IJISET - International Journal of Innovative Science, Engineering & Technology, Vol. 1 Issue 10, December 2014.
www.ijiset.com
ISSN 2348 – 7968

A Survey: Fast & Approximate Algorithm of Depth Image Based Rendering Process for Application of 2D to 3D Conversion

Neetesh Nema[1], Bragesh Patel[2]

[1]MTech Schlolar, Department of CSE, SRIT, Jabalpur (M.P.), RGPV, India
[2]H.O.D, Department of CSE, SRIT, Jabalpur (M.P.), India
Email: nitesh.beats@gmail.com[1]

ABSTRACT - Three- dimension (3D) technology increases the visual quality as compared to two-dimensional (2D) technology. In present era every multimedia device needs 3D technology. So for generation of 3D content there is need of Depth image based rendering (DIBR) process which will generate left and right image through depth image and original image. Basically DIBR is following the concept of actual 3D recording camera setup. Through original camera setup there is virtual camera formula is generated which will create left and right image. Using both image 3D content is created. As we already know for any image processing application time complexity is main issue. So in this work I will propose a fast and approximate DIBR algorithm which will reduce the time complexity issue. For image quality measurement there is some scientific parameter will use which will check the quality of generated left and right image through proposed DIBR algorithm. Those parameters are like PSNR, SSIM, RFSIM and FSIM. Algorithm will implement on Matlab.

KEYWORDS — Depth image based rendering, Adaptive smoothing, Terrestrial DMB, 3D service, Image Processing and Analysis.

INTRODUCTION:

The 3D video signal processing has received considerable attention in visual processing. Given advances in 3D display technology, humans aspire to experience more realistic and unique 3D effects. Depth-Image-Based-Rendering (DIBR) is a key technology in advanced three dimensional television system (ATTEST 3D TV System) [1][2]. Depth image based rendering contains three steps [3].Pre-processing of depth map is applied for reducing the sharp horizontal transitions on depth map. Second, 3D image warping renders left and right images based on the pre-processed depth map and intermediate color image. The traditional 3D game has some information of the third dimension, but it is still two-dimensional when it is displayed on screen, while the stereoscopic game has stereoscopic view objects in the images stand out of the screen. the depth preprocessing method is applied before transmission. Once both the original image and the accompanying depth image are received, the auto stereoscopic image is produced by applying 3D Wrapping, hole filling, and merging simultaneously.

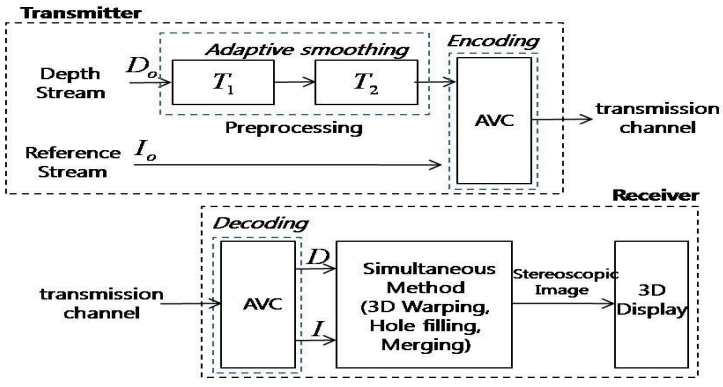

Fig-1 Block diagram of the DIBR technique

The main function for 2D-3D image conversion consists of two parts: (1) Depth map information generation and (2) 3D Rendering based on the depth information.

ISSN 2348 – 7968

Depth-Image Based Rendering (DIBR) :

Depth-image-based rendering (DIBR) is the process of synthesizing “virtual” views of a scene from still or moving images and associated per-pixel depth information. Conceptually, this novel view generation can be understood as the following two-step process: At first, the original image points are reprojected into the 3D world, utilizing the respective depth data. Thereafter, these 3D space points are projected into the image plane of a virtual camera, which is located at the required viewing position[4][5].

(SIC) Stereoscopic Image Creation :

On a stereoscopic- or auto stereoscopic 3D-TV display, two slightly different perspective views of a 3D scene are reproduced simultaneously on a joint image plane. The horizontal differences between these left- and right-eye views, the so-called *screen parallax* values, are interpreted by the human brain and the two images are fused into a single, three-dimensional percept.

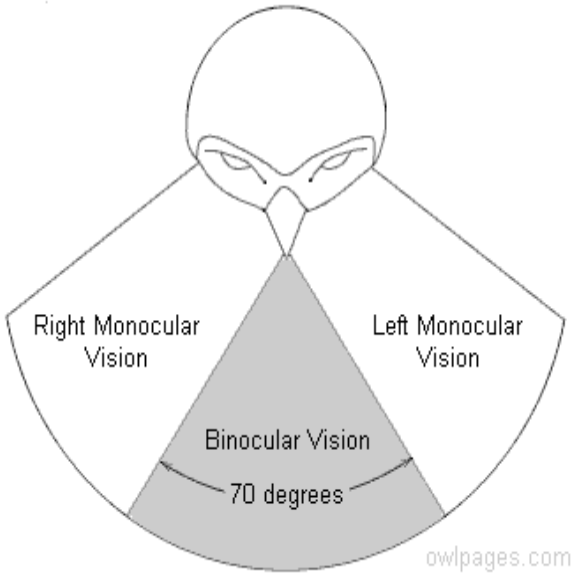

Fig - 2 Binocular depth reproduction on a stereoscopic 3D-TV display. Two different perspective views, i.e. one for the left eye and one for the right eye, are reproduced (quasi)simultaneously on a joint image plane.

