A REAL-TIME AUGMENTED VIEW SYNTHESIS SYSTEM FOR TRANSPARENT CAR PILLARS

Yu-Lin Chang, Yi-Min Tsai, and Liang-Gee Chen

DSP/IC Design Lab,
Graduate Institute of Electronics Engineering,
National Taiwan University, Taipei, Taiwan
Email: {ylchang,firstpeople,lgchen}@video.ee.ntu.edu.tw

ABSTRACT

In this paper, a real-time augmented view synthesis system is proposed. With real-time consideration and augmented reality property, the proposed system provides a novel application for making car pillars transparent to enlarge the eyesight of the drivers. Thanks to the proposed trinocular depth estimation, online depth generation becomes possible through trinocular fast dense disparity estimation. With the proposed texture mapping free viewpoint depth image based rendering on the GPU, the processing speed for view interpolation achieves real-time. The computational power doesn't cost much so that this real-time system is achievable on common computers. The experimental results show that the proposed system is able to project view synthesized video which is integrated with the background to make the user perceive seamless outside scene inside their car.

Index Terms— augmented reality, view synthesis, dense disparity estimation, depth estimation, transparent car

1. INTRODUCTION

As the population of the 3D display device is growing up, the demand for 3D video content is increasing. The MPEG MVC and MPEG FTV group put their focus on coding multiview video sequences and generating free viewpoint video. Both of them adopt "view synthesis" for the prediction of non-existing viewpoint or generating video from an arbitrary viewpoint. View synthesis is also used on many augmented reality system such as digital archives and augmented panorama. Most of the applications of view synthesis are off-line operation type. The content providers generate the 3D information of the objects which are going to be shown to viewers. The object information is reduced to achieve rendering in real-time. If the whole view synthesis process achieves real-time operation, the applications can be much more broadened.

One of the applications for real-time view synthesis comes from the see-through pillar of NISSAN Pivo Concept. The concept car NISSAN Pivo equipped with a see-through pillar which put a LCD on it's A pillar and a wide-angle camera outside the pillar. However, the view angle of every driver differs; it is hard for the see-through pillar to deliver the outside video from a correct viewpoint. Another way to achieve the see-through function is to use optical camouflage in the transparent cockpit[1]. The idea was to create a vehicle where the interior seemed transparent. Thanks to a special viewing devices, retro-reflective projection technology, the view changed as you changed the position of your head and angle where you looked.

The application of the transparent car pillars suffers from two problems: the viewpoint of the driver and the special viewing device. A real-time augmented view synthesis system is proposed in this paper to solve these two problems in a feasible way. As the real-time augmented view synthesis system is popularized, more applications can be put into practice to make life convenient.

2. PRIOR ARTS

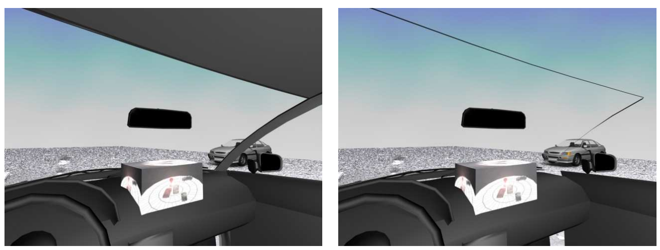
View synthesis, as known as view interpolation, consists of two procedures – finding pixel correspondence and interpolating correspondence [2]. Ohm proposed a disparity controlled projection algorithm for incomplete 3D representation of video objects[3]. A multi-view image compression algorithm along with a view interpolation method is proposed by Chang[4]. Fusiello further refines the transform between uncalibrated cameras for view synthesis[5]. Zitnick et al provide a framework focusing on video view interpolation which achieves real-time view interpolation for bullet-time effect on computers[6]. However, the applications of the previous designs only localize in the multiview video and film-making areas.

Most of the view synthesis algorithms need the disparity estimation for finding pixel correspondence. The disparity estimation is discussed more in stereo matching and multi-view video coding areas. It is divided into two classes: one is block-based disparity estimation, the other is dense disparity estimation. The block-based disparity estimation is usually adopted in the multi-view video coding system. The dense disparity estimation is also called per-pixel disparity estimation, since its disparity vector density is one disparity vector per pixel. It is more suitable for generating a depth map from per pixel information.

