EFFICIENT HARDWARE ARCHITECTURE FOR LARGE DISPARITY RANGE STEREO MATCHING BASED ON BELIEF PROPAGATION

Sih-Sian Wu, Student Member, IEEE, Chen-Han Tsai, Liang-Gee Chen, Fellow, IEEE

DSP/IC Design Lab, National Taiwan University, Taiwan
{benwu, phenom, lgchen}@video.ee.ntu.edu.tw

ABSTRACT

This paper introduces an efficient hardware architecture for the belief propagation(BP) algorithm especially for large disparity range stereo matching applications. BP is a popular global optimization algorithm for labelling problems which is hardware friendly. There are few researches focus on BP implementation in large disparity range stereo matching problems, since traditional belief propagation hardware implementations suffer from a server trade-off between hardware efficiency and short critical path while the disparity range is larger than 64. In this paper, we eliminate the redundancy of previous BP implementation and propose an efficient architecture without introducing any delay overhead which is more suitable for large disparity range cases. As a result, the hardware complexity is reduced from $O(L^2)$ to $O(L\log_2 L)$, where L is the disparity range. We use a time-area term to demonstrate the trade-off between various architectures, results show that the proposed one can reach 49.6% and 71.2% reduction compared to the state-of-the-art implementation[1] with disparity ranges 64 and 128 respectively.

Index Terms— BP-M, tile-based BP, Hardware implementation, stereo matching, disparity estimation

1. INTRODUCTION

Recently, 3D model reconstruction, simultaneous localization and mapping, robotic vision, 3D interaction, and an increasing number of emerging accurate applications rely on real-time, accurate disparity estimation. Among those stereo matching optimization algorithms, loopy belief propagation (BP) [2], graph cut [3], tree re-weighted message passing [4] and dynamic programming method [5] are discussed widely because of their promising results. Within those methods, BP algorithm, with its regularity and simple arithmetic computations is most suitable for hardware implementation. Not only for disparity search, BP is also widely applied in other computer vision tasks such as optical flow[6], denoising[7] and image inpainting [8].

While the disparity range, L, is more than 64, previous BP hardware implementations will encounter problems like large on-chip memory and a impractically long critical path. In our previous study [9], we found the belief propagation hardware implementation suffering from high memory overhead. We proposed a memory-efficient architecture inspired from the tile-based architecture discussed in [8].

In this paper, we focus on how processing units compute the minimum quantity for each disparity level efficiently. As the disparity range is increased, BP hardware implementations suffer from a severe trade-off between short critical path and hardware efficiency. Table 1 summaries different architectures of BP. In general, path-based methods [10][1] provide better hardware efficiency but with a longer critical path, and tree-based methods[8][2] offers a shorter delay but less hardware efficiency. Whether tree-based methods or path-based ones, prior architectures are not suitable for large disparity range cases because of large hardware area and a long critical path. In this paper, we provide an efficient hardware architecture to reach better trade-off between area and critical path especially for large disparity range scenarios.

The paper is organized as follows: In Section 2, we address the problem we are facing and briefly introduce several previous works. The discussion of the proposed architecture is shown in Section 3. The experimental result and conclusion are presented in Section 4 and 5, respectively.

2. PREVIOUS WORK

BP is an global optimization algorithm through a iterative process. Which finds the optimal label to minimized energy as defined by data cost and smoothing constrain as

$$E = \sum_{p\in P} E_p\left(l_p\right) + \sum_{(p,q)\in N_p} E_{pq}\left(l_p, l_q\right). \qquad (1)$$

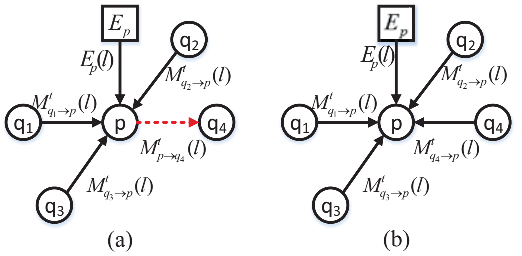
The first term, $E_p(l)$, is the data cost of each node p. $E_{pq}\left(l_p, l_q\right)$ is the discontinuity penalty to constrain the smoothness. The smoothness term for a given pixel, is based on messages from four-connected neighboring pixels which are denoted as N_p.

