Face Tracking as an Augmented Input in Video Games: Enhancing Presence, Role-playing and Control

Shuo Wang* Xiaocao Xiong Yan Xu*** Chao Wang** Weiwei Zhang* Xiaofeng Dai* Dongmei Zhang***

* Microsoft Research Asia, Center for Interaction Design
5F, Sigma Center, No.49, Zhichun Rd. Haidian Dist., Beijing 100080, P. R. China
{shuowang, weiweiz, dongmeiz, xdai}@microsoft.com
** Tsinghua University, Haidian Dist., Beijing 100080, P. R. China
{xxc, wchao04}@mails.tsinghua.edu.cn
*** Renmin University of China, Haidian Dist., Beijing 100872, P. R. China, vivi.xy@gmail.com

ABSTRACT

Motion-detection only games have inherent limitations on game experience in that the systems cannot identify the player's existence and identity. A way of improvement is by introducing information such as a player's face or head into the system. We designed and implemented two game prototypes that apply real-time face position information as intrinsic elements of gameplay to enhance game experience. The first prototype augmented a typical motion-detection-based game. Face information was designed to enhance the sense of presence and role-playing. In the second prototype, face tracking is applied as a new axis of control in a First Person Shooter (FPS) game.

Although Face detection and tracking technology has started being utilized in game scenarios, there was little systematic research on how game experience is leveraged by applying face information to video games. The results of our user tests on comparing camera-based video games with and without face tracking demonstrated that using face position information can effectively enhance presence and role-playing. In addition, an intuitive control that augmented by face-tracking in the FPS game also got positive feedbacks from the test.

AUTHOR KEYWORDS

Face tracking,, motion detection, camera-based games, presence, role-playing, game control, First Person Shooter (FPS).

CHI 2006, April 22–27, 2006, Montréal, Québec, Canada.

ACM Classification Keywords

H5.m. Information interfaces and presentation (e.g., HCI): Miscellaneous.

INTRODUCTION

Mainstream commercial camera-based game systems [1, 3] that use motion-detection technology take a player's physical existence for granted. Our research focuses on augmenting input of video game systems with real-time face or head information. We achieved this in order to:

- Enhance the sense of human presence in games that are based only on motion detection;
- Visually enhance role-playing in a compelling way by employing "virtual props."

We also conducted research on enriching the FPS game experience by using physical movements as game controls. This research aims to demonstrate how the use of a mixture of traditional and physical controls in video games can enhance a user's game-playing experience.

Problems with Current Motion-detection Games

Camera-based entertainment has shown great potential in recent years, as revealed by commercial game titles like Sony Eyetoy® in Sony PlayStation2 [1, 3].

In such game systems, a player's physical motions are observed by a camera and processed by a computer as computer vision. The movements can then be used to control the games. The camera is typically permanently placed on the display facing towards the person standing and moving in front of it. Throughout all the activities, the player views a video mirror image of himself immersed in a virtual graphic world. Interaction with this world is undertaken by moving his body or by touching hotspots in the virtual environment.

Despite a multitude of game packages and graphic treatments, the mechanism behind all these games that utilize motion detection can be mainly divided into two categories:

(1) Triggering pre-defined hotspots: players react to hotspots through body movements to receive feedback. Types include boxing (*Virtua Fighter*, Figure 1), dancing (*Boogie Down*), exercise (*Eyetoy Kinetic*), and board games (*Eyetoy Chess*).

(2) Controlling an avatar: directional control of a running/flying avatar through motion, such as *Billy Hatcher* (Figure 2 a). The avatar is controlled by waving one's hands to activate virtual triggers to determine an avatar's orientation.

Using face and motion-tracking technologies, Eyetoy *Antigrav* (Figure2 b) also shows another way of controlling an avatar: players control their on-screen character exclusively with their body movement: leaning, ducking, and reaching. These movements are translated into player's on-screen character copying a player's movements. Compared to motion-detection only games, *Antigrav* explored the promising use of face tracking specifically in controlling an avatar. It proved that face tracking has more to offer for camera-based video games.

(a) **(b)**

Figure 1. Screen shots of "*Virtua Fighter*": (a) The initial screen of the game limits player's position with an outline. (b) The game screen shows hotspots marked with "Attack" and "Guard."

(a) **(b)**

Figure 2. Screen shot of Eyetoy games: (a) *Billy Hatcher*. (b) *Antigrav*.

All those types of games, especially the first kind, have limitations on sensing presence. For example, the game instructions in *Virtua Fighter* (Figure 1) states: "Defeat your opponents with skillful attacks while defending areas of the screen." [1] This flaw appears when the virtual boxer continues to punch the area defined with the player's outline after the player has left to answer the door, and the player loses. Motion-detection only games are often hard to sustain long-term use since they require intense physical effort to sustain play.

