A 320M PIXEL/S VLSI ARCHITECTURE DESIGN OF WEIGHTED MODE FILTER FOR 4K ULTRA-HD DEPTH UPSAMPLING

Bo-Hsiang Yang, Li-De Chen, and Chao-Tsung Huang

National Tsing Hua University
Department of Electrical Engineering
Hsinchu, Taiwan

ABSTRACT

High-quality and high-resolution depth maps have opened tremendous possibilities for various applications, such as AR/VR display, 3D reconstruction, image refocusing, and view synthesis. But high-resolution depth estimation requires heavy hardware resources. Depth upsampling with weighted mode filtering is an efficient way to overcome this challenge. However, its hardware implementation has two major design issues: large on-chip memory for storing high-precision depth labels and high logic cost for computing adaptive range weight. In this work, we present two techniques, histogram candidate mapping and binary range weight kernel, which can reduce on-chip memory size and logic gate count by $46.9\,\%$ and $64.3\,\%$ respectively. Furthermore, we also implement a VLSI circuit for 4K Ultra-HD depth video upsampling using TSMC 40nm technology. It has 25.5-KB SRAM and 420K-gate logic, and the core area is $1.1 \times 1.1\ \mathrm{mm}^2$. When operating at 200 MHz and 0.9V, it delivers 320M pixel/s to support 4K Ultra-HD depth video at 40 fps, and consumes 104 mW based on post-layout simulation.

Index Terms— Weighted mode filter, depth upsampling, VLSI architecture, 4K-Ultra HD

1. INTRODUCTION

As the trend of computer vision arises, accurate depth maps are widely used in many computer vision fields, such as robotics, including navigation, manipulation, object recognition, and human pose analysis, and other applications like 3D modeling, virtual/augmented reality display, view synthesis, and image refocusing. Depth estimation solutions can be divided into two classes, active and passive techniques. However, both of these methodologies suffer from the limited spatial resolution and measuring noise.

To address the resolution problem, upsampling techniques have been proposed, and they are mainly categorized into two classes. The first class is motivated by the idea of the image super-resolution, which explicitly considers the low-resolution image formation process. Multiple depth maps upsampling [1, 2, 3] requires multiple sensors or time sequences. Nevertheless, the alignment between depth maps is necessary but extremely challenging and requires precise camera motion estimation.

The second class solves the upsampling task by filtering. It deals well with noisy low-resolution depth map since it often considers a registered high-resolution texture image assuming the correlation between depth and texture structures. Cost-volume-based methods [4, 5] perform the 2D filtering of each depth candidate on the 3D cost volume. Though they perform well for large upsampling scale, the computational cost is heavy. On the other hand, joint bilateral filter [6] and guided filter [7] have been adopted directly on the depth map. They sometimes give inaccurate edges on depth map which result in considerable depth bleeding artifacts due to the inconsistency between color and depth variations. To handle such artifacts, weighted median [8] and weighted mode filters [9] were proposed. Instead of averaging on the filtered output values, they seek the median and mode values from a local weighted histogram respectively, and they can successfully preserve object boundaries. Considering both quality and computational efficiency, the weighted mode filter (WMoF) is employed in this work. Chen *et al.* [10] proposed a VLSI architecture of WMoF, but it suffered from large memory size and logic gate count.

In this work, we propose a VLSI design and implementation for the WMoF under TSMC 40nm technology. It delivers the throughput to 320M pixel/s, 4K Ultra-HD depth video at 40 fps, operating at 0.9V and 200 MHz. We first introduce WMoF and analyze the design challenges in Section 2. In section 3, we introduce the two major proposed architectures: binary range weight and histogram candidate mapping, and describe how they address the design challenges. Then the corresponding implementation details and results are discussed in Section 4. Finally, we conclude our work in Section 5.

This work was supported by Novatek Microelectronics Corp.