There are two processing modes in a BP algorithm which are the passing mode and the deterministic mode as shown in Fig. 1. The Passing mode minimizes the outgoing messages from the pixel and the deterministic mode determines which disparity has minimum energy for the pixel.

2374-7390/16 $31.00 © 2016 IEEE
DOI 10.1109/SiPS.2016.49

Table 1. Summary of various architectures. L is the disparity range and T is he truncated number for smoothing term.

Categories	Path-based		Tree-based		
Architecture	Efficient BP[10]	ISSCC 2015[1]	BP-M[2]	Tile-based BP[8]	Proposed
Feature	Hardware efficiency		Low critical path		Low critical path and Hardware efficiency
Critical Path	$O(L)$	$O(L)$	$O(log_2L)$	$O(log_2L)$	$O(log_2L)$
Area	$O(L)$	$O(L)$	$O(L^2)$	$O(L^2)$	$O(Llog_2L)$



(a)

(b)

Fig. 1. (a)The passing mode and (b)the deterministic mode for a BP algorithm.

$M^t_{p\to q_4}(l)$ is the optimized message passing from the pixel p to the pixel q_4 which represents the disparity l in t-th iteration. To find minimum energy for each disparity, the minimized message is defined as

$$M^t_{p\to q_4}\left(l\right) = \min_{l'\in L}\left\{E_s(l,l') + E_p(l') + \sum_{q'\in N_p\setminus q_4} M^{t-1}_{q'\to p}(l')\right\}. \tag{2}$$

$E_s(l,l')$ is the penalty to keep the smoothness. $E_p(l')$ and $M^{t-1}_{q'\to p}(l')$ are the matching cost of the pixel p and latest updated messages from neighboring pixels except q_4, respectively. For simplicity, last two terms are combined as $H\left(l'\right)$ which is defined as,

$$H\left(l'\right) = E_p(l') + \sum_{q'\in N_p\setminus q_4} M^{t-1}_{q'\to p}(l'). \tag{3}$$

According to regularity, the truncated linear model is friendly to hardware implementation [8][10][1], which is defined as,

$$E_s\left(l,l'\right) = \min_{l'\in L}(\lambda\left|l-l'\right|, \lambda T), \tag{4}$$

where λ is the weight of the smoothness and T is the parameter that constrains quantity from increasing. When $|l-l'|\leq T$, the smoothing term is in untruncated region. The value increases as a difference of labels l and l' increases. Otherwise, the smoothing term is constant when the label l' is in the truncated region. Thus, Eq. 2 can be transformed to

$$M^t_{p\to q4}\left(l\right) = \min_{l'\in L}\left\{E_s(l,l') + H\left(l'\right)\right\}. \tag{5}$$

Each passing includes a set with L optimized directional messages. Each message represents the minimum value during L possibilities (hypothesises).

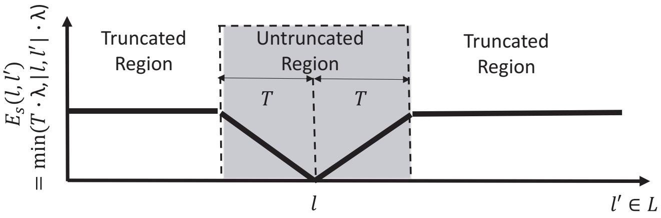


Fig. 2. Smoothing term with the truncated linear model.

The complexity of traditional hardware implementation [2] with L parallelism is $O(L^2)$. The approach is not practical to implement in hardware when the disparity range, L, is large because of the huge gate count. Whose architecture is shown in Fig. 3(a). An efficient BP implementation [10] is proposed to improve hardware efficiency. Two-pass method is adopted to find the minimum envelop of the disparity set. Even though there is an efficient hardware approach, the critical path is directly proportional to the disparity range as presented in Fig. 3(b). To shorten the critical path, specialized hardware modification improves the implementation [1] by modifying the two-pass method into an one-pass one in eight-stage pipelined architecture with a cost of performance loss as shown in Fig. 3(d). However, implementation [1] is impractical when label count is larger than 64.

