Architecture Design of Stereo Matching Using Belief Propagation

Chao-Chung Cheng, Chung-Te Li, Chia-Kai Liang, Yen-Chieh Lai, and Liang-Gee Chen

Graduate Institute of Electronics Engineering, National Taiwan University, Taiwan

***Abstract*—We propose a new architecture for stereo matching using belief propagation. The architecture combines our fast, fully-parallel processing element (PE) and memory-efficient tile-based BP (TBP) algorithm. On the architectural level, we develop several novel techniques, including a three stage pipeline, a message forwarding scheme, and a boundary message reuse scheme, which greatly reduce the required bandwidth and power consumption without sacrificing performance. The simulation shows that the architecture can generate HDTV720p results at 30 fps when operating at 227MHz. The high-quality depth maps enable real-time depth image based rendering and many other important applications in the 3D TV industry.**

I. Introduction

The 3D video signal processing has become an active topic in the visual processing field. As 3D display technology matures, human aspires to experience more reality. One typical way to generate 3D digital content is to use binocular vision, as shown in Fig. 1, where the depth map is generated using stereo matching, and then the multiple-view video streams are rendered by depth image based rendering (DIBR). In this method, the performance of stereo matching directly determines the quality of the depth map, and consequently that of rendered images.

Stereo matching has been extensively studied for decades [1], and most start-of-the-art methods are based on the global optimization framework [2]. In this framework, the optimal solution is obtained by minimizing the user-specified energy function using graph cuts [3] or belief propagation (BP) [4]. BP is a powerful method to estimate the optimal hidden states from the observation in a graphical model. Here the hidden states are the per-pixel depth/disparity values, and the observation is the left/ right image pair. While the original BP can generate results much better than greedy local methods [10], it requires a great number of computations, memory, and bandwidth, and thus it is not suitable for real-time or streaming applications [5]. Fortunately, there are many recent improvements addressing these issues [6][7][11], making BP become an affordable choice.

In this paper, we extend those software approaches to design efficient hardware architecture. This architecture combines our fast, fully-parallel processing element (PE) [8] and memory-efficient tile-based BP(TBP) algorithm [6]. On the architectural level, we develop several novel techniques, including a three-stage pipeline, PE retiming, and boundary message forwarding and reuse to achieve high efficiency. Compared with the previous design, the new architecture achieves full utilization, consumes smaller bandwidth and chip area, and generates high-quality results with lower power consumption. When operating at 227MHz, it generates 30fps HDTV720p results.

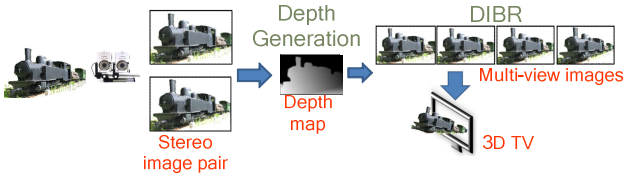

Fig. 1. The flow of generating multiple-view vide from stereo video.

II. Preliminaries of Previous Work

In this section, we first give a brief review on the stereo matching problem and belief propagation. Then we describe our acceleration PE and TBP algorithm.

Given a left/right image pair I_l and I_r, the stereo matching (or disparity estimation) problem seeks a correspondence assignment d_p for each pixel p in I_l such that the pixel $p+d_p$ in I_r and p in I_l map to the same physical point in the scene [1]. While the per-pixel local matching would fail due to the lack of texture/edge information in smooth areas, the state-of-the-art algorithms formulate this problem in a global optimization framework. That is, the optimal disparity value $\{d_p\}$ of all pixels should minimize a specified energy function:

$$\{d_p\} = \arg\min\left\{\sum_{p\in P} E_d(d_p) + \sum_{(p,q)\in G} E_s(d_p, d_q)\right\}, \quad (1)$$

where P is the set of all pixels, G is the set of all neighboring pairs, E_d is the per-pixel cost function, encoding the similarity between $I_l(p)$ and $I_r(p+d_p)$, and the E_s is the smoothness cost function, encoding the prior assumption that the disparity field should be smooth.

