Low-Cost Hardware Architecture Design for 3D Warping Engine in Multiview Video Applications

Pin-Chih Lin, Pei-Kuei Tsung, and Liang-Gee Chen
DSP/IC Design Lab, Graduated Institute of Electronics Engineering
National Taiwan University
Taipei, Taiwan
{lpg, iceworm, lgchen}@video.ee.ntu.edu.tw

***Abstract*—As the history of television industry goes, multiview video (MVV) and its applications draw more and more attentions by the realistic 3D scene it can bring. In these applications, virtual view synthesis is required for providing free view point sequences so as to fulfill a real-time display system. In this paper, a low-area architecture is proposed. By employing the linear-interpolated approximation algorithm, the large area requirement due to the synthesis parameters is resolved. In addition, redundant information for fraction bits of parameters is further reduced by precision fitting analysis. As a result, 95.9% of area for matrix parameter rendering stage and 69.5% for vector transform stage are reduced with only 0.0059 dB overhead of PSNR performance.**

I. INTRODUCTION

Multiview video (MVV) can provide viewers a complete 3D world scene with capturing multiple video sequences from different view-points. Furthermore, several emerging MVV applications, such as 3DTV [1] and free view point TV (FTV) [2], are conceptually proposed and widespread discussed. These applications bring the epochal change with an innovative media that enables viewers to view real world scenes as if they were there by freely changing their view points. However, to capture and transmit infinite video sequences from all free view-points are critical challenges for such applications. Due to the project, only parts of anchor videos need to be captured and transmitted. Then, the virtual view synthesis process plays an important role to synthesize all the other views. Thus, the view synthesis reference software (VSRS) is released by MPEG-FTV group as a research platform since November 2008 [3]. The concept of the virtual view synthesis process is to warp two original images to synthesize the intermediate virtual image by 3D space projection. Depth-image-based rendering (DIBR) is a well-accepted method of pixel-to-pixel 3D space warping by providing per-pixel depth information [4], which is more general than 1D-parallel disparity synthesis [5]. In the virtual view synthesis process, DIBR occupies a quite major proportion of computational cost because matrix-based multiplications and scalar divisions are included in per-pixel process. Figure 1 shows the runtime profiling result of general mode view synthesis procedure in VSRS on Windows32 platform with Microsoft Visual Studio .NET. It can be observed that more than a half of the processing time is consumed for DIBR warping. Owing to the feasibility of 3DTV and FTV concept for real-time display system, hardware implementation for view synthesis acceleration has been mentioned in the previous work [6]. Similarly, the acceleration of DIBR warping is undeniably the most important task to deal with. Intuitively, a fast and low-cost hardware design of pixel-to-pixel warping engine is indispensable for a FTV display and an essential issue to be discussed. In this paper, we propose a low-cost hardware design of warping engine. Firstly the simpler DIBR method, homographic transform, is implemented with a further reduction of the required parameter matrices to save up to 95.9% of the storage area cost by linear-interpolated manner. Then, the suitable precisions of binary fraction of all parameters are analyzed to optimize the design while keeping the same performance with only 0.0059 dB overhead of PSNR.

Rest of this paper is organized as follows: methods of DIBR warping and hardware design challenges are introduced with more mathematical details in Sec. II. Section III describes our proposed scheme and analysis. Simulation results are shown in Sec. IV and conclusions are in Sec. V.

