Efficient Message Reduction Algorithm for Stereo Matching Using Belief Propagation

Yen-Chieh Lai[1], *Chao-Chung Cheng*[1], *Chia-Kai Liang*[2], *and Liang-Gee Chen*[1]

[1]Graduate Institute of Electronics Engineering, National Taiwan University, Taiwan
[2]Army of Taiwan, R.O.C.

ABSTRACT

Belief propagation (BP) is a popular global optimization technique in computer vision. However, it requires huge bandwidth and memory in hardware implementation because it iteratively processes messages between the neighboring nodes. In this paper, we propose an efficient message reduction algorithm that greatly reduces the bandwidth and memory consumption. Compared with the original message passing operation, we successfully reduce the memory and bandwidth with similar quality. For stereo matching of a VGA input where the disparity range is 64, the proposed algorithm can achieve 93.75% message memory reduction with about only 0.2-2.2% bad pixels quality degradation. The proposed algorithm greatly reduces the memory requirement and is suitable both for hardware and software realization.

Index Terms— Belief propagation, global optimization, stereo matching.

1. INTRODUCTION

Stereo vision is a popular issue in computer vision. Stereo vision can be applied in many fields, such as robotic, 3D video signal processing, etc. We can apply stereo matching technique to acquire the depth information from a stereo image pair. In computer vision and image processing, the stereo matching is regarded as the problem of finding an optimal disparity vector assignment for every pixel. Finding an optimal assignment can be formulated in the Markov random field (MRF) framework as an energy minimization problem. The energy minimization problem can be defined as:

$$\{l_p\} = \arg\min\left\{\sum_{p\in P} E_d\left(l_p\right) + \sum_{(p,q)\in G} E_s\left(l_p, l_q\right)\right\}, \tag{1}$$

where P is the set of all nodes and G is the specified neighborhood system. We assign the optimal label set $\{l_p\}$ to minimize the energy function. The energy function has two terms: a data term E_d and a pairwise smoothness term E_s. E_d is the cost to penalize the inconsistence between the disparity value and the observed data. E_s is the discontinuity cost between the neighborhood nodes.

Several efficient global optimization techniques such as graph cuts [1] and belief propagation [2][3](BP) can find the stronger optimal solution than local methods. BP has high potential in hardware implementation: it is highly parallel and only uses simple arithmetic operations. However, it requires huge memory and bandwidth due to the iterative message passing operation. In the hardware implementation, it is prohibitive due to enormous memory and bandwidth.

In the previous work, we have proposed a tile-based belief propagation algorithm [6][7]. By only storing and passing the boundary`s messages of the tiles, tile-based belief propagation algorithm greatly reduces the memory and bandwidth requirements and still converges to a solution of similar quality. In [7][8], a fast message computation method is explored to reduce the computation complexity.

After these algorithms are employed, we found that the remaining major bottlenecks for hardware implementation are *on-chip memory consumption* and *on-chip bandwidth consumption*. It still needs huge on-chip memory and bandwidth to store the messages between nodes. To conquer this problem, we propose a new message passing operation to reduce the memory and bandwidth consumption.

The rest of the paper is organized as follows. In Section 2, we review tile-based belief propagation. In Section 3, we present the proposed message passing operation in detail. In Section 4, we perform the proposed message passing operation to the stereo matching and compare the result with the original message passing operation. Finally, we conclude this paper in Section 5.

