Reversi Game Using Vision Interaction

Jinpeng LI#1, Yinwei ZHAN#2

#*Faculty of Computer, Guangdong University of Technology*
Guangzhou, 510006, China
[1]jinpengli@yeah.net
[2]ywzhan@gdut.edu.cn

***Abstract*—this paper elaborates a vision-based interface (virtual touch screen) via an inexpensive webcam and an ordinary monitor for reversi game. Three main steps, calibration, segmentation and finding the finger tip, are presented in our model. Compared with GUI-based methods which is an efficient and effective way for office use and the classical intuitive methods using expensive device such as wearable camera, the simple model we propose is capable of efficacious.**

***Keywords*— Reversi Game, Human Computer Interaction, Virtual Touch Screen, Visual Screen, Bilinear Transformations.**

I. INTRODUCTION

GUI(graphic user interface)-based interaction, mainly employing mice and keyboards, makes computer more convenient to use, especial for office tools. Indeed, with the dramatically development of hardware, increasing their speed and capacity, GUIs cannot fully meet users' needs particularly in entertainment [6]. Instead of using conventional mice and keyboards, many amazing and intuitive human-computer interaction technologies have been presented such as virtual reality, augmented reality, machine vision, etc. But some methods have a strict demand for special facilities, such as wearable camera which is too luxurious for most users and developers in [1][5][2]. In [3][4] an interaction has been introduced with a projection screen and camera-tracked laser pointer which has similar structure with the model we propose. But this vision-based interface is more suitable for media classroom which resorts to convenient, but costly, projector. Therefore we establish a reversi game based upon inexpensive a webcam and a CRT monitor to illustrate vision-based interfaces.

Several works on finger pointing can be found in [9][8][7][1]. [9] consists of two systems: visual screen and visual panel, and we apply visual screen to reversi game. Visual screen means a user can manipulate computer by his/her finger tip touching on image plane. System can automatically transform the position of tip into the cursor position on screen by the bilinear ([8][7]) transformations lookup table with the aid of camera capturing and motion analysis techniques.There are many applications for visual screen where mice and keyboards are not convenient and accessible. For instance, shopping mall can build visual screen game or interaction advertisement with a projector and a plastic white board for visitors. Unlike the conventional touch screen which users can easily make insensitive, users cannot destroy visual screen directly.

Moreover many games do not call for accurate pointing. For example, when users play reversi games or chesses, they have to point to a box of chessboard, shown in Fig. 3, not a precision point on chessboard. As a result, our system can detect which box, not a precision point, has been touched to improve its efficiency and robustness.

This paper is organized as follows. Part II introduces restrictions and key techniques such calibration and segmentation of reversi game. Experiment results are given in Part III. At last, we give the conclusion and future work in Part IV.

II. REVERSI GAME

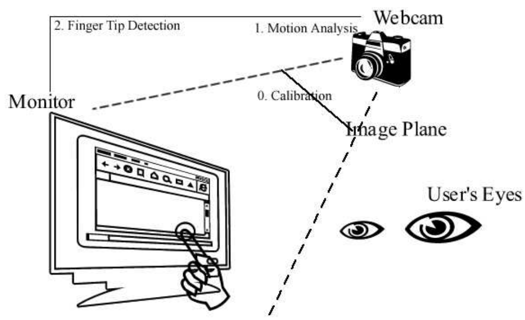

Fig. 1 Overview of Reversi Game

As shown in Fig. 1, we adopt a webcam with 320x240 resolution and a CRT monitor. Because the webcam cannot adapt well to varying light of environment, system should be placed in soft light environment. Besides the light, the angle between the webcam, the finger tip and user's eyes cannot be very large. Because the larger the angle becomes, the bigger the perspective distortion and the occlusion occur. In addition to the restriction, segmenting finger from changing content of chessboard is an important concern.

Three basic characteristics of reversi game should be introduced at first: (1) only one hand on the screen can be playing reversi game at one time; (2) there is a short period between two turns of players; (3) the only way that the finger points to box is crossing the edge of the screen that seen from image plane. In other words, all changes in screen without crossing or being adjacent to the edge of screen are chessboard's changes.