The reasons for those limitations are because the motion-only system does not have enough information to answer questions below: Is the player left or standing still in front of the camera? Is it a human player who is producing the detected motion? Is the player overreacting and falling out of the camera's view? How many people are in the game? Is the player cheating with helpers? Is the player too close to or too far from the camera? Most of the limitations can be solved with face tracking as shown in Table 1.

Motion Detection versus Face Tracking in Games

Motion information is cheap to acquire and utilize, yet poor in fidelity. It is not constant and is unlikely to enable richly detailed feedback because it is wholly dependent on a player's motion. Associated algorithms are unable to accurately represent the strength, orientation, velocity or acceleration of a subject's actual physical motion [4, 23]. For example, regardless of how hard a player smashes the wooden board in the bonus round of the *Eyetoy Kung-Foo*, identical bonus points are earned (Figure 3a). This may greatly discourage players who go all out in trying to achieve bonus points. Likewise, pure motion detection does not stop players from cheating with a big paper bag to clean the window in *Wishi Washi* instead of using their bare hands (Figure 3b).

(a) **(b)**

Figure 3. Screenshot of Eyetoy games: (a) The bonus round of *Kung-Foo*. (b) Cheating using a big paper bag in *Wishi Washi*.

On the other hand, face tracking systems can solve many of these issues. Unlike motion detection, the information used in face tracking is constant and does not require constant motion from the user. It is also able to provide higher fidelity. Once the system detects and tracks the face area, its potential uses are wide and varied.

Note that though face detection and tracking algorithms are based on human face information: face pattern and color histogram. The output is a rectangle that encloses a face (or multiple rectangles enclose a face in each). This rectangle keeps updating position and size in real-time according to the face(s) detected in the scene. Facial features like eyes and mouth are not analyzed in our current work. However, the rough positions of facial features could be inferred from

the rectangle by the heuristics on a relative scale of facial features.

Although compared to motion detection, face tracking has more dependencies like background (disturbing patterns), skin tone and lighting condition, it could still be considered as an augmented input (See Table 1).

	Motion detection	Face tracking
Assumptions	Some games require players to stay within an outline; All detected motions are valid.	Player face represents his/her identify; Players usually face toward camera.
Fidelity	No human presence detection; Poor on detect the orientation, velocity, and acceleration.	The existence of a player and the number of players; Real-time face position.
Sustainability	Need to generate motion.	Effortless.
Visualization of the output	Render the area of motion detected.	Face area is rendered and feature positions inferred.
Identity of multiple players	Not supported.	Given that players have no occlusion, they hold a consistent identity.
Mechanism as a controller	On/off switches.	Pointing device, orientation controller.
Reliability	Less dependencies.	Higher risks of failure (more dependencies).
System requirements	Less computational resource cost.	More restrictions: background, skin tone and lighting.

Table 1. Characteristics of motion detection vs. face tracking in video game systems.

FACE TRACKING IN FPS GAMES AND BEYOND

Taking research a step further, we have come to believe that face tracking could also benefit conventional video games by augmenting them with another input axis. The only effort required by players for acquiring this input is simply to be present. A typical behavior pattern is for players engaged in video games to exhibit physical behavior not necessary for game control, such as leaning when steering and dodging when being "attacked". Those body movements can be translated into face position changes, and can thus be readily utilized as augmented input.

We implemented an FPS game prototype to realize physical peek-and-dodge movements with the intension of testing usability and experience of the hybrid model of control. As this trend of a camera-based input method becomes more accessible, conventional video game control models based on the input devices like a mouse, keyboard and game pad are being reconsidered. This is a logical direction that game designers could go when considering additional camera functionality within their games.

RELATED WORKS

Inspired by the pioneer work *VideoPlace* by Krueger, et al. [12], Interactive video mirrors and processed video of the user have been used in games and art installations in which two dimensional (2D) video images of users interact with computer-generated, animated characters.

There are also commercial applications of this approach, such as the Intel® *Play Me2Cam* [11], Eyetoy camera-based games [1, 3], and games presented by GestureTek [2], among other similar ones.

The Alive system is an example of a more sophisticated, three-dimensional (3D) interactive video mirror in which users can interact with virtual character agents by using gestures [22].

The performance of face detection and the robust nature of object tracking are two barriers preventing such computer-vision technologies from being implemented into video games. Recently, those two issues have greatly improved [19]. They are mature enough for practical use in enhancing gaming experiences and situational awareness, immersive qualities and personal identity, among other game qualities.

Although the prior face tracking systems apply face or head information in real-time in a robust and reliable manner [9, 8, 15], user experience was not a focus of their research.

In summary, although these commercial and research systems have had a major impact in the field, there was little systematic research on how user experience is leveraged by applying face information to video games.