While finding the exact solution of (1) is a NP-hard problem, BP can efficiently find the approximate solution. In BP, assume the number of possible disparity values is L, for each pixel p and its neighbor q, a message M_{pq} of length L is stored and updated iteratively. At each iteration, M_{pq} is updated based on messages incoming to p. When the messages converge or become stable, a local belief is calculated for each pixel, and the optimal disparity value is the extreme of the belief. These operations and the corresponding equations are shown in Fig. 2. While BP is an efficient algorithm, solving the global optimization is still a relatively expensive task. Generally, for an input of N pixels, BP requires $O(NL)$ memory, $O(NLT)$ bandwidth, and $O(NL^2T)$ computations, where T is the total number of iterations [5].

We proposed several algorithm-level improvements to reduce the costs of BP in [5][6]. First, we developed a tile-based

The project is partially sponsored by Himax Inc.

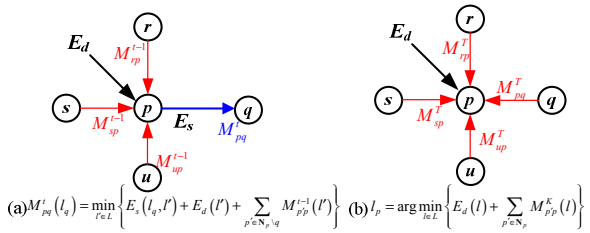

Fig. 2. (a) A message at iteration t from p to q is constructed using the messages from r, s, and u to p at iteration t–1. (b) The node p collects all messages from the neighbors to decide the best label.

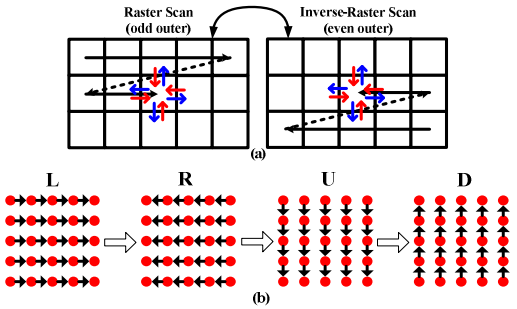

Fig. 3. The scan order in the TBP. (a) The boundary message passing scheme of outer iteration. (b) The in-tile passing scheme of one inner iteration.

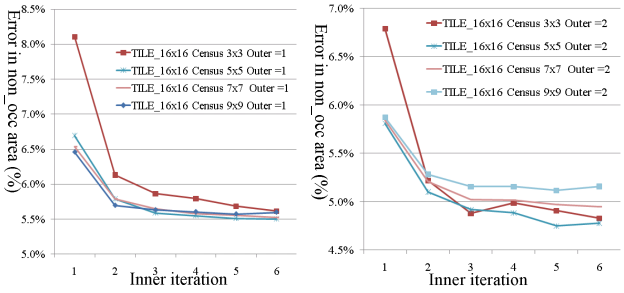

Fig. 4. The percentage of the bad-pixels with different tile sizes, census transform sizes and inner iterations. (a) outer iteration=1, (b) outer iteration=2. The number is the average bad pixels in the non-occluded area of four benchmark images in the Middlebury website [2].

belief propagation (TBP) algorithm. Instead of storing all messages, the TBP splits the image into many non-overlapping $B\times B$ tiles and only keeps the messages in-between the tiles in the memory. By processing the tiles sequentially in a raster and inverse-raster scan order shown in Fig. 3 (a), the TBP generates results similar to those by the original BP, but only takes $O(NL/B)$ memory and $O(T_oNL/B)$ bandwidth, where the number of outer iteration T_o is a fraction of T. The order of the message passing with a tile can be chosen arbitrary, and here we use the classic BP-M order in [9] (Fig. 3 (b)) due to its fast convergence property.

We also exploited the redundancy in updating messages to reduce the computations. When the smoothness cost function E_s in (1) is a robust function (e.g., truncated linear or quadratic functions), many temporary variables are repetitive, and can be reused for many times. Follow the observation, we developed a new algorithm to reduce the per-node message update cost from $O(L^2)$ to $O(L)$. The algorithm removes all dependency in-between message entities and enables full parallelism. In [8], we designed an $O(1)$ fast PE based on this algorithm. It can generate a message (L=64) using *only one cycle*.

