Video Game Interface Using Image Identification Technique

Chin-Ho Chung[1], Wen-Yuan Chen[2], Jheng-Jhao Gu[3], and Ciou-Yu Yan[4]
[1]*Department of Electronic Engineering,*
Ta Hwa Institute of Technology, Hsin-Chu, Taiwan, R.O.C.
chc@thit.edu.tw
[2,3,4]*Department of Electronic Engineering,*
National Chin-Yi Institute of Technology, Taichung, Taiwan, R.O.C.
[2]cwy@ncut.edu.tw, [3]mike0256x@yahoo.com, [4]r74125874@yahoo.com

***Abstract*—In this paper, we propose a video game interface scheme using certain image identification techniques. We use an LED (light emitting diode) gun, instead of a mouse, to aim at the target on the screen of the video game. This method obviates the problem of players sitting for a long time in one position and abandoning the game due to fatigue. We use a PC with a Webcam and an LED gun (instead of a mouse or a joystick) to play this shooting video game. Our procedure comprises six main steps: 1) capturing the image; 2) filtering the image and discarding the large region noise; 3) finding the targeted region quickly; 4) calculating the center of the targeted region; 5) detecting shooting and, 6) refilling cartridges. After running a simulation, we obtained experimental results which showed that our proposed scheme was successful in quickly and precisely locating the position of the target point, and detecting the firing of the gun.**

***Keywords*—Light Emitting Diode (LED), Webcam, Targeted Region(TA), Interface, Image Identification.**

1. INTRODUCTION

With the advance of technologies and the ubiquity of computers, development of new computer games is a prime field. Duan et al. [1] offered a unified framework for semantic shot classification in sports videos. They ran simulations on five sports video games: tennis, basketball, volleyball, soccer, and table tennis games. Their results showed 85%–95% accuracy in classification (over 5500 shots in one 8 hour period).

The traditional peripheral devices associated with computer games such as joysticks, and other specially made consoles, have been extensively explored in the literature [3-5]. Robot soccer games and robot basketball games are treated in papers [6 & 7] respectively. These games use traditional consoles, but they need an extra microcomputer. Thus, the price of the game is relatively expensive.

In this paper we present a scheme that uses image processing and recognition [8-13] to find the targeted position on the screen. We use a PC with a Webcam and an LED gun (instead of a mouse or joysticks) to play this shooting video game. As shown in Fig. 1, in traditional video games, we move the mouse to aim at the birds, press the right key of the mouse to shoot, and then press the left one to refill cartridges. In our scheme, standing in front of the computer, we hold an LED gun which is 1.5 - 2 meters distance away from the screen to play the game. We aim at and lock on to the target, and then shoot at the screen.

Fig. 1 a frame of shooting bird video game

The remainder of this paper is organized as follows: Section 2 describes the detection algorithm and its process details. The experimental results are given in Section 3, and Section 4 is the conclusion.

2. Detection Algorithm

This paper presents a method for shooting targets on a regular computer monitor screen. Fig. 2 shows the flow chart of our detection algorithm. In the first stage, we capture 5 color frames per second from the webcam of the PC. Then the Br and Bg image are generated by using Eq.(1) and Eq.(2), respectively. Binarization is applied in the next stage; binary images in red and green component are obtained. Next, we label the objects of Bg and Br, and denote them as Lg and Lr. Using accumulated numbers of Lr and Lg, the targeted location is easily obtained through the process denoted as Shooting Location Decision (SLD). This method discerns where the gun is actually pointing – the Targeted Region (TR) – as Fig. 3(d) or Fig. 4(d) shows, and computes the center of TR. We then filter the blue component image and detect if the gun is shooting. If results indicate "shooting" we output a signal. Finally, we apply a "detect refilling cartridges" (DRC) stage, in which we can rapidly detect if the players are actually shooting or refilling cartridges. The details are described in the following section.