DESIGN FOCUS

In this section, we introduce our approach of enhancing video game experiences on three levels: presence, role-playing, and control. These aspects are related to typical cognitive processes when a player is exposed to game systems. We express this process in these ways: Presence: Am I in the game (is the fact of "existence" convincing)? Role-playing: Whom am I playing? (What should I do in the name of the game character)? Control: What can I control? (Are the controls intuitive and easy to perform)?

Presence is defined as the subjective experience of being in a place or environment, even when one is physically

situated in another [5]. We refer to it as the primary level of existence in a game world.

A system should be aware of a player's existence inside the game world and provide qualitative feedback so the player will acknowledge this existence. For example, the system should be able to identify where the player is standing. In addition to this, the intrinsic difference of the head area from the body (like vulnerability, using one's head to reach objects) can be emphasized for a far more convincing presence.

Compared to presence, role-playing allows a player a higher level of identity awareness. When a player takes on the identity of a game character, the player also accepts the characteristics associated with that identity: outfits, assignments, and other attributes. Face tracking enables more visually appealing role-playing with add-on graphics.

Control describes a player's actions in the game world. We focused our efforts more on intuitive controls based on natural movement.

In all, the sense of presence, role-playing and intuitive control are varied aspects of the gaming experience that face-tracking technology can enhance.

Sense of Presence

Presence is a fundamental aspect of the gaming experience. Games with a high sense of presence are thought to be highly entertaining and more fun (greater enjoyment) [14]. A sense of presence may also facilitate players' game performances [14].

Real-world stimuli, or stimuli that is perceived as such, are likely to elicit a greater degree of attention compared to stimuli that are readily perceived as mediated presentations of real-world stimuli [18].

Video game face tracking can contribute to presence and stimuli in terms of identifying a vulnerable head area, hitting a bonus with one's head, and view-dependent controls through physical movements like leaning sideways or other body movements.

Role-playing

Players are interested in game characters because (1) they identify with the player; (2) they are interesting to the player, (3) the characters develop further as action occurs [6]. Other factors also affect how players get into a character. Indeed, props can be considered to be much more than just the outer appearance of a character. Costumes, hairstyles, jewelry and other props strongly affect body language [21].

Using add-on graphics superimposed on the face, players could visually take on identity and become involved as the game character. Compared to raw video, composited game characters are associated more with game context, and enable additional visual hints from changes in graphical props.

Control

In traditional video games, actions need to be mapped to a controller. In the real world, users have expectations of how their surrounding environment works. The game world should match such a model.

As Crawford pointed out, actions that are simple and obvious using other technologies become arcane with a computer [6]. He also gave an interesting example: Given a bat and told that the goal in baseball is to hit the ball, few would have problems deciding that swinging the bat at the ball should be their goal. A computer baseball game is not so easy to decipher, however. Should one hit H for "hit" or S for "swing" or B for FPS "bat"? Should one press the START key or press the joystick trigger? Without directions, the goal remains unclear.

Game heuristics, varying activities and controlling pace during game play minimizes a player's fatigue [10]. Face tracking allows for more varied activities because it tracks real-time face position. It is important to get the player involved quickly and with ease [10]. Interface with the player should be as non-intrusive as possible. Controls should be intuitive and mapped in a natural way [10].

PROTOTYPE SPECIFICATION

With the design focus proposed above, we created two game prototypes that used face tracking as inputs.

The Diver: Hitting Bonuses and Protecting Your Mask

The goal of this game prototype is to use face tracking to enhance sense of presence and role-playing.

An Oxygen mask is superimposed on the head of the player and continues to track and follow his face. The objective of the game is to prevent fish from hitting the Oxygen mask. Players punch the fish away to protect the mask and increase their score. They can also hit bonus oxygen hot-spots with their heads and can enhance their health. However, winning bonuses also exposes players to attacks.

In addition, event-driven emoticons are introduced into the game to stimulate emotional responses.

Technically, this two-dimensional (2D) game iteratively detects collisions of the sprite graphics of the fish and bonus (2D image or animation that is integrated into a larger scene) every 40 milliseconds with the projected face. It also detects collisions using the motion of the player. Game events such as being attacked, eliminating fish, and acquiring bonuses are then triggered by the collisions of the above two.

Fish Chasing Diver's Head

Instead of attacking the outline predefined in *Eyetoy* type of gesture games, fish in the Diver game will chase players' heads while players move. This is designed for better sense of presence: the system knows where one is projected in the game scene at any given moment. When a player escapes

out of the camera's field of view, a system message: "Don't be a wimp, Face it!" will be displayed on the screen to encourage the player go back to the game. At the same time, fish cruise back and forth since there is no target to attack.

Differentiate Head from Body

The head is the only area to be tracked and needs protection through the use of body motion (waving hands before fish can gain access to the diver's head). The bonus is acquired with player head movement.