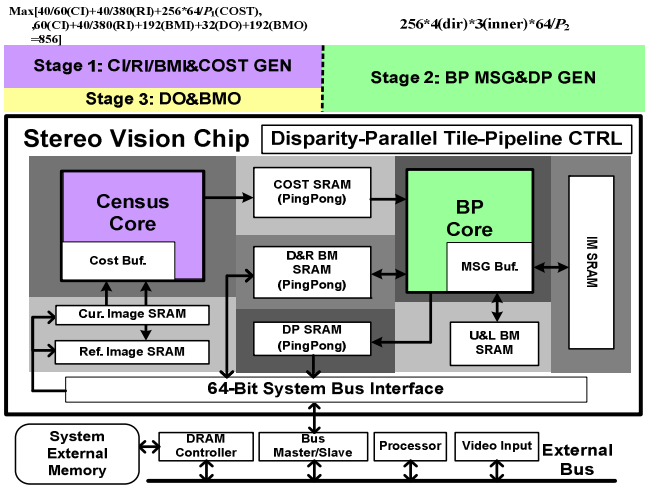

Fig. 5. The proposed three-stage Tile-pipeline architecture.

A VLSI circuit described in [6] shows the hardware feasibility of those algorithm-level improvements. However, this proof-of-concept implementation requires a large and complex on-chip memory routing and control system, which is not suitable for practical applications or SoC design. By carefully examining the hardware dataflow and target applications, we develop a new architecture which achieves a better balance between performance, bandwidth, and costs. The details of the new architecture are described in the next section.

III. Proposed Architecture

The systematic view of the architecture is shown in Fig. 5. The system contains two processing cores. The **Census Core** uses census transform to generate the data cost in (1); the **BP Core** performs the message construction and disparity decision operations. The census data cost between two pixels in two images is the Hamming distance of their census vectors. As verified in [15], among many popular methods, census transform is most robust to radiometric distortions and noises.

Before describing the design analysis and the implementation in the next section, we give the parameters of the design here. Follow the quality/memory tradeoff analysis in [5], we set the tile size to 16×16. The disparity range is 64. Each entity in the message vector is truncated to 63, and a message takes 384bits (6bit×64). In Fig. 4 we show the percentages of bad pixels in the depth maps generated with different inner iteration numbers and neighbor size. It shows that the 5×5 block size with 2 or 3 inner iterations can generate good results without taking too many computations.

The proposed architecture has three key features. Firstly, the cost generation core and BP core are three-stage pipelined to improve computation efficiency. Secondly, the fast message computation PE is carefully designed to achieve 100% utilization. Thirdly, the boundary message is efficiently reused. In the following sub-sections, we will describe details.

A. Three-Stage MB Pipeline Architecture

For high speed stereo matching application, we propose a three-stage Tile-Pipeline architecture. In Stage 1, pixel data input and the boundary messages are loaded, and data cost are

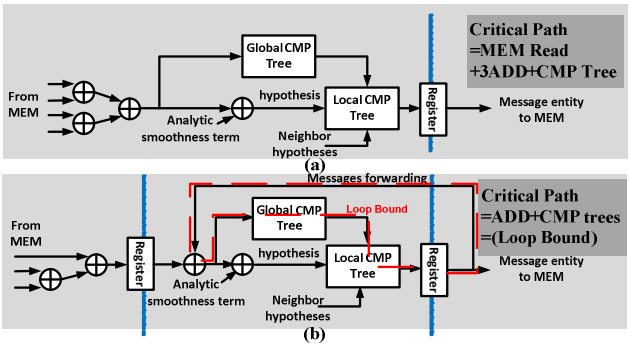

Fig. 6. The PE design, (a) the dataflow of the original PE design in **[8]** with message forwarding, (b) the PE of message forwarding + pipelining. The dotted blue lines indicate the pipeline.

generated. Then, BP and disparity decision are performed in Stage 2. In Stage 3, the boundary messages (BMO) and disparity values (DO) are stored during along with the computations in Stage 1. Therefore, Stage 3 can share the I/O cycles of Stage 1.