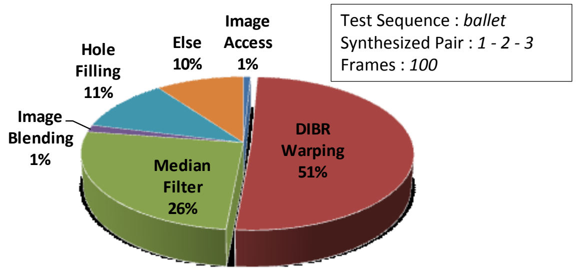

Figure 1. Runtime profiling of VSRS general mode

II. Preknowledges and Design Challenges

To synthesize a virtual view image from an original view with different other view-point, DIBR contains the concatenation of an image-to-world point reprojection and the subsequent world-to-image projection utilizing the respective depth data just as Fig. 2 depicts. In this figure, ***u*₁*(x*₁*,y*₁*,1)*** of original view is warped to the corresponding point ***u*₂*(x*₂*,y*₂*,1)*** on the synthesized image plane and the depth value *Z* locates the real world object ***V****(X,Y,Z)* during the projection process. Matrix-based 3D pixel projection is the conventional model for DIBR [4]. Homographic transform is another approach in camera graphics and employed in VSRS software. These two practically equivalent models will be introduced as follows.

1) Matrix-based 3D pixel projection: The corresponding image point on the synthesized view can be obtained by two projective matrix-based equations shown as follows:

$$\begin{bmatrix} X & Y & Z \end{bmatrix}^T = \alpha_1 \mathbf{R}_1^{-1} \mathbf{A}_1^{-1} \begin{bmatrix} x_1 & y_1 & 1 \end{bmatrix}^T - \mathbf{T}_1 \qquad (1)$$

$$\begin{bmatrix} x_2 & y_2 & 1 \end{bmatrix}^T = \alpha_2 \left(\mathbf{A}_2 \mathbf{R}_2 \begin{bmatrix} X & Y & Z \end{bmatrix}^T + \mathbf{A}_2 \mathbf{R}_2 \mathbf{T}_2 \right) \qquad (2)$$

Symbols with bold font in (1) and (2) stand for matrices, and where **A** denotes a 3-by-3 intrinsic matrix of camera, **R** denotes a 3-by-3 rotation matrix, and **T** denotes a 3-by-1 translation vector according to the camera arrangement. The index 1 or 2 represents that parameter is for original view or synthesized view, respectively. During the calculation of these two equations, α is the scaling variable and needed to be figured out. That means divisions are inevitable for both equations, concerned with adopting hardware dividers that we wish to avoid because of their large area overheads.

2) Homographic Transform (HT): A 3D-vectors transform is represented by a non-singular 3-by-3 matrix **H**:

$$w'_2 \begin{bmatrix} x_2 \\ y_2 \\ 1 \end{bmatrix} = \begin{bmatrix} x'_2 \\ y'_2 \\ w'_2 \end{bmatrix} = \begin{bmatrix} h_{XX} & h_{XY} & h_{XI} \\ h_{YX} & h_{YY} & h_{YI} \\ h_{IX} & h_{IY} & 1 \end{bmatrix}_{\mathbf{H}(Z)} \begin{bmatrix} x_1 \\ y_1 \\ 1 \end{bmatrix} \qquad (3)$$

In camera graphics [7], there is a homogenous transform from all ***u*₁** among image region on original view to ***u*₂** on the synthesized view while depth value *Z* is fixed. Equation (3) is the mathematical model for HT which contains only one matrix multiplication process and the bottom-right entry of **H** is always 1. Since the depth value *Z* in MVV applications is evaluated as an 8-bit number from 0 to 255, 256 homographic matrices are required for transform with *Z*. Divisions are also regarded because we have to scale the vector *(x'₂,y'₂,w'₂)* letting the third-coordinate value becomes 1. That is, *x₂=x'₂/w'₂* and *y₂=y'₂/w'₂*. Another form of equation model can be introduced by non-matrix manner.

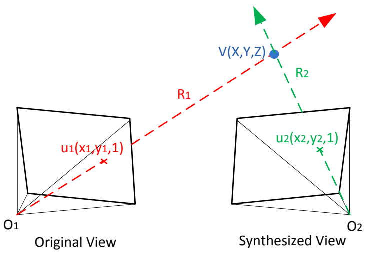

Figure 2. Pixel-based 3D warping process

$$x_2 = \frac{h_{XX} x_1 + h_{XY} y_1 + h_{XI}}{h_{IX} x_1 + h_{IY} y_1 + 1}, \; y_2 = \frac{h_{YX} x_1 + h_{YY} y_1 + h_{YI}}{h_{IX} x_1 + h_{IY} y_1 + 1} \qquad (4)$$