In our model, firstly we find four corners of screen (only for first-time user) for configuration and use bilinear transformations to calibrate the camera view, and we propose a fast model, which is suitable for the CRT monitor with aforementioned conditions, to find finger tip in runtime.

A. Calibration

As shown in Fig. 2, four corners, P'_0, P'_1, P'_2 and P'_3 denoted by (x'_0, y'_0) to (x'_3, y'_3) respectively, on image plane can be found manually or detected automatically by flashing four corners of the screen. In the system, of course, the screen position P_0, P_1, P_2 and P_3, which are defined by (x_0, y_0) to (x_3, y_3) respectively, can be read from the platform API (application programming interface).

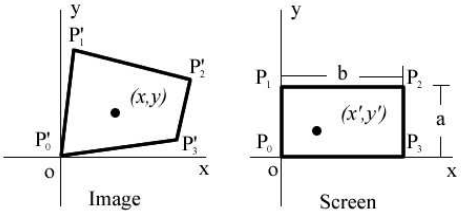

Fig. 2 Bilinear Transformations

Since the webcam is positioned where it can see the whole view of the screen of a monitor, the projection object of rectangleshaped from screen to image is an arbitrary quadrangleshaped. Thus, we use the bilinear transformations [7] to find the mapping, and it can be described by

$$\begin{aligned} x' &= \alpha^T m \\ y' &= \beta^T m \end{aligned}, \tag{1}$$

where $\alpha^T = [a_0, a_1, a_2, a_3]$, $\beta^T = [b_0, b_1, b_2, b_3]$ and $m^T = [1, x, y, xy]$. In order to determine α and β, we have to provide 4 couples of corresponding known points. Here, we can use the four corners' points. Let

$$\begin{aligned} (X')^T &= \alpha^T M \\ (Y')^T &= \beta^T M \end{aligned}, \tag{2}$$

where $(X')^T = [x'_0, x'_1, x'_2, x'_3]$, $(Y')^T = [y'_0, y'_1, y'_2, y'_3]$ and

$$M = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 0 & 0 & b & b \\ 0 & a & a & 0 \\ 0 & 0 & ab & 0 \end{bmatrix}. \tag{3}$$

We use *a* and *b* to denote the height and the width of the screen respectively.

Because $a > 0$ and $b > 0$,

$$|M^{-1}| = \begin{vmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1/a & 0 \\ 0 & 0 & 0 & 1/(ab) \\ 0 & 1/b & 0 & 0 \end{vmatrix} \neq 0. \tag{4}$$

So

$$\begin{aligned} \alpha^T &= (X')^T M^{-1} \\ \beta^T &= (Y')^T M^{-1} \end{aligned}. \tag{5}$$

Indeed, due to positions of a webcam and a monitor are invariant, we can calculate each pixel of screen for image-screen mapping in advance for the first-time use, and save the mapping into the lookup table in memory for real-time calibration.

In most cases, screens are not flat. We can solve this distortion by drawing a gird of circles on the screen and computing the centroid of corresponding ellipses projected by circles according to [9]. For a more accuracy method, you can read the [7]. But aforementioned bilinear transformations applied to our system is enough.

B. Segmentation

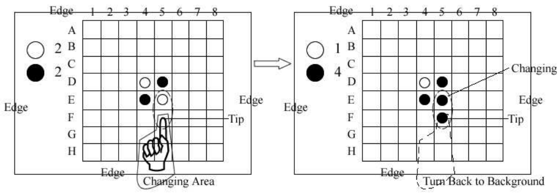

Fig. 3 Chess Board Changing

In terms of [9], they use the colour segmentation which can be described in brief as follows: 1. modeling the background's colour intensity as a Gaussian distribution without the hand since pixels in the image to some extent are invariance in colour space; 2. selecting a polygonal bounding area of the hand on the screen manually and modeling the hand in the same way; 3. segmenting the hand from background by a standard Bayers classifier.

