iSense3D: A Real-Time Viewpoint-Aware 3D Video Synthesis System

Sung-Fang Tsai, Pei-Kuei Tsung, Kuan-Yu Chen, Chung-Te Li, *Member, IEEE*, and Liang-Gee Chen, *Fellow, IEEE, Graduate Institute of Electronics Engineering, National Taiwan University*

***Abstract*-- In this paper, a real-time 3D video synthesis system is proposed. The system achieves more real 3D effect by dynamically adapts the synthesized view to user's viewpoint. There are two major parts: 6D viewpoint parameter extraction, and real-time Free-Viewpoint View Synthesis(FVVS). 6D viewpoint parameter extraction is done by 3D object tracking over image and depth. Various techniques for FVVS are proposed and implemented on GPU to achieve real-time. The system is demonstrated on multi-core system with programmable GPU. Real-time performance up to 1280×720p with 30fps is achieved.**

I. INTRODUCTION

3D technology provides vivid watching experience and is becoming increasingly available in next-generation consumer display systems. Many interactive applications such as interactive gaming and virtual reality also start to support 3D display. In these applications, user's view position is likely to change. Synthesized views should also change to match depth perception from motion parallex and that from stereopsis for holographic 3D experience. Fixed-disparity rendering systems cannot provide such experience since disparity does not change while viewer's position and orientation is changing. To overcome this problem, the rendered view should be adapted to user's viewpoint by Free-Viewpoint View Synthesis(FVVS)[1]. We propose iSense3D, a real-time viewpoint-aware 3D video synthesis system with free viewpoint capability. Viewpoint invariant interactive 3D experience is achieved in this system. Viewer tracking finds the 6D viewpoint parameters for both eyes by depth-based 3D face tracking. Videos for both eyes are rendered at real-time on GPU by FVVS with 6D viewpoint parameters. Various optimization technique is used. The final system achieves real-time performance at 1280x720 resolution and 30fps on a 4-core notebook computer with programmable GPU.

II. PROPOSED SYSTEM

In order to provide viewpoint invariance in interactive 3D, we need to synthesize stereoscopic images corresponding to eyes' viewpoint. The proposed system is shown in Fig. 1. System consists of two major parts: 6D viewpoint parameter extraction, and 6D Free-Viewpoint View Synthesis(FVVS). The detail is shown below.

A. 6D Viewpoint Parameter Extraction

In 6D Viewpoint Parameter Extraction, both eyes' 3D translation and 3D rotation (6D) viewpoint parameters are extracted. We use 3D object tracking to extract both eyes' 3D positions in space. Object tracking is initialized by face recognition. Relative positions of both eyes are extracted. After that, particle filter tracker continuously tracks the face and find eyes' positions. Tracking observers in the particle filter use both color and depth image to locate 3D position of the face. The observation probability is generated by fusion the probability for both image and depth trackers.

To derive 6D viewpoint parameters, we assume that user is mostly looking at the screen. 3D rotation parameters is then derived by calculating line of sight when user is looking at screen center. Besides, view transformation parameters for eyes and recording are different. The parameters are adjusted in advance according to viewer's position and the actual dimension of the display region.

To reduce the complexity of FVVS warping, we derive warping matrixes with 6D viewpoint parameters in advance. Warping requires per-pixel Homographic Transform (HT). To reduce computation of HT matrices for every pixel, we save the computed per-depth HT matrices and model them with linear-interpolation (LI) scheme[6]. Furthermore, we project the matrixes to each epipolar line. As a result, original complex HT computations are simplified as computing bilinear equation.

B. Real-time 6D Free-Viewpoint Virtual View Synthesis

Since the user's viewpoint may change dynamically, we need to synthesize the image according to detected user viewpoint at real-time. FVVS[1] uses complicated warping technique. Two adjacent pixels in the reference view may be warped to arbitrary distant positions in the virtual view according to depth and epipolar geometry. Previous methods[4] can only support 1D movement. Since the pixels along a horizontal line in original view is not likely to lie along a single horizontal line in synthesized view, fragment memory access in this case will also reduce memory access efficiency.

To perform FVVS on real-time, we use several techniques to reduce overhead of the algorithm and parallelize it on GPU, as shown below:

1) Warping-based View Selection & Switching

In VSRS[2], multiple reference views are warped and merged to generate one virtual view. This way is impractical for real-time rendering, since the reference view number can be quite high. Some test sequences have 100 views[5]. To reduce the required views, we use a warping-based view selection. Firstly, we try to detect occlusion regions from depth discontinuous points. If two adjacent pixels in the reference view maps to a large region in virtual view, we mark it as an occlusion region. For the occlusion region, we will switch to the reference that mapped to a slight larger region for the same