Chang *et al*. [11] observed the redundancy of implementation [2] and derived an efficient architecture from the conventional implementation with the following formula,

$$M_{p\to q4}\left(l\right) = min\left(M^{Local}_{p\to q4}\left(l\right), M^{Global}_{p\to q4}\right). \tag{6}$$

Where $M^{Local}_{p\to q4}\left(l\right)$ is the minimum value within the untruncated region which is computed in a local tree.

$$M^{Local}_{p\to q4}\left(l\right) = \min_{l-T\leq l'\leq l+T}\left\{E_s(l,l') + H\left(l'\right)\right\}. \tag{7}$$

$M^{Global}_{p\to q4}$ computed in the global tree is the global minimum constraint whose value is independent of the disparity level.

$$M^{Global}_{p\to q4} = \min_{l'\in L}\left\{T\lambda + H\left(l'\right)\right\}. \tag{8}$$

The designed structure is applied in [8] due to hardware efficiency and short critical path properties as shown in Fig. 3(c).

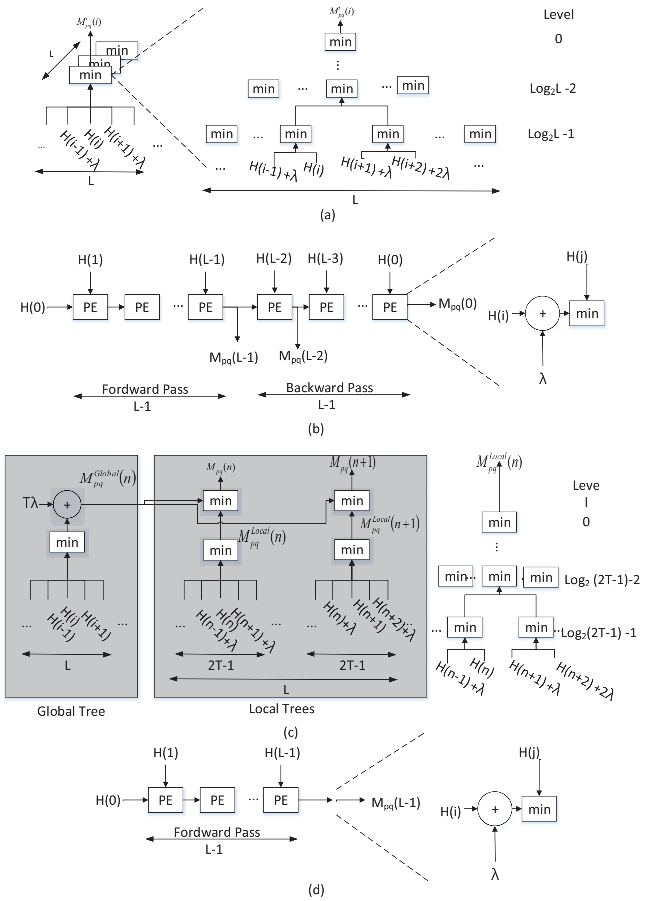


Fig. 3. System structure for (a)a original hardware implementation [2], (b)an efficient hardware implementation [10], (c)the proposed architecture in [11], and (d)the proposed architecture in [1].

Despite hardware efficient and short critical path properties, the scenario is limited in the extremely small untruncated region such as $T \leq 2$. In this paper, we proposed an architecture based on [8] with generalized truncated region and increased hardware efficiency via eliminating the redundancy in the previous one.

3. PROPOSED SYSTEM

3.1. Observations

Unlike a stereo matching criteria, disparity with sub-pixel accuracy[6][8] and other application[7] is not appropriate to limit to untruncated region into such small area [8] while applying BP algorithms. When T is too small, the updated minimum messages are more likely to be in the disparity with the smallest data cost. In other words, the smoothing effect is unapparent and the advantage of BP optimization is lost if T is unreasonably limited. For the generality, we simply set $T = \frac{L}{8}$ during the following discussion. That is, $T \propto L$.