Oxygen Mask

A virtual Oxygen mask is designed to be a game-related graphics add-on. It is super-imposed on player through the use of mirror video.

- Round-shaped: The mask cannot be rotated according to the orientation of a player's face, as this information is not available from the face-tracking module. The symmetrically radiated shape is visually reasonable no matter the orientation of a player's head.
- Status: When the player's head comes under attack, a glow accompanied by new cracks is rendered on the mask. This status information is designed for reminding the player to pay more attention to protecting himself.
- Visibility: The mask is rendered only when there is face detected in the mirrored video. If a player falls out of the camera scope, the mask will disappear along with the face.
- Fixed-size: The size of the mask cannot be adapted according to a player's distance to the camera. This information is not reliable from our current face-tracking algorithm. The mask graphic is 200 x 200 pixels in resolution and filled with radical gradient transparent color.
- Variations: The mask concept could be generalized with extra variations associated with specific game designs, such as a halo, a crown, or other items.

Emoticons

Animated emoticons are superimposed on video mirror images of the player to stimulate greater intensity in emotional responses and provide additional feedback (Figure 4). This is also designed for enhanced role-playing.

- Implementation: The emoticons are inferred on a relative scale of facial features (such as eyes and mouth) according to their position within a rectangle from the face-tracking module that marks the face's position. Animated emoticons are then rendered in those positions. The life span of the emoticons is a 3 second period.
- Animation: Animation makes the emoticons more expressive. Interestingly, from a technological standpoint, it is hard to always provide reasonably inferred positions, especially when the player goes back and forth toward the camera. Taking this into consideration; animated emoticons seem more tolerable on inferred positions compared to static ones with fixed sizes.
- Event-driven: Animated emoticons that express fear, excitement, and pride, that are superimposed on player's faces are spawned entirely by events, such as when a player is under attack, hits a bonus hotspot, or keeps his mask safe for a 20-second increment.

Fish

The mask or head of a player is like a magnet field, attracting big and small fish. Their attacks differ in that the former hit the mask and disappear while the latter will stick to the mask and keep attacking it for 5 seconds till body motions swipe them away.

Bonus Hotspots

Bonus hot spots are shown every 10 seconds at random places close to the margins of the screen lasting for 3 seconds. Their display encourages players jumping up or dropping down to retrieve the bonuses.

For research purposes, we also implemented two prototypes with the same game package: motion only and invisible mask. Those two are for testing the differences in presence and role-playing that face tracking could possibly make.

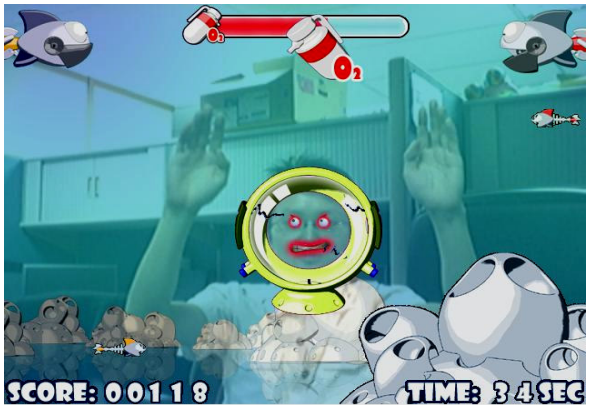

Figure 4. Screen shot of Diver with Protective Oxygen Mask on head from attacking and earning bonus Oxygen supply using head.

Bullet Time: Physical Dodging and Peeking in FPS Games

Players engaged in video games often exhibit physical behaviors not necessary for game control, such as leaning when steering during racing game or dodging bullets in shooting-related scenarios.

In these cases, upper body movement can be translated through head movements. Consequently, face tracking can be applied to reflect players' intentions.

Bullet Time is such a games prototype. It employs face tracking as an additional axis of control in a conventional

FPS game modality. Typical movements of the upper body cause head to roll (tilting right or left), yaw (turning right or left) and to pitch (looking up or down). We focused on rolling in this game prototype.

Technically, leaning left and right is mapped to a Direct 3D rendering camera movement in a 3D game scene. A player's physical dodges are captured through tracking his head position. These images are obtained in real-time and used to update the player's view. The player in the Direct 3D scene is modeled as a straight and rigid object. The bottom point is defined to be its center of rotation in the plane shown (Figure 5, left). When the player moves his head to the left, the horizontal movement of A is projected to segment B on the screen.

The movement path of the rendering camera in the plane (Figure 5, right) is a half circle. The view of the player is rotated by a certain angle when the player moves his head sideways (Figure 5, left). The rotation is linearly proportional to the amount of horizontal movement of the player's head.