2. WEIGHTED MODE FILTER AND CHALLENGES FOR IMPLEMENTATION

The weighted mode filter focus on developing a post-processing algorithm with the local histogram $H(t,d)$, which means that each bin represents an occurrence of neighboring pixels inside a source window.

$$H_w(t,d) = \sum_{s \in N(t)} w(s,t)\delta(D(s)-d) \tag{1}$$

$$w(s,t) = G_S(s-t,\sigma_S)G_R(I_s - I_t,\sigma_R) \tag{2}$$

In the local histogram of weighted mode filter, the histogram is biased, which means depth occurrence inside the region is adaptively counted on its corresponding bin by using the weight function $w(s,t)$ which is evaluated with the difference of the target pixel and source pixels. The weight function $w(s,t)$ considers not only the spatial weight $G_S(\cdot)$ but also the intensity weight $G_R(\cdot)$ between the target pixel t and the source pixel s within the source window $N(t)$. $G_S(\cdot,\sigma_S)$ and $G_R(\cdot,\sigma_R)$ represents the Gaussian functions with the corresponding standard deviations to measure the data difference for spatial and range weight respectively, where $s=(x_s,y_s)$ and $t=(x_t,y_t)$ are the position of each pixel and I_s and I_t stand for the intensity of pixel s and t. After constructing the local weighted histogram, the bin with the maximum weighted sum in the histogram is the final mode result. Finally, a simple quadratic curve fitting is applied around the mode bin to further improve the precision of the result.

In this work, we aim to propose a real-time upsampling engine to increase the resolution of depth map to 4K Ultra-HD at 30 fps. To achieve such high throughput, a large number of target pixels should be processed simultaneously. Each target pixel possesses its own histogram which is not shareable. That is, the tremendous memory cost for histogram is one of the design challenges. Besides, the other important design issue is the large gate counts due to the complex histogram updating process. To address these problems, we propose the binary range weight kernel to reduce both the memory cost and the gate counts. Then, the histogram candidate mapping architecture is introduced to further reduce the memory cost.

3. DESIGN OF PROPOSED ARCHITECTURE

We propose a VLSI circuit of weighted mode filter to support 4K Ultra-HD depth video upsampling at 30 fps. The whole system architecture is shown in figure 1. The system consists of memory managing unit and WMoF engine group. The memory managing unit is responsible for I/O image data arrangement, and external memory arbitration. To meet throughput requirement, 16 pieces of the WMoF engines are employed in the WMoF engine group.

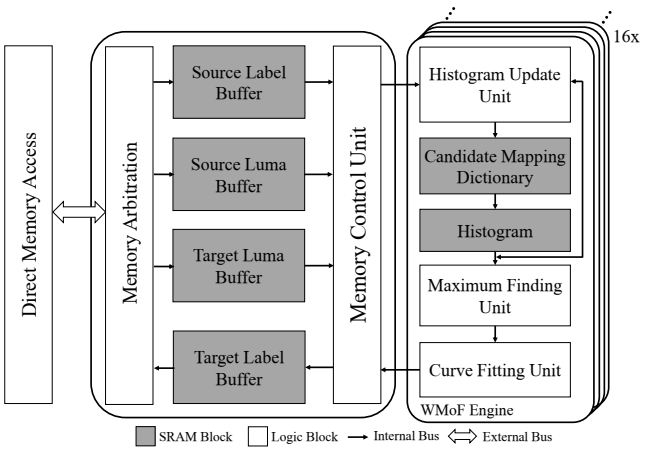

Fig. 1. System architecture.

3.1. Binary Range Weight Kernel

The Histogram Update Unit evaluates weight for each occurrence inside the local histogram. The weight kernel considers not only the spatial difference but also the intensity dissimilarity. Moreover, since each pixel should be calculated independently, the histograms of every target pixel cannot be shared. The gate count is seriously affected by the two factors, even if the kernel evaluation is implemented in look-up table, the histogram update unit accounts for about $34\,\%$ of total area as reported in [10].

A trivial implementation to calculate the weight is combining two Gaussian look-up tables, for range weight and spatial weight respectively. To reduce the gate count, we introduce binary range weight kernel. The detailed architecture of Histogram Update Unit is illustrated in figure 2. The candidate is only counted when the intensity difference is below the threshold. Otherwise the candidate is considered as invalid. On the other hand, the spatial weight is still implemented with a Gaussian look-up table. Though the binary range weight kernel could slightly affect the quality as shown in figure 3, the quality loss is acceptable.

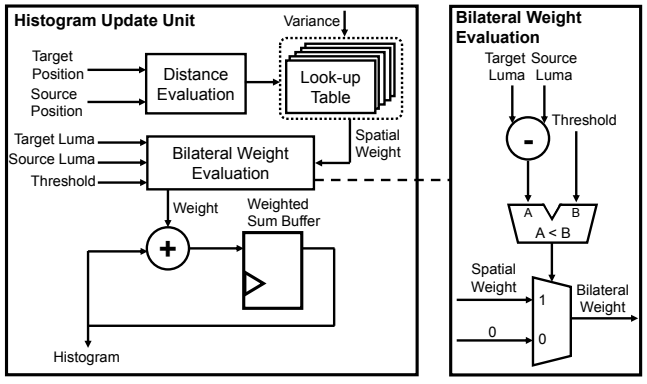

Fig. 2. Hardware design of the histogram update unit.