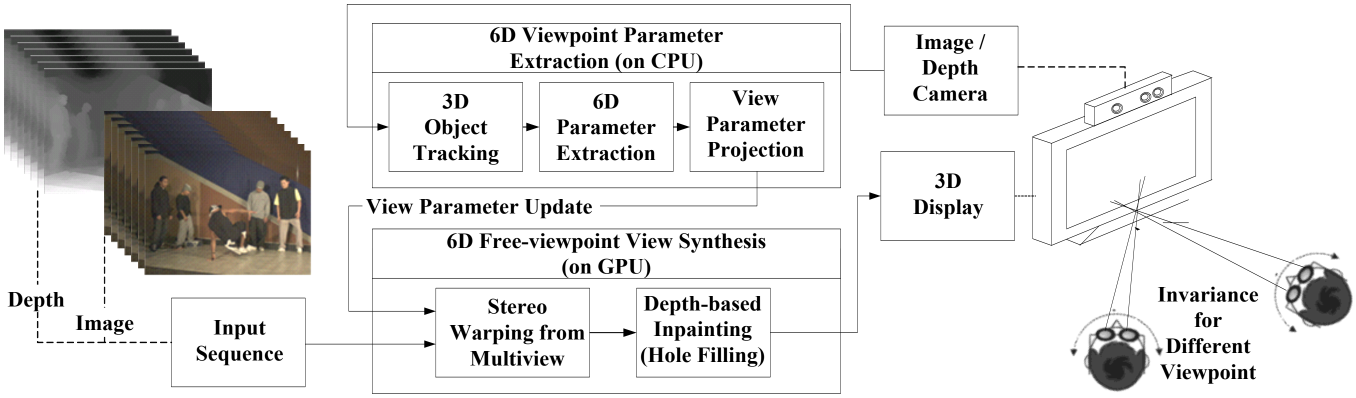


Fig. 1. iSense3D System Overview

position. This method would cover the occlusion.

2) Epipolar Stripe-based Rendering

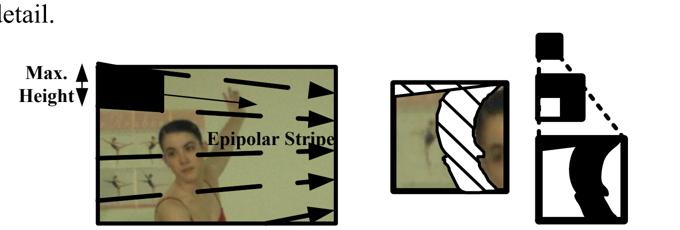
Warping algorithm in VSRS [2] requires a depth frame buffer. To reduce the required buffer, we use epipolar geometry. Depth buffer access is localized to a single line by rendering along the epipolar lines. However, the epipolar lines are neither horizontal nor parallel to each other. That results in fragment memory transactions and makes line buffering difficult. To make it performable, we proposed a epipolar stripe-based rendering. The rendering scheme is shown in Fig. 2(a). Workload is divided into stripes with the limited maximum height. The required buffer is allocated with maximum height. Data input and output is performed block by block and buffered on-chip. Since the data access is block-based, the fragment access problem is reduced.

3) Parallel Depth Test Algorithm with Weak Execution Ordering

View warping conflicts may occur if multiple pixels with different depth warped to the same position. Our previous work[4] uses parallel algorithm to avoid conflict. Since the conflict is not frequently happened, we propose a faster depth test algorithm based on weak execution ordering. Simple synchronous write-and-check is done iteratively until the depth is equal or larger than current depth. Actual color filling is done afterwards.

4) Depth-based Inpainting with Hierarchical Hole Map

After the warping, the unfilled occlusion regions are fixed by depth-based inpainting. To reduce redundant scanning of the whole frame, a hierarchical hole map is used as shown in Fig. 2(b). Only the blocks marked as 'black' will be scanned in detail.



(a) Epipolar Stripe-based Scheduling

(b) Image with holes and the corresponding Hierarchrical Hole Map

Fig.2. Schemes for Epipolar Stripe-Based Rendering

III. IMPLEMENTATION RESULTS

The system is implemented on a 3D Notebook with a 4-Core CPU and a programmable GPU. Color and image sensors are connected to the notebook to track user's 6D viewpoint. Traditional fixed disparity system will show strange motion parallex when moving. By using 6D viewpoint parameter extraction, the system will adapt the synthesized images to user's position, showing better 3D effects.

To evaluate the system performance, we compare our system with ViSBD[3] and a software implementation based on VSRS[2]. The performance comparison is shown in Table. I. With proposed optimization techniques, the system can achieve up to 30fps at 1280×720p. Real-time performance is important since it guarantees fast response for user's movement and provide holographic 3D effects during moving.

TABLE I
PERFORMANCE COMPARISON

Algorithm	Average Performance
VSRS-based implementation	407160 ms / frame
ViSBD(*)	2953 ms / frame
Proposed Algorithm	31 ms / frame

The ViSBD only works on 1D Parallel Multi-view sequences.

IV. CONCLUSION

This work presents iSense3D, a real-time viewpoint-aware 3D video synthesis system. By using 6D viewpoint parameter extraction, we can find the user's viewpoint in 3D space and synthesize the best view for the user. With the proposed real-time optimization techniques, we demonstrate the usability of free-viewpoint technology on consumer devices, enabling more realistic 3D effects for many interactive 3D application.

REFERENCE

[1] MPEG-FTV Group, "Draft Report on Experimental Framework for 3D Video Coding," in ISO/IEC JTC1/SC29/WG11 MPEG2010/N11273, April. 2010.

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

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

[4] Sung-Fang Tsai, Chao-Chung Cheng, Chung-Te Li, Liang-Gee Chen, "A real-time 1080p 2D-to-3D video conversion system," IEEE Transactions on Consumer Electronics, vol.57, no.2, pp.915-922, May 2011.

[5] M. Tanimoto, T. Fujii and K. Suzuki, "Multi-view depth map of Rena and Akko & Kayo", ISO/IEC JTC1/SC29/WG11 M14888, Shenzhen, China, October 2007.

[6] Pei-Kuei Tsung, et.al, "A 216fps 4096×2160p 3DTV set-top box SoC for free-viewpoint 3DTV applications," 2011 IEEE International Solid-State Circuits Conference (ISSCC), pp.124-126, 20-24 Feb. 2011.