Journal of Mobile Multimedia, Vol. 7, No.4 (2011) 285-304

A 3D INTERACTIVE SYSTEM FOR 3D DISPLAY

PEI-JUN LEE

Electrical Engineering Department, National Chi Nan University, Taiwan, R.O.C
pjlee@ncnu.edu.tw

EFFENDI

AU Optronics Corporation, Taiwan, R.O.C.

YUNG-YEN SU

Electrical Engineering Department, National Chi Nan University, Taiwan, R.O.C

Received Dec 10, 2011
Revised March 24, 2012

This paper proposes an interactive 3D picture system for 3D display application which includes the techniques of 3D content synthesis based on 2D image with its depth image and the interaction between the user, picture, and environment. The conversion technique uses depth image based rendering (DIBR) to synthesis 3D image. An adaptive depth map pre-processing method is proposed for DIBR to solve the hole occurring problem on virtual view images. The interaction is achieved based on the measurements of the sensors installed around the system and the methods of interaction tool recognition and speed estimation. To enhance the interaction, some sensors are installed around the picture system so that the scene in the picture can change corresponding to its surroundings. Experimental results show that the proposed system can convert any 2D image into a 3D image with good quality and achieve real-time interaction between the user, picture, and environment successfully. This proposed system can provide a good achievement for 3D display application.

Key words: 2D to 3D conversion, adaptive depth map pre-processing, depth image based rendering, interactive stereo picture, 3D display

1 Introduction

Recently, 3D technology has become the main focus for creating virtual reality experiences. Some studies have examined aspects of 3D image visualization, such as 3D displays, 3D content, and 3D space, which support realistic experiences [1–5]. The implementation of 3D technology has also been realized in real life, for example, in 3D TV systems, virtual conferences, health systems, and games. Many digital pictures with high image quality are already publicly available. Currently, digital pictures are used primarily in digital photo frames, artistic paintings, and digital posters. However, most of them still provide 2D content and lack interaction between the user and picture. To address the above

problems, this paper designs an interactive 3D picture system and the proposed system applies two techniques, 3D content synthesis and interaction between the user, picture, and environment. In some point of view, the system can be seen as a combination of art and technology.

This study designs an interactive stereo picture system that provides the user with a realistic experience through a combination of 3D content synthesis and the interactions between the picture system, the user, and the environment. The system contains two subsystems, the 3D content synthesis subsystem and the interaction subsystem. Figure 1 shows the structure of the proposed picture system. Using a technique based on DIBR which uses the original 2D video and its depth map synthesis multi-view video. The interaction subsystem detects the changes in the environment (such as weather, humidity, and vibration) by the sensors around the display, and then performs speed estimation for the interaction toll. The 3D scene in the picture system reacts to the signals from the interaction subsystem.

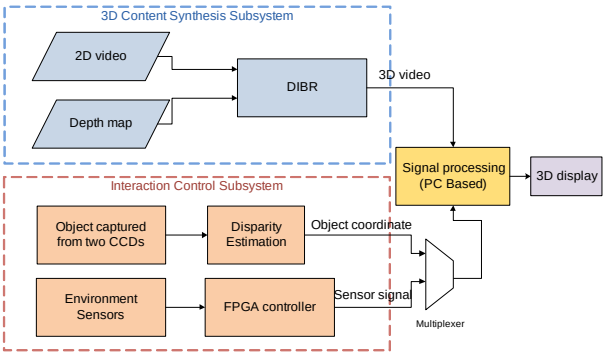


Figure 1 Structure of the proposed interactive picture system

The traditional method for producing a 3D image is to use two or more cameras to capture the required images for the left and right eyes, respectively. Although this method can provide correct view images for each eye with high 3D quality, it still has drawbacks. For example, a multi-cameras setup is costly; high transmission and data storage bandwidth are required for two or more color bitstream images which are applicable only for a single display configuration at the receiver. Furthermore, the depth effect is not easily customized. Depth image based rendering (DIBR) uses an original 2D image (intermediate image) with a corresponding depth map to generate virtual 3D images [6-8]. A depth map is a 2D gray-level image in which each pixel represents the depth of the corresponding color pixel in the intermediate image [9]. Since the depth of each color pixel is known, virtual views can be synthesized by per-pixel image mapping according to the corresponding depth value. However, because the 3D images generated from DIBR are virtual views, high quality is the most important requirement. The major problem in DIBR is that, after image warping, the holes occur in the virtual view images and degrad the quality of the 3D images. The papers [12] preserve the depth effect of the 3D image and require less computational time than those in [10] and [11]. However, some geometric distortions are still visible in the regions with vertical lines. These problems will make the user to be incomfortable when he watches the 3D image in the system. In this paper, an adaptive

smoothing method for depth map pre-processing in DIBR is proposed to reduce geometric distortion and to provide high quality 3D image form 2D content in the system.

The picture system in this study uses many sensors to detect changes in the environment. To illustrate the interaction between the user and the picture system, this study provides a ball-throwing scene as follows. A virtual dog in the picture chases the ball after the user throws it (see Figure 2). To implement the interaction subsystem, two CCDs are used to capture two slightly different scenes. The ball's speed and direction are estimated according to the calculated disparity between its motions in the two scenes. The scene of the picture system contains the interaction between users, the dog in the picture, and the environment around the display. Here, the ball is an interaction tool. When the user throws a ball toward the picture, the dog chases the ball according to the ball's speed and position. The selection of the 3D scenes is determined by the PC that processes a signal transmitted from the interaction subsystem.