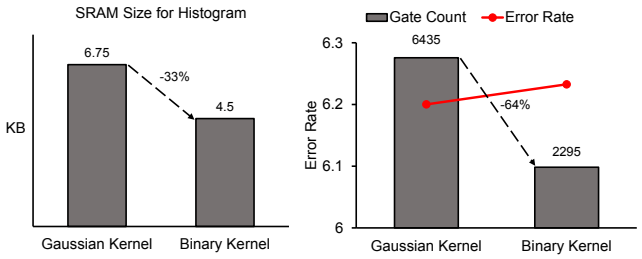

Fig. 3. Evaluation between binary kernel and Gaussian kernel. The experiment was conducted on the test images generated by semi-global-matching (SGM)[11] from Middlebury Stereo Vision[12]. The average error rate contains 12 different data sets.

Compared to Gaussian weight kernel as shown in figure 3, the error rate of binary one increases only $0.5\,\%$, but the gate counts decrease $64.3\,\%$. Besides gate counts, binary weight kernel technique can also save the memory cost. Since the range weight is calculated by binary weight kernel and the spatial weight is still calculated by Gaussian weight kernel, when updating the weighted histogram, only spatial weight requires be accumulated. For our design with binary range weight kernel, 6-bit spatial weight and 12-bit weighted sum are adopted. If Gaussain range weight kernel is adopted, extra 6-bit for weighted sum is required. Therefore, the binary range weight kernel can save the memory cost for histogram by $33.3\,\%$.

It is worth noting that if configurable variance for weight kernel is required, the hardware will suffer overhead from look-up table implementation. The Gaussian kernel will require more look-up tables to support various variance. But for binary kernel, the 8-bit threshold comparator can support full coverage for all possible intensity differences. In our design, we provide 5 different variances for Gaussian spatial weight.

3.2. Histogram Candidate Mapping

Every target pixel needs a histogram SRAM, so the histogram SRAM accounts for up to $90\,\%$ of the on-chip memory. In [10], the histogram is directly mapped to SRAM, that is, the histogram SRAM is designed to have the entries with the same amount of histogram bins to cover all possible depth candidates as shown in figure 4(a). The SRAM requirement for the direct mapping can be formulated as

$$(d \cdot w_{ws}) \cdot n_t \text{ bits}, \tag{3}$$

where d is the amount of depth candidates, n_t is the number of total target pixels, and w_{ws} is the bit-width of the weighted sum. For wide depth range application, the direct mapping architecture requires large amount of SRAM since the memory size linearly increases with depth range. In our target scenario (depth range of 128), the SRAM requirement for the direct mapping will take 48 KBytes.

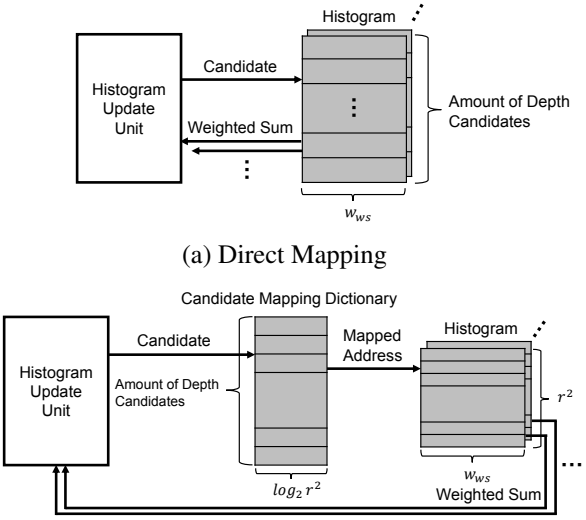

Fig. 4. Histogram updating method.

Since the number of depth labels is more than that of window size for our design, even all candidates within the support window possess different depth values, there will still be redundant bins. Therefore, we utilize the histogram memory redundancy, and we propose the histogram candidate mapping method to reduce the histogram memory usage. In figure 4(b), we demonstrate the histogram candidate mapping method. An additional candidate mapping dictionary is introduced to register the mapping between the source pixel, which ranges from 0 to d, and the address, which ranges from 0 to r^2 and r represents the radius of the support window, for histogram. The total memory requirement can be formulated as

$$(d \cdot log_2 r^2) \cdot n_g + (r^2 \cdot w_{ws}) \cdot n_t \text{ bits}, \tag{4}$$

where $n_g = \frac{n_t}{upsampling\ factor^2}$ is the total target groups that need updating, the first term represents the candidate mapping dictionary, and the second term stands for the histogram. Note that the target pixel group shares the same source candidate, but the pixels in each target pixel group cannot share their histogram.