Both the **BP Core** and **Census Core** can be implemented in parallel, and their degrees of parallelism should be balanced such that the whole system can work at full utilization. However, because we can still adjust the number of inner iterations to tradeoff the quality, the cycles in Stage 2 may change. To balance the cycles between Stage 1 and Stage 2, the degrees of parallelism of the **Census Core** should be designed with respect to the inner iteration number and the parallelism of the **BP Core**.

Specifically, the memory access in Stage 1 takes a fixed amount of cycles (60(CI)+380(RI)=440 in our implementation), and Stage 2 takes (256×64×4×#inner) operations, to balance the cycles, parallelism-degree (P_1) of the **Census Core** and that (P_2) of the **BP Core** should be designed such that

$$\frac{256\times\ \underset{\#\text{pixel}}{}\ \frac{64}{\#\text{disparity}}}{P_1}+440\cong\frac{\left(256\times\ \underset{\#\text{pixel}}{}\ \frac{64}{\#\text{disparity}}\times\ \frac{4}{\#\text{direction}}\times\#\text{inner}\right)}{P_2}. \quad (2)$$

Here we set P_2 = 64 to adopt an improved version of the PE in [8], as described in the next sub-section. Hence, one inner iteration takes 1024 cycles. According to (2), P_1 is set to 32-, 16-, and 8- parallel for 1, 2, and 3 inner iterations, respectively.

B. Fast Message PE Design

In [8] we developed a message construction PE based on an efficient and parallel algorithm for robust smoothness functions [6]. The PE uses parallel adder arrays and comparator trees to achieve full parallelism, that is, one stream for each message entity. For example, in Fig. 2(a) the message M_{pq} can only be computed after all messages toward p are computed and ready in the register.

We solve this problem by properly *forwarding and pipelining* the PE such that the overall latency is reduced. The basic idea is to split the adder arrays into multiple stages and forwarding the message dependent from the previous computation in the last stage. A dataflow comparison between the PE in [8] and new PE is shown in Fig. 6. In Fig. 6(a), the computation begins after all messages are ready, and the critical path includes memory reading, 3 adders and comparator trees. However, because only one of three incoming messages depends on the previous operation, we can move the addition involving static messages to the first pipeline stage and reduce the critical path to the loop bound, as shown in Fig. 6 (b). In this way, the PE can run at a faster speed. In our simulation using the UMC90um process, the critical path is reduced from 5.3ns [8] to 4.4ns. Note we only address the dependent processing order of two consecutive pixels in Fig. 3(b) The loop bound value can be further reduced by parallel processing of multiple data-independent nodes.

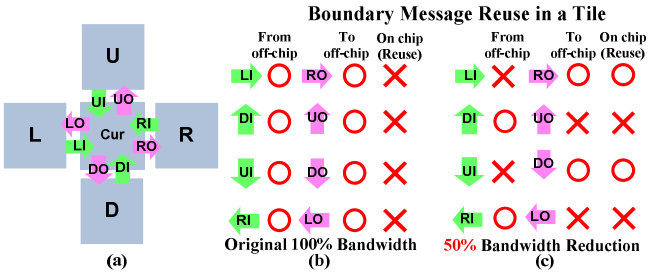

Fig. 7. (a) Boundary messages used and generated by the current tile. The memory access of BM's in (b) the design of [6], and (c) the new design.

C. Boundary Message Reuse

In [6] we showed that the TBP can greatly reduce memory/bandwidth costs by only keep the boundary messages (BM) in-between tiles. Here, we show that the bandwidth can be further reduced by reusing the BM's and exploiting the regularity of the scan order in TBP.

An example of BM access is shown in Fig. 7 (a). Originally, the current tile **Cur** reads four groups of BM's and writes four other groups of BM's after processing. From the figure we can clearly see that the BM group **RO** can be reuse for tile **R**, and **DO** can be reused for tile **D**, at the cost of a tile width buffer for **RO** and a frame width buffer for **DO**. In this way, we save 25% of total bandwidth.