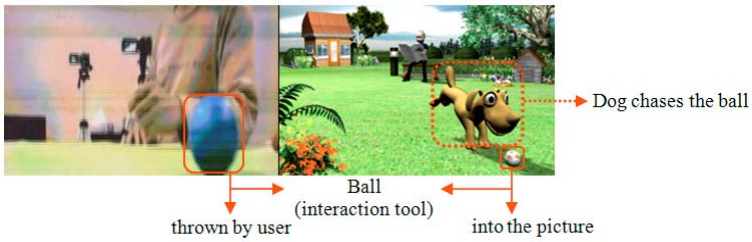


Figure 2 Illustration of the ball-throwing interaction

The main challenge in this study is the real-time implementation for the speed estimation based on disparity measurement. Foggia *et al.* [13] used two cameras and regional differences for disparity measurement and obstacles detection. Mühlmann *et al*. [14] proposed a sliding window method and the sum of absolute difference (SAD) to reduce the computational time for disparity estimation. Using the concepts of previous studies, this paper proposes a fast speed estimation technique for the interactive picture system. The proposed method detects and calculates the distance and the speed for the ball such that the user can interact with the picture system in real time. Therefore, the contributions of this paper are the conversion of 2D image to 3D image in the interactive system and the implementation of the real time 3D interaction subsystem.

This study is organized as follows. Section 2 describes the 3D content synthesis subsystem; An interaction subsystem for the 3D picture system is described in Section 3. The scene design and experimental results that demonstrate the system's performance are presented in Section 4. Section 5 concludes the paper.

2 3D Content Synthesis Subsystem

To synthesis the 3D images need two or more than two view images (two-dimensional images) by take in different view angles. In this study is based on a DIBR technique which creates the two view images by a 3D image warping process. Depth map pre-processing is applied to reduce sharpness of depth

transitions in the depth map, and to reduce the number of holes after image warping. The remained holes are filled by a hole-filling process. Figure 3 shows a block diagram of the DIBR technique [9], [15].

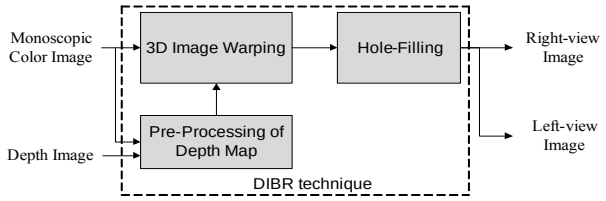


Figure 3 Block diagram of the DIBR technique

Depth map pre-processing reorganizes the depth values so that fewer holes appear in the virtual view images after image warping. Smoothing the whole depth map before image warping can significantly reduce the number of holes, but it also leads to geometric distortions, so that the quality of the virtual images is degraded and the computational time is increased. An adaptive edge-oriented smoothing method is proposed to solve this problem. The proposed pre-processing detects the hole regions before the smoothing filter is applied to the depth map. It also checks the vertical lines in the hole regions, since the vertical lines are the main cause of geometric distortion. The most suitable smoothing filter is determined depending on the presence of vertical lines and is employed to detect the hole regions. Figure 4 shows the block diagram of the proposed method.

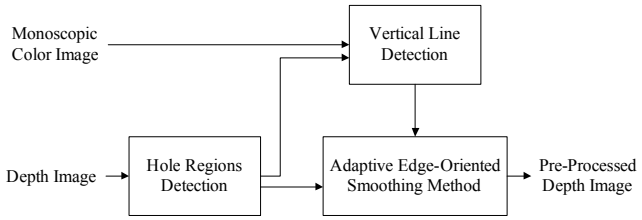


Figure 4 Block diagram of the proposed depth map pre-processing

2.1 Hole Regions Detection

To reduce the required computational time and preserve the depth quality in non-hole regions, it is essential to only smooth the hole regions in the depth map. Holes occur due to sharp depth discontinuities in the depth map. For the left view image, holes appear in the regions with a sharp low-to-high depth transition, and vice versa for the right view image [12] (see Figure 5). Based on the depth transition concept, a pixel is considered as a hole pixel after image warping if the depth value $s(x, y)$ satisfies the condition (1) and thus it is labeled.

$$P_L(x,y)=\begin{cases}1, \text{ if } s(x+1,y)-s(x,y)>\text{Th}_0 \text{ for left view}\\ \quad \text{or if } s(x-1,y)-s(x,y)>\text{Th}_0 \text{ for right view}\\ 0, \text{ otherwise}\end{cases} \tag{1}$$

This study sets the threshold value Th0 of sharp depth transition to be 20. Since the depth value of background pixels is always smaller than that of foreground pixels, we can use this knowledge to determine the background

(a) (b) (c) (d)

Figure 5 Analysis of holes occurrence in the "Interview" sequence. (a) intermediate color image, (b) depth map, (c) virtual right view image, (d) virtual left view image.

2.2 Vertical Line Detection

Any vertical line on object boundaries in a color image with sharp depth discontinuities will become geometric distortion artifacts obviously after image warping. To reduce geometric distortion, vertical lines must be detected by an edge detection operator.

The edge detection only calculates the pixels around the previously predicted hole pixel which belongs to background scene PL(x,y)=1 in (2).

$$V_d(x,y+q)=\begin{cases}1, \text{ if } \left|P_L(x,y+q)*E\right|>\text{Th}_1, \text{ where } P_L(x,y+q)=1,\\ \\ 0, \text{ otherwise}\end{cases} \tag{2}$$

where $q\in\{1,2,\text{L},r\}$ or $q\in\{-1,-2,\text{L},-r\}$. "$*$" is the convolution operator, r is the checking window size for vertical line detection and E in (2) is the 3×3 vertical Sobel convolution kennel. Then, from (2), we can use $V_L(x, y)$ to denote the existence of the vertical line around the hole pixel with coordinate (x, y). The vertical line detection is performed on the upward or downward direction of the hole pixel. Let

$$V_L(x,y)=\begin{cases}1, \text{ if } \dfrac{\sum_{q=1}^{r}V_d(x,y+q)}{r}\geq Th_2 \text{ or } \dfrac{\sum_{q=-1}^{-r}V_d(x,y+q)}{r}\geq Th_2\\ 0, \text{ otherwise}\end{cases} \tag{3}$$

where $V_L(x, y)=1$ denotes that the vertical line exists. Edges are considered as a line when the continuous edge number inside the checking window is larger than the threshold value Th_2.