With the same window size and throughput, as the amount of depth candidates increases, the memory saving percentage is more significant. The high-resolution depth map often requires a wider range for depth labels. Therefore, the histogram candidate mapping architecture can be efficiently adopted in the application scenarios with small window size and wide depth range.

Overall, with upsampling factor 4, both binary range weight kernel and histogram candidate mapping architectures can reduce the memory cost as shown in figure 5. Binary range weight kernel reduce the cost by $33.3\,\%$, and histogram candidate mapping technique reduce the cost by $47\,\%$. With

Table 1. Implementation result on the proposed architecture.

Technology	TSMC 40 nm
Supply Voltage	Core 0.9V, I/O 2.5V
Chip Size	1.67×1.67 mm^2
Core Size	1.09×1.09 mm^2
Gate Count	420k (2-input NAND)
On-Chip Memory	25.5 KByte
Maximum Throughput	320M pixel/sec @ 200MHz
Power Consumption	104 mW
Maximum Output Depth Map Size	3840×2160 @ 40fps
Upsampling Factor	4
Number of Depth Bin	128
Window Size	8×8

Table 2. Implementation performance comparison.

	Histogram-based		Window-based		
	[13]	[14]	[15]	[10]	Ours
Filter Type	Joint Bilateral Filter	Guided Filter	Weighted Median Filter	Weighted Mode Filter	Weighted Mode Filter
Implementation Method	UMC 90nm	TSMC 90nm	Intel i7 3.4GHz CPU	TSMC 40nm	TSMC 40nm
Gate Count (2-input NAND)	356k	92.9k	-	247k	420k
Throughput (pixels/sec)	124M @ 200MHz	62.2M @ 100MHz	4.6M	67.5M @ 200MHz	320M @ 200MHz
On-chip Memory (KBytes)	23	3.2	-	5.4 (2-port)	25.5
Window Size	31×31	31×31	10×10	8×8	8×8
Number of Bins	64	-	256	128	128

both methods, we can reduce the memory cost by 65 % in total as compared to the conventional implementation.

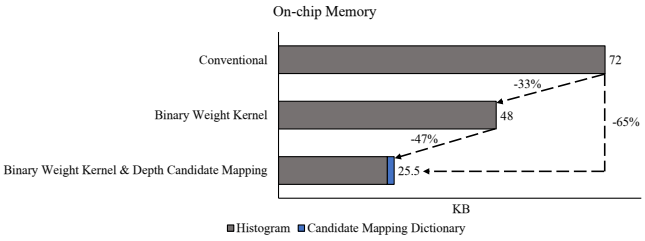

Fig. 5. On-chip memory reduction by the proposed method.

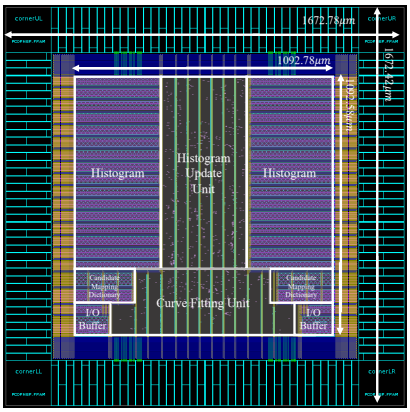

Fig. 6. Chip layout.

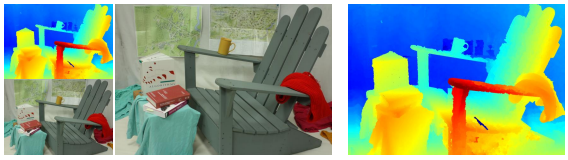

Fig. 7. Result of the proposed architecture with input data sets from Middlebury Stereo Vision[12].

4. IMPLEMENTATION

The proposed architecture of the weighted mode filter has been implemented with Verilog-HDL and synthesized under the TSMC 40nm technology process, and the result are summarized in the table 1. The final layout is shown in the figure 6. The bit-width of the input depth label is 7, and the design further increases its precision to 9 bits with the curve fitting unit. The total power consumption is 104 mW at 0.9 V and 200 MHz, and it is worth noting that the on-chip memory accounts for up to 74 %.