When the setting for T is not much smaller than L, the complexity of [8] returns to $O(L^2)$. We observed that with the truncated linear smoothing model the computation of the message update operation becomes more regular. The straightforward hardware implementation to find minimum value is based on binary tree structure as shown in Fig. 3(c). In the conventional architecture, each local tree is independent with other ones. According to the following equation

$$\begin{aligned} &min(H(n-2), H(n-3)+\lambda)+2\lambda \\ &= min(H(n-2)+2\lambda, H(n-3)+3\lambda), \end{aligned} \tag{9}$$

we discovered that there are some redundant operations in the previous implementation. We derived our architecture to utilize regularity instead of direct mapping. To leverage regularity of the truncated linear model, the modified minimum operator can provide the same result of direct mapping one. Most of minimum operators inside local trees can be removed by regularity as presented in Fig. 4. Most important is the modified mapping operator can be shared within different local trees. Note that the critical path is not affected due to the delay is dominated by the global trees as shown in Fig. 3(c).

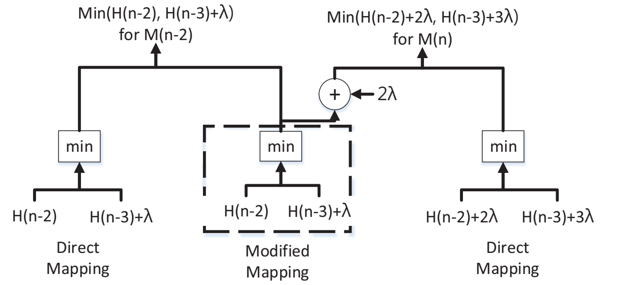


Fig. 4. Original mapping and modified mapping for different minimum operations.

The regular property is only sustained within the same side of the untruncated region.

$$\begin{aligned} &min(H(n-1)+\lambda, H(n)) \\ &\neq min(H(n-1), H(n)+\lambda) \end{aligned} \tag{10}$$

To further increase the hardware efficiency, both sides should be designed separately.

3.2. Proposed Architecture

Leaves in each local tree is the size of untruncated region, $2T+1$, which means there are $\lfloor \frac{2T+1}{2} \rfloor$ min. operators in the lowest level within a traditional local tree. According to properties we observed above, we noticed that the number of min. operations in the bottom level which are actually required is $2 \times \frac{L}{2}$ for two sides of untruncated region. Each local tree contributes one operator and collect other information from other trees.

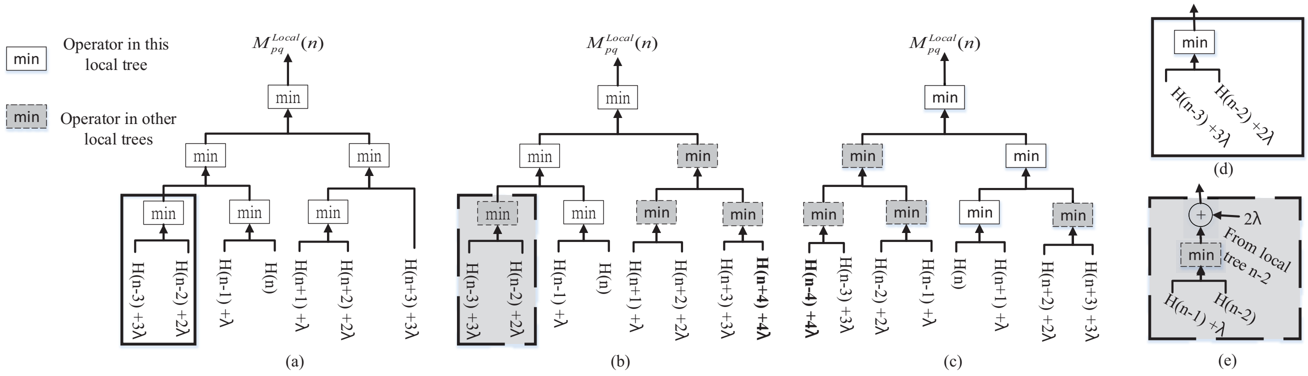


Fig. 5. Local trees structures (a)Original mapping in [8]. (b)Proposed tree architecture for n is an even number. (c)Proposed tree architecture for n an is odd number. (d)detail in direct mapping operator and (e) detail in shared operator.

In the upper level, regularity is preserved. The conventional architecture requires $\lfloor 2 \times \frac{2T+1}{2^2} \rfloor$ operations in each local tree and the proposed one required $2 \times \frac{L}{2}$ operations for ”all” local trees. It is interesting that total operators for local trees in each level is equal. Each local tree contains only one min operator in each level. The height of a local tree is $\lceil log_2(2T+1) \rceil$. That is, the number of total operators inside a local tree is equal to the height of the local tree, $\lceil log_2(2T+1) \rceil$. We mentioned that $T \propto L$, the complexity of a BP processing element is reduced from $O(L^2)$ to $O(Llog_2(L))$. The problem is how to design the local tree to reach the efficient we desired.

In contrast, the proposed local tree architecture utilizes the aforementioned property which are shown in Fig. 5(b) and (c) instead of direct mapping [8] as Fig. 5(a). In each local tree, only one minimum operation is required in a single level. In Fig.5, we use $T = 3$ as an example to demonstrate the difference between original architecture and the proposed one. In the figure, we color operators in other local trees. In the given instance, min. operators in original architecture and the proposed one is 6 and 3, respectively. To further increase the hardware efficiency, proposed local trees are divided into even and odd groups. Since the left-right-side unequal property, one of groups contributes the left side and other group contributes right side. As a result, the required min. operators for each local tree is reduced from $2T$ to $\lceil log_2(2T+1) \rceil$. When the T is not limited much smaller L, such as 1 or 2, the complexity is reduced from L^2 to $Llog_2(L)$. In spite of the fact that the proposed tree has longer critical path than [8] does, the total delay is dominated by the global tree. The proposed architecture does not introduce delay overhead for the whole BP processing unit.

One may notice that the number of leaves between conventional and proposed architecture in Fig 5 are not equal. We add an additional node in the leaf of the proposed local trees which makes the architecture become more efficient. The proposed local tree contributes a min. operator per level and collects information provided by other trees. More specifically, each local tree collects information from lower level of other trees which represents the minimum value of other hypotheses. The information from other local trees represents the minimum value among hypothesises, where the number of hypothesises is power of 2. To fully utilize the information, we re-allocate notes with adding an additional hypothesis which is not inside the untruncated region. With this modification, the The correctness is not affected since the Eq. 6 can be rewritten as

$$M_{p\to q4}\left(l\right) = \\ min\left(M_{p\to q4}^{Local}\left(l\right), H(n \pm (T+1)) + (T+1)\lambda, M_{p\to q4}^{Global}\right). \tag{11}$$

If the minimum value is occurred in the additional node $H(n \pm (T+1)) + (T+1)\lambda$, the min value is determined by the global tree, which defined in Eq.8, as

$$H(n \pm (T+1)) + (T+1)\lambda < M_{p\to q4}^{Global} \\ \Rightarrow H(n \pm (T+1)) + (T+1)\lambda < H(n \pm (T+1)) + T\lambda. \tag{12}$$

The additional node is added for further hardware efficiency and without affecting the correctness.

4. EXPERIMENTAL RESULTS

We implemented the proposed architecture and counterpart designs with TSMC 40nm technology. All synthesis results are under the same condition, the ”slow” operation condition and the ”slow” wire load model. Table 1 illustrates the analysis comparison between prior hardware architectures and the proposed one. The short delay characteristic of tree-based method[2][8] retains in the proposed architecture. Compared to path-based methods[10][1], tree-based architectures are more suitable for large disparity range scenarios since the shorter delay. Furthermore, the proposed efficient structure reduces the complexity from $O(L^2)$ to $O(Llog_2L)$ through eliminating redundancies. On the other hand, the critical path are proportional to the disparity range in architecture [10] and [1] both are not suitable for BP hardware implementation with large disparity range.

Table 2. The minimum operations comparison.

Architecture	[2]	[10]	[8]	[1]	Ours
L = 64 T = 8	4032	126	1023	63	383
L = 128 T = 16	16256	254	4095	127	895

The number of minimum operators of different disparity ranges with various hardware architectures are presented in Table 2. Noted that minimum operators are dominated the area of the BP processing element. According to the data, path-based architecture[10][1] do require less operators, but these approaches are not proper for large disparity range due to long critical paths. Among tree-based methods, the proposed architecture requires less operators reflects its reduction in the gate count.