Grammalidis et al took multiocular system into consideration[7] and proposed an area-based disparity estimation method for the coding of occlusion and disparity information. They concentrated on the image data reproduction through disparity and occlusion information. Strecha presented a PDE-based Multi-view depth and disparity estimation method[8]. But the PDE-based method does not produce smooth results while processing natural images. Huang et al proposed a three-view dense disparity estimation method with Gabor filter[9]. The disparity estimation method with Gabor filter corrects most of the occlusion but costs too much computational power.

For interpolating correspondence, depth image based rendering(DIBR) is used in our design. It is proposed along with the advanced three dimensional television system (3D TV System)[10]. Traditional DIBR methods are all image-shifting algorithms. Chen et al proposed a DIBR hardware architecture design which is by far the first image-shifting DIBR hardware accelerator in the

Fig. 1. The Proposed View Synthesis System block diagram for transparent car pillar



(a) (b)

Fig. 2. Simulated Transparent Car Pillar Scenario (a) Original look out scene with A pillar (b) Look out scene with transparent A pillar

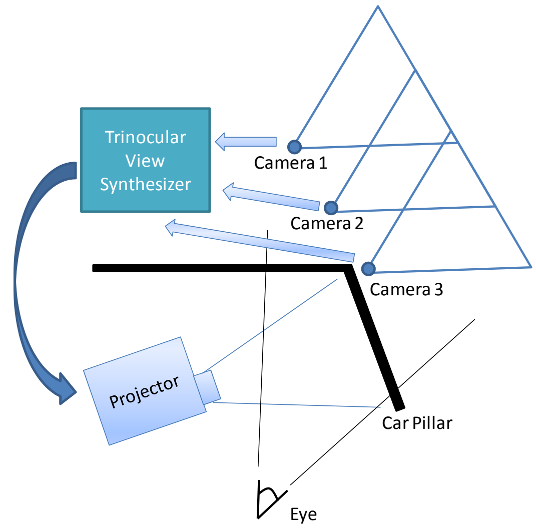


Fig. 3. The function diagram of the transparent car pillar

literature[11]. However, the image-shifting DIBR algorithms undergo the hole filling problem.

In this paper, a real-time augmented view synthesis system is proposed to realize the see-through car pillar function without arbitrary view angle from driver's position. The proposed view synthesis system is described in section 3. The computational costs analysis and augmented experimental results are shown in section 4. At last, the conclusion remarks the proposed system.

3. PROPOSED AUGMENTED VIEW SYNTHESIS SYSTEM

An augmented view synthesis system is proposed in this section. For the view synthesis, the depth information from the outside world must be gotten. As the system block diagram shown in Fig.1, we used trinocular camera configuration for the input, trinocular depth estimation for depth map generation, view transform for transforming the original view video to an arbitrary view, and texture-mapping free viewpoint depth image based rendering for texture synthesis of a new viewpoint.

Fig.2 presents the transparent pillar scenario for a car. In Fig.2 (a), the original look out scene in a car shows that the A pillar might occlude the driver's line of sight. What we proposed is to utilize the view synthesis technique to make the A pillar transparent, as shown in Fig.2 (b).

View synthesis needs original views for new view prediction. The more the original views are, the better the prediction does. The view synthesis system for transparent car pillars functions as Fig.3. Trinocular cameras(Camera 1, 2, 3) are used for trinocular depth estimation. The trinocular view synthesizer consists of view transform and depth image based rendering which provide the generation of the video from the driver's viewpoint. After the video of the new viewpoint is generated, the projector projects the video onto the car pillar. The projected video should be integrated with the objects outside the pillar, then the drivers will have the feeling of seeing through the car pillar. If the generated new viewpoint is not the same with driver's viewpoint, adjustment can be done through eye gaze tracking or user-defined eye direction, just like the adjustment of the traditional rear mirror in a car. Here we realize the "adjustment" through view transform and texture mapping free viewpoint depth image based rendering.