An important design challenge for implementing DIBR warping model as hardware is the large computational complexity and circuit cost for its arithmetic model. In comparison, simpler calculations are taken to find warped points by HT than matrix-based 3D pixel projection method. Table I indicates both arithmetic operator costs and runtime ratio for two DIBR models above. The comparative results of computational costs are estimated by software and show that the runtime of 3D pixel projection model is 5.92 times longer than HT. As for hardware design, there is a great reduction of arithmetic operator amount especially for dividers and multipliers by HT. Unfortunately, some barricades are placed on the road of hardware implementation of HT warping engine such as the unacceptable storage area of 256 homographic matrices, which is up to 200,000 gate counts for 64-bit matrix entry each and being stored by dual-port SRAM, which is estimated by UMC 90nm technology. Moreover, large and redundant fraction accuracy of parameters can fatally drag down the design efficiency by increasing timing path and area with respect to hardware architecture.

III. Proposed Scheme

A. Linear-Interpolated Approximation

As mentioned above, although the HT warping engine can be fulfilled with less arithmetic operators, considerable amount of parameters for H matrices makes the design area explode. Thus, the optimization for reducing stored parameters is quite necessary. For this reason, we try to dig out the relationship between different entries and matrices with different depth value *Z*. It has been observed that values of the same entry are closed to be linearly distributed among *Z*. In statistical result this issue is also testified since the square of correlation coefficient can reach almost 0.99 for every entry in **H**. Table II lists the average difference rate between real value *hᵢ* and estimated value *hᵢ,linear* by linear regression of least squares error method. Even the largest average error of linear regression is under 1%, and it is sufficient to stand up for evaluating **H** by linear approach to prevent the huge cost of storing them all.

TABLE I. Arithmetic Operator And Runtime Cost Comparison

Warping Method	Arithmetic Operator Cost			Computing Runtime Cost Ratio
	add/subtract	*multiplication*	*division*	
3-D Pixel Projection	18	17	3	5.92x
Homographic Transform	6	6	2	

TABLE II. Average Error Rate for Linear Regression

Sequence	Synthesized Pair (OV-SV)*	Average Error Rate = $\underset{i\in[0,255]}{avg}\left(\left\|\tilde{h}_{i,linear} - h_i\right\|/h_i\right) \times 100\%$							
		h_{XX}	h_{XY}	h_{XI}	h_{YX}	h_{YY}	h_{YI}	h_{IX}	h_{IY}
ballet	1 - 2	0.002	0.007	0.538	0.001	0.001	0.003	0.003	0.659
breakdancers	3 - 4	0.001	0.024	0.823	0.001	0.001	0.013	0.004	0.022
pantomime	37 - 38	0	0	0.061	0	0.001	0.046	0	0

OV-SV : original view number – synthesized view number

Linear-interpolated approximation (LIA) denotes the method of only knowing the head matrix **H***(Z=0)* and tail matrix **H***(Z=255)*, then providing linear interpolation for approximating all other intermediate matrices. In spite of preparing all information of 256 matrices, LIA needs just two matrices with an interpolation calculation. Besides, the accuracy of LIA model can be boosted by interpolating matrices among shorter intervals. LIA-*n* defines the proposed LIA model by cutting depth value *Z* into *n* intervals. For each interval, head and tail matrices are known. We also regulate *n* to orders of 2 due to binary characteristic of hardware designing, i.e. LIA-2, LIA-4, and LIA-8.

$$\mathbf{H}_{\mathbf{LIA\text{-}1}}(Z)=\mathbf{H}_{\mathbf{base}}+\frac{Z}{256}\cdot\mathbf{H}_{\mathbf{inc}} \tag{5}$$

$$\mathbf{H}_{\mathbf{LIA\text{-}2}}(Z)=\begin{cases}\mathbf{H}_{\mathbf{base,0}}+\dfrac{Z}{128}\cdot\mathbf{H}_{\mathbf{inc,0}} & ,\ Z<128\\[2ex] \mathbf{H}_{\mathbf{base,1}}+\dfrac{(Z-128)}{128}\cdot\mathbf{H}_{\mathbf{inc,1}} & ,\ Z\geq 128\end{cases} \tag{6}$$

