization strategies. One of both techniques reduced the number of message passing iterations, the second technique halved message computation.

Recently we integrate the ordering constraint in the BP algorithm, and we also plan to design an adaptive algorithm which calculates discontinuity penalties based on image intensities of frames.

The provided image sequences allowed a much more careful analysis than just by using a few image pairs. This improved the confidence in derived conclusions, but also showed more cases of unsolved situations. This is certainly just a beginning of utilizing such a very useful data set for more extensive studies.

Acknowledgment: We acknowledge the support of Daimler AG, Germany (research group of Uwe Franke) of providing those seven rectified night-vision stereo sequences. Tobi Vaudrey (Auckland, on a research internship at Daimler AG) prepared those sequences, and his careful work and good selection of different scenarios is appreciated.

References

1. Y. Boykov and V. Kolmogorov. An experimental comparison of min-cut / max-flow algorithms for energy minimization in vision. *IEEE Trans. Pattern Analysis Machine Intelligence*, **26**:1124–1137, 2004.
2. Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. *IEEE Trans. Pattern Analysis Machine Intelligence*, **23**:1222–1239, 2001.
3. P. F. Felzenszwalb and D. P. Huttenlocher. Efficient belief propagation for early vision. *Int. J. Computer Vision*, **70**:41–54, 2006.
4. U. Franke, C. Rabe, H. Badino, and S. Gehrig. 6D-vision - fusion of stereo and motion for robust environment perception. In Proc. *DAGM*, pages 216–223, 2005.
5. B. J. Frey, R. Koetter, and N. Petrovic. Very loopy belief propagation for unwrapping phase images. In *Advances in Neural Information Processing Systems 14* (T. G. Dietterich, S. Becker, and Z. Ghahramani, editors), Cambridge, Massachusetts, MIT Press, 2001.
6. R. Klette, S. Stiehl, M. Viergever, and V. Vincken (editors). *Performance Evaluation of Computer Vision Algorithms.* Kluwer Academic Publishers, Amsterdam, 2000.
7. Z. Liu and R. Klette. Performance evaluation of stereo and motion analysis on rectified image sequences. CITR-TR-207, The University of Auckland, Computer Science Department, Auckland, 2007.
8. J. Sun, N. N. Zheng, and H. Y. Shum. Stereo matching using belief propagation. *IEEE Trans. Pattern Analysis Machine Intelligence*, **25**:787–800, 2003.
9. M. F. Tappen and W. T. Freeman. Comparison of graph cuts with belief propagation for stereo, using identical MRF parameters. In Proc. *IEEE Int. Conf. Computer Vision*, pages 900–907, 2003.
10. Y. Weiss and W. T. Freeman. On the optimality of solutions of the max-product belief propagation algorithm in arbitrary graphs. *IEEE Trans. Information Theory*, **47**:723–735, 2001.
11. J. Yedidia, W. T. Freeman, and Y. Weiss. Understanding belief propagation and its generalizations. In *Exploring Artificial Intelligence in the New Millennium* (G. Lakemeyer and B. Nebel, editors), pages 239–236, Morgan Kaufmann, San Mateo, 2003.