The screen is divided into nine vertical strips for better control and head-tracking smoothness (Figure 5, right). This enhances the reliability and visual effects of physical dodging.

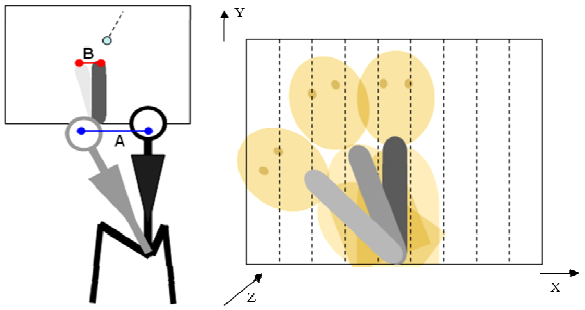

Figure 5. Mapping of physical dodging by Direct 3D camera movement.

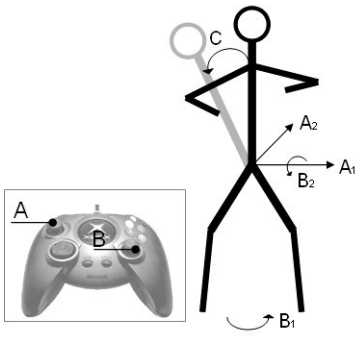

Figure 6. Physical dodging adds extra axis of control.

Control: Head as another Axis of Control

Although peeking and dodging are all forms of sideways movements, they differ in intention and purpose. The peeking action is initiated with the intention of checking out a hidden threat while the dodging action is believed to be unconscious feedback as the player undergoes attack.

Game pad Controls

- Conventional controls: Walk, perspective adjustment, and fire all remain the same. Given upper body without leaning, Bullet Time is just another standard FPS game (Figure 6). Left Joystick: Move left and right along the A1, i.e. X axis; move forward and backward along the A2, i.e. Z axis. Right Joystick: Aim using two rotation controls B1 and B2. Right Trigger: Fire. Left Trigger: Another axis of control is dodging with upper body movement C that results in a rotation about the A2, i.e. Z axis.
- View controls: Players' views of the game environment are controlled by three actions. Move (Left joystick): players' views move in the XY plane (Figure 5, right). Aim (the avatar head turning). Right joystick: a player's view rotates along the Y and X axes as the avatar turns his head around. Dodging (a player's upper body moves sideways): a player's view rotates along the Z axis when the player's upper body moves sideways.

Offset Value

Threshold: The range of movement on the left and right side spans is from 6.3 to 30 degrees. The player tilts right or left and does not start to see the view change until he rotates over 6.3 degrees.

Bullet

Movie-style exaggerated motion and slow-motion in video games such as Max Payne. The speed of bullet is about 14 pixel/sec (the coverage of the game scene is 585 x 1850 pixels), and the bullets are rendered with a trajectory that provides a visual hint for dodging.

Figure 7. Screen shot of Bullet Time.

IMPLEMENTATION

Both prototypes run on a 1.8GHz Pentium4 and 512M memory laptop with ATI Radeon 9800. Each is equipped with a USB webcam that is placed on the computer's monitor facing toward the player. The Bullet Time game also requires a USB game pad. DirectShow is used for camera processing and DirectX 9.0 is used for graphics.

The architecture of the two prototypes consists of two modules: camera process and game logic. The camera process module retrieves camera video from the webcam and provides face detection and tracking, and motion detection. The game logic module takes the input from extracted face and motion information, and the game pad, to use in game play, such as hitting the fish, or dodging the bullet.

Technology Overview

Three vision technologies were used in current video games: motion detection, face detection and object tracking. Our innovations are mainly on the latter two.

In order to know whether the player triggered the hotspot graphics, player's movements should be extracted from the live video stream as the foreground, and the rest is so-called background. The technology for determining which part of the video has motion is known as motion detection. We applied a robust algorithm: *non-parametric background subtraction* [4] to distinguish salient motions (e.g., from a player) from trivial motions (e.g., from a waving curtain behind a player).

When achieving a bonus or dodging the bullet, the game engine should know "where the player's head is." However, without extra knowledge, a computer cannot identify a human face from any other thing – both are a bulk of color pixels. The face-detecting process [20] is used to figure out a rectangle, which covers the face components from the image. The process can be divided into two stages:

(1) Learning whether the image is a face from thousands of face samples, and saving that knowledge is the criteria. Note that this stage is very slow (typically 4-5 hours) and thus can be made available forehand.

(2) Using the criteria to determine the proper rectangle that contains a face. This stage is much faster (0.1s, when resolution is 320 x 240 pixels) and can be done real-time.