2.3 Adaptive Edge-Oriented Smoothing Method

Based on the predicted hole regions and vertical line information, an adaptive smoothing filter is selected to reduce the number of holes and the geometric distortion. Figure 6 shows a flowchart of the proposed smoothing process.

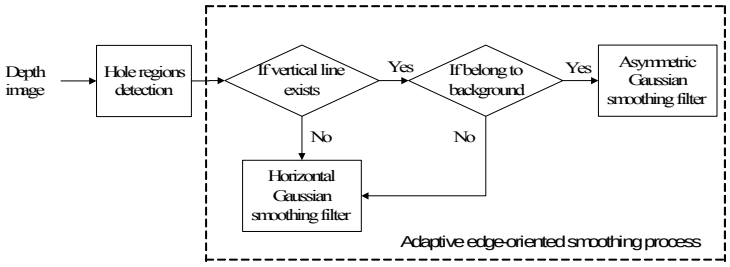


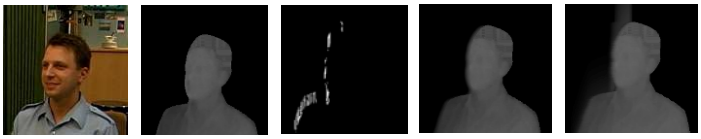
Figure 6 Flowchart of the proposed depth smoothing process

To correct the geometric distortion caused by vertical lines on the depth map in the regions with sharp depth discontinuities, the asymmetric Gaussian smoothing filter from [11] is employed. However, using the filter for non-hole regions may increase the computation time and does not preserve the depth information. Therefore, the asymmetric Gaussian smoothing filter is applied to hole regions only that have vertical lines and belong to the background scenery. Thus, users can be easy to experience the 3D effect for closer objects because the depth of the foreground scene is preserved. Because a parallel camera configuration is used for the image warping, holes occur only in the horizontal direction. Therefore, only the horizontal Gaussian smoothing filter ($g(\mu,\sigma_\mu)$) is applied to the predicted hole pixel (s(x,y)) without vertical lines as shown in (4); this also reduces the pre-processing computational time.

$$\hat{s}(x,y)=\frac{\sum_{\mu=-\frac{w}{2}}^{\frac{w}{2}}\left(s(x-\mu,y)\,g(\mu,\sigma_\mu)\right)}{\sum_{\mu=-\frac{w}{2}}^{\frac{w}{2}} g(\mu,\sigma_\mu)}, \tag{4}$$

Figure 7 illustrates the complete procedure for the proposed smoothing method. Note that the depth regions with sharp depth transition contain the vertical edge information belonging to the background scene which is smoothed by the asymmetric Gaussian smoothing filter.

Applying the proposed smoothing method to the depth map greatly reduces the number and size of holes in the virtual view images. Since small holes are remained in the virtual view image, the average filter is employed for hole-filling. After hole-filling, nine virtual view images without holes are generated as a 3D scene.



(a) (b) (c) (d) (e)

Figure 7 Smoothing procedure for the "Interview" sequence. (a) intermediate color image, (b) depth map, (c) edge buffer for the left-view image, (d) depth map for the left-view image after pre-processing by the horizontal Gaussian smoothing filter, (e) depth map for the left-view image after further pre-processing by the asymmetric Gaussian smoothing filter.

3 Interaction Subsystem Design

In this section we present the design of a 3D interaction subsystem. To demonstrate the interaction between the user, picture and envirement, there are a story and a scene to be introduction as follows.

1. In the initial state, an old man and a dog appear in the picture. If the picture is shaken in any way, the old man and the dog can feel the vibration, so they run away from the scene in the picture.

2. On sunny days (low humidity) in summer (hot weather) or in autumn (cool weather), when a user stands in front of the picture, the dog will run to the center of the picture or else it will run back to its doghouse.

3. On rainy days (high humidity) in summer or in autumn, the old man and the dog will run into their houses to avoid being drenched.

4. On a sunny day, when a user stands in front of the picture and throws the ball to the dog, the dog will chase the ball and then return it to the user. The user can throw the ball to the eight regions shown in Figure 10, where regions 1–6 are inside the width range of the picture. The regions 7 and 8 are outside the picture. This interaction between the user and the dog is achieved in real time by using the proposed speed estimation algorithm.

5. Finally, in winter (cold weather), regardless of the signal from the other sensors, the picture shows a snowy scene.

To achieve the above interaction subsystem, all signals from the sensors are processed by the microprocessor and transferred to the PC through the wireless transmission module. Moreover the speed information of interaction tool is estimated by the disparity estimation. The control interface handles all signals and selects the most suitable 3D interaction scene. Then 3D images are shown on the 3D display by image synthesis techniques. Figure 8 shows the flowchart of the interaction subsystem.

This study uses a 3D LCD display to show the picture scenes. Two CCDs are installed at the bottom of the display to detect the ball's speed and direction. Moreover, IR, temperature, humidity, and vibration sensors are installed around the display. The IR sensor is placed on the top of the CCD. The temperature and humidity sensors are placed on the right and left of the picture, respectively. The vibration sensor is installed at the back of the display. Figure 9 shows the arrangement of all sensors.

