SYSTEM AND VLSI IMPLEMENTATION OF PHASE-BASED VIEW SYNTHESIS

Han-Chih Huang, Yu-Chih Wang, Wei-Chih Chen, Ping-Yen Lin, and Chao-Tsung Huang

National Tsing Hua University
Department of Electrical Engineering
Hsinchu, Taiwan

ABSTRACT

View synthesis is one of the important techniques utilized in 3D TV devices. Traditional methods such as depth image-based rendering usually rely on accurate depth maps which require computation-intensive stereo matching. In this work, we present a hardware system of phase-based view synthesis that is able to convert stereoscopic videos to multi-view content with low-resolution depth maps. When compared to the view synthesis reference software, the phase-based method does not suffer from severe arifacts on object boundaries and provides higher quality views in our experimental result. There are two major contributions in our implementation. First, we propose a cross-band disparity correction scheme that not only enables the usage of low-resolution disparity maps but also improves the quality of novel views. Second, we propose a hardware-friendly wavelet re-projection engine to reduce the hardware complexity. We implemented a VLSI circuit for 8-view 4K Ultra-HD (UHD) 3DTV in TSMC 40nm technology. It delivers 30 frames per second (fps) for UHD display when operating at 200MHz. It uses 228-KB SRAM and 2M-gate logic . We also implemented the system on FPGA, and it can provide 4K UHD multi-view content at 12 fps.

***Index Terms*—** View synthesis, phase-based processing, VLSI architecture, 3D TV

1. INTRODUCTION

Conventional view synthesis algorithms mainly apply depth image-based rendering (DIBR) [1][2] that re-projects images to new viewpoints according to dense depth maps. However, DIBR suffers from artifacts especially on object boundaries. In addition, it requires accurate depth maps for image warping and additional inpainting skills for hole filling.

Recently, a novel approach based on phase manipulation was proposed. This method is inspired from the observation that motion can be encoded in phase difference. Didyk et al. first implemented this phase-based approach for view synthesis in [3]. They decompose signals into Gabor-like wavelets by applying a steerable pyramid [4] and then manipulate disparity based on local phase differences instead of a per-pixel disparity map. New viewpoints are rendered without disparity estimation, but this framework is limited to small disparities. Kellnhofer et al. proposed an Eulerian-Lagrangian method to solve this issue by combining the phase-based method with DIBR in [5]. This approach takes disparity maps as input and re-projects each wavelet to increase the disparity range.

In the original algorithm of [5], it adopts a wavelet re-projection filter with heavy computation. To reduce the hardware complexity of this step, we replace it with the Lanczos filter which requires less complexity. To improve the quality of novel views, we propose a cross-band disparity correction scheme to further refine the per-wavelet disparity map. Based on our modification to [5], we design a VLSI circuit and implement it with TSMC 40nm technology and FPGA respectively. We will first introduce the framework of the original algorithm [5] in Section 2. In Section 3, we introduce our proposed system and VLSI design. Then the implementation details and results are discussed in Section 4. Finally, we conclude our work in Section 5.

2. EULERIAN-LAGRANGIAN VIEW SYNTHESIS

The key to this algorithm is a wavelet representation incorporating with initial disparity maps to estimate per-wavelet disparity. The overall algorithm consists of three main stages. Input signals are decomposed into a wavelet pyramid in the first stage. The second stage improves the quality of initial disparity maps by phase differences. Finally, wavelets are re-projected to new positions according to the per-wavelet disparity map. Novel views are then obtained by pyramid reconstruction.

Wavelet decomposition. In [5], rectified stereo images and their corresponding disparity maps are sent into the system. Since input views are rectified, input signals can be processed in scanline fashions. The steerable pyramid [4] is then applied to perform 1D decomposition on input signals I, and the transfer function of the filter banks resemble Gabor-like wavelets. By using the filter banks Ψ_f of [4], a single wavelet coefficient is computed for a given location x and frequency f as: $A_{fx} = (\Psi_f * I)(x)$. To perform decomposition, dis-

This work was supported by the Ministry of Science and Technology, Taiwan, under Grant MOST 106-2221-E-007-120.

crete fourier transform (DFT) is applied to convert I into $\tilde{I}$ and then different frequency bands are obtained by multiplying $\tilde{I}$ with $\tilde{\Psi}_f$ in frequency domain. The transfer functions of these bandpass filters are designed to contain only positive frequencies. Thus, real signals can be converted into complex signals after the decomposition.