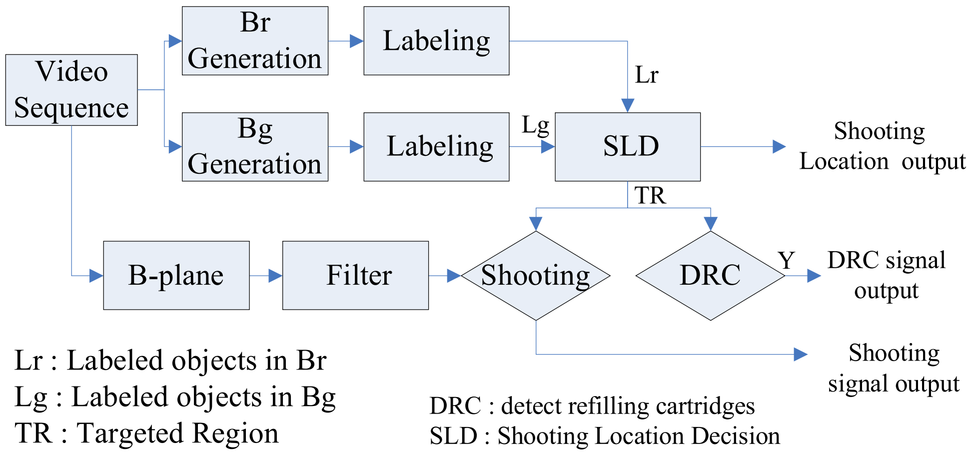

Fig. 2 The flow chart of the targeted point tracking algorithm

$$B_r(x,y)=\begin{cases}1 & if \quad (R>160)\cap(G<80)\cap(B<80)\\ 0 & otherwise.\end{cases} \tag{1}$$

$$B_g(x,y)=\begin{cases}1 & if \quad (R<60)\cap(60<G<100)\cap(B<50)\\ 0 & otherwise.\end{cases} \tag{2}$$

Where $0\le R\le 255, \quad 0\le G\le 255, \quad 0\le B\le 255$

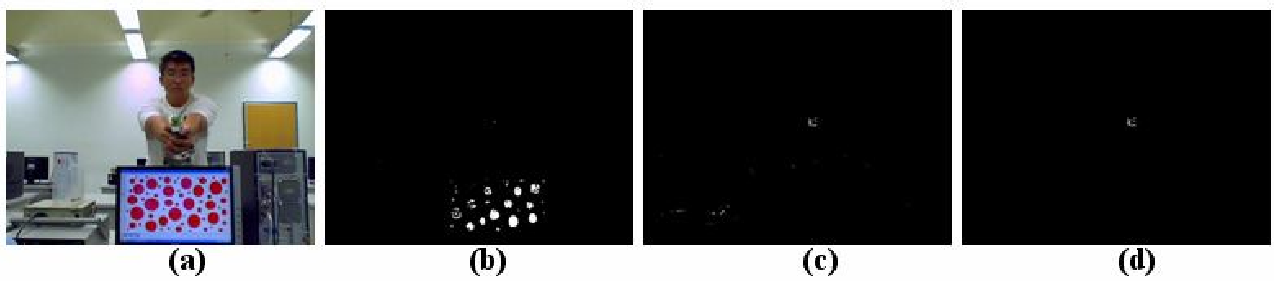

Fig. 3 The background image with more red pixels: (a) Original image; (b) binary Br image of (a); (c) the binary Bg image of (a); (d) the resulting image of TR.

Fig. 4. The background image with more green pixels: (a) Original image; (b) binary Br image of (a); (c) the binary Bg image of (a); (d) the resulting image of TR.

2.1. LED Gun

It is simple and easy to produce a LED gun. We used a toy gun, a three-color LED, a resister, and batteries to assemble our LED gun. Since people are particularly sensitive to RGB, in order to increase the speed of image processing, and to eliminate the background noise, we make the gun as Fig. 7(b) shows. A piece of paper with a red circle surrounded by a green circle in the outer area is attached to the muzzle of the gun. We use a blue light radiating device, located below the muzzle, to detect if the gun goes off.

2.2. Noise Reduction in Background

We select 5 frames per second from the Webcam, and filter large size noise blocks which contain red or green component areas whose pixels > 100. Fig. 3 shows the background image with more red pixels, and Fig. 4 shows the background image with more green pixels. Fig. 3(a) and Fig. 4(a) are the original color image. Fig. 3(b) and Fig. 4(b) are the binary image Br and Bg which show the noise filtered. We set two conditions which let the TR image be passed, accepted and identified, and filter out other background noises.

