

A new Procedure for 2D to 3D conversion using Depth Array Conversion

¹B.Yugandhara Chary ,Assistant Professor, CSE(AIML), yugandhar.bandla@gmail.com
Swarna Bharathi institute of science and technology,
Khammam

²Dr.K.Spurthi, Associate Professor , CSE(AIML), Kolluspoorthy03@gmail.com
Swarna Bharathi institute of science and technology,
Khammam,

³P.Anusha, Assistant Professor, CSE, peddapakanusha@gmail.com
Swarna Bharathi institute of science and technology,
Khammam,

Abstract

The actual world's entertainment industry has a golden age that continues for decades. Television and projection technology have come a long way, allowing for the most lifelike visuals and cinematic experiences ever. Stereoscopic translation of 2D program content into 3D realistic is now being developed by a number of researchers and research forums. We suggested a novel technique to transform a 2D array of video sequences into stereoscopic 3D as 3D videoscopic displays are becoming more scarce. This method relies on depth array conversion of 2D video backgrounds and subtracting active objects from each scene at each frame to get depth information from the video. It's a real-time computation heavy process, but it was accomplished by Cryengine with AMD QUADCORE A6 CPU and Radeon graphics. The experimental findings demonstrate a 3D recognition rate of up to 78% for a single picture inside a 2D video series, with a display rate of 29 frames per second for that specific sequence. Included are experimental findings showing that the converted sequence may achieve a 3D display rate of up to 92% accuracy and 32 frames per second in FULL HD (1080 p). Subjects: 2D to 3D conversion, depth array conversion, stereoscopic, 2D array, 3D array, FULL HD, 1080p, Cryengine, quad core, AMD, A6 CPU.

Introduction

Multimedia, augmented and virtual reality, three-dimensional television, and three-dimensional motion pictures are just a few of the real-world uses for three-dimensional imagery. More and more people are interested in 3D these days due to the

realistic effects it produces [3]. As a result, 3D movies are hard to record since the devices needed to do so are cumbersome to transport and set up. Conversion technology is driving the evolution of recent 3D videos. The process of preparing a 3D instance is still in its early stages, but it involves converting 2D films into a 3D sequence [4][5]; the end user will get this 3D sequence as their final output. In this work, we will develop a novel method for transforming 2D arrays into 3D ones. converting arrays to depth arrays. The many 2D color models that are available with supporting formats are shown in Table 1. The many 2D color models that are available with supporting formats are shown in Table 1.

Colour model	2D		
	Accuracy	Jitter	Format
RGB	High	Yes	.avi,.vob,.avi
CMYK	High	Yes	.avi
HSV	High	Yes	.MPEG

Table1: Supportive 2D Colour model along with their supportive formats

Colour model	3D		
	Accuracy	Jitter	Format
RGB	High	Yes	.mkv,.avi
CMYK	High	No	.mkv
HSV	Low	Yes	.MPEG

Table2: Supportive 3D Colour model along with their supportive formats DIBR method This approach uses Luminance Intensity to encode both

the pixel intensity and its depth information from the outset. Luminance intensity is used to encode the depth information of every pixel in the supplementary photos; values that are closer to the camera are represented by brighter values, while values that are further away are represented by darker values. Figure 1 clearly shows the transformation from 2D to 3D rendering, as the equation represents the perspective of the left and right eyes on a virtual camera set at. To put it simply, 2D pictures are just 2D arrays that have been rearranged into rows and columns in accordance with the image's pixel grid. The extension is used for the different picture formats. For example, when applying transformations to pictures, JPEG files play a crucial role in lossless compression. Other file extensions include PNG, GIF, BMP, and TIFF. The use of 3D rendering allows for more accurate stereoscopic visuals, since these 2D arrays fail to provide realistic outperformance.

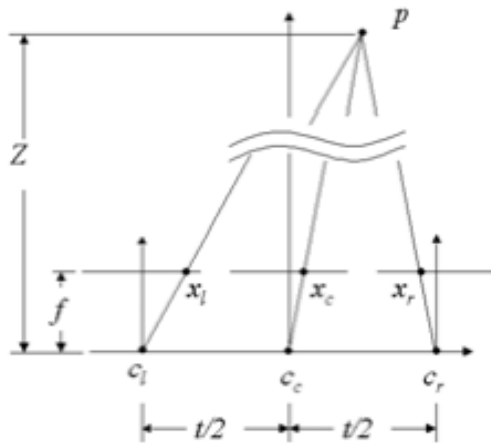


Fig1: Denotes the DIBR – virtual stereoscopic images (adopted from [2] [3])

$$x_{left} = x_c + \frac{t f}{2 Z}$$

$$x_{right} = x_c - \frac{t f}{2 Z}$$

The demonstration of stereoscopic vision with human visual - binocular processing of two variant different pictures is the essential premise concerning 2D to 3D conversion. Originally developed to produce varying depths of objects beyond the 2D plane, actual perception now focuses on the little subtraction of discrepancies between the left and right eye views. Therefore, horizontal pixel shifting to generate new pictures is the basic technique for

2D to 3D conversion[3]. The distance of the feature object in the stereoscopic image at that pixel is used to create the horizontal shift [3]. Table 3 shows the supported color model and formats for converted 3D sequences.