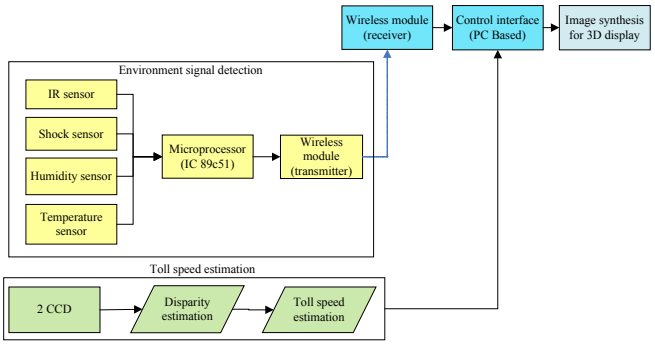
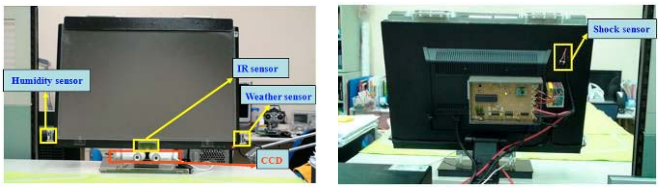


Figure 8 Block diagram of the interaction subsystem



(a) Fornt (b) Back

Figure 9 Sensors arrangement for the interaction subsystem

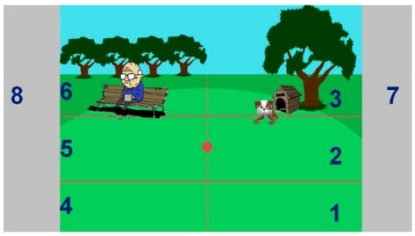


Figure 10 Regions to which users can throw the ball

The interaction subsystem includes 3 parts, the signal processor, tool speed estimation, and image synthesis for 3D display. The design detail is described as follows.

3.1 Signal processing

All environment sensor signals are processed by the microprocessor and transferred to the PC through the wireless transmission module. The control interface handles all the signals and selects the most suitable 3D interaction scene. In this subsection, the following two items are needed to be presented.

3.1.1 Sensors

The picture can react to any changes of the environment detected by the sensors. Table 1 shows the types of sensors and their functions.

Table 1 Sensors and their purposes

Sensors	FUNCTIONAL PURPOSE
IR sensor	To detect the existence of user in front of the picture
Temperature sensor	To detect the seasons change around the picture
Humidity sensor	To detect the weather change around the picture
Vibration sensor	To detect any vibrations to the picture

3.1.2 Controller Interface

The selection of 3D scenes is determined by a PC that processes signals transmission in the interaction subsystem. A microprocessor (IC 89c51) is used for the connection between the user and the interaction subsystem. When all the sensor signals are collected, the microprocessor sends them to the PC through a Zigbee wireless transmission module. Figure 11 shows the installation of the microprocessor and wireless module Zigbee on the circuit board

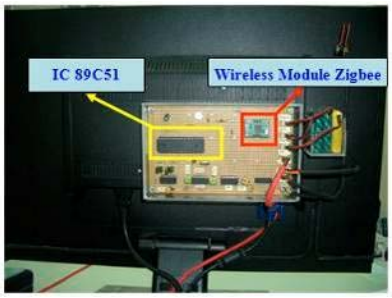


Figure 11 Installation of microprocessor and Zigbee

3.2 Interaction Tool Speed Estimation

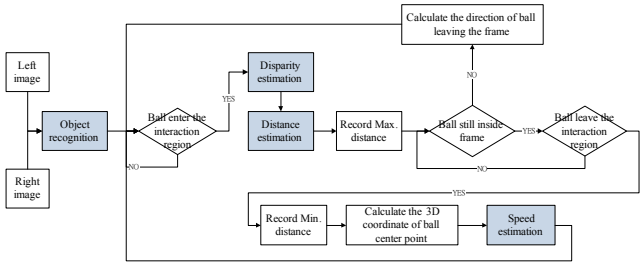


Figure 12 Flowchart of the proposed interaction toll speed estimation

The real-time detection of the interaction toll's speed is based on the proposed speed estimation. The proposed method uses two CCDs to recognize the interaction tool (ball) and calculate the corresponding disparity value [14] such that the distance between the display and ball, and the speed are estimated. Figure 12 shows a flowchart for the estimation of the ball's distance and speed.

3.2 .1 Interaction Tool Speed Estimation

To reduce the computation time, object recognition should be done before the disparity calculation. The object recognition method consists of four steps: background difference, color conversion (RGB to CgCbCr), color recognition, and component connection.

1. Background difference.

The background difference is employed to significantly reduce the possibility of mistaking other object as the ball when the object and the interaction toll (ball) are of the same color. First, a reference background image without the user and ball is saved to the database. Thus, the ball is clearly recognized.

2. Color conversion

The CCD captures RGB color images. However, RGB color domain is not efficient for color recognition since it is easily influenced by luminance changes in the environment. Therefore, this study converts the RGB pixel domain into the C_g (greenness), C_b (blueness), and C_r (redness) color domain from (5), since the latter is not sensitive to the luminance changes.

$$\begin{bmatrix} Cg \\ Cb \\ Cr \end{bmatrix} = \begin{bmatrix} 128 \\ 128 \\ 128 \end{bmatrix} + \frac{1}{256} \begin{bmatrix} -81.085 & 112 & -30.915 \\ -37.945 & -74.494 & 112 \\ 112 & -93.768 & -18.214 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \tag{5}$$

3. Color recognition

To achieve the best color recognition, we first save the initial C_g, C_b, and C_r pixel values of the ball and then check (6) to find the ball. If (6) holds, the ball exists.

$$\begin{aligned} (I_{Cg} - 20) < B_{Cg} < (I_{Cg} + 20) \\ (I_{Cb} - 20) < B_{Cb} < (I_{Cb} + 20) \\ (I_{Cr} - 20) < B_{Cr} < (I_{Cr} + 20) \end{aligned} \tag{6}$$

where I_{Cg}, I_{Cb}, and I_{Cr} are the initial pixel values of the ball. B_{Cg}, B_{Cb}, and B_{Cr} represent the current ball's color. Once the ball's color satisfies (6), the image is binarized such that the ball's pixel value equals 255 and the other values are zero.