In order to get the position of a face, a straightforward approach is used in face detection on each frame. However, the upper boundary of frame rates in this mode is 10 frames per second, which is inefficient for a video game. On the other hand, it is reasonable to see that the face-searching process would be much faster to use the last known position, under the assumption that the face will not move unreasonably fast. In fact, even in the most intense FPS games – those similar to "Bullet Time" – players would not likely move their head more than 4m/s. This technology is known as "Object Tracking" in the computer vision field. In our implementation, the optimized algorithm comprised by *Condensation Filter* [16] and *MeanShift* [7] enables the system to process a 320 x 240 pixels image in 0.016s, which is much faster than face detection.

In conclusion, we took advantage of both the accuracy of face detection and the efficiency of face tracking to retrieve information of player's face movements. The face detection is conducted every 15 frames (about 0.5s) to maintain accuracy. Object tracking is conducted continuously in each frame (about 0.03s) after the face is detected to provide the efficiency. Through this combination, the system can achieve better balance in both accuracy and efficiency, which provides a satisfying game experience.

System Performance and Constraints

Both the two prototypes run smoothly in the system configurations mentioned above. The system is robust: it could handle high-frequency noise such as screen flashes, and can adopt arbitrary background in motion detection. In terms of face tracking, the system can also handle variable poses, scale of face, rapid motion and partial occlusions.

On the other hand, the system has constraints: (1) The camera is required to be static. (2) Extreme illumination (too bright or too dark) should be avoided. (3) Avoid profile face. (4) Color and lighting should not change dramatically.

USER STUDIES

We conducted formal user studies for evaluating the Diver (Figure 8) and the Bullet Time game (Figure 9).

Participants

Volunteer participants included 36 people (32 males and 4 females) who were student interns from our lab, all majoring in computer-related fields. They ranged from 21 to 28 years of age, averaged 25. None of them had previous experience with these prototypes. None was familiar with camera-based games. All of them played video or computer games at least once a month. Eighteen rated themselves as veteran players of FPS games. They participated in return for a gift of a stopwatch.

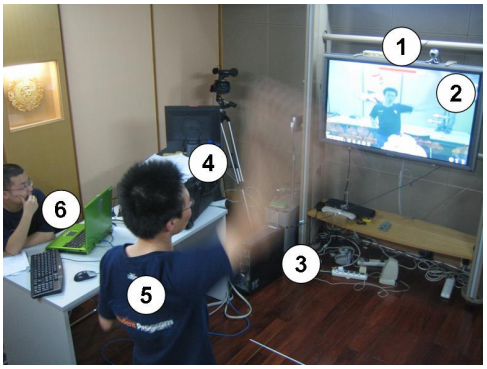

Figure 8. Test setup of Diver, showing camera (1), plasma display (2), computer with game (3), video taping system (4), player (5) and instructor (6).

Figure 9. Test setup of Bullet Time, showing camera (1), plasma display (2), computer with game (3), joystick (4) and player (5).

Comparisons

In order to evaluate face tracking as an augmented element to current motion-detection based computer and FPS games, we conducted three comparison tests (Table 2 and 3):

A versus B, B versus C and D versus E.

Prototype #	Motion detection	Face tracking	Mask and emoticons
A	Yes	No	No
B	Yes	Yes	No
C	Yes	Yes	Yes

Table 2. Prototype variations of the Diver game.

Prototype #	Control with Xbox game pad	Face tracking
D	Yes	No
E	Yes	Yes

Table 3. Prototype variations of the Bullet Time game.

C is the original *Diver* game. A, B were variations of C, sharing the same the game package and the same parameter values, including the number and speed of fish attackers, the life span of a bonus.

B is C without showing the mask and emoticons. A is B without face-tracking (motion-detection only).

Unlike A, B utilized face tracking. In turn, this leads to two differences between A and B:

(1) The attacked area: In A, it was the head area of a fixed human profile, which located at the center of the screen; In B, the area was the player's head on the screen. (2) The method of acquiring bonuses: In A, bonuses were acquired by motion; while in B, hitting the bonus with one's head.

A and B were compared to study how face tracking make a difference in presence.

The only difference of B and C was whether or not showing a mask and emoticons over the mirrored video of the player.

B and C were compared to demonstrate the effect of graphics superimposed on face areas in the sense of role-playing.

E is the original Bullet Time prototype. D is the same as E without the feature that allows users to peek and dodge using sideway movements. D represented the standard FPS games using conventional controls.

Hypothesis

There are three hypotheses in terms of presence, role-playing, and control:

Hypothesis 1 (H1): A versus B
Compared to motion detection, face tracking could effectively enhanced player's sense of presence through the given designs.

Hypothesis 2 (H2): B versus C
Game-related graphical add-ons (a mask and animated emoticons) could enhance role-playing by raising the levels of engagement and positive emotional response during game play.

Hypothesis 3 (H3): D versus E
Conventional physical control enhanced by face tracking could augment conventional FPS video games and provide a preferable hybrid control model.