Colour model	3D Converted sequence		
	Accuracy	Jitter	Format
RGB	High	Yes	.avi, MPEG, MP
CMYK	Low	Yes	EGII, .vob,
HSV	Low	Yes	.mkv, .FLV, .mpg , .mp4, .wmv

Table3: Supportive Colour model for converted 3D sequence with their supportive formats

Video Classification

Prior to using any conversion methods, video categorization must be performed. This is essential for improving the accuracy rate and computational speed of the final product. At first, we estimate the depth cues based on the motion activities of all comparable frames. Following this, we divide each frame in half and then examine and categorize the content's complexity.

- Big gap between frames
- Extremely strong jitter and a big gap between frames
- A big gap between frames with little jitter
- A big gap between frames with a complex background
- A big gap between frames with a simple background
- A big gap between frames with a complex background and a lot of jitter
- A big gap between frames with a complex background and little jitter
- Minimum frame disparity
- Very little frame difference accompanied by high jitter
- Very little frame difference accompanied by low jitter
- Very little frame difference accompanied by high complexity of the background
- Very little frame difference accompanied by low complexity of the background
- Very little frame difference accompanied by high jitter and relatively complex of the background
- Very little frame difference accompanied by relatively simple of the background

The content complexity is solved by calculating texture pitch at each level. The difference between each picture is recorded at regular intervals and is characterized as

$$X^d = \frac{1}{M_H M_W M_S} \sum_x \sum_y \sum_t V(I(x, y, t) - I(x, y, t-1) - T_d)$$

Where V (*.*) denotes frame function and M_H & M_W denotes the height and width of the frames, T_d denotes the threshold rate of the frames.

Parameter Evaluation

A key concept in conversion methods is parameter evaluation. It is a problem that arises often in situations when there are a lot of errors and jitters. Cue, saturation, depth perspective, active object, texture pitch (gradient), and parallax assessment are the basic parameters. Approach to Work Converting Depth Arrays Discovering the most important spot for determining the depth of a two-dimensional picture is a huge scientific boon. The focal length of the camera and the viewpoint towards the focus are used to calculate the depth of an image. This means that the depth actually acquired from the focal length of the camera is dependent on the modeling perspective towards the active item. We apply the defocus operator once again to get the depth information and focal length/distance of the depth, and then we use inverse filtering to get the object's depth of focus. The high-frequency area will undergo non-zero wavelet transformation and analysis, while the low-frequency region will have a large number of parameters, and these parameters are calculated using local cue and depth perception parameters with multi-resolution wavelet analysis. You may use luminance intensity values to forecast local regions with plenty of textures on the background and sharp edges; unfortunately, this approach only works with the RGB color model; HSV and CMYK do not work with these kinds of sequences. To analyze the texture gradient and its intensity, which might result in a stereoscopic picture of certain active objects, texture pitch parameters are used. Here, we classify a single input picture and use that classification to drive how we analyze the depth information from a geometric perspective.

$$d^{background}(x, y, t) = 255 * (y - y_{vp}) / (N_H - y_{vp})$$

Procedure for applying Depth array conversion:

```
Pseudo 2dto3d1080p(unsigned char*
depth,2D,filter,DIBR,transform,fgd,bkgd,vie
w)
```

```
Input:2D,filter,DIBR,Depth,transform
```

```
Output:3D array,Frames,stereoscopic view
```

```
Class 3D
```

```
3D 3 dobj=new 3D(inputreader);
```

```
Dilate(3 d);
```

```
Erode(3 d);
```

```
Img.growth(3 d);
```

```
Img.Boundary(local,global,perception);
```

```
Imgout=Img.Transform(3 d,DWT);
```

```
Img.Thres(Imgout);
```

```
Frgd=thres(Imgout,bkgd);
```

```
Grad=Gradient(frgd);
```

```
Img.write(grad);
```

```
End
```

```
Class Morphology
```

```
Object1=Classify(grad,frgd);
```

```
ApplyLI(object1);
```

```
// li---- luminance intensity
```

```
Lere=Convert(grad→view(lefteye,righteye))
```

```
DepthfocusImg=DIBR(lere);
```

```
Depthinfo=Depthfocus(DepthfocusImg);
```

```
Out=convert(Depthinfo→stereoscopic);
```

```
3D=Frame(out);
```

```
Display(3D);
```