The reconstruction is done similarly in the frequency domain. Therefore, for hardware design of the decomposition and reconstruction, several Fast Fourier Transform (FFT) [6] engines are required to convert signals between frequency and spatial domains. To reduce the complexity of FFT engines, we propose to merge their delay buffers and perform a thorough precision analysis.

Disparity Refinement. In [5], per-wavelet disparity maps of different levels are calculated by considering input disparity maps and phase differences. Phase differences are estimated between corresponding wavelets in the stereo image pair. For wavelet ψ_{rx} in right view, the corresponding wavelet $\psi_{lx'}$ can be found in the left view with $x' = x - d_{rx}$, where d_{rx} is the initial wavelet disparity. After that, a refined phase difference ϕ_{diff} between them can be computed by:

$$\phi_{diff} = atan2(sin(\phi_{rx} - \phi_{lx'}), cos(\phi_{rx} - \phi_{lx'})), \quad (1)$$

where ϕ_{rx} and $\phi_{lx'}$ are the phase of ψ_{rx} and $\psi_{lx'}$ respectively. The positional shift Δd, which corresponds to the spatial displacement, can be calculated by scaling the phase difference by ω:

$$\Delta d = \frac{\phi_{diff}}{\omega}, \quad (2)$$

where $\omega = 2\pi f$. Finally, the wavelet disparity can be updated by added Δd back to d_{rx}.

In our implementation, we found that the phase difference fails to correct the wavelet disparity when the initial disparity is significantly incorrect. To solve this issue, we propose a cross-band disparity correction scheme which corrects wavelet disparity of each level.

Novel Views Reconstruction. The position of each wavelet is modified similarly to the pixel warping in DIBR. Each wavelet is re-projected to its new position according to the per-wavelet disparity map. The new position of each wavelet at location x and disparity d is computed as $x + \alpha d$, where α controls the novel view position. This method modifies the position of each wavelet while traditional phase-based method only adjusts phase differences. After that, a non-uniform Fast Fourier Transform (NUFFT) is used to convert the displaced non-uniform wavelets into uniform grid. It first converts wavelets into an oversampled grid. Then, a down-sample filter is used to down-sample wavelets back into the original grid. The reader is referred to the original paper [7] for more details. Finally, a pyramid is reconstructed by combining lowpass residuals and wavelets through all the frequency bands.

In the hardware implementation, we observed that the original re-projection filter requires large computational cost. We propose a 7-tap Lancozs filter to reduce the computational complexity. Another issue of the hardware design is that several line buffers are required to store the re-projected wavelets. We eliminate this need by careful scheduling between re-projection and reconstruction engines.

3. SYSTEM AND VLSI IMPLEMENTATION

3.1. Cross-band Disparity Correction

To correct wavelet disparity of each level, we have two assumptions. First, the high frequency wavelet moves in a similar way to the corresponding low frequency one. Second, we assume that the phase difference between corresponding wavelets in the stereo image pair is not greater than $\pi/2$ since it is derived after aligning the corresponding wavelets. Thus, when the phase difference in the current level is greater than $\pi/2$, we correct the wavelet disparity to the corresponding disparity from the lower level.

Since the source code of [5] is unavailable, we built our own implementation for the phase-based view synthesis. We compare the quality of novel views between input disparity maps of different resolutions. Fig. 1 shows that our implementation is less sensitive to the resolution of disparity maps after we apply the cross-band disparity correction. Furthermore, the quality of novel views are also improved by the disparity correction. We also compare our implementation to the view synthesis reference software (VSRS) [8] algorithm which is a popular DIBR method. Our implementation has better PSNR and SSIM than VSRS. Although novel views generated by our implementation do not show sharp edges as VSRS does, they do not suffer from severe artifacts near boundaries as shown in Fig. 2.

3.2. System Overview

We propose a VLSI circuit for phase-based view synthesis to support 4K UHD 3D TV at 30 fps. The whole system diagram is shown in Fig. 3. Our design consists two stages. The first stage aims to refine the initial disparity. It contains the decomposition and the disparity refinement engines. The three channels of stereo views are decomposed into wavelets simultaneously by six decomposition engines which mainly consist of FFT engines to perform filtering in the frequency domain. The wavelet disparity is refined by the disparity refinement engine according to the desired phase difference. Finally, we store the wavelets and the wavelet disparity information into SRAM.