2.3. Shooting Location Decision Algorithm

Fig. 5 shows the system by which we can quickly find the x and y coordinates of the point targeted by the gun. We label the Binary image (Br) of the red component, and the Binary image (Bg) of the green component, respectively. Next, we accumulate the number of objects in Br and Bg to obtain Nr, Ng, respectively. Then we compare Nr and Ng, to determine which is larger. If Nr is larger, we start searching for the targeted point from Bg, or, if Br is larger, from Br. When we find the region in which the colors red and green are close together, that is the TR, as shown Fig. 3(d), or Fig. 4(d). Finally, we compute the center of the TR. Because the TR image is irregular, we determine its coordinate center by approaching from four corners of the TR image. We can quickly obtain four intersects (corners) of the TR. We record the LED gun region from left to right, and from top to bottom, (as shown in Fig. 6); these coordinates are given by $(X_{\min}, Y_{\min})$, $(X_{\max}, Y_{\min})$, $(X_{\min}, Y_{\max})$, $(X_{\max}, Y_{\max})$. We then use Eq. (3) and (4) to calculate the coordinate center (Xc, Yc) of the TR, and give the shooting location output.

$$Xc = (X_{\min} + X_{\max}) / 2 \quad (3)$$

$$Yc = (Y_{\min} + Y_{\max}) / 2 \quad (4)$$

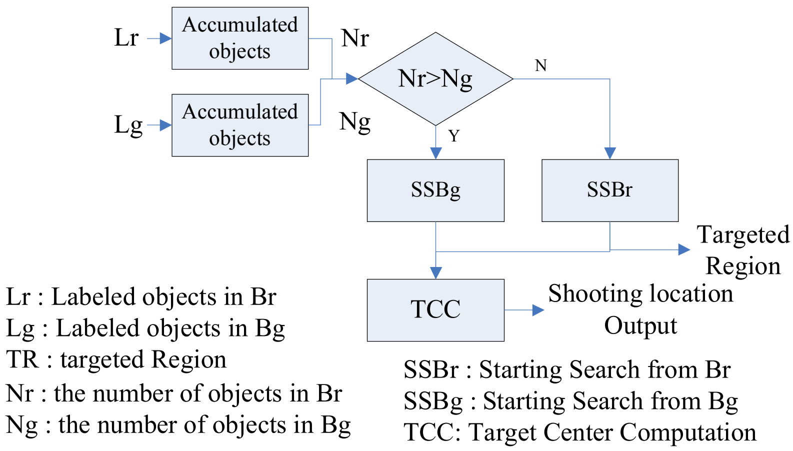

Fig. 5 Shooting Location Decision Algorithm

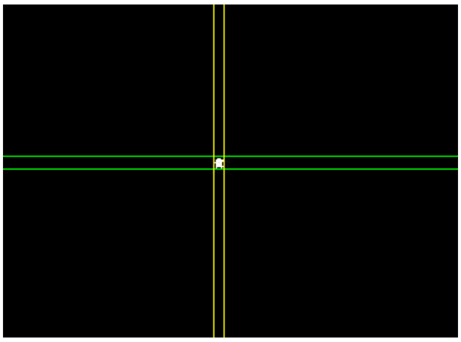

Fig. 6 TA is surrounded by two yellow lines and two green ones.

2.4. shooting Detection

After we find the TR and its center, we need to detect if the player is shooting. When the player is shooting, blue light is emitted, but since it is located under the TR, we need to extend the image range for this searching process. In Fig. 6 we compare the distances between two green lines and two yellow ones, and denote the larger one as ℓ. Then, we use the center of the TR as the center of the extended square image, and use $2.5*\ell$ as the side of extended square image (as Fig. 7 shows), which includes the range of the radiated image.

We search the extended square image instead of the whole image so that we can quickly detect when the player is shooting. Fig. 7(a) and Fig. 7(b) show the color image of the player, non-radiated and radiated, in the extended square image, respectively. Fig. 7 (c) and Fig. 7(d) show the binary image corresponding to Fig. 7(a) and Fig. 7(b), filtered by Eq.(5) to obtain the blue component image, respectively.

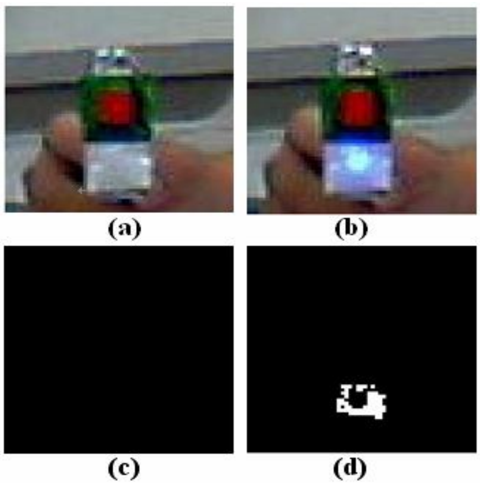

Fig. 7 The extended square image displays if the players launch or not: (a)non-radiated; (b)radiated; (c) the binary image corresponding to (a); and (d) the binary image corresponding to (b).