Process

We conducted three independent tests. Each followed within-subject design. Each participant was asked to accomplish two tasks. The order effect was counterbalanced by switching the order of the tasks.

Each test started with a trial-and-error session. We used the "think aloud" method to uncover participants' understanding of the games as novice players. Participants were then given the rules and asked to finish the training session. Selected parameters were logged to reveal general trends on performance and to reflect playability.

We gathered participants' feedback on the prototypes by using a questionnaire and interview after each test. We also observed their behavioral patterns.

Experimental Quantitative Evaluation:

Measurements for H1 (A versus B)

The measurements for evaluating a player's sense of presence are: performance data (hit rate, earned bonus) and movement data (accumulated horizontal movements of the player's head). These measurements reflected how players

feel their existence in the game scene and how much they were engaged.

Measurements for H3 (D versus E)

The measurements for evaluating the controllability of the game are: performance data (hit rate, bullets used, time of task) and accumulated sideway time (the time period when the player's upper body was leaning larger than 6.3 degrees) These measurements can tell the likelihood that players apply head movements in game play, either as a natural reaction or as a new control axis.

Query Evaluation: Questionnaires

For H1, our questionnaire was designed based on ITC-Sense of Presence Inventory (ITC-SOPI) questionnaire to measure presence [13]. Items included valence, arousal, and involvement, with a 5-point Likert scale, rating from 1 (strongly disagree) to 5 (strongly agree).

For H2, the emotional response was adopted to evaluate the effects of add-on graphics. The qualitative data were adopted from previous work on evaluating emotional responses in video games [17]. Items include valence, arousal, joy, pleasant relaxation and depressed mood.

H3: Emotional responses like valence, arousal, presence, and involvement were measured in the comparison of D and E.

RESULT

The results confirmed the three hypotheses. Log data, questionnaires, interviews, and observations were all consistent and validated one another. The paired-sample T-test was the major statistics model adopted.

A versus B

All 12 participants preferred B. Participants' feedback included statements about their overwhelming acceptance being due to "The fish chasing me makes the game more realistic," "Hitting the bonuses with my head is more challenging than with body movement," and "B makes me feel more like I'm in the game."

The sense of presence of B was significantly better over A (mean: A=3.7955, B=4.1591, t=2.951, p<0.15, 2-tailed).

The sense of presence was measured by the combination of four items: the valence of being in the game, the intensity of the emotional responses, and the involvement and joy.

Players failed more in A to protect the attacked area (t=3.484, p<0.006, 2-tailed). This result validated that participants were more aware of the vulnerable head area in B. It reflected that sense of presence in B was significantly larger than A.

B encourages participants to acquire significantly more bonus points, with their heads rather than their body motion (t=-6.314, p<0.001, 2-tailed).

In B, participants' accumulated horizontal movement was significantly larger than that in A (t=-3.362, p<0.007, 2-tailed).

B versus C

Ten out of 12 participants (83.33%) preferred C. Typical feedback includes, "I felt more like a diver during play," "The mask shows the status and it helps," and "I associated more with the game scenario."

C and B were significantly differentiated by joy ratings; Players got more joy in C (t=2.872, p<0.015, 2-tailed). Although C caused more frustration than B, the difference was not significant. With this context, we concluded that players could have more positive emotional responses with the add-on graphics, such as masks and emoticons.

D versus E

All participants wanted to have the option of the augmented control.

Three of them (21.4%) preferred to have it on all the time, 11 others (78.6%) preferred to make it optional. Some reported that it was easy to learn but takes time to master. That could be the main reason for the hesitation in completely accepting the control.

The time duration of upper body sideways movement was not significantly different in both modes (t=1.228, p<0.241). This was quantitative evidence that players subconsciously moved their upper body regardless of the input method.

With the face control on, there was no significant difference found in player's performance between D and E. Player's performances included the number of bullets used, life losses and completion times. The average bullet number and life lost in E were less than those in D (t=-0.855, p=0.408 and t=-1.149, p=0.271). For the average completion times, E was longer than in D (t=0.792, p=0.442). This result proved that this new input axis was intuitive because it did not harm user's performance when participants play with it for the first time.

The sense of presence in E was significantly higher than D (t=4.660, p<0.001). This result came from the combination of two factors ("I am in the game scene," and "I am involved in the game"): On average, self-evaluation rates for sense of presence were 3.93 for D and 3.32 for E.

SUMMARY

Augmenting current camera-based games with face tracking is a promising approach to enhance the sense of presence, role-playing and intuitive nature of controls.

Face-tracking technology can be readily combined into current game systems and the potential of the hybrid controls could have a big impact on the game experience.

Although the face-tracking algorithms mentioned in this paper are based on the patterns abstracted from the face, the information of facial features and facial expressions cannot be provided. For most of the cases, effect of face tracking mentioned is equal to "head tracking." This leaves challenges for both vision technology and game design.