2. TILE-BASED BELIEF PROPAGATION

Belief propagation iteratively performs the message passing operation between nodes. In message passing operation, each node p sends a $|L|$-dimensional message M_{pq}^t to its neighbor q at the iteration t. Each entity $M_{pq}^t(l)$ in the message is defined by (Figure 1(a)):

$$M_{pq}^{t}\left(l_q\right) = \min_{l'\in L}\left(E_s\left(l_q, l'\right) + E_d\left(l'\right) + \sum_{(p,p')\in N_p\setminus q} M_{pp'}^{t-1}\left(l'\right)\right), \tag{2}$$

where L is the set of all labels, $|L|$ is the number of labels, and N_p denotes the neighbors of node p. Node p first scans all labels l' and decides which one having the greatest sup-

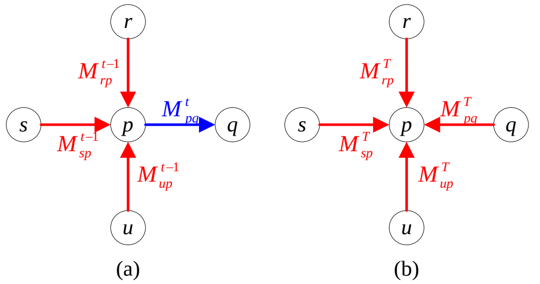


Figure 1. (a) A message at iteration *t* from node *p* to node *q* is constructed by using the messages from node *r*, *s*, and *u* to node *p* at iteration *t – 1*. (b) The node *p* collects all messages from the neighbors to decide the best label.

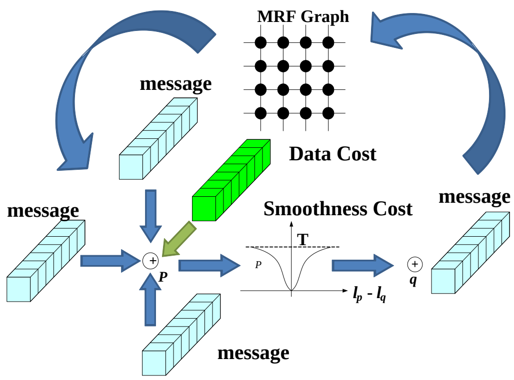


Figure 2. The illustration of the original message passing operation.

port for assigning *l'* to *p* based on the smoothness cost, the data cost and the messages from the neighbors of node *p* except node *q* (Figure 2). During BP, all nodes exchange messages about the label assignment and through iterations. The messages are iteratively propagated. At each iteration, each node loads three-dimensional messages and generate the outgoing message to propagate to the neighbored node. After enough number of iterations, say *T*, we utilize min-sum algorithm to determine the label of *p* based on the local likelihood and the messages from neighbors (Figure 1.(b)):

$$l_p = \arg\min_{l\in L}\left\{E_d(l) + \sum_{(p,p')\in N_p} M^{t-1}_{p'p}(l)\right\}. \qquad (3)$$

In before work, we propose tile-based belief propagation performs the message passing procedure locally at a small tile [7][8]. After the message passing procedure is performed inside the tile, we abandon all messages inside the tile but only save the outgoing boundary messages for the next tile. In this way, the tile-based BP can not only maintain the accuracy in performance but also greatly reduce the memory and bandwidth consumption.

However, during performing message passing inside the tile, the tile-based BP still have to store all messages within the current tile. In the next section, we propose a new message passing operation which can further reduce the memory and bandwidth consumption.

3. MESSAGE REDUCTION ALGORTIHM

In this section, we propose a new message passing operation which provides similar performance but requires much less memory and bandwidth compared with the original message passing operation. We divide the proposed message passing operation into two stages: preservation of minimal label messages and reconstruction of the unknown messages.

3.1 Preservation of minimal label messages

In the proposed message passing operation, we pass fewer message elements to reduce the memory size and bandwidth but still preserve the accuracy in message representation. The main idea is to preserve the *n* minimal message elements. Each node will select a candidate with minimal cost value. Since the labels with less cost value have higher probability to be selected as a best candidate, we keep the minimal *T* candidate labels to reduce the memory and bandwidth complexity, typically $T = 4$ is enough for $L = 64$. It means only four candidates are needed to be transferred and stored rather than 64 in the proposed algorithm.

In selecting the best preserved labels for efficient representation for messages, we use the min-sum algorithm. Generally speaking, the messages with smaller value are more important messages. After calculating the message of one node, all the messages are sorted from the minimum to maximum. Depending on the system resource, the *n* minimum messages are transferred to the temporal message buffer. The message buffer only requires the number of the preserved minimum label messages. Moreover, we can select the number of preserved minimum label messages dynamically to achieve the flexible message reduction ratio under different system resource and quality consideration.

3.2 Reconstruction of the unknown messages

Before computing the message for the next node, because we do not have all message elements in the temporal message buffer, the missing elements in the incoming messages should be reconstructed. Therefore, a model is proposed to reconstruct the other incoming message elements based on the preserved ones.

In the message reconstruction process, we assume that the reduced message element has an absolute distance from the preserved ones, and the distance is based on the difference between the labels. Therefore, we assign the absolute distance being linearly increasing by the absolute difference between the labels and bounded by a constant T_s.

$$M_{con}(l_q) = M_{pre}(n) + \lambda \cdot \min\left(\left|l(n) - l_q\right|, T_s\right), \qquad (4)$$

Function Reconstruct_messages($M_{incoming}$, $I_{incoming}$, $M_{preserved}[t]$, *min_para*)
$M_{incoming} = INT\_MAX;$
for $t = 0, 1, \ldots,$ *min_para*
$I_{preservedv}[t]$ = the label index of $M_{preserved}[t]$;
$M_{constructedv}[t] = M_{preserved}[t] + \lambda * V[I_{incoming}][I_{preserved}[t]]$;
$M_{incoming} = \min(M_{incoming}, M_{constructed}[t])$;
end for
**min_para* denotes the number of preserved messages in sorting list, $V[\,][\,]$ is the distance between label function truncated by threshold T_s. We set λ as 20 and T_s as 2 in our implementation.

Figure 3. The pseudocode of the reconstruct unknown message elements.

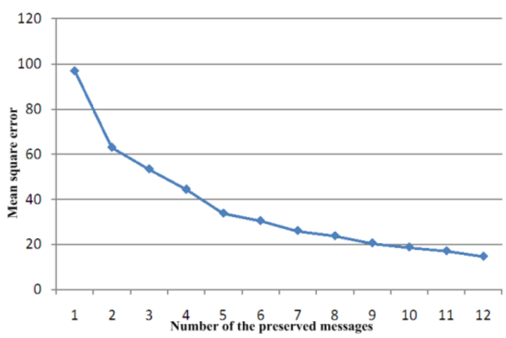


Figure 4. MSE in the different numbers of preserved minimum label messages.

where M_{con}^{n} is the reconstructed message, $M_{pre}(n)$ is the preserved message with the sorting index n, and $l(n)$ is the label of the $M_{pre}(n)$. Then, the minimum value of all the reconstructed value from the n preserved candidate values are selected as the incoming messages.

$$M_{inc}(l_q) = \min_{n=1,2,\ldots,T} M_{con}^{n}(l_q), \tag{5}$$

where T is the number of preserved messages in the sorting list. The pseudo code of reconstruct messages function is listed in Figure 3.

Next, we show that our method can preserve the accuracy in message representation. We compute the Mean Square Error (MSE) between the messages without any loss and the messages reconstructed by our method (Figure 4.). As the number of preserved minimum label messages increases, we get the lower MSE value. When we set the linear increasing function parameter T_s =2, λ= 20, and the number of preserved minimum label messages is four, the MSE value is only 44 with the possible largest error 1600. We will show the stereo matching performance analysis and the final disparity map in Section 4.