Equation (5) and (6) indicate how linear approximation models work of LIA-1 and LIA-2. LIA-4 and LIA-8 are similar to LIA-2 with more conditions. For example of (5), the number of matrix information needed is down to 2 from 256, where $\mathbf{H}_{\mathbf{base}}$ = **H***(0)* and $\mathbf{H}_{\mathbf{inc}}$ = *256***H***(255) /255*. Here we do not implement the conventional linear interpolation equation with weighted sum of head and tail matrices, because (5) can be constituted by less multiplication operators than it. Figure 3 illustrates the hardware architectures of pixel-to-pixel DIBR warping engines. Since HT algorithm is chosen for proposed implementation, architecture can be divided into two stages, H. matrix rendering stage and vector transform stage. The former obtains the correspondence **H** with input value *Z* and the latter performs matrix multiplication and division as in (3). In the direct implementation, a 200,000-gate-count H. matrix LUT SRAM is implanted like Fig. 3(a). Figure 3(b) shows that the proposed LIA dwindles the design of H. matrix rendering stage by only a linear interpolation block and a selector of $\mathbf{H}_{\mathbf{base}}$ and $\mathbf{H}_{\mathbf{inc}}$ in our proposed scheme.

When the LIA-2 scheme in Eq. (6) is chosen, double of matrix information will be stored, but smaller approximation error is achieved than LIA-1 as we observed from Fig. 4. Note that the average pixel error is derived from the pixel location difference on the synthesized view while using the LIA-estimated **H** matrix or the original **H** matrix with every single depth value *Z*. It makes sense that in Fig. 4 there are mountain-shaped curves among each interval for all LIA models. In brief, more intervals we split for LIA model, smaller location error we get. On the other hand, the storing area for required matrix information grows as a tradeoff, whereas it still costs lower for hardware design compared with the original architecture.

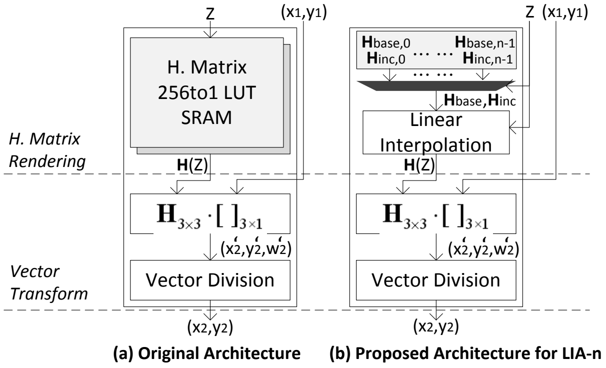

Figure 3. Hardware architecture of 3D warping engine

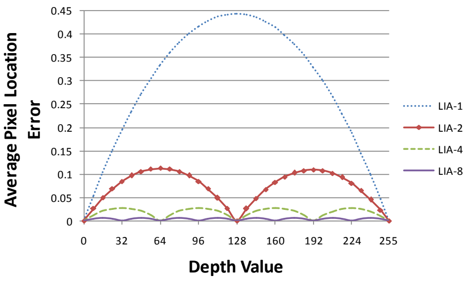

Figure 4. Average error curve for different LIA models

B. Precision Fitting

The issue of fractional precision for warping parameters has been neglected since the 3D warping process is usually accomplished with software by far. However, it becomes more serious in hardware respect while the redundant fraction accuracy not only gains the information needed to be stored, but also worsens both area and timing for arithmetic operators. In order to optimize the trade-off between the accuracy and hardware cost, a precision fitting methodology is proposed to analysis the required precision for all parameters.