FUTURE WORK

For the presence of the player, it would be interesting to study the scenario of multiple side-by-side players with each holding a unique identity. This could significantly evolve the "playability" of camera-based games.

For control, physical sideways peeking and dodging actions inspire a new integration model of physical presence into video games.

Face tracking is based on 2D image sequencing, and thus the third dimension: the precise distance between player and camera is unavailable. Once take this dimension into account, the player will influence the game world in real 3D. A study around this aspect would report interesting findings.

ACKNOWLEDGMENTS

This work has been greatly supported by Visual Computing and UI group of the lab. The authors would like to thank Dave Vronay, Jian Wang, Rong Xiao, Rania Ho, Xiang Cao, Dwight Daniels, Leizhong Zhang, Neema Moraveji, Zhuohao Wu, Haoyue Yu, Wenli Zhu and all who have played the games and helped with their comments.

REFERENCES

1. Eyetoy games developed by Sega. http://www.sega.com/gamesite/segasuperstars/.
2. GestureTek, http://www.gesturetek.com/.
3. Sony Eye-toy, http://www.eyetoy.com/.
4. A. Elgammal, R. Duraiswami, D. Harwood, and L. S. Davis, Background and foreground modeling using nonparametric kernel density estimation for visual surveillance, *Proc. of the IEEE*, 90 (2002), 1151-1163.
5. B.G., & Singer, M.J., Measuring presence in virtual environments: A presence questionnaire. *Presence: Teleoperators and Virtual Environment, 7* (1998).
6. Chris Crawford, *the Art of Computer Game Design*. http://vancouver.wsu.edu/fac/peabody/game-book/
7. D. Comaniciu, V. Ramesh, and P. Meer, Kernel-based Object Tracking, IEEE Trans. *Pattern Anal. Machine Intell., 25* (2003), 564-577.
8. D. Gorodnichy. On importance of nose for face tracking, *Proc. IEEE Intern. Conf. on Automatic Face and Gesture Recognition* (2002).
9. Gary R. Bradski, Computer Vision Face Tracking for Use in a Perceptual User Interface, *Intel Technology Journal* (1998).
10. Heather Desurvire, Martin Caplan, Jozsef A. Toth, Using heuristics to evaluate the playability of games, *CHI* (2005), 1511.
11. Herman D'Hooge and Michael Goldsmith, Game Design Principles for the Intel® Play™ Me2Cam Virtual Game System, *Intel Technology Journal* (2001). http://www.intel.com/technology/itj/archive/2001.htm.
12. Krueger, M., Gionfriddo, T., Hinrichsen, K. VIDEOPLACE: An Artificial Reality, *CHI* (2004), 35-40.
13. Lessiter, J., Freeman, J., Keogh, E., and Davidoff, J. A Cross-Media Presence Questionnaire: The ITC-Sense of Presence Inventory. *Presence: Teleoperators & Virtual Environments, 10* (2001), 282-298.
14. Lombard, M. and Ditton, T. At the heart of it all: The concept of presence. *Journal of Computer Mediated Communication* (1997).
15. Marian Stewart Bartlett, Gwen Littlewort, Ian Fasel, Javier R. Movellan, Real Time Face Detection and Facial Expression Recognition: Development and Applications to Human Computer Interaction. *Conference on Computer Vision and Pattern Recognition Workshop, 5* (2003), 53.
16. M. Isard and A. Blake, Condensation: Conditional Density Propagation for Visual Tracking, *Intern. Journal of Computer Vision* (1998), 5-28.
17. Niklas Ravaja et. al, Emotional Response Patterns and Sense of Presence during Video Games: Potential Criterion Variables for Game Design, *NordiCHI* (2004), 339-347
18. Pugnetti, L., Meehan, M., and Mendozzi, L. Psychophysiological correlates of virtual reality: A review. *Presence: Teleoperators & Virtual Environments, 10* (2001), 384-400.
19. P. Viola and M. Jones, Robust Real-Time Face Detection, *Intern. Journal of Computer Vision* (2004), 137-154.
20. R. Xiao, L. Zhu, and H. Zhang, Boosting Chain Learning for Object Detection, *ICCV* (2003), 709-715
21. Terhi Säilä, Beyond Role and Play Tools, Toys and Theory for Harnessing the Imagination, http://www.ropecon.fi/brap/.
22. The ALIVE system: wireless, full-body interaction with autonomous agents, Pattie Maes, Trevor Darrell, Bruce Blumberg, Alex Pentland, *Multimedia Systems, 5* (1997), 105-112.
23. W. Grimson and C. Stauffer, Adaptive background mixture models for real-time tracking, *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, *1*(1999), 22-29.