3.1. Trinocular Depth Estimation

The trinocular depth estimation part is the most important part for the view synthesis system. The flow chart in Fig.4 describes that the trinocular depth estimation is constituted of the intrinsic camera parameter calibration, trinocular dense disparity estimation, extrinsic camera parameter estimation, depth estimation, depth fusion, and finally the output of the depth map. The intrinsic camera parameter calibration and extrinsic camera parameter estimation are carried out by the setup of the trinocular cameras. The trinocular dense disparity estimation consumes a lot of computational power. We adopted a fast dense disparity estimation algorithm from Tsai's work[12]. The symmetric trinocular property is also used here for the correction of the calculated disparity map. The Sum of Absolute Difference (SAD) is used for the matching criterion of this dense disparity estimation method. It is expressed as the following equation:

$$SAD_m = \sum_{i,j \in MB}(|f(i-MV_{kx}, j-MV_{ky}, n-1) \\ -f(i,j,n)| + |f(i+MV_{kx}, j+MV_{ky}, n+1) - f(i,j,n)|) \quad (1)$$

The function f(i, j, n) denotes the intensity of the input image. i and j are the horizontal and vertical axes, and n describes the camera number. The first SAD calculation describes the relationship between the left and middle camera, and the second SAD calculation describes that between the middle and right camera. This symmetric trinocular property helps to find more accurate disparity vectors. If it is combined with fast search mechanism, real-time processing is able to be achieved. Here we use the 1D fast search algorithm to speed up the matching.

The disparity map is converted to depth map through triangulation from the disparity vectors and extrinsic camera parameters. The dense disparity estimation and depth map conversion is combined as the depth estimation step mentioned in Fig.4. This step provides the view synthesis system with the depth map it needs.

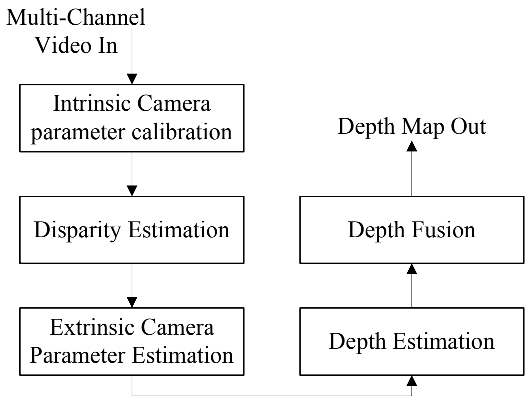


Fig. 4. The Flow Chart of the Proposed Trinocular Depth Estimation

3.2. View Transform

After obtaining the depth map from the trinocular depth estimation, view transform combined with depth information is adopted to construct the relationship between camera position and viewer's eye. In Fig.5, we look the driver's eye as a second camera with new position and different parameters(focal length, image plane size, etc). In order to further transform the real object to the image plane of the second camera, the translational vector from the camera to the eye helps to form the translation matrix $\overrightarrow{T} = \overrightarrow{C_1} - \overrightarrow{E}$. The position mapping of the objects between different image planes is described in eqn. 2. P_1 and M_1 denote projected object position and projection matrix in camera C_1. P_E and M_E denote projected object position and projection matrix in the driver's eye. The mapping from P_1 to P_E can be obtained from the last equation in eqn. 2.

$$
\begin{aligned}
&P_1 = M_1(\overrightarrow{P} - \overrightarrow{C_1}) \\
&P_E = M_E(\overrightarrow{P} - \overrightarrow{E}) \\
&M_1^{-1}P_1 + \overrightarrow{C_1} = \overrightarrow{P} \\
&P_E = M_E(M_1^{-1}P_1 + \overrightarrow{C_1} - \overrightarrow{E}) = M_E(M_1^{-1}P_1 + \overrightarrow{T})
\end{aligned}
\quad (2)
$$

Here we left the adjustment of the eye position to the driver. The driver controls the eye location and angle by himself/herself, just like controlling the rear mirror of a car. Since interactive controlling is needed, real-time free viewpoint depth image-based rendering is required.