$$B_b(x,y)=\begin{cases}1 & if \quad (R<100)\bigcap(G<100)\bigcap(150<B<255)\\ 0 & otherwise.\end{cases} \tag{5}$$

Where $0 \le R \le 255,\quad 0 \le G \le 255,\quad 0 \le B \le 255$

2.5. Refilling Cartridges Detection

In test image sequences, there are 640*480 images acquired from the Webcam in a given video sequence. Suppose the top-left corner on the screen is original point (0,0), and the center of TR is expressed as coordinates (Xc, Yc) in the image. If the coordinate (Yc) of the TR image is greater than 450 as Fig. 8(a) shows, and the TR in the next image acquired has disappeared (as shown in Fig. 8(b)) this is interpreted to mean that the player has lowered the muzzle and is preparing to refill cartridges.

Fig. 8 two consecutive image for refilling cartridges: (a) a Yc value that the coordinate of TR image is greater than 450; (b) the TR of next image acquired is disappear

3. EMPIRICAL RESULTS

Several video sequences of the test image with frame size (640×480) were used in a simulation for purposes of demonstrating the performance of the proposed scheme. Fig. 9 shows the player standing in front of the computer and holding an LED gun. It is 1.5 - 2 meter's distance from the screen. Fig 9(a) – Fig. 9(d) are color images, and Fig 9(e) – Fig. 9(h) are the corresponding images which are generated by our scheme, respectively. We set the original point (0,0) at the top-left corner on the screen, and the second row of Fig. 9 shows from left to right with coordinates (349,146), (336,146), (321,150), and (304,148), respectively. After simulation from above, we can precisely discern the coordinates of the center of the targeted point in the image.

In our experimental test, we adopted an Intel CPU operating in a 2.0G Hz, PC platform, with RAM of 1.0G Byte per second to compose a 5 array image. In our procedure, using Matlab 7.1 edition, the track of the luminous point on the screen is moved as the muzzle. We were unable to find out if there was any delay taking place in the picture. After verifying our results, it was evident that our method is really fast, and definitely feasible. We were able to filter the various kinds of miscellaneous interference of background noise, and we could correctly detect the penetration of the signal as shown in Fig. 7 and the procedure of refilling cartridges (as in Fig.8) for magazine movement

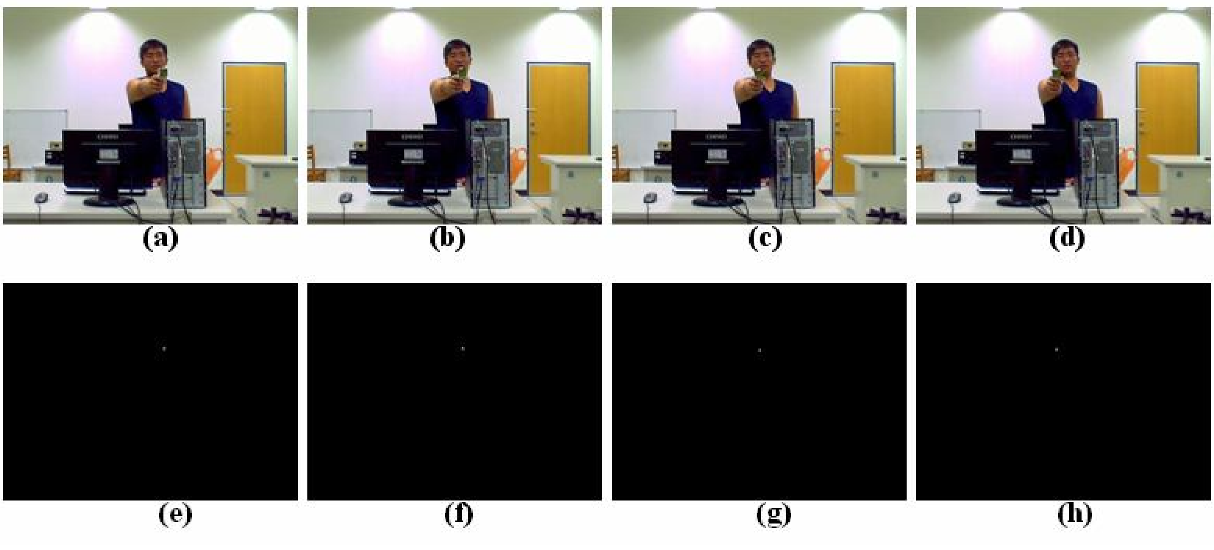

Fig. 9 The first row, Fig(a-d) are original color image, and the second row, Fig(e-h) , are the corresponding images using our scheme generated, the coordinates from left to right are (349,146), (336,146), (321,150), (304,148), respectively