In the HT scheme, matrices **H** are only parameters needed to put emphasis on. Although 8 non-constant entries play all different roles during HT computation, some similarities and relationships still exist when we take mathematical model (4) into account. First of all, equations for evaluating x_2 and y_2 are symmetric, that means correspondence parameter pairs possess the same characteristic. Secondly, the requirement of accuracy for h_{XX} and h_{XY} is stricter than h_{XI} cause that the truncation error on h_{XX} and h_{XY} will be magnified by x_1 and y_1. Since the values of x_1 and y_1 can be as large as 2^{10} according to test sequences, the required number of binary fraction bit of h_{XX} and h_{XY} has to be at least 10 more than h_{XI}. After then, accuracy of h_{IX} and h_{IY} may also differ from others due to performing as factors of denominator in mathematical model. To summarize all, it is reasonable to partition 8 matrix entries into 3 groups. Group A is h_{XX}, h_{XY}, h_{YX}, and h_{YY}. Group B is h_{XI} and h_{YI} while h_{IX} and h_{IY} are included in group C.

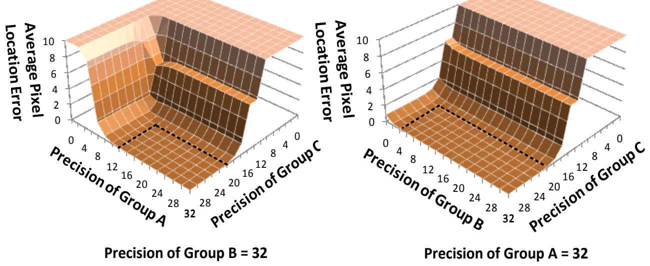

Figure 5. Precision analysis for group A,B and C

Parameters in the same group are assigned for the same bit number for binary fractional part. In Fig. 5, the average warping errors with 100 frames of *"ballet 1-2"* are calculated for both charts. Notice that the error values over 10 are not showed to make the curve near zero-plane more observable. Both quadric surfaces have the tendency to become flat to zero after the convergent boundaries marked by thick dash lines in Fig. 5. That is, optimal precision bit numbers of group A, B and C can be acquired near convergent boundaries, which can save the most area while keeping the computation accuracy.

IV. Experimental Analysis

To verify the proposed method and estimate performance of our hardware design, we implement multipliers and dividers using *DW02_mult* and *DW_div_pipe* modules from DesignWare building block IP of Synopsys Inc. We synthesize the hardware in UMC 90nm technology setting all designs to 200MHz. Figure 6(a) reveals the performance comparison among LIA models. In this figure, we can see that LIA-2 reaches the minimal area cost even compared with LIA-1 and the average location error for test sequences is small enough to be omitted. Similar results are shown in Fig. 6(b). In the analysis, near convergent numbers of bit for three groups there is also an optimal sample, where the most suitable numbers of fraction bit of group A, B, and C are 15, 5, and 24, respectively. The curve of gate count in Fig. 6(b) depicts that the area grows rapidly right after the optimal sample point and curves of error are about to converge. Based on above statistical results, we determined the best solutions of proposed architecture: LIA-2 model for H. matrix rendering stage with fitted fraction bit-widths for all entries as Table III.

The hardware performance of our proposed scheme is shown in Fig. 7. Considerable area is saved for both two architecture stages by LIA and precision fitting analysis. In H. matrix rendering stage, compared with implanting H. matrix LUT SRAM to fetch correspondence matrix parameters, 95.9% of area can be preserved with LIA-2 approximation model. As to vector transform stage, we preserve 69.5% of area by precision fitted HT, rather than the conventional matrix-based 3D pixel projection method with 32-bit fraction bit-width considering. We also compare the computational quality. Table IV displays PSNR results for both synthesized views obtained by VSRS and our work. The overhead is only 0.0059 dB in average and the results demonstrate that our proposed work makes no difference for synthesis quality.