End

Processing the input photos in a 2D series allows one to extract depth information. Reduce out-of-focus areas and bring the depth of field into focus. To get the depth information, use inverse filtering and simulate the object's perception. Locate the immediate area and determine the depth perception. At last, use the multi-resolution wavelet transform to separate the high- and low-frequency bands. acquire the local area and the backdrop surface region to acquire the foreground and background textures, then remove the foreground from the background using the cut off threshold. Use texture pitch to get a gradient, then make a stereoscopic view to foretell what objects will be active. After that, sort the input according to its depth using texture analysis and morphological operations. The last step in creating

stereoscopic 3D video is to apply a depth-based threshold to the incoming data. Convert the input pictures into left eye view and right eye view images; get the brightness intensity of each photos. Retrieve the array's depth data for the recently converted pictures (both eyes). Before converting the 3D array into frames, apply DIBR (input left and right eye views) and Depth of focus to create a stereoscopic view. Create stereoscopic 3D videos. The following function is used to produce the final output sequence once the left and right pictures have been defined. The foreground and background images are then subjected to depth array conversion methods. When the converted disparity background is used to create a realistic presentation for human perception, the baseline parameter of the s-sequence is employed. By using our suggested approach, the DIBR [1] technique is used to construct the whole video sequence.

Findings from the experiments and evaluation of performance Using the video sequence's converted pictures, frame size, and other parameters, we demonstrate the suggested process.



Fig2: 2D array of an input image

After converting a video sequence, the experimental findings are in line with the visual sense of 3D reality. The 3D array is shown in Figures 2 and 3, and the transformed sequence is then represented using the CMYK color model. When a better accuracy rate is required, the CMYK color model is used. Fig. 4 displays the 3D converted pictures that were synthesized from row interlaced images; these images indicate a 92% accuracy rate in rendering the 3D video sequence at the same quality (1080p) with minimum loss rate. Ref. 2-6



Fig3: Converted 3D stereoscopic image of an 2D image



Fig4: Right eye view image



Fig 5: left eye view image



Fig 6. Converted 3D stereoscopic images

Conclusion

In this article, we presented a novel approach to transforming 2D video sequences into 3D ones. Each parameter is taken into account during assessment (cue, saturation, depth perspective, active object, texture pitch (gradient), motion evaluation (parallax)) in our suggested approach for depth array conversion, which deals with improved accuracy in converting 2D arrays into 3D arrays. As a result, the converted sequence's depth array is generated by arranging the right (3D image) and left (3D image) pictures. In figures 2 and 3, the converted sequence's left and right images are clearly marked. You can make stereoscopic films using only these two picture sequences (left and right). using a 92% success rate (obtained using CRYENGINE) in 1080p 3D video at 32 frames per second, we can say that the Depth array conversion process is superior. Upcoming Improvement The enhanced buffer rate is a result of these requirements, since depth array conversion approach employs both the left and right eye views. Since deploying a high number of apps causes the buffer to overflow, we used our technique in CRYENGINE to address this problem. Better management of buffer overflows in the absence of a separate resource engine, such as "CRYENGINE," will be the focus of our future efforts. Without putting our model in an extra buffer, we would want to improve our work with HSU, a monochromatic color model with shifting intensity in red and blue.

References

- [1] Cheng-Ying Yeh, "Stereoscopic Conversion Technique Based on Monocular View Source", National Chung Cheng University, July 2008.
- [2] L. Zhang and W. J. Tam, "Stereoscopic image generation based on depth images for 3D TV", IEEE Trans. Broadcasting, vol. 51, 2005.
- [3] Wa James Tam, Liang Zhang, "3d-Tv Content Generation: 2d-To-3d Conversion", Communications Research Centre Canada, ICME 2006, IEEE, 2006.
- [4] R. Fergus, P. Perona, and A. Zisserman. "Object class recognition by unsupervised scale-invariant learning". In Proc. IEEE Conference on Computer Vision and Pattern Recognition, pages 264–271, 2003.
- [5] Gerald Fritz and Lucas Paletta and Horst Bischof; "Object Recognition Using Local Information Content", ICPR'04, IEEE conference, 2004.
- [6] Jae-Han Park, Seung-Ho Baeg, Jaehan Koh, Kyung-Wook Park, Moon-Hong Baeg; "A New Object Recognition System for Service Robots in the Smart Environment", International Conference on Control, Automation and Systems 2007, IEEE conference, 2007.
- [7] R.C. Veltkamp, M. Tanase, "Content-Based Image Retrieval Systems: A Survey", Technical Report UU-CS-2000-34 (extended version), 2002.
- [8] Adiat kola Abdul-Nafiu, et.al, "Effects of Electrode of 2D resistivity imaging to Delineate Subsurface features" American Journal of Applied Science, Vol. 10(1), PP-64-72, 2013.
- [9] Liang Zhang, Carlos Vázquez, and Sebastian Knorr, "3D-TV Content Creation: Automatic 2D-to-3D Video Conversion", Vol. 57, No. 2, 2011.
- [10] Al-Marshadi, A.H., "New procedure to improve the order selection of autoregressive time series model". J. Math. Stat., 7: 270-274., 2011.