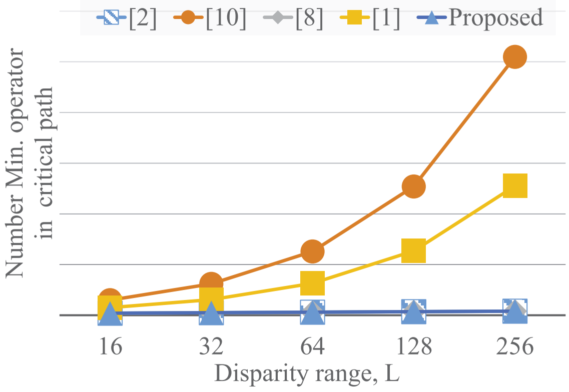


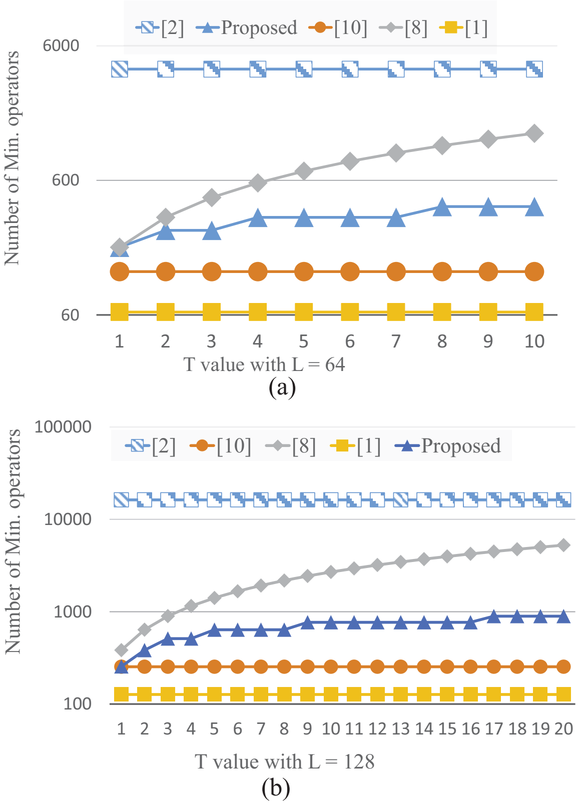
Fig. 6. Number of minimum Operations in critical path with different architectures in various label count (L) and $T = \frac{L}{8}$.

To illustrate the critical path issue, the number of minimum operators on the critical path for each architecture is presented in Fig. 6. Unlike tree-based implementations[2][8] and proposed one, the operators in the critical path proportionally increases with the disparity range increase. The critical path of tree-based architectures are dominated by the global tree which has $log_2 L$ operators in the critical path.

The time-area term is used to illustrate the trade-off between different architectures, where the term is defined as

$$time - area = criticalpath \times gatecount. \quad (13)$$

Numbers of minimum operators in different hardware architectures with various T values is analysed as shown in Fig. 7(a) and (b) whose disparity ranges are 64 and 128, respectively. While T is larger than 1, the proposed architecture requires less minimum operators than [8]. Note that, implementation [10] and [1] require less min. operators but long critical paths which are not friendly to large disparity range conditions. Among those architectures, only the proposed one and [8] are effected with different truncated parameter, T. Since the proposed architecture has removed redundancies inside the implementation[8], the proposed one always has less operators except $T = 1$. When $T = 1$, the proposed requires equal operators as [8] does. Tile-based BP [8] can be seen as a special case of our proposed architecture with $T = 1$.



(a)

(b)

Fig. 7. Minimum Operations comparison for different architectures with various T value and (a) label count is 64 (b) label count is 128.

Table 3. Synthesis report of the BP processing element under L = 64 and T = 8 condition.