3.3 Bandwidth and memory analysis

Compared with the original message passing operation, the main advantage of the proposed algorithm is that the memory and bandwidth consumption is greatly reduced. In each iteration, we do not store and pass the messages of all the labels to the memory. And the dropped messages can be reconstructed by the message reconstruction process. For the BP-M method[2], the cost of memory becomes

$$\underbrace{N|L|}_{\substack{Data\\ terms}} + \underbrace{4N|T|}_{\substack{Message\\ terms}} \tag{6}$$

where N is the number of nodes, and T is the number of preserved message elements. Compared with the original message passing operation which needs $5N|L|$ memory, if we select T as 4 and L as 64 for a VGA image, we can dramatically reduce 93.75% message memory size and 75% total memory size. On the other hand, the bandwidth becomes

$$\underbrace{4N|L|}_{\text{Data Terms}} + \underbrace{3NT}_{\substack{\text{Preserved}\\ \text{Messages}}} + \underbrace{NT}_{\substack{\text{Outgoing}\\ \text{Messages}}}. \tag{7}$$

The bandwidth is reduced from $5N|L|$ to $N|L| + 4NT$, where $T << |L|$.

For the tile-based BP algorithm, the original message passing operation will preserve the message of all labels on the boundary to a temporal buffer with the length equal to the number of all possible labels for the next tiles. Therefore, in the original message passing operation, the memory buffer amount to store the tile boundary`s messages is $2B|L|$, where B is the width of tile size. The bandwidth for the boundary messages is $4B|L|$ [6]. In our proposed messages passing operation, we only need $2BT$ memory buffer and the bandwidth is only $4BT$, where $T << |L|$.

4. EXPRIMENTAL RESULTS

In previous section, we have shown that the proposed algorithm can significantly reduce the memory and bandwidth in hardware implementation. In this section, we analyze the quality performance of different numbers of preserved messages. We use datasets from the Middlebury vision website [9]. We adopt the data and smoothness term definition according to [4]. The data term is based on the Census measure method [10]. The census data cost between two pixels in two images is the Hamming distance of their census vectors. As verified in [11], among many popular methods, census transform is most robust to radiometric distortions and noises.

Two BP methods, the BP-M method and the tile-based BP [6], are used to examine the quality performance. In the tile-based BP, we set the tile size as 16-by-16. Different numbers of preserved messages in message passing algorithm are simulated from one to twelve. Compared with the ground truth, we found the bad pixels rate is around the limit when the number of preserved messages is four, as shown in Figure 5. There is no significant improvement when the number of preserved messages increases more than four candidates. When the number of preserved messages we select is three, the average non-occlusion bad pixels is only

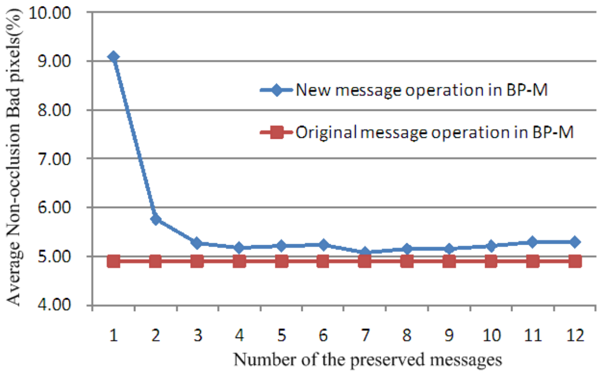


Figure 5. The average bad pixels rate with different message operation in BP-M.

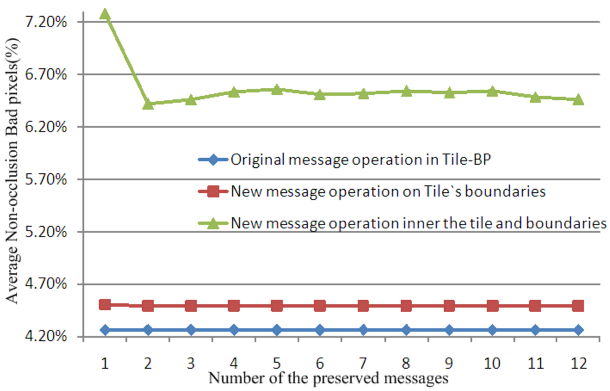
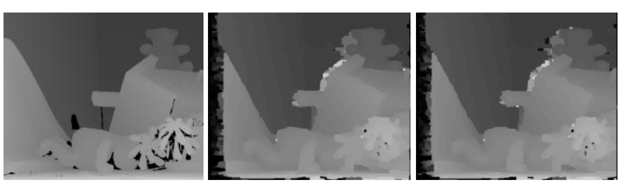


Figure 6. The average bad pixels rate with different message operation in tile-based BP.