In order to support an 8-view 3D TV, we provide seven novel views and one existing right view. In the second stage, we generate the three channels of seven novel views simultaneously. Thus, there are 21 reconstruction and 21 wavelet re-projection engines in total. For the reconstruction engine, it also consists of the FFT engine. For the wavelet re-projection

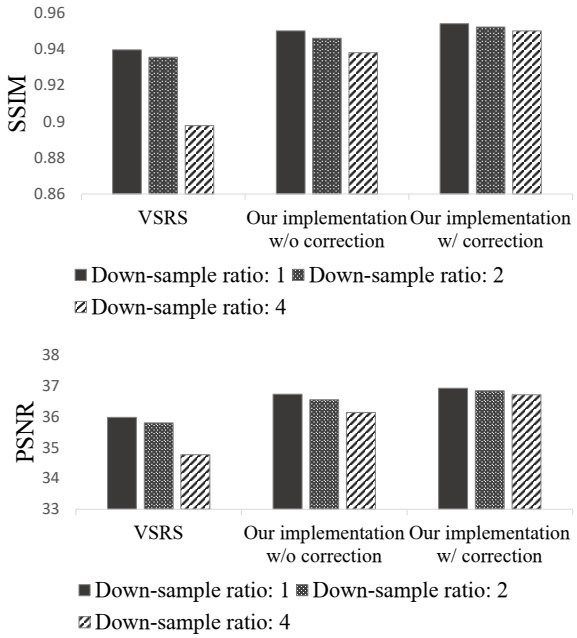


Fig. 1. Evaluation of different resolutions of input disparity maps. We compare ground truth images from Heidelberg Collaboratory for Image Processing Light Field (HCI) [9] datasets. The quality of novel views are evaluated by Strutural Similarity Index (SSIM) and peak signal-to-noise ratio (PSNR).

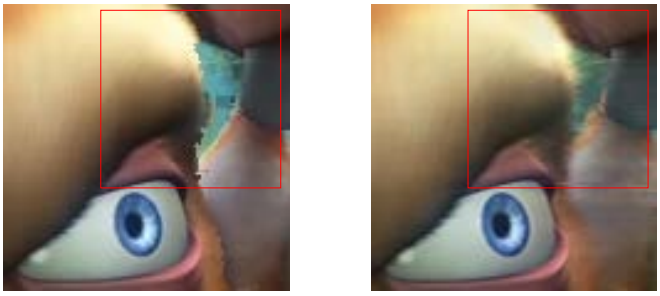
engine, it finds the new position of each wavelet and performs Lanczos filtering.

3.3. FFT Design

As shown in Fig. 3, there are totally 27 sets of FFT engines which consume large hardware resource in our design. To reduce the complexity, we adopt the SRAM-based single path delay feedback (SDF) structure [10] to implement FFT because it provides high throughput and requires less memory. We further merge the delay buffers of FFT engines in the decomposition and reconstruction engines, respectively, to make them more compact. This merging reduces the area of the system by 9.5 % compared to a direct implementation. Furthermore, we do experiments on the fixed point precisions to guarantee that novel views do not suffer from severe quality loss as shown in Fig. 4. We finally choose 14-bit for the FFT wordlength and 9-bit for the twiddle factor.

3.4. Wavelet Re-projection Engine

To reduce the complexity of the wavelet re-projection engine, we replace the NUFFT and down-sample filter by the Lanczos filter in our implementation. We do experiments on filter tap number to guarantee the quality of novel views as shown in Fig. 5. Finally, we select the seven-tap Lanczos filter as it



(a) Synthesized by VSRS (b) Synthesized by our implementation

Fig. 2. Quality comparison of novel views synthesized by VSRS and our implementation.

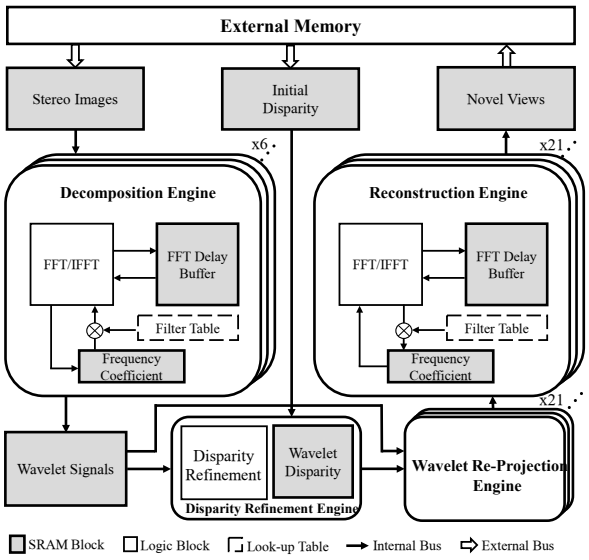


Fig. 3. System diagram.

saves 63 % of the computation while remaining similar quality when compared to the NUFFT and seven-tap down-sample filter.