Moreover, due to interleave scan order of outer iteration, in the raster scan pass, the **UO** and **LO** BM groups are never accessed; in the inverse-raster scan pass, the **UO** and **LO** are re-computed at each tile (see Fig. 3). Therefore, without affecting the algorithm, we can drop groups **UO** and **LO** in the raster scan, and groups **RO** and **DO** in the inverse-raster scan. Combining both techniques, we can greatly reduce the bandwidth for BM's by 50%. For example, for a 720p@30fps video with 2 outer iterations, the overall system bandwidth is reduced 41% from 810MB to 478 MB.

IV. Implementation Results and Discussion

We follow the architecture design principles described in the previous section, to develop a chip using the UMC 90 nm technology. In our design, the maximal number of disparity values is 64, which is larger than most existing GPU, VLSI, and FPGA implementations [12]. The Cost Core is 32 disparity-parallel and the BP Core is 64 disparity-parallel to balance the Tile-pipeline architecture. Because for each message processing, 5*384bit is need from U, D, L, R and Cost memories. Due to the limitation of memory port, in BP Core, per

cycle one pixel is processed. Table I lists the gate counts and the layout view of the synthesized circuit. Table II lists the on-chip memory buffers. The chip performance and spec are listed in Table III. The chip consumes 242 mW when working at 227 MHz. It consists of 1.33M gates and the die size is 9 mm^2. Compared to the proof-of-concept design in [6], the new design can work at a higher clock-rate but use less power and area.

The circuit generates messages at a constant rate, 248M messages per second. The speed of the disparity estimation and the quality depend on the parameter settings. If we set (T_i, T_o) = (3, 2), we can process VGA-sized image at 30 fps. Better results can be obtained by increasing the iteration numbers and lower the frame rate. Compared with the GPU implementation in [7][14], our chip is 6-40 times faster in terms of the number of disparity estimations per second. Several disparity maps in our simulation are shown in Fig. 8. We can see that due to the use of census transform, we can generate plausible results with only a few iterations.

V. Conclusion

In this paper, we have proposed an efficient architecture for stereo matching using belief propagation. In the new architecture, we use the three-stage pipeline and the modified PE to improve computational efficiency, and a boundary message reuse scheme to reduce the bandwidth. Compared to the previous design, the new design can work at a faster clock-rate, consume less power and small area, and generate better depth maps.

References

[1] D. Scharstein and R. Szeliski, "A taxonomy and evaluation of dense two-frame stereo correspondence algorithms", *IJCV*, vol. 47, no. 1-3, pp. 7-42, 2002.

[2] The Middlebury stereo vision web pages: http://vision.middlebury.edu/

[3] Y. Boykov, O. Veksler, and R. Zabih, "Fast approximate energy minimization via graph cuts," in *IEEE Trans PAMI*, vol. 23, no. 11, pp. 1222-1239, 2001.

[4] W. Freeman, E. C. Pasztor, and O. T. Carmichael, "Learning the low level vision," *IJCV*, vol. 70, no. 1, pp. 41-54, 2000.

[5] C.-C. Cheng, C.-K. Liang, Y.-C. Lai, H. H. Chen, L.-G. Chen, "Analysis of belief propagation for hardware realization," in *Proc. of SiPS* 2008.

[6] C.-K. Liang, C.-C. Cheng, Y.-C. Lai, H. H. Chen, and L.-G. Chen, "Hardware efficient belief propagation," in *Proc. of CVPR*, 2009.

[7] Q. Yang, et al., "Real-time global stereo matching using hierarchical belief propagation," in *Proc. BMVC*, 2006.

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

[9] R. Szeliski, et al., "A comparative study of energy minimization methods for Markov random fields with smoothness-based priors," *IEEE Trans PAMI*, vol. 30 no. 6, pp. 1068-1080, June 2008.

[10] J. Sun, N. N. Zheng, and H. Y. Shum, "Stereo matching using belief propagation," *IEEE Trans. PAMI*, vol. 25, no. 7, pp. 787-800, July 2003.