4. Component connection

To distinguish the ball from other objects that have the same color with the ball, this study uses the connected component technique to categorize each object. After each object is categorized, the object size ratio (height and width) is measured. Only the object with the size ratio closest to one is then determined to be the ball.

3.2 .2 Speed Estimation for the Interaction Tool

To achieve interaction between the user and the picture system, this study uses disparity information to calculate the ball's speed and position. The disparity value (D*v*) is obtained from the minimum distance between the pixels in the left and right images measured by the SAD from (7).

$$SAD(x,y,d) = \sum_{i,j=-n}^{n} \left| L(x+j, y+i) - R(x+j+d, y+i) \right|, \; d = 0,1,2\text{L} \; d_{\max}$$
$$\text{disparity value} = Dv = \arg_{d} \{\min SAD\} \tag{7}$$

where n is the matching region size, (x, y) is the disparity pixel coordinate, is the horizontal search region, disparity is the smallest value of d ($\arg_{d}\{\min SAD\}$) after the SAD operation. L(x,y) and R(x,y) are the left and right binary image pixels in the position (x,y), respectively.

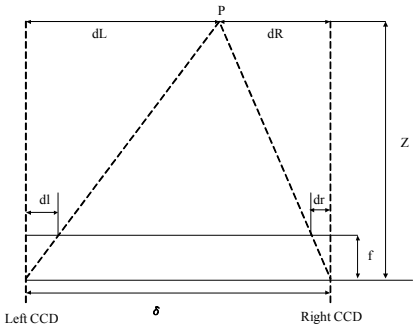


Figure 13 Illustration of the ditance perception

Since this study uses the disparity technique to realize an interaction subsystem, a real time process for the system is essential. Therefore, the proposed ball estimation method first uses the object recognition algorithm to capture the interaction tool (ball) and calculate only the disparity value of the ball's center point. The distance between the ball and the display can then be calculated. Figure 13 illustrates the distance perception using the disparity information

$$\delta = dR + dL = \frac{Z}{f}(dr + dl) \tag{8}$$

$$Z = \frac{\delta \times f}{(dr + dl)} = \frac{\delta \times f}{Dv} \tag{9}$$

where δ in (8) is the distance between the left and the right CCDs. dL and dR denote the horizontal distance from the left CCD to the object point (ball) P and the distance from the right CCD to P,

respectively. The disparity value (D*v*) is defined as (dl + dr) which is obtained from (7). The distance Z between the ball and the picture system can be obtained from (9).

From (9), D*v* and Z are inversely related. However, the experiment results in Figure 14 show that when Z is big, the relationship between Dv and Z is not linearly inverse, so (9) is no longer accurate. This is because a small image resolution is used in the experiment, and the alignment of two CCDs is not exactly parallel.

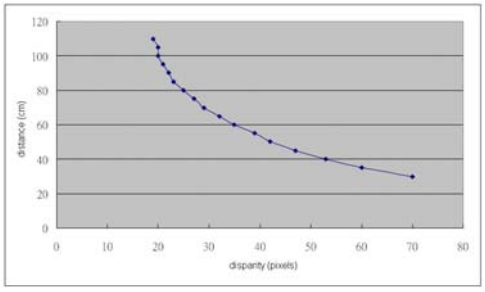


Figure 14 Disparity versus distance

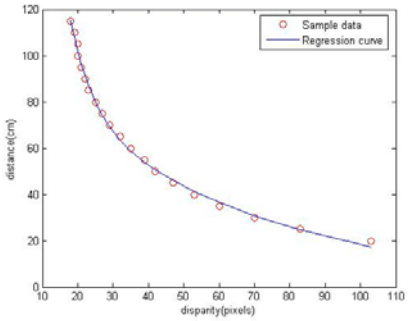


Figure 15 Curve for disparity to distance (simulated by MATLAB)

To efficiently calculate the ball's speed, this study uses a nonlinear inverse function to simulate the relationship between disparity and distance. The mathematical model of a nonlinear inverse curve is shown in (10).

$$Z = f(Dv) = a_1 e^{\lambda_1 Dv} + a_2 e^{\lambda_2 Dv} \tag{10}$$

where D*v* is the disparity value, and Z is the distance between the ball and CCDs. To increase the accuracy of the speed estimation, additional information from outside the interaction region is added to the calculation. Figure 15 shows the most accuracy relation curve for the disparity and distance calculated from (11).

$$Z = 102.1133 e^{-0.0171 Dv} + 695.7365 e^{-0.1582 Dv} \tag{11}$$

Thus, the refined distance Z can be obtained by inserting the disparity value (D*v*) of the ball into (11).

As the ball enters the interaction region, the system activates the timer (t_{max}) and records the distance as the maximum distance (Z_{max}). The timer will start when the ball begins to move toward the picture. When the ball leaves the picture, the timer stops (t_{min}) and records the last distance the ball reaches as the minimum distance (Z_{min}). Finally, the ball's speed is estimated by using (12). Table 2 shows the degree of speed and the corresponding region of the ball-throwing.

$$Ball\ speed = \frac{|Z_{\max} - Z_{\min}|}{|t_{\min} - t_{\max}|} \ (\text{m}/\text{sec}) \tag{12}$$

Table 2 Degree of ball speed

Ball speed	DEGREE	THROW BALL REGIONS (FIGURE 10)
> 2 (m/sec)	Fast	3 or 6 (based on ball direction)
0.8(m/sec)~2 (m/sec)	Medium	2 or 4 (based on ball direction)
< 0.8 (m/sec)	Slow	1 or 3 (based on ball direction)

3.2 Image synthesis for 3D display

To provide a better 3D experience, the image synthesis is a process to generates two view images to 3D display for the user with glasess, or nine-view images to 3D display without wearing 3D glasses, from one monoscopic color image and its corresponding depth map. This study uses a slanted lenticular technology for the 3D display, then the user can see 3D images without wearing 3D glasses. To provide a more comfortable and natural 3D effect, this study generates nine virtual view images. The nine-view images are then synthesized into one stereoscopic image and transmitted to the 3D display.

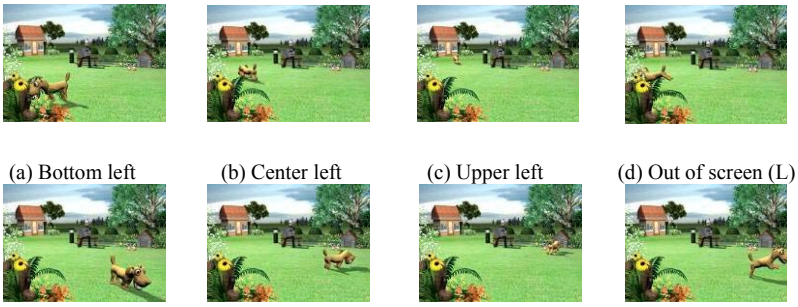
4 Experimental Results

4.1. Virtual Environment Design

This study designs a virtual world (animation) to simulate interaction between real-life events and the picture. The high-end 3D rendering software Maya is used to design the video sequences in this study. Some of the 3D models and textures are generously provided in [16–25]. The story line for the picture centers on an old man and a dog in a park. Three seasons are represented in the picture shown in Figure 16, summer (hot weather), autumn (cool weather), and winter (cold weather). In summer and autumn, rain is created to simulate humidity changes in the environment. Figure 17 shows sequences of eight images of ball throwing and the dog's actions on a sunny day (low humidity) in summer, where the user is standing in front of the picture. Finally, when the picture has any shaking, the old man and the dog will run away from the scene. Both of them will return to the picture when the shaking stops.

(a) Autumn season (b) Summer season (c) Winter season

Figure 16 Three seasons with their corresponding depth maps, respectively, in the interactive stereo picture system



(a) Bottom left (b) Center left (c) Upper left (d) Out of screen (L)

(e) Bottom right (f) Center right (g) Upper right (h) Out of screen (R)

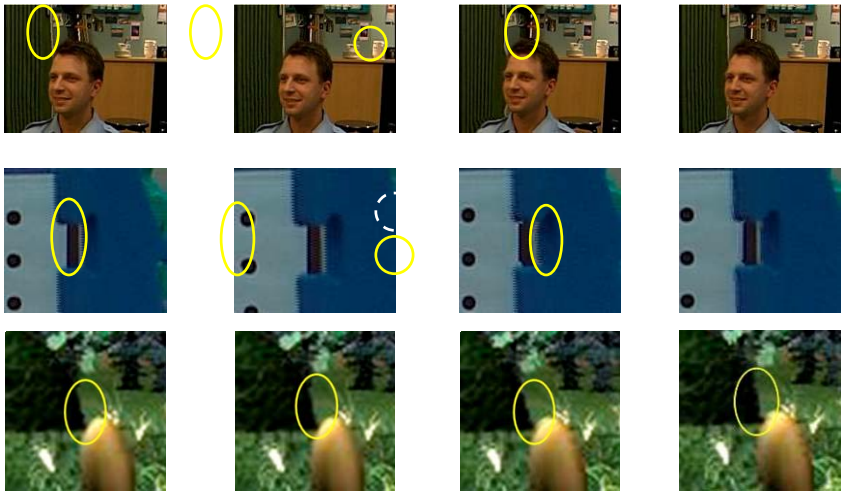
Figure 17 Eight-image sequence of ball throwing and the dog's chasing in summer

(a) Interview (b) Orbi (c) Dog

Figure 18 Three test video sequences and their corresponding depth maps

(a) [10] (b) [11] (c) [12] (d) Proposed

Figure 19 Evaluation of first view images for the three test sequences



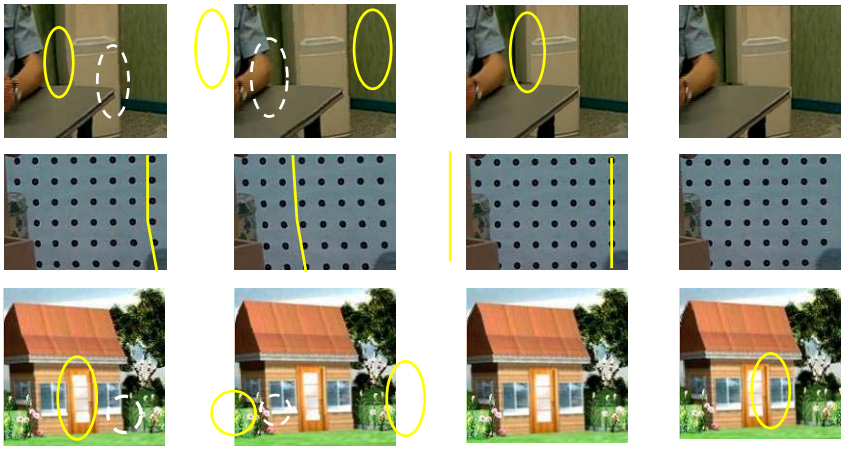
(a) [10] (b) [11] (c) [12] (d) Proposed

Figure 20 Enlarged segments of a hole region from Figure 19

4.2. Experiment Results of 2D to 3D Conversion

To present the performance of the proposed 2D to 3D conversion algorithm, the 3D image and some virtual video sequences are tested experimentally. Figure 18 shows the test video sequences with the corresponding depth maps. From left to right, they are "Interview," "Orbi," and "Dog" sequences. The first two sequences are "real" video sequences provided by the MPEG 3DAV group [27], [28]. These color videos were generated by using a real camera, and the depth maps were generated by using the IR-sensor camera ZcamTM in (OK or not?) 3DV systems. The third sequence is one of the video sequences for the picture system. The image resolutions of the "Interview" and "Orbi" sequences are 720×576 pixels, and the "Dog" sequence is 740×540 pixels.

The main objective here is to compare the image distortion performance of the previous studies [10]–[12] with the proposed method. Since the first view image has the largest baseline distance, which results in a bigger hole size than that in the other view images. The quality is evaluated on the basis of the first view image. Figure 19 shows the first view images generated by the methods in [10]–[12] and the proposed method. Enlarged segments of hole and non-hole regions from the first view images are shown in Figs. 20 and 21, respectively. The results demonstrate that the proposed method provides the best solution for image distortion in both hole and non-hole regions.



(a) [10] (b) [11] (c) [12] (d) Proposed

Figure 21 Enlarged segments of a non-hole region from Figure 19

4.3. Interaction Subsystem Proformance Evaluation

In this study, two CCDs are used to estimate the ball speed based on the disparity value. The distance between the two CCDs is 5 cm. The spatial resolution for both images is 320×240 pixels. The disparity evaluation for the proposed method is performed at different distances (see Figs. 22–24).

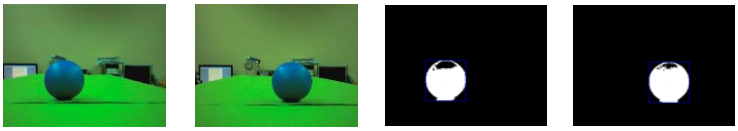


Figure 22 Distance evaluation: ball's distance away from two CCDs is 30 cm, disparity value is 70 pixels

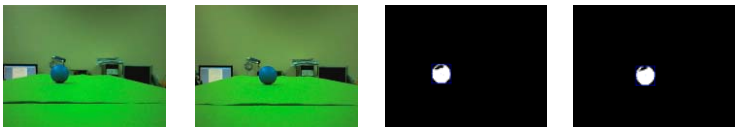


Figure 23 Distance evaluation: ball's distance away from two CCDs is 65 cm, disparity value is 32 pixels

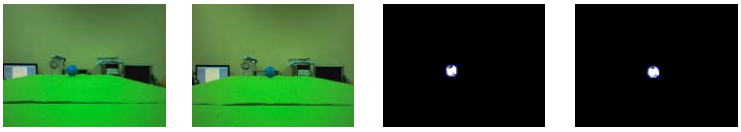


Figure 24 Distance evaluation: ball's distance away from two CCDs is 110 cm, disparity value is 19 pixels

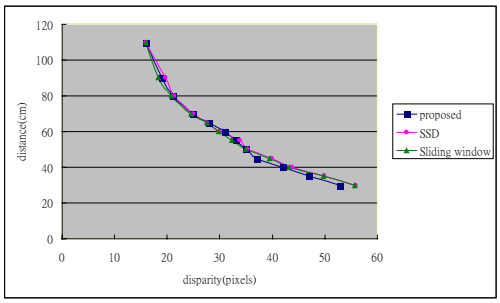


Figure 25 Disparity versus distance for the proposed, SSD, and sliding window methods

Figure 25 shows that the relationship between the disparity and the distance calculated from the proposed method is almost the same as those from the SAD and sliding window methods. Table 3 shows the accuracy of the calculated ball's speeds by the proposed method (P), SAD (S) method, and sliding window (SW) method [28] at different distances. Taking the SAD method as the standard for comparison, the error difference of the proposed method and the sliding window method are calculated by (13) and (14), respectively

$$\Delta P = \frac{P - S}{S} \times 100\% \tag{13}$$

$$\Delta SW = \frac{SW - S}{S} \times 100\% \tag{14}$$

Table 3 Comparison of the results of the speed of the ball

Distance (cm)	Proposed (pixels)	SSD (pixels)	Sliding windows (pixels)	Diff. (ΔP)	Diff. (ΔSW)
30	53	55.88	55.67	-0.05%	0.00%
35	47	49.87	49.76	-0.06%	0.00%
40	42	43.84	43.26	-0.04%	-0.01%
45	37	39.89	39.43	-0.07%	-0.01%
50	35	35.3	34.96	-0.01%	-0.01%
55	33	33.8	32.29	-0.02%	-0.04%
60	31	29.98	29.77	0.03%	-0.01%
65	28	27.58	27.58	0.02%	0.00%
70	25	24.98	24.5	0.01%	-0.01%
80	21	21.3	20.77	-0.01%	0.02%
90	19	19.67	18.26	-0.03%	-0.07%
110	16	15.98	15.88	0.01%	-0.01%

The experimental results show that the proposed method provides better accuracy than the sliding window method [15] when the distance is larger than 65 cm, whereas the sliding window method provides better accuracy if the distance is smaller than 65 cm. This is because the surface of the ball is obvious when the ball is close to the CCD, so each pixel may have a different disparity value. The SAD and sliding window method calculate the ball's disparity value for all pixels of the ball and then average it to obtain one disparity value. The proposed method assumes that the ball's surface is planar. Therefore, when the ball is closer to the CCDs, the sliding window method provides better accuracy than the proposed method. However, as the distance becomes larger and the image becomes more blurred, the number of ball pixel candidates for the sliding window method also decreases, and error matching occurs in disparity estimation. The experimental results show that the difference between the proposed method and the sliding window method is not more than 0.06%. The proposed method also reduces computation time considerably and achieves real-time processing by calculating the speed only of the interaction toll (see Table 4).