4. CONCLUSIONS

As demonstrated by the above experimental results, we put the player "personally on the scene" by replacing the Genius mouse with the LED rifle. Because our method is fast and accurate, the luminous point moving over the screen and indicating the muzzle direction can be followed with sufficient speed. To sum up, our method has the following advantages:(1) It is simple to make; (2) It can really strain out the miscellaneous noise of the background; (3) The picture is smooth (luminous point is followed with sufficient speed); and (4) It can calculate the coordinate position of the targeted point with accuracy.

ACKNOWLEDGMENT

This work was partly supported by the National Science Council, Taiwan (R.O.C.) under contract NSC 97-2221-E-167-019.

REFERENCES

[1] L. Y. Duan, M. Xu, O. Tian, C. S. Xu, and J. S. Jin, "A unified framework for semantic shot classification in sports video ," *Multimedia, IEEE Transaction*, Vol. 7, Issue 6, pp.1066 – 1083, Dec. 2005.

[2] S. Cass , "Mind games" [computer game AI] *Spectrum, IEEE*, Vol. 39, Issue 12, pp. 40-44, Dec. 2002.

[3] Ming et al., " A low-cost force feedback joystick and its use in PC video games," *Consumer Electronics, IEEE Transactions* , Vol. 41, Issue 3, pp. 787-794, Aug. 1995.

[4] I. Fabrizio, F. Luca, R. Roberto, D. Giusti, A. Scebba, "Game Console Controller Interface for People with Disability Complex, " *Intelligent and Software Intensive Systems*, 2008. CISIS 2008. International Conference on, pp. 757 – 762, March 2008.

[5] K. S. Chang, G. B. Kim, T. Y. Kim " Video Game Console Audio: Evolution and Future Trends, " *Computer Graphics, Imaging and Visualisation*, 2007. CGIV '07, pp. 97 – 102, 14-17 Aug. 2007.

[6] R. C. Liu, I. F. Lin, " Fuzzy shooting control of car-like soccer robot " Electrical and Electronic Technology, 2001. TENCON. *Proceedings of IEEE Region 10 International Conference* on Volume 1, pp 447- 452, 19-22 Aug. 2001.

[7] C. Y. Cheng, H. Y. Liu, and C. C. Lo, " Toward a Robot Basketball Game "; *The 33rd Annual Conference of the IEEE Industrial Electronics Society (IECON),* pp.3013-3017, Nov. 5-8, 2007, Taipei, Taiwan.

[8] L. Rueda, B. J. Oommen, "On optimal pairwise linear classifiers for normal distributions: the two-dimensional case," *Pattern Analysis and Machine Intelligence, IEEE Transactions*, Vol. 24, Issue 2, pp. 274 – 280, Feb. 2002.

[9] Y. G. Pi, H. l. Shu, T. C. Liang, "The Frame of Cognitive Pattern Recognition", *Control Conference*, 2007. CCC 2007. pp. 694 – 696, Chinese, July 26-June 31 ,2007.

[10] S. J. Wang, X. Chen, "Biomimetic (topological) pattern recognition - a new model of pattern recognition theory and its application," *Neural Networks, 2003. Proceedings of the International Joint Conference* on Volume 3, pp. 2258 – 2262, 20-24 July 2003.

[11] J. H. Yang, M. S. Yang, "A control chart pattern recognition system using a statistical correlation coefficient method," *Computers & Industrial Engineering* , Vol. 48, Issue 2, pp. 205-221, March, 2005.

[12] Skomorowski, Marek, " Syntactic recognition of distorted patterns by means of random graph parsing," *Pattern Recognition Letters*, Vol. 28, Issue: 5, pp. 572-581, April 1, 2007.

[13] R. C. Gonzalez, R. E. Woods, *Digital Image Processing*, chapter 9-12, by Prentice-Hall Inc, New Jersey. 2002.