	[2]	[10]	[8]	[1]	Ours
Gate Counts(M)	4.43	0.38	1.09	0.13	0.55
Compare to [8] (%)	406.4	34.9	100	11.9	50.4
Critical Path(ns)	17.21	212	17.21	106	17.21
Compare to [8] (%)	100	1232	100	616	100
Time-area term (%)	406.4	429.97	100	65.2	**50.4**
Reduction(%)	X	X	0	38.8	**49.6**

The synthesis results for each architecture with a disparity range of 64 are presented in Table 3. The critical paths of each hardware implementations are listed in the tables as well. The proposed architecture can reduce required area by 49.6% compared to [8]. The synthesis results for each architecture with a disparity range of 128 are presented in Table 4. The proposed architecture can reduce required area by 71.2% compared to [8].It is worth to mention that the reduction in minimum operators is approximately equal to the chip area reduction and the area reduction increases as the label count increases. In Table 4, 71.1% time-area reduction is achieved. When the disparity range is larger than 64, proposed architecture can reach the nearly reduction compared to implementation[1] and [8] which are state-of-the-art implementation of path-based and tree-based architecture. Results show that the proposed structure reach better trade-off between area and critical path while disparity is larger than 64. While the disparity range rises, the trade-off advantage of our proposed architecture is more obvious. The time-area term reduction are 49.6% and 71.2% for disparity range 64 and 128, respectively.

Table 4. Synthesis report of the BP processing element under L = 128 and T = 16 condition.

	[2]	[10]	[8]	[1]	Ours
Gate Counts(M)	14.61	0.43	3.58	0.39	1.03
Compare to [8] (%)	408	12.0	100	10.9	28.8
Critical Path(ns)	23	300	23	210	23
Compare to [8] (%)	100	1304	100	913	100
Time-area term (%)	408	156.48	100	99.52	**28.8**
Reduction (%)	X	X	0	0.48	**71.2**

5. CONCLUSION

An efficient architecture for BP algorithm with large disparity range is proposed. Compare to prior implementations, the proposed one requires less area without increasing the critical path. The proposed architecture is designed to utilize regularity that we observed and the redundancy of the previous architecture is removed. The hardware complexity is reduced from $O(L^2)$ to $O(L \log_2 L)$. In the case of a disparity range of 64, a 49.6% time-area term reduction is achieved. Furthermore, when the disparity range increases to 128, a 71.2% reduction in time-area term is reached. Experimental results specify that the disparity range is larger the more time-area reduction is achieved by the proposed architecture. Therefore, the proposed architecture is more suitable for large disparity range scenarios and applications.

6. REFERENCES

[1] H.H. Chen, C.T. Huang, S.S. Wu, C.L. Hung, T.C. Ma, and L.G. Chen, "23.2 a 1920x1080 30fps 611 mw five-view depth-estimation processor for light-field applications," in *IEEE International Solid- State Circuits Conference (ISSCC)*, Feb 2015, pp. 1–3.

[2] J. Sun, N.N. Zheng, and H.Y. Shum, "Stereo matching using belief propagation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, no. 7, pp. 787–800, July 2003.

[3] Y. Boykov, O. Veksler, and R. Zabih, "Fast approximate energy minimization via graph cuts," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, no. 11, pp. 1222–1239, Nov 2001.

[4] V. Kolmogorov, "Convergent tree-reweighted message passing for energy minimization," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 10, pp. 1568–1583, Oct 2006.

[5] O. Veksler, "Stereo correspondence by dynamic programming on a tree," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2005, vol. 2, pp. 384–390 vol. 2.

[6] Yu Li, Dongbo Min, Michael S Brown, Minh N Do, and Jiangbo Lu, "Spm-bp: Sped-up patchmatch belief propagation for continuous mrfs," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 4006–4014.

[7] Frederic Besse, Carsten Rother, Andrew Fitzgibbon, and Jan Kautz, "Pmbp: Patchmatch belief propagation for correspondence field estimation," *International Journal of Computer Vision*, vol. 110, no. 1, pp. 2–13, 2014.

[8] C.K. Liang, C.C. Cheng, Y.C. Lai, L.G. Chen, and H.H. Chen, "Hardware-efficient belief propagation," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 21, no. 5, pp. 525–537, May 2011.

[9] S.S. Wu, H.H. Chen, C.H. Tsai, and L.G. Chen, "Memory efficient architecture for belief propagation based disparity estimation," in *IEEE International Symposium on Circuits and Systems (ISCAS)*, 2015, pp. 2521–2524.

[10] Pedro F Felzenszwalb and Daniel P Huttenlocher, "Efficient belief propagation for early vision," *International Journal of Computer Vision*, vol. 70, no. 1, pp. 41–54, 2006.

[11] 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 *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2009, pp. 745–748.