Another design issue is that the re-projected wavelets need to be stored into several line buffers before reconstruction. To save the line buffers, we design the wavelet re-projection engine to generate novel wavelets consecutively in every cycle. As the design of FFT is an SDF structure that takes serial input, we can save 40.8 KBytes on-chip memory by interlacing the reconstruction and wavelet re-projection engines without line buffers.

4. IMPLEMENTATION RESULTS

4.1. TSMC 40 nm Synthesis Result

We target to produce contents for an 8-view 4K (4096×2160) automultiscopic display, where each output view has a resolution of 1024×1080. The proposed design has been imple-

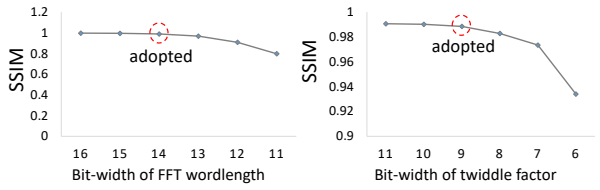


Fig. 4. Quantitive comparison between different precisions on HCI dataset.

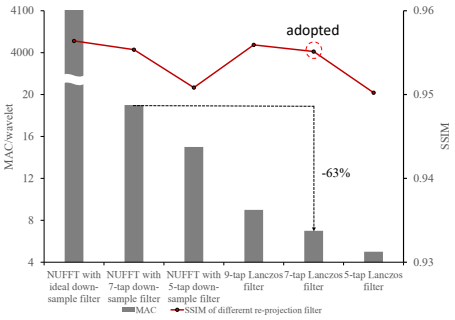


Fig. 5. Comparison between different re-projection filters on HCI dataset. We compare Lanczos filters of different taps with the NUFFT and its down-sample filters.

mented with Verilog-HDL and synthesized with TSMC 40nm technology process. In Table 1, it is shown that our design has comparable logic complexity when compared to [11] which is a high throughput VLSI design with 65nm process for image domain warping (IDW). Compared to the VSRS implementation [12], our design can synthesize higher-quality novel views but requires higher hardware complexity. The complexity issue comes from the adoption of several FFT engines. One possible future extension of this work is to design low-cost spatial filters to replace those FFT engines.

Table 1. Implementation performance comparison.

	[11]	[12]	This work
Algorithm	IDW	VSRS	Phase-based
Technology Process	UMC 65 nm	UMC 90 nm	TSMC 40 nm
Operating Frequency	260MHz	200MHz	200MHz
Throughput	477 Mpixel/s	67 Mpixel/s	266 Mpixel/s
Logic Gate Count (two-input NAND)	2.3M	268.5K	2M
On-chip Memory (KBytes)	294.4	69.3	228
Output views	8	1	8

4.2. FPGA Implementation Result

Our hardware design was implemented on Xilinx ZC706 platform. It occupies 142K LUTs, 58K Registers, and 228 KBytes of BRAM. It operates at 80MHz and provides UHD multi-view content at 12 fps. In Table 2, we compare our implementation result to the result of [5] which uses high level synthesis to construct hardware blocks. The registers and BRAM usage of the previous work are relatively high compared to ours. The proposed work mainly improves the BRAM utilization by 88 % and increases the throughput by 100 % even with a slower working frequency. Finally, we build an FPGA demo system in Fig. 6.

Table 2. FPGA implementation comparison.