In the reversi's case, we can segment the hand automatically by game rules' features. As shown in Fig. 3, the adjacent edge pixels of screen are changeless. Therefore when the hand is entering into chess board, only the changing pixels adjacent to edges can be regarded as a hand bounding area. Similarly, when the hand disappears, the bounding area come back to pixels of background except changes of chessmen. As a result, all changes far from edges can be defined as the background. In the program, according to the second characteristic mentioned at the beginning of section 2, when the hand leaves the chessboard, we update the background image during the vacant period. So the hand can be obtained by simple comparison of the runtime frame and the updated background.

In fact, a problem, the background cannot adapt well to luminance changing, occurs due to the webcam's poor ability of adapting varying environment luminance. For instance a person passes the system and its shadow will influence it significantly. Two methods can solve this problem. (1) One is to increase the threshold of variance for detecting motion. (2) The another is to define each pixel two statuses, changing (hand) and unchanging (background), and we set a variable for each pixel for the "lifespan" (for example 5 seconds) on their changing status. Since the hand can not be on screen for a long period, if a pixel keeps changing status more than lifespan, we can set a pixel into unchanging status. Another important function of the method 2 is that when the system enters an unknown situation that it does not work correctly, we can wait for the lifespan time to reset all pixels' status.

C. Finger Tip Detection

In our implementation, the definition of the tip point only is a position, not includes a direction which is useless for reversi game. According to another characteristic that the only way, the hand points to the box, is crossing over the edge. So the bounding area of the hand, which can be easily got from the approach mentioned above, should be adjacent to edges and the tip has the maximum Euclidean distance from the adjacent edge.

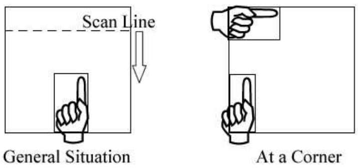

Fig. 4 Different Hand Poses

As shown in Fig. 4, we can define H as the set of a hand's coordinates (h_1, h_2, h_3...) and E as the set of points (e_1, e_2, e_3...) for the crossing edge. So the tip point

$$t = h_i \text{ where } h_i = \max_{h_i \in H, e_j \in E} (|h_i - e_j|). \quad (6)$$

For implementation, we can use a scan line to find the furthest point from the edge.

There is a another robust finger tip locating method mentioned in [9] and its definition of tip point including the direction is different from us, which is easily confused at a corner situation in our model. In short, the [9]'s method is to calculate the horizontal histogram for a cumulative total of the number of pixels that belong to the hand, and find the first appear and increase in the histogram as identified scan line. Then a number of lines above and below the identified line are selected for finding each center pixel between the start and the end of a hand in horizontal direction. Furthermore the intersection of the line, which is fit through the center points, and the identified line is the finger tip.

For the event design, we can create a click event directly by a finger tip stopping over the screen for a short period. Many actions can be implemented by the click event. Here the click event stands for placing a chessman on the chessboard. Until now, all the vital techniques have been introduced, we can play a reversi game now.

III. Experiment Results

Fig. 5 Poingting

As shown in Fig. 5, we can see a finger clicking the chessboard, and the background is changing in the right image. These pictures are taken by the same type of webcam with resolution 320x240. Reversi game is running on the Pentium IV 1.5 GHz with an average frame rate of 40 fps. In implementation, image is capturing by directshow and the user graphic interface is drawn by OpenGL in windows XP.

Different implementations have different speeds. Using VFW for webcam capturing and OpenGL drawing without any calculation for the reversi game, only 20 fps can we get. Rendering the chessboard by OpenGL, we can get above 130 fps. Only directshow capturing have about 60 fps. Therefore, it is more suitable to use directshow to capture the image.

Some issues are found in our experiments.

(1)Sometimes the reflection light from the surface of screen is very strong which results in chaos in detecting the finger tip.

(2)Furthermore, as mentioned before, if a person passes the screen, the system occurs the same problem like the first issue, and sometimes we even can see the passing person shadow in the screen because of the reflection. For these two problems, the only thing we can do is keep the soft and invariance environment luminance for system due to webcam's poor adaptability.

(3)What is more, the angle between webcam, screen and user's eyes becoming lager will lead to much more distortions and occlusions. For the occlusion exactly no methods can find the finger tip since the system cannot see it at all.

In a turn of reversi game, we missed average 2 clicks and the 5 error clicks with 53 correct clicks. For the missing clicks, the system is in chaos which means the click event occurs in invalid position, and we have to wait for 5 seconds to reset the system. For the error clicks, these often occur near edges of the screen because the occlusion often happen near edges and the finger here is too small to detect in image plane.

For the aforementioned problem that the screen is not flat, it is not affected evidently. So we can use bilinear transformations to calculate the mapping lookup table directly.

In order to improve the accuracy of pointing, the system can give a feedback for users such as a red point drawn on the screen to let user know where s/he point to.

IV. Conclusion and Future Work

In this paper, we describe a method for a two-player reversi game based upon vision interaction using inexpensive webcam and an ordinary CRT monitor. The webcam captures the image of the monitor and we calibrate the arbitrary quadrangle projection from the screen with bilinear transformations. Then the hand is segmented and the finger tip is found by bounding area. We consider a click event as a finger tip stopping over the screen for a short period (such as 1 second). This click event is employed to implement the reversi game. It imitates a function like a touch screen. So we can call it visual screen.

For the issues mentioned above, we can improve abilities of devices. For example, we use a high resolution camera to recognize the finger tip. It is unreasonable to detect the finger tip with lower than 320x240 webcam (because the screen cannot fully take up the full image plane) in 1024x768 screen.

For the occlusion, multiple views can be applied to solve this problem. We can put two cameras to top left and top right above the monitor symmetrically. Some multiple view techniques can be employed to analyze to finger tip position. Furthermore the imaging unit can use CCD instead of CMOS.

Although some problems have been found, but we are success to build this intuitive interface for us to play the reversi game. This system is built on a very cost-efficient device and it can be widely employed to advertisement and entertainment. For the advertisement, we can use a projection screen such as a plastic white board and a projector to play some interactive advertisement.

REFERENCES

[1] J. Coutaz, J. L. Crowley, and F. Berard, "Things that see: Machine Perception for Human-computer Interaction", Communications of the ACM, Vol. 43, No. 3, pp. 54-64, 2000.

[2] M. Hoch, "Object Oriented Design of the Intuitive Interface", 3D Image Analysis and Synthesis, pp. 17-19, 1997.

[3] C. Kirstein, H. Muller, "Interaction with a Projection Screen Using a Camera-Tracked Laser Pointer", Multimedia Modeling Proceedings, IEEE Computer Society, pp. 192-192, 1998.

[4] D. R. Olsen Jr., T. Nielsen, "Laser pointer interaction", Proceedings of the SIGCHI conference on Human factors in computing systems, Conference on Human Factors in Computing Systems, pp. 17-22, 2001.

[5] T. Starner, S. Mann, S. Mann, B. Rhodes, J. Levine, J. Healey, D. Kirsch, R. W. Picard, and A. Pentland, "Augmented Reality Through Wearable Computing", Presence, Special Issue on Augmented Reality, Vol. 6, No. 4, 1997.

[6] M. Turk, "Perceptive Media: Machine Perception and Human Computer Interaction", Chinese Computing Journal, 2001.

[7] S. Wangsiripitak, "Cursor Position Estimation Model for Virtual Touch Screen Using Camera", Proceedings of World of Academy of Science, Engineering and Technology, Vol. 5, pp. 30-33, Apr. 2005.

[8] Z. Zhang, "A Flexible New Technique for Camera Calibration", IEEE Transactions on Pattern Analysis and Machine Intelligence, 2000.

[9] Z. Zhang, "Vision-based Interaction with fingers and Papers", Proc. International Symposium on the CREST Digital Archiving Project, Tokyo, pp. 83-106, May 2003.