[11] P. F. Felzenszwalb and D. P. Huttenlocher, "Efficient belief propagation for early vision," *IJCV*, vol. 70, no. 1, pp. 41-54, 2006.

[12] J. Díaz, E. Ros, R. Carrillo, and A. Prieto, "Real-time system for high-image resolution disparity estimation," *IEEE Trans. Image Processing*, vol. 16, no. 1, pp. 280-285, 2007.

[13] A. Criminisi, A. Blake, and C. Rother, "Efficient dense stereo with occlusions for new view-synthesis by four-state dynamic programming," *IJCV*, vol. 71, no. 1, pp. 89-110, 2007.

[14] M. Gong and Y.-H. Yang, "Near real-time reliable stereo matching using programming graphics hardware," in *Proc. CVPR*, 2005.

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

TABLE I. LIST OF GATE COUNT

UMC90nm@ 4.4ns (L=64)	Gate count
BP Core	77.4K
CENSUS Core	73.5K
MSG Buffer	36.7K
Cost Buffer	32.1K
CTRL	65.5K
Total Processing Cores	**285.0K**
Memory	**1.04M**
Total	**1.33M**

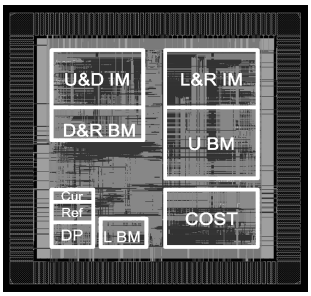

Table II. LIST OF MEMORY

	Memory size
Cur. Image SRAM	60Word×64bit
Ref. Image SRAM	380Word×64bit
Cost SRAM	256Word×384bit×2
U&D&L&R IM SRAM	256Word×384bit×4
D&R BM SRAM(DI/DO/RI/RO)	64Word×384bit×2
U BM SRAM(UI)	(width)1280Word×384bit
L BM SRAM(LI)	16Word×384bit
DP SRAM	32Word×64bit×2

Table III. SUMMARY OF THE DEVELOPED STEREO MATCHING CHIP

Technology	UMC 90 nm 1.0V	
Power consumption	242mW (at 227MHz)	
Chip area	3 x 3 (mm^2)	
Disparity range (L)	64/128(128 are 2:1downscale to 64)	
Processing Speed (at 227 MHz)		
Image resolution	(T_i, T_o)*	fps
CIF	(3, 1)	185.5
640×480	(3, 2)	30.6
D1 720×576	(2, 2)	33.9
HDTV 720p	(2, 1)	30.55
Number of constructed messages per second		248M

*T_o =1 means one raster scan, and T_o >1 means interleaved raster and inverse-raster scans for high quality mode.

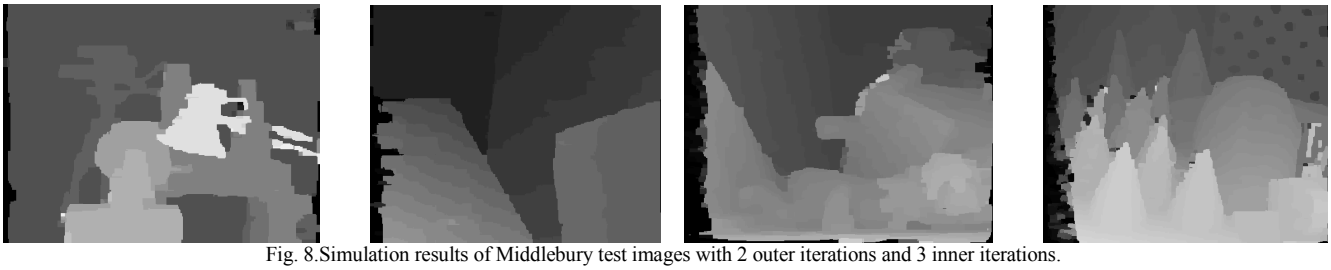

Fig. 8.Simulation results of Middlebury test images with 2 outer iterations and 3 inner iterations.