	[5]	This work
FPGA	Xilinx ZC706	Xilinx ZC706
Clock	150MHz	80MHz
[LUT, Registers]	[101K, 117K]	[142K, 58K]
BRAM	1927 KBytes	228KBytes
DSP utilization	59 %	100 %
Throughput	53MPixel/s	106MPixel/s

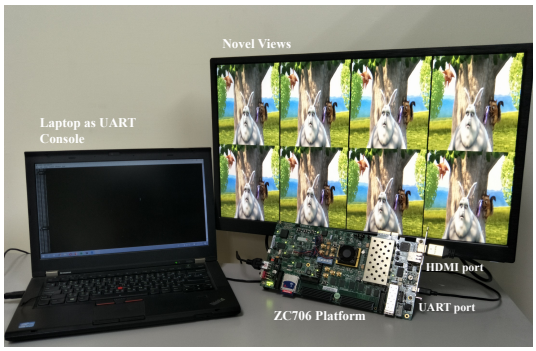


Fig. 6. Photograph of the demo system: After booting, the system starts to read stereo videos and their corresponding wavelet disparity maps from the SD card. We combine eight novel views into a frame and show it on an external display through HDMI. A console UI is implemented through UART. Users can adjust the parameter $\Delta\alpha$ which controls the distance between novel view through the console.

5. CONCLUSION

In this work, we propose a VLSI design of phase-based view synthesis. To our knowledge, this is an early work that implemented the phase-based method into hardware. To save the computational cost of the re-projection filter, we adopt a 7-tap Lanczos filter. To lower the sensitivity to the resolution of input disparity maps, we propose the cross-band disparity correction. With the proposed hardware design, the system delivers 266M pixel/s at 200MHz with 2M gate logic and 228 KB on-chip memory. We hope that this paper bring more hardware design discussions on the phase-based method for real-time applications.

6. REFERENCES

[1] P. Kauff, N. Atzpadin, C. Fehn, M. Müller, O. Schreer, A. Smolic, and R. Tanger, "Depth map creation and image-based rendering for advanced 3DTV services providing interoperability and scalability," *Signal Processing: Image Communication*, vol. 22, no. 2, pp. 217–234, 2007.

[2] C. Riechert, F. Zilly, P. Kauff, J. Güther, and R. Schäfer, "Fully automatic stereo-to-multiview conversion in autostereoscopic displays," in *Proc. IBC*, 2012, pp. 8–14.

[3] P. Didyk, P. Sitthi-Amorn, W. T. Freeman, F. Durand, and W. Matusik, "Joint view expansion and filtering for automultiscopic 3D displays," *ACM Transactions on Graphics (TOG)*, vol. 32, no. 6, pp. 221, 2013.

[4] E. P. Simoncelli and W. T. Freeman, "The steerable pyramid: A flexible architecture for multi-scale derivative computation," in *Proceesings of International Conference on Image Processing*. IEEE, 1995, vol. 3, pp. 444–447.

[5] P. Kellnhofer, P. Didyk, S. P. Wang, P. Sitthi-Amorn, W. T. Freeman, F. Durand, and W. Matusik, "3DTV at home: Eulerian-Lagrangian stereo-to-multiview conversion," *ACM Transactions on Graphics (TOG)*, vol. 36, no. 4, pp. 146, 2017.

[6] J. W. Cooley and J. W. Tukey, "An algorithm for the machine calculation of complex Fourier series," *Mathematics of computation*, vol. 19, no. 90, pp. 297–301, 1965.

[7] Q. H. Liu and N. Nguyen, "An accurate algorithm for nonuniform fast Fourier transforms (NUFFT's)," *IEEE Microwave and Guided Wave Letters*, vol. 8, no. 1, pp. 18–20, Jan 1998.

[8] *Test Model 8 of 3D-HEVC and MV-HEVC*, ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11 document JCT3V-H1003, Apr. 2014.

[9] S. Wanner, M. Sven, and B. Goldluecke, "Datasets and benchmarks for densely sampled 4D light fields.," in *Annual Workshop on Vision, Modeling and Visualization: VMV*, 2013, pp. 225–226.

[10] S. He and M. Torkelson, "A new approach to pipeline FFT processor," in *Proceedings of International Conference on Parallel Processing*, April 1996, pp. 766–770.

[11] M. Schaffner, P. Greisen, S. Heinzle, F. K. Gürkaynak, H. Kaeslin, and A. Smolic, "Madmax: A 1080p stereo-to-multiview rendering ASIC in 65 nm CMOS based on image domain warping," in *Proceedings of the ESSCIRC (ESSCIRC)*, Sept 2013, pp. 61–64.

[12] Y. R. Horng, Y. C. Tseng, and T. S. Chang, "VLSI architecture for real-time HD1080p view synthesis engine," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 21, no. 9, pp. 1329–1340, Sept 2011.