3.3. Texture-mapping Free Viewpoint Depth Image Based Rendering

The proposed free viewpoint depth image based rendering(FVDIBR) utilizes the GPU in computers for real-time consideration. The proposed texture mapping FVDIBR is described in Fig.6. The input data are the middle view picture and the depth map. First we have to convert the depth map to the vertices with Z value, as the grid picture shown in Fig.6. Each of the image point in the original picture is going to be assigned a Z value and a color value of the original image point . With so many vertices, triangles and polygons are drawn. The neighboring four vertices form two triangles. After the polygonization, the rendering process is delivered to the GPU. The GPU looks the whole picture as an image plane. So the holes induced by the depth change are going to be compensated by its shading methods.

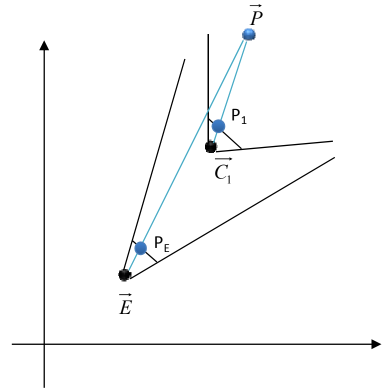


Fig. 5. The relationship between camera position and viewer's eye

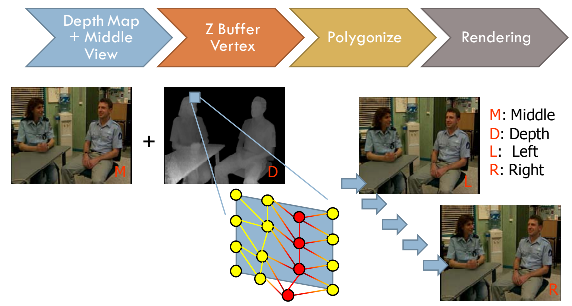


Fig. 6. Texture mapping Free Viewpoint Depth Image Based Rendering Algorithm

After FVDIBR, the video from the driver's viewpoint is rendered and sent to the projector. The projector then projects the video onto the car pillar to augment with the outside background. The augmented view synthesis is done while the driver adjusts the view transform parameters to his/her suitable values.

4. EXPERIMENTAL RESULTS

4.1. Computational Costs

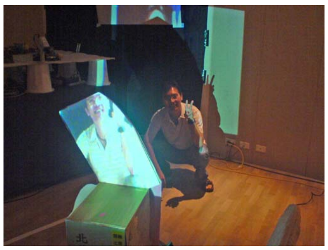
There are two main computational intensive parts in our design: the dense disparity estimation and the free viewpoint depth image based rendering. For the dense disparity estimation we adopted from our previous car radar design[12], the computational speed achieves 30FPS on a Pentium-4 2.4GHz computer since the 1D fast search really reduces the search candidates and find correct disparity vectors. For free viewpoint depth image based rendering, the computational cost and image quality is shown in Table 1. The instruction count is decreased because the bottleneck of graphics - the shading - is carried out by Gouraud shading, which is a linear interpolation algorithm. Moreover, most of the shading jobs are processed by the GPU. For a GPU with more than 24.8MTriangles per sec capability, the FVDIBR is able to run in real-time.

4.2. Augmented Subjective View Results

An augmented view synthesis experiment has been done with the proposed view synthesis system. A white board is setup as a virtual car pillar to show the occlusion in a car. In Fig.7 (a), a man

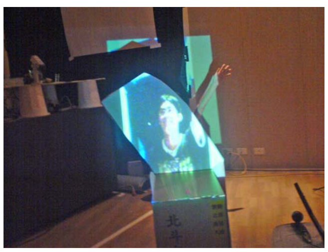
Table 1. Comparison between the Depth Image Based Rendering Methods

Functions	Traditional DIBR	Chen's Work[11]	This Work
GIPS	190.55	7.08	44
Speed Up	1	26.9	4.33
Average PSNR	36.3	42.1	36
Support Direction	Horizontal	Horizontal	Full Angle



(a)

(b)



(c)

(c)

Fig. 7. Real-Time experimental results which make a big white pillar transparent (a)(b) A man hiding behind the pillar (c)(d) A man standing behind the pillar

is hiding behind the virtual pillar. From incorrect viewpoint, the view synthesis system cannot augment the synthesized video with the background. Fig.7 (b)(c)(d) shows the augmented view synthesized results with the camera's viewpoint. The video projected on the pillar is integrated with the background seamlessly. The proposed view synthesis system provides the car manufacturers a new thought about the tradeoff between the driver's viewing range and car rigidity.