Table 2 compares specification between our implementation and other different filters. Tseng *et al.* [13] proposed the joint bilateral filter with large supported window and high throughput, but the number of the depth candidate is low. Kao *et al.* [14] proposed VLSI architecture design of guided filter which can deliver high throughput with low hardware cost. However, joint bilateral filter and guided filter are not suitable for depth map upsampling. Compared to Chen *et al.*[10], although they have smaller SRAM size, our design provides 4.7x higher throughput and uses single-port SRAM rather than two-port SRAM for overall SRAM area and power consumption concern.

5. CONCLUSION

In this work, after analyzing the depth quality and hardware complexity, we propose an VLSI architecture design of weighted mode filter for 4K Ultra-HD depth map upsampling. Our two major contribution: binary range weight kernel and histogram candidate mapping address the gate count and memory cost issues. The binary range weight kernel reduce the gate counts by 64 % and the histogram memory cost by 33 %. Then, the histogram candidate mapping architecture can further reduce the histogram memory by 47 %. With the proposed architecture, the system delivers 320M pixel/s at 0.9V and 200MHz with a gate count of 420k and 25.5 KB on-chip memory.

6. REFERENCES

[1] S. Schuon, C. Theobalt, J. Davis, and S. Thrun, "LidarBoost: Depth superresolution for tof 3D shape scanning," in *2009 IEEE Conference on Computer Vision and Pattern Recognition*, June 2009, pp. 343–350.

[2] S. Schuon, C. Theobalt, J. Davis, and S. Thrun, "High-quality scanning using time-of-flight depth superresolution," in *2008 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, June 2008, pp. 1–7.

[3] Y. Cui, S. Schuon, D. Chan, S. Thrun, and C. Theobalt, "3D shape scanning with a time-of-flight camera," in *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, June 2010, pp. 1173–1180.

[4] Q. Yang, R. Yang, J. Davis, and D. Nister, "Spatial-depth super resolution for range images," in *2007 IEEE Conference on Computer Vision and Pattern Recognition*, June 2007, pp. 1–8.

[5] J. H. Cho, S. Ikehata, H. Yoo, M. Gelautz, and K. Aizawa, "Depth map up-sampling using cost-volume filtering," in *IVMSP 2013*, June 2013, pp. 1–4.

[6] Johannes Kopf, Michael F. Cohen, Dani Lischinski, and Matt Uyttendaele, "Joint bilateral upsampling," *ACM Trans. Graph.*, vol. 26, no. 3, July 2007.

[7] K. L. Hua, K. H. Lo, and Y. C. F. Frank Wang, "Extended guided filtering for depth map upsampling," *IEEE MultiMedia*, vol. 23, no. 2, pp. 72–83, Apr 2016.

[8] Z. Ma, K. He, Y. Wei, J. Sun, and E. Wu, "Constant time weighted median filtering for stereo matching and beyond," in *2013 IEEE International Conference on Computer Vision*, Dec 2013, pp. 49–56.

[9] D. Min, J. Lu, and M. N. Do, "Depth video enhancement based on weighted mode filtering," *IEEE Transactions on Image Processing*, vol. 21, no. 3, pp. 1176–1190, March 2012.

[10] L. D. Chen, Y. L. Hsiao, and C. T. Huang, "VLSI architecture design of weighted mode filter for Full-HD depth map upsampling at 30fps," in *2016 IEEE International Symposium on Circuits and Systems (ISCAS)*, May 2016, pp. 1578–1581.

[11] H. Hirschmuller, "Accurate and efficient stereo processing by semi-global matching and mutual information," in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, June 2005, vol. 2, pp. 807–814 vol. 2.

[12] "Middlebury stereo," vision.middlebury.edu/stereo/.

[13] Y. C. Tseng, P. H. Hsu, and T. S. Chang, "A 124 Mpixels/s VLSI design for histogram-based joint bilateral filtering," *IEEE Transactions on Image Processing*, vol. 20, no. 11, pp. 3231–3241, Nov 2011.

[14] C. C. Kao, J. H. Lai, and S. Y. Chien, "VLSI architecture design of guided filter for 30 frames/s Full-HD video," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 24, no. 3, pp. 513–524, March 2014.

[15] Q. Zhang, L. Xu, and J. Jia, "100+ times faster weighted median filter (WMF)," in *2014 IEEE Conference on Computer Vision and Pattern Recognition*, June 2014, pp. 2830–2837.