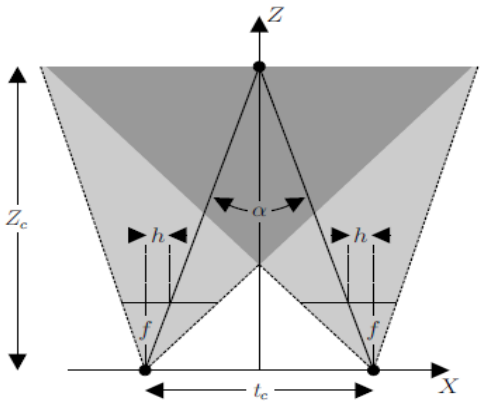

Fig- 3. Shift-sensor stereo camera setup. In a shift sensor stereo camera setup, the convergence distance is established by a shiftof the camera’s CCD sensors

DIBR ALGORTHIM :

DIBR is the process of generating virtual views from monoscopic video and the associated depth information[5]. Consider an arbitrary 3D space point M, the two perspective projection equations in two views result to:

$$\tilde{m} = AP_n\tilde{M}, \tilde{m}^* = A^*P_nD\tilde{M}.$$

Where *m* and *m∗* represent two 2D image points in the left and right view images, respectively. The matrix D contains the rotation matrix and the translation matrix *t* that transforms the 3D point from the world coordinate system into the camera coordinate system. The matrices A and *A∗* specify the intrinsic parameters of the cameras. The identity matrix P_n designates the normalized perspective projection matrix. Fig shows a virtual camera setup for rendering of virtual views. The parameters *f* and *tc* represent the focal length and the baseline distance between two virtual camera *Cl* and *Cr*, respectively. *Zc* means the depth value of the ZPS (Zero Parallax Setting). The parallel camera setup rather than the convergent camera setup is used not to generate vertical disparities and it’s much easier to implement with DIBR technically. Since the position of the 3D space point M depends on its depth value, the 3D image warping equation can be obtained:

ISSN 2348 – 7968

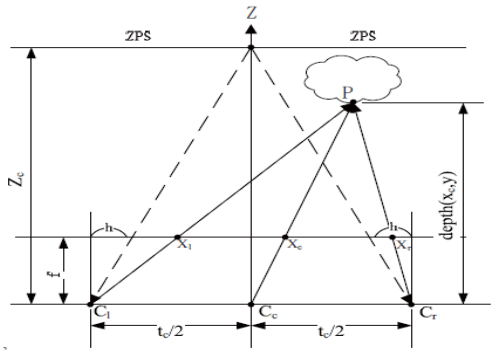

Fig- 4. Camera configuration used for generation of virtual stereoscopic

$$\tilde{m}^* = \tilde{m} + \frac{A^* t}{depth} + \begin{bmatrix} h \\ 0 \\ 0 \end{bmatrix}, with \quad t = \begin{bmatrix} t_x \\ 0 \\ 0 \end{bmatrix}.$$

So, the pixel position (*x, y*) of each warped image point can simply be calculated as:

$$x^* = x + \frac{\alpha_u t_x}{depth} + h, with \quad y^* = y$$

Where *αu* is a parameter related to left-right-eye distance and eye-screen distance. The pixel position (*xc, y*), (*xl, y*) and (*xr, y*) of the reference view and two virtual views corresponding to the point *P* with *depth* have the following relationship:

$$x_l = x_c + \frac{t_c f}{2depth(x_c, y)} + h,$$
$$x_r = x_c - \frac{t_c f}{2depth(x_c, y)} - h$$

The offset h between reference view and target view can be computed by:

$$h = -\frac{t_c f}{2Z_c}$$

ALGORITHM EXPANSION:

The whole left-and-right view images would be translated horizontally by *h* pixels to replace the addition and subtraction of *h* in (6). And as the division operation is time and area consuming by hardware implementation, the DIBR algorithm is optimized to eliminate the division operation. In (4), 1*/depth* ranges between 0 and 1. Since the value of 1*/depth* only represents the relative distance of the pixels but not real distance, and that people are only sensitive about the objects which have smaller depth value, (256−*depth*)*/*256 could be used to replace 1*/depth*, and then we get (4)..

$$x_l = x_c + k\frac{t_c f}{2}\frac{256 - depth(x_c, y)}{256},$$
$$x_r = x_c - k\frac{t_c f}{2}\frac{256 - depth(x_c, y)}{256}$$