(a) (b) (c)

Figure 7. The disparity maps of *Teddy* : (a) Ground truth, (b) the original message passing operation in the tile-based BP, (c) the proposed message passing operation applied only on the tile boundary with $T = 4$.

increased by 0.2%. Considering the hardware cost and quality performance, the suitable number of preserved messages for BP-M is three.

In the tile-based BP, if we only apply the new message passing operation on the tile boundary to reduce the off-chip memory size and bandwidth, the percentage bad pixel rate is very close to the original message passing operation. When the number of preserved messages is three, the average non-occlusion bad pixel rate is only increased by 0.2%. If we apply the new message passing operation both on the message of tile boundary and node inside the tiles, we can save both the off-chip memory and the on-chip memory, and the performance of average non-occlusion bad pixels only increased only 2.2%. Considering the hardware cost and quality performance, the suitable number of preserved messages for the tile-based BP is three. The result is shown in Figure 6. The disparity maps of the original tile-based BP and proposed algorithm are shown in Figure 7. We can see that the result of proposed method with $T = 4$ is similar to the result of the tile-based BP.

5. CONCLUSION

BP is a powerful technique for solving energy minimization problems. Compared to the local minimization methods, it can obtain a better solution, but is also difficult for hardware implementation due to the high requirement of memory and bandwidth . In this paper, we have proposed an efficient message passing operation that can greatly reduce the memory and bandwidth requirement without serious quality degradation. From the hardware implementation perspective, our proposed algorithm is a good candidate to save the hardware resources. The proposed algorithm makes BP more practical in hardware implementation.

6. REFERENCES

[1] Y. Boykov, O. Veksler, and R. Zabih, “Fast approximate energy minimization via graph cuts,” in *IEEE transactions on Pattern Analysis and Machine Intelligence* , vol. 23, no. 11, pp. 1222-1239, 2001.

[2] W. T. Freeman, E. C. Pasztor, and O. T. Carmichael, “Learning the low level vision,” in *International Journal of Computer Vision* , vol. 70, no. 1, pp. 41-54, 2000.

[3] P. F. Felzenszwalb and D. R. Huttenlocher, “Efficient belief propagation for early vision,” *International Journal of Computer Vision*, vol. 70, no. 1, October 2006.

[4] R. Szeliski et al, “A comparative study of energy minimization methods for Markov random fields with smoothness-based priors,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, no. 6, pp. 1068-1080, 2008.

[5] T. Yu, R.-S. Lin, B. Super, and B. Tang, “Efficient message representations for belief propagation,” in *Proc. International Conference of Computer Vision*, 2007.

[6] C.-C. Cheng, C.-K. Liang, Y.-C. Lai, Homer H. Chen, L.-G. Chen, “Analysis of belief propagation for hardware realization,” in *Proc. SIPS*, 2008.

[7] C.-K. Liang, C.-C. Cheng, Y.-C. Lai, H. H. Chen, and L.-G. Chen, “Hardware-efficient belief propagation,” in *Proc. IEEE CS Society Conf. CVPR*, pp. 80-87, 2009.

[8] C.-C. Cheng, C.-K. Liang, Y.-C. Lai, H. H. Chen, and L.-G. Chen, “Fast belief propagation process element for high-quality stereo estimation,” in *Proc. ICASSP*, pp. 745-748, 2009.

[9] The Middlebury computer vision pages. http://vision.middlebury.edu/

[10] R. Zabih and J. Woodfill, “Non-parametric local transforms for computing visual correspondence,” in *Proc. ECCV*, pp. 150-158, 1994.

[11] H. Hirschmuller and D. Scharstein, “Evaluation of stereo matching costs on images with radiometric differences,” in *IEEE Trans PAMI*, 2009.