5. CONCLUSION

A real-time augmented view synthesis system is proposed in this paper for using on transparent car pillars. Traditional view synthesis algorithms mainly focus on offline depth generation and online view interpolation. This restricts the application within multi-view video coding and film-making. The proposed trinocular depth estimation provides the ability for online depth generation through trinocular fast dense disparity estimation. With the proposed texture mapping free viewpoint depth image based rendering on GPUs, the processing speed for view interpolation achieves real-time. By giving users the freedom to control the translational and rotational vectors in the view transform, the users may adjust the viewpoint for the view synthesis system on their transparent car pillars interactively. It is as easy as adjusting the rear mirror direction in a car. The experimental results show that the proposed system really generate a view synthesized video which is integrated with the background to make the user perceive seamless outside scene inside their car. With real-time consideration, view synthesis is able to be adopted into more applications and make augmented reality more practicable.

6. REFERENCES

[1] Takumi Yoshida et al, "Transparent cockpit," *International Conference on Computer Graphics and Interactive Techniques, ACM SIGGRAPH 2007 emerging technologies*, 2007.

[2] Shenchang Eric Chen and Lance Williams, "View interpolation for image synthesis," *Proceedings of the 20th annual conference on Computer graphics and interactive techniques*, pp. 279–288, 1993.

[3] Jens-Rainer Ohm and Karsten Muller, "Incomplete 3-d multiview representation of video objects," *IEEE Transactions on Circuits and Systems For Video Technology*, vol. 9, no. 2, Mar. 1999.

[4] Gan-Chieh Chang and Wen-Nung Lie, "Multi-view image compression and intermediate view synthesis for stereoscopic applications," *Proceedings of the IEEE International Symposium on Circuits and Systems*, May 2000.

[5] Andrea Fusiello, "Specifying virtual cameras in uncalibrated view synthesis," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 17, no. 5, May 2007.

[6] C. Lawrence Zitnick, Sing Bing Kang, Matthew Uyttendaele, Simon Winder, and Richard Szeliski, "High-quality video view interpolation using a layered representation," *ACM Transactions on Graphics (TOG)*, vol. 23, 2004.

[7] Nikos Grammalidis and Michael G. Strintzis, "Disparity and occlusion estimation in multiocular systems and their coding for the communication of multiview image sequences," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 8, no. 3, pp. 328–344, Jun. 1998.

[8] Christoph Strecha and Luc Van Gool, "Pde-based multi-view depth estimation," *Proceedings of the First International Symposium on 3D Data Processing Visualization and Transmission*, 2002.

[9] Xiaodong Huang and Eric Dubois, "Two-dimensional regularized disparity estimation based on the gabor transform," in *Visual Communications and Image Processing 2006*, John G. Apostolopoulos and Amir Said, Eds., 2006, vol. 6077 of *Proc. SPIE*.

[10] Andr Redert, Marc Op de Beeck, Christoph Fehn, Wijnand IJsselsteijn, Marc Pollefeys, Luc Van Gool, Eyal Ofek, Ian Sexton, and Philip Surman, "Attest: Advanced three-dimensional television system technologies," *Proceedings of the First International Symposium on 3D Data Processing Visualization and Transmission (3DPVT.02)*, 2002.

[11] Wan-Yu Chen, Yu-Lin Chang, and Liang-Gee Chen, "Real-time depth image based rendering hardware accelerator for advanced three dimensional television system," *Proceedings of the 2006 IEEE International Conference on Multimedia and Expo (ICME 2006)*, 2006.

[12] Yi-Min Tsai, Yu-Lin Chang, and Liang-Gee Chen, "Symmetric trinocular dense disparity estimation for car surrounding camera array," *SPIE 19th Annual Symposium on Electronics Imaging, Visual Communications and Image Processing (VCIP)*, 2007.