In the implementation, for each depth value between 0 and 255, the average deviation between (4) (the practical value) and (6) (the theoretical value) is about 3.6%. Then, the division operation in (4) can be simply implemented by shift operation in hardware. In the implementation, the white color in the depth map (i.e. 255) represents the nearest plane while the black color (i.e. 0) represents the farthest. It is opposite to the real depth given in (2). We use *D* to represent the value that obtained from the depth map and get:

$$D = 256 - depth$$

L_offset and R_offset are used to represent the value of offset in the left-and-right view images, respectively. So while implementing the DIBR algorithm in hardware, use to compute left-view offset l offset and right-view offset r offset:

$$\begin{cases} l\_offset = D \cdot pos/256 \\ r\_offset = pos - l\_offset. \end{cases}$$

Where *pos* = *k tcf* 2 is a parameter related to the eye-screen distance and the screen size. *pos* is set to 1/32 of the width of the screen [4]. This will give a good visual experience and can be simply calculated by shifting operation. After the *l offset* and *r offset* are calculated, the left-view image and the right-view image can be obtained.

$$\begin{cases} lpic(x,y) = pic(x - l\_offset, y) & left - view \\ rpic(x,y) = pic(x - r\_offset, y) & right - view. \end{cases}$$

Where *lpic* and *rpic* represent the left-view and rightview images, respectively. *pic* is the original 2D image, and (*x, y*) means the pixel position in the images. Pre-Processing of depth image is usually a smoothing filter. Because depth image with horizontal sharp transition would result in big holes after warping, smoothing filter is applied to smooth sharp transition so as to reduce the number of big

ISSN 2348 – 7968

hole. However, if we blur the whole depth image, we will not only reduce big holes but also degrade the warped view.

3D IMAGE WRAPPING:

3D image warping maps intermediate view pixel by pixel to left or right view according to pixel depth value. In the other words, 3D image warping transforms pixel location according to depth value. The 3D image warping formula is as following:

$$x_l = x_c + (\frac{t_x}{2}\frac{f}{Z}),$$

$$x_r = x_c - (\frac{t_x}{2}\frac{f}{Z})$$

The x*l* is the horizontal coordinate of the left view, and *xr* is the horizontal coordinate of the right view. Besides, *xc* is the horizontal coordinate of the intermediate view. *Z* is depth value of current pixel, *f* is camera focal length and *t x* is eye distance. The formula shows that 3D warping maps pixel of intermediate view to left and right view in horizontal direction. 3D data service based on the proposed DIBR technique can be provided through both packet mode data path and stream mode data path of T-DMB with keeping backward compatibility with conventional 2D service [6].

(a)Intermediate View

(b)Depth Map

(c) Left View

(d) Right View

APPROXIMATION & CONCLUSION:

The approximation can increase performance and reduce the time complexity issues especially in image processing applications. These applications related to human senses, approximate arithmetic can be used to generate sufficient results rather than absolutely accurate results. Approximate design exploits trade-off accuracy in computation versus performance. However, required accuracy varies according to applications. In present era for multimedia application hardware complexity and memory conception are main issue. Human eye sense accuracy of 90-95% so why we need accurate algorithm for reduction of complexity issue we can apply approximation unit.

REFERENCES:

[1] A. Redert,M. Op de Beeck, C. Fehn,W. IJsselsteijn,M. Pollefeys, L. Van Gool, E. Ofek, I. Sexton, and P. Surman, “Attest vadvanced three-dimensional television system,” *Proc. Of 3DPVT*, pp. 313–319, 2002.

[2] Chung J. Kuo, Ching Liao, and Ching C. Jin, “Stereoscopicimage generation based on depth images,” *ICIP*, pp. 2993–2996, 2002.

[3] Wan-Yu Chen, Yu-Lin Chang, Shyh-Feng Lin, Li-Fu Ding,and Liang-Gee Chen, “Efficient depth image based rendering with edge dependent depth filter and interpolation,” *ICME*, 2005.

[4] W. R. Mark, *Post-Rendering 3D Image Warping:Visibility, Reconstruction and Performance forDepth-Image Warping*, PhD thesis, University ofNorth Carolina at Chapel Hill, 1999.

[5] L. McMillan, *An Image-Based Approach on Three-Dimensional Computer Graphics*, PhD thesis, Universityof North Carolina at Chapel Hill, 1997.

[6] *KwangHee Jung, Young Kyung Park, Joong Kyu Kim, Hyun Lee, Kugjin Yun, Namho Hur, and Jinwoong Kim,* Depth image based rendering for 3d data service over t-dmb, School of Information and Communication Engineering, Sungkyunkwan University,Suwon 440-746, Republic of Korea.

[7]http://cgit.nutn.edu.tw:8080/cgit/PaperDL/WSY_100717143701.PDF

[8]http://www.owlpages.com/articles.php?section=Owl+Physiology&title=Vision.