Table 4 Comparison of computational times

Method	Time (sec/frame)
Proposed	0.0469
SSD	18.42
Sliding window	0.36

5 Conclusion

This paper describes the implementation of a stereoscopic digital picture system which includes video animation design, 3D realization, and real-time interaction. The video sequence is designed to simulate the conditions of the real world. The proposed 2D to 3D conversion method is employed to convert the video animation into high-quality 3D video. This study used sensors to detect environmental changes around the picture system, which reacts on the display with appropriate scenery. The system also applies a proposed speed estimation method that allows users to interact with the picture system in real time. Interaction between the user and the object in the picture is represented by the user throwing a ball and a dog chasing it. Future studies will focus on how to widen the system's application to smaller and wireless digital devices. The realistic experiences and real-time interaction with an interactive stereo picture system may become available for different devices.

References

1. Smolic, K. Mueller, P. Merkle, C. Fehn, P. Kauff, P. Eisert and T. Wiegand, 3D Video and Free Viewpoint Video - Technologies, Applications and MPEG Standards. IEEE International Conference on Multimedia, 2006, Expo. 2161-2164.
2. L. Onural, A. Smolic, and T. Sikora, An overview of a new European consortium: Integrated three-dimensional television—Capture, transmission and display (3D TV). Proceedings of European Workshop on the Integration of Knowledge, Semantics and Digital Media Technology, 2004 (EWIMT), 25-26.

3. A. Smolic and D. McCutchen, 3DAV exploration of video-based rendering technology in MPEG. IEEE Transactions on Circuits and Systems for Video Technology, 2004 14(3), 348-356.
4. A. Smolic and P. Kauff , Interactive 3-D video representation and coding technologies. IEEE Int. Special ISSUE on Advances inVideo Coding and Delivery, 2005 93(1), 98-110.
5. L. Onural, T. Sikora, J. Ostermann, A. Smolic, R. Civanlar, and J. Watson, An assessment of 3DTV technologies. Proceeding of NAB 2006, Las Vegas, 2006 NV, 456-467.
6. C. Fehn, Depth-Image-Based-Rendering (DIBR), Compression and Transmission for a New Approach on 3D-TV. International Society for Optical Engineering Conf. Stereoscopic Displays and Virtual Reality Systems, 2004 XI, 93-104
7. L. McMillan, An Image-Based Approach to Three-Dimensional Computer Graphics . PhD thesis, University of North Carolina at Chapel Hill, Chapel Hill, NC, USA, 1997.
8. W. R. Mark, Post-Rendering 3D Image Warping: Visibility, Reconstruction, and Performance for Depth-Image Warping. PhD thesis, University of North Carolina at Chapel Hill, Chapel Hill, NC, USA, Apr.1999.
9. C. Fehn, K. Hopf and Q. Quante, Key Technologies for an Advanced 3D-TV System . International Society for Optical Engineering Three-Dimensional TV, Video and Display III, Philadelphia, PA, USA, 2004, 66-80.
10. W. J. Tam, G. Alain, L. Zhang, T. Martin, and R. Renaud , Smoothing Depth Maps for Improved Stereoscopic Image Quality. International Society for Optical Engineering Conf. Three-Dimensional TV, Video, and Display III, 2004, 162-172.
11. L. Zhang, and W. J. Tam, Stereoscopic Image Generation Based on Depth Images for 3DTV. IEEE Transactions on Broadcasting, 2005, 191-199.
12. W. Y. Chen, Y. L. Chang, S. F. Lin, L. F. Ding, and L. G. Chen, Efficient Depth Image Based Rendering with Edge Dependent Depth Filter and Interpolation. IEEE International Conference on Multimedia and Expo. , 2005, 1314-1317.
13. P. Foggia, A.Limongiello, and M.Vento, A real-time stereo-vision system for moving object and obstacle detection in AVG and AMR applications. Proceedings of Seventh International Workshop on Computer Architecture for Machine Perception, 2005 CAMP, 58-63.
14. K. Mühlmann, D. Maier, J. Hesser, and R. Männer, Calculating Dense Disparity Maps from Color Stereo Images, an Efficient Implementation. International Journal of Computer Vision, 2002, 79-88.
15. C. J. Kuo, C. Liao, and C. C. Jin, Stereoscopic Image Generation Based on Depth Images. International Conference on Image Processing, 2004 Singapore, 2993-2996.
16. Photoshop textures: http://www.photoshoptextures.com/
17. Free textures got3D: http://free-textures.got3d.com/
18. Free textures CG-India: http://cg-india.com/free_textures.html
19. Highend3D: http://www.highend3d.com/maya/downloads
20. Autodesk: http://area.autodesk.com
21. Fudge Graphics Textures: http://www.fudgegraphics.com/2008/11/free-hi-res-autumn-textures-tree-silhouettes-leaves/
22. Free Texture Maps: http://freetexturemaps.com/
23. 3D model: Dog(TOMMY) originally created by Prashant Chandra , provided by http://www.highend3d.com/maya/
24. 3D model: Old man (Alfred_1.2) originally created by Rodri Torres, 2006 (http://rodri.aniguild.com)
25. RobinWood Textures: http://www.robinwood.com/Catalog/FreeStuff/Textures/TexturePages/BallMaps.html
26. C. Fehn, K. Schr, I. Feldmann, P. Kauff, and A. Smolic, Distribution of ATTEST test sequences for EE4 in MPEG 3DAV. ISO/IEC C1/SC29/WG11 MPEG02/M9219, December 2002.

27. A. Redert, M. Op de Beeck, C. Fehn, W. IJsselsteijn, M. Pollefeys, L. Van Gool, E. Ofek, I. Sexton, and P. Surman, ATTEST—advanced three-dimensional television system techniques. In Proc. 3D Data Processing Visualization and Transmission, Padova, Italy, 2002, 313-319.
28. I. Daribo, C. Tillier, and B. Pesquet-Popescu, Distance dependent depth filtering in 3D warping for 3DTV. IEEE International Workshop on Multimedia Signal Processing, Crete, Greece , 2007, 312-315.