TABLE III. Proposed Fraction Bitwidths for All Parameters

Parameter	h_{XX}	h_{XY}	h_{XI}	h_{YX}	h_{YY}	h_{YI}	h_{IX}	h_{IY}
# of fraction bit	15	15	5	15	15	5	24	24

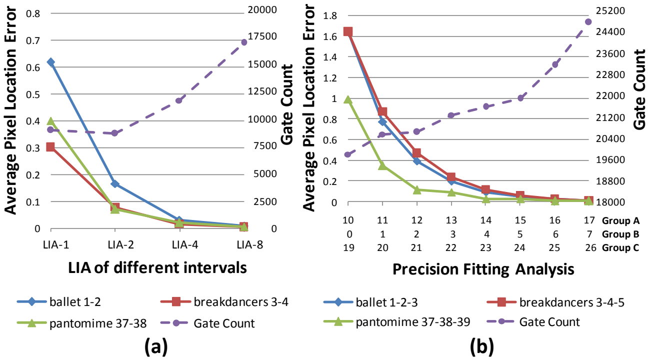

Figure 6. Error and area cost analysis for LIA models and precision fitting

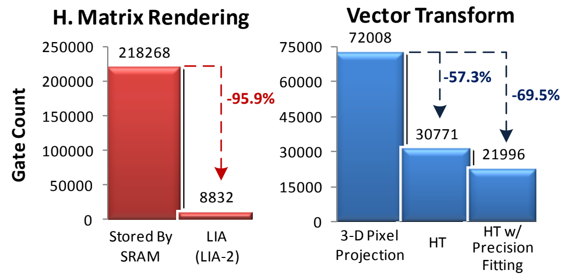

Figure 7. Comparison and improvement of proposed hardware design

TABLE IV. PSNR Performance

Sequence	Synthesized Pair	PSNR		
	(OV1-SV-OV2)	VSRS	Our Work	Difference
ballet	1 - 2 - 3	31.0209	31.0246	+ 0.0037
	3 - 4 - 5	33.8459	33.8299	- 0.0160
breakdancers	1 - 2 - 3	35.0433	35.0046	- 0.0387
	3 - 4 - 5	34.5078	34.5312	+ 0.0234
pantomime	37 - 38 - 39	38.9396	38.9241	- 0.0155
champagne_tower	39 - 40 - 41	39.2844	39.2820	- 0.0024
akko & kayo	47 - 48 - 49	29.5919	29.5786	- 0.0133
rena	44 - 45 - 46	31.1413	31.1532	+ 0.0119
Average				**- 0.0059**

V. Conclusion

This paper has presented a new 3D warping engine model that achieves large hardware area preserving while keeping the same quality as the original algorithm. We show that the rationality and low-cost characteristic for LIA are alluring and suitable for hardware design. In addition, the redundant information for fraction bit of parameters is further reduced by the precision fitting scheme. By doing so, our architecture obtains great improvement where 95.9% and 69.5% of area are saved for H. matrix rendering and vector transform stage, with the negligible 0.0059 dB overhead of PSNR.

References

[1] A. Smolic and P. Kauff, "Interactive 3-D video representation and coding technologies," Proceedings of the IEEE, vol. 93, no. 1, pp 33-36, January, 2005.

[2] M. Tanimoto, "Free viewpoint television - FTV," Proceedings of Picture Coding Symposium, 2004.

[3] MPEG-FTV Group, "View synthesis reference software (VSRS) 3.0," ISO/IEC JTC1/SC29/WG11, May, 2009.

[4] Christoph Fehn, "Depth-image-based rendering (DIBR), compression and transmission for a new approach on 3D-TV," Proceedings of SPIE Stereoscopic Displays and Virtual Reality Systems XI, vol. 5291, pp. 93-104, January, 2004.

[5] Dong Tian and et al., "Improvements on view synthesis and LDV extraction based on disparity (ViSBD 2.0)," ISO/IEC JTC1/SC29/WG11 MPEG2008/M15883, October, 2008.

[6] P. K. Tsung, P. C. Lin and L. G. Chen, "Single iteration view interpolation for multiview video applications," in Proceedings of 3DTV conference, May, 2009.

[7] Richard Hartley and Andrew Zisserman, "Multiple View Geometry in computer vision," Cambridge University Press, pp. 32–37, 2003.