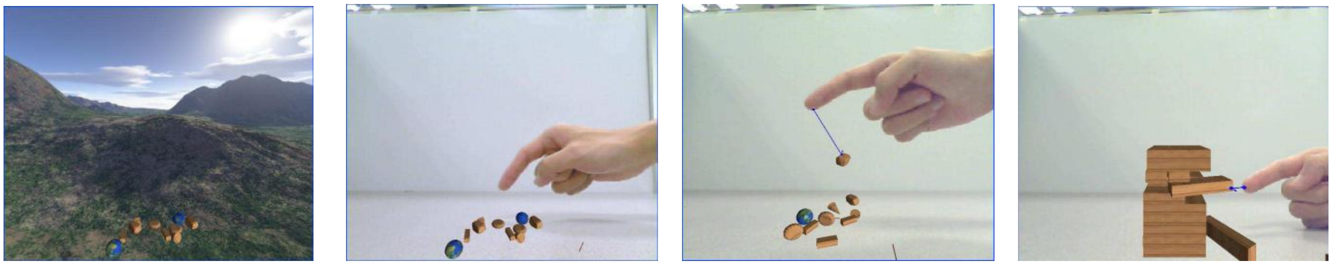
Vision-based 3D Finger Interactions for Mixed Reality Games with Physics Simulation

Peng Song[1], Hang Yu[1] and Stefan Winkler[2]

[1] *Institute for Infocomm Research, Singapore*
[2] *Symmetricom Inc.*



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

Fig. 1. A Virtual Reality game and its counterpart Mixed Reality games based 3D Finger Interactions as implemented. A virtual reality game background with modeled space is shown in figure (a), while in figures (b-d) a similar mixed reality game is shown with 3D finger interactions.

Abstract—**Mixed reality applications can provide users with enhanced interaction experiences by integrating virtual and real world objects in a mixed environment. Through the mixed reality interface, a more realistic and immersive control style is achieved compared to the traditional keyboard and mouse input devices. The interface proposed in this paper consists of a stereo camera, which tracks the user's hands and fingers robustly and accurately in the 3D space. To enable a physically realistic experience in the interaction, a physics engine is adopted for the simulating the physics of virtual object manipulation. The objects can be picked up and tossed with physical characteristics, such as gravity and collisions which occur in the real world. Detection and interaction in our system is fully computer-vision based, without any markers or additional sensors. We demonstrate this gesture-based interface using two mixed reality game implementations: finger fishing, in which a player can simulate fishing for virtual objects with his/her fingers as in a real environment, and Jenga, which is a simulation of the well-known tower building game. A user study is conducted and reported to demonstrate the accuracy, effectiveness and comfort of using this interactive interface.**

Index Terms—**finger interaction, finger tracking, mixed reality, physics simulation**

I. INTRODUCTION

With the development of highly advanced technologies, computerbased entertainment systems make extensive use of multimedia elements, such as graphics and sound, which help to create a stimulating atmosphere. However the interactions provided in these applications are usually constrained by standard input devices, such as keyboard, mouse and joystick, which lack the intuitive and natural interactions with real environments.

Mixed reality is a possible solution to this problem, as it creates an environment that supports interactions with both virtual and real objects. With this technique, players are able to interact with virtual and real information in the same environment for a more immersive experience. To achieve this, physical and virtual world need to be seamlessly merged through the accurate registration between the image captured by a camera and the virtual world, which allows us to place virtual objects on top of real objects when required. Fig. 1 gives an example of Virtual Reality game scene with comparison to its counterpart Mixed reality game scene. A finger fishing game scenario created in a virtual world is shown in Fig. 1(a). A mixed reality game scene with the camera captured image registered well with virtual objects is shown in Fig. 1(b). One of the most important issues in mixed reality systems are the interfaces they can provide. According to the work by Wanderley et al. [22], the interfaces can be classified into the following three categories:

1. *Classical Interfaces* usually adopt certain tracking devices, such as gloves, wands and infrared sensors. Typical examples inlcude ARQuake [17] and Human Pacman [4] using wearable computers and positioning devices such as Head-Mounted Displays (HMDs) and GPStracked laptops. Although the interfaces offer a practical and direct way of interactions to users, they are not only expensive but also intrusive as cables and sensors have to be carried or worn by the users.

2. *Tangible Interfaces* make use of graspable physical objects to represent and control digital models. The virtual objects are manipulated through a one-to-one mapping relation to the physical metaphors or background objects. Motivated by the project of Tangible Bits [8], tangible interfaces excel in that

Manuscript received on 30 June 2008.
E-Mail:psong@i2r.a-star.edu.sg.

users can feel the touch, weight and control of tangible objects. More recent examples include mulTetris [1] which uses a graspable interface to manipulate bricks in the traditional Tetris game, and MRI [20] which is a novel input device that provides control of virtual targets through the operation on real-world objects. A further extension from physical object metaphors is the use of markers attached to any background objects. Through vision-based tracking, virtual objects can be displayed on top of these markers. Examples include Magic-Book [2], a mixed reality game displaying virtual pictures on physical books, and the Harbour Game [11], an urban planning game for harbour areas on a physical board with markers. These interfaces provide user-friendly interactions that people without computer knowledge can handle well. However, the physical metaphors may not be natural enough, and the markers must be specially designed for calibration and tracking purpose.

3. *Bare-hand Interfaces* rely on the hands of the user for interaction. Compared to the above two, bare-hand interfaces [6, 22] are more natural as hands are commonly used in real life. The interactions are achieved by the detection of the bare hands of users without wearing any devices. A great advantage is that it provides a more natural and simple control style. Predesign of artifacts such as markers is also not necessary using these methods.

The interactions in the above mentioned mixed reality games are mostly pre-defined, *i.e.* the player will always have the same experience for a specific interaction in the game. The recent trend is that real-time physics simulation was added into mixed reality games to enhance the realism in the interactions. Lee *et al.* 23] presented a tangible interface using a real-time 3D avatar. The avatar is created using 3D visual hull reconstruction algorithm and integrated with physics simulation for full body interaction with virtual objects.

Based on the above analysis, in this paper we proposed a 3D finger based interaction interface for mixed reality games with physical simulation. The interface is implemented with an improved robust and accurate vision-based finger tracking technique, thus all the operations are performed by finger gestures. The interface consists of a stereo camera to track the 3D location and direction of a user’s fingers. To provide additional realism to the manipulation of the virtual objects, a physics engine is also well integrated into the system to handle the physics-based interactions. When the user manipulates a virtual object with his finger, gravity and collision effects are simulated. If the object is released by the user, it will fall under gravity and collide with other objects as what would happen in the real world. To study these bare-hand interactions in the mixed reality space, we have applied our interaction techniques to two games: finger fishing and Jenga. The former is a fishing simulation game purely based on the player’s finger gestures, while the latter is based on the original Jenga, a well-known tower building game using a collection of wooden blocks. We have also conducted a user study to report on the accuracy, effectiveness and comfort of using our interface.

In the next section, the basic system design will be presented. The robust 3D finger tracking algorithms will be discussed in Section 3, followed by the design of interactions in Section 4. The userability study results will be discussed in Section 5. We will discuss the future works and the conclusions in the last section.

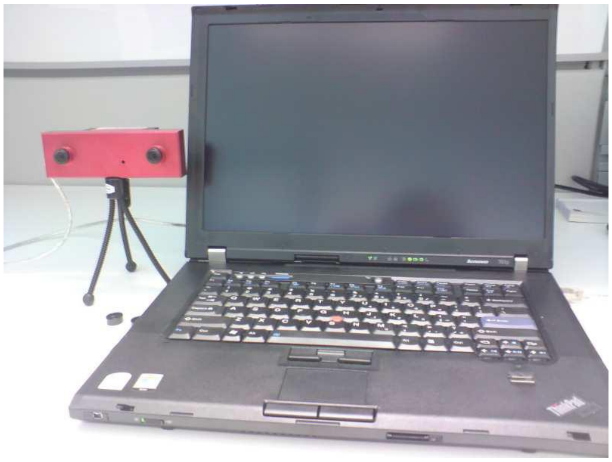


Fig. 2. The basic system setup.

II. System Design

We proposed a system for 3D finger interactions with virtual objects. This is motivated by the fact that in real life, we manipulate and interact with objects using our hands and fingers. The barehand interactions are implemented with the passive vision-based approach without attaching any extra intrusive devices to the users. This system can be used for mixed reality games, *i.e.* the game player is presented with virtual objects embedded in the physical world. Thus there is no requirement for a detailed model of the game space. To enrich the experience of the interactions, the mixed reality games are integrated with physics engine to simulate the real experience in our physical world.

The system is comprised of an lenovo T61p laptop, a STOC stereo camera from Videre Design which is shown in Fig. 2. A clean background in the camera view is desired for accurate tracking of finger tips. However, it is not strictly necessary for the system setup. The provided interactions between hand and virtual objects can be carried out within the view of the stereo camera.

The proposed system is comprised of 3 components: video input, finger tracking and finger interactions. The video input is provided by the stereo camera. In the following sections, finger tracking and interactions will be described respectively.

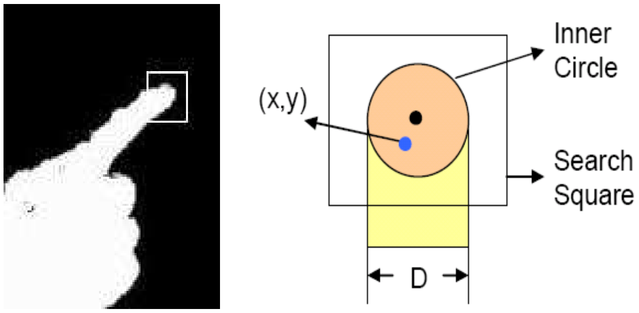
III. Finger tracking

Because in the mixed reality games we provide finger interactions with the virtual objects, the finger of the user needs to be tracked in the camera view such that interactions can be enabled accordingly. There are many vision-based techniques that can track fingers. However, there are many constraints on these methods: methods based on color segmentation [9] need users to wear colored gloves for efficient detection; wavelet-based methods [19] are computationally expensive and non real-time; contour based methods [15] work only on restricted backgrounds; infrared segmentation based methods [13, 14] require expensive infrared cameras; correlation-based methods [5] require an explicit setup stage before the tracking starts; the blob model based method [8] imposes restrictions on the maximum speed of hand movements. In order to provide robust real-time hand and finger tracking in the presence of rapid hand movements and without the need of initial setup stage, we adopted an improved finger tracking from our previous paper [16] based on Hardenberg's fingertip shape detection method [6] with more robustness and accuracy.

3.1 Fingertip Shape Detection

Hardenberg [6] proposed a finger tracking algorithm using a single camera. With its smart image differencing, the user's hand can be easily distinguished from the static background. Fingertips are then needed to be detected for interaction purposes.

In a difference image, the hand is represented in filled pixels, while the background pixels are all unfilled, as shown in Figure 3(a). If a square box is shown around a fingertip, as in Figure 3(b), the fingertip shape is formed by circles of linked pixels around location (*x; y*), a long chain of non-filled pixels, and a short chain of filled pixels.



(a) (b)

Fig. 3. Fingertip Shape Detection. (a) Difference image shows the fingertip shape of filled pixels on a background of non-filled pixels. (b) The fingertip at position (x; y) can be detected by searching in a square box (see text).

In order to identify the fingertip shape around pixel location (*x; y*), 3 criteria have to be satisfied:

1. In the close neighborhood of position (*x; y*), there have to be enough filled pixels within the searching square;

2. The number of filled pixels has to be less than that of the non-filled pixels within the searching square;

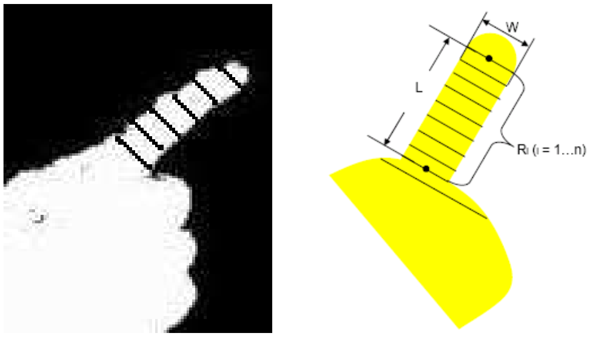
3. The filled pixels within the searching square have to be connected in a chain.

The width of the chain of filled pixels can be expressed as *D* and is the diameter of the identified finger. *D* needs to be preset/adjusted in order to detect fingers of different diameters in the camera view.

This method works well under normal lighting conditions, but not robust enough. Sometimes it may even lose track. In order to detect the fingertips more robustly, a finger shape detection method is proposed in the following section.

3.2 Finger Shape Detection

The above detection method may produce false detections on the finger end in connection with one's palm, as shown in Fig. 4(a), because of its similar shape. However, these false detections can be eliminated if the shape of the finger is taken into consideration. As a fingertip always has a long chain of filled pixels connected to the palm, and the width of the chain of filled pixels will be greatly changed only at the connection from the finger to the palm, a more robust finger detection algorithm based on the shape of the finger is proposed as follows.



(a) (b)

Fig. 4. Finger Shape Detection. (a) Difference image showing the finger shape of a long chain of filled pixels connected with the palm's chain of filled pixels. The finger is labeled by rows of black pixels orthogonal to the direction of the chain of filled pixels detected in Hardenberg's method (b) A finger can be represented by a long chain of filled pixels. Along the finger, the width W of chain of filled pixels orthogonal to the direction of the detected chain of filled pixels will change dramatically at the connection of a finger to its palm.

When a fingertip is detected from the method used in Section 3.1, record the center of the fingertip, move further along the direction of the chains of filled pixels. As shown in Figure 4(b), the width *Wi* (*i* = 1 ... *n*) of the i^{th} row of filled pixels (*Ri* (*i* = 1 ... *n*)) orthogonal to the direction of the detected chain of filled pixels is computed. If the width of row *Wi*+1 is comparable with *Wi*, move along the direction of the chain of filled pixels, until the width increase between row *Wn* and *Wn*-1 increases dramatically.

The length of the potential finger *L* can be derived by computing the distance between the center pixel of row *R*1 and that of row *Rn*-1. A finger is confirmed if *L* is sufficiently long, since false detected fingertips normally do not have long fingers attached.

Through employing the finger information, his algorithm effectively eliminates the false fingertip detections based on the fingertip shape information.

3.3 Finger Detection in 3D Space

The localization of fingers in 3D space can be achieved through stereo triangulation based on the 2D finger tracking results derived from our proposed 2D finger detection approach. The stereo triangulation can be implemented using a stereo camera which is straightforward, and will be discussed here.

IV. INTERACTIONS

With the robust tracking of fingers in the 3D space, we then further integrate this interface with a physics engine to enable interactions with more realistic virtual object movements in mixed reality games. Two mixed reality games were implemented using our 3D finger tracking methods together with physics engine: finger fishing and Jenga. In these two applications, players can freely use their fingertips to interact with virtual objects. The physical characteristics for the movement of virtual objects are simulated with rigid body dynamics, collision detection and inertial forces. When a virtual object is released from the finger of the player, it will move according to the simulated gravitational and inertial forces. If a virtual object collides with any other object, collision simulation will be performed.

4.1 Physics Engine

A physics engine is a software often used in computer games and animations to simulate Newtonian physical models and phenomena, such as gravity, collision and friction. The objective of using a physics engine in our application is to enhance the degree of realism by simulating the behavior of objects in the physical world.

There are several well-known physics engines, such as Bullet [3], ODE [12], Havok [7], Newton Game Dynamics [10], Tokamak [1], etc. To demonstrate the realism added to the user exprience, we choose Newton Game Dynamics, an open-source engine to be integrated with the above mentioned two mixed reality games.

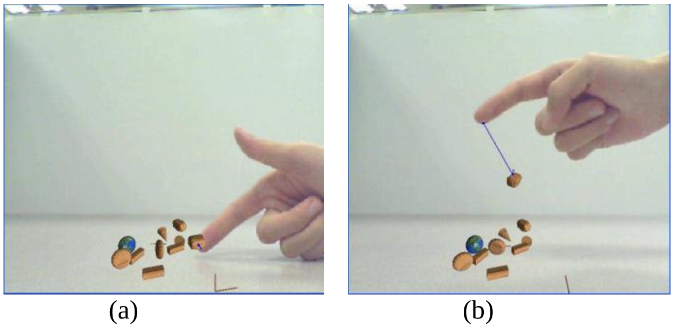


Fig. 5. Interactions in the Finger Fishing Game. An index finger selecting a virtual fishing target with the confirmation of a thumb stretched out is shown in (a). After the target is selected, the player can pull his finger "pole" freely from which the target is hooked by a blue elastic rope, as shown in (b).

4.2 Finger Fishing

A fishing simulation game controlled by finger gestures is shown in here. The game scenario is depicted in Fig. 5. A collection of geometrical objects is distributed randomly on a virtual plane. These objects represent the fishing targets and have different shapes and sizes. The player can use his index finger as the fishing pole. The interaction is simple and intuitive: a player catches the "fish" using his index fingertip, then pulls back his finger "pole" to get the "fish" out of water. The scores are calculated based on the difficulty of the fishing. The smaller the target is and the more irregular its shape is, the harder the fishing, and thus the higher the score if the fishing is successful.

4.2.1 Selecting a Target

The gesture for selecting a target consists of two steps. In the first step, the player stretches out the forefinger to hover above the targets area. By touching an object with his index finger he can pick the target for fishing. In the second step, the selection is confirmed by stretching out his thumb naturally. A small blue patch will appear on the target confirming the selection. Figure 5(a) shows the step of picking a target.

4.2.2 Pulling the Pole

After the target is selected, the player can move his finger freely to simulate the operation for pulling the pole. This gesture is displayed in figure 5(b). A blue elastic rope is shown connecting the fingertip and the selected virtual object to simulate the fishing line. The "catch" can be released if the player bunches his fingers into a fist gesture.

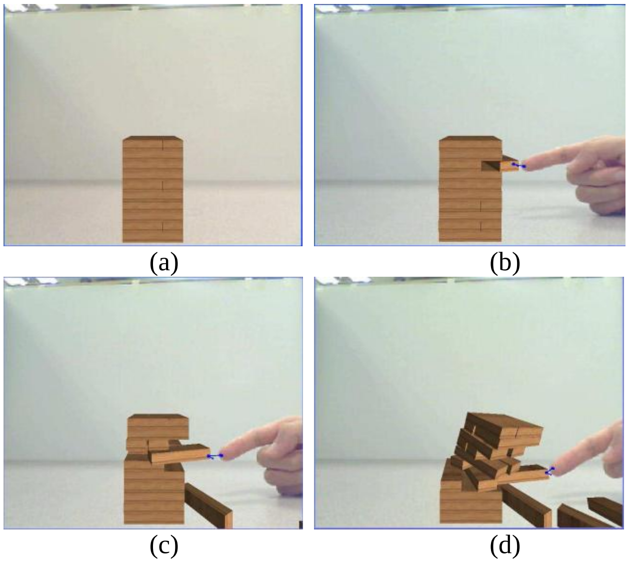


Fig. 6. Moving blocks in Jenga. A virtual tower of building blocks is shown in (a); Blocks has been selected by the player's index fingertip and pulled out of the tower, as shown in (b) and (c) respectively; The tower collapses when the player pulls out the last "balancing" block, as shown in (d).

4.3 Jenga

The original Jenga is a well-known game in which players have to build a tower using a collection of wooden blocks. The players are required to extract blocks from the tower and put them on top without toppling the tower. The one who makes the tower collapse loses. In our simplified version of Jenga game, the player only needs to extract blocks from a tower using his index finger. The game ends when the player pulls out the last "balancing" block and the tower collapses. The interactions are very similar to those introduced in the finger fishing game.

4.3.1 Selecting a Building Block

Similar to the action of selecting a fishing target in the finger fishing game, in this game the user also needs to stick out his index finger to touch any building block first. After that, by stretching out his thumb, the selected is confirmed.

4.3.2 Pulling the Building Block

When a building block is selected, the player can then pulls his index finger to drag the building block out of the tower, as shown in Fig. 6(b, c). The direction and speed of the dragging may affect the tower structures, according to physics. Whenever the tower loses balance and collapses, the game will end, as shown in Fig. 6(d).

V. USER STUDIES

We conducted user studies on 57 people (38 males and 19 females) with their age ranging from 14 to 34 years old. Most of them played games occasionally and have played computer games in the past 5 years. Before answering our questionnaire, the users were asked to play two rounds of Fishing/Jenga game. The first round was conducted using our finger interaction interface, and the second round was conducted using the traditional mouse and keyboard interface. The mouse click corresponds to the selection of targets, while the mouse click and drag corresponds to the pulling of the targets. When the mouse click is released, the target will be released as well.

5.1 Subjective Tests

In the subjective tests, users were asked with a few questions and some of the results are selected to be shown here. The users were asked to rate mouse/keyboard interactions vs. finger interactions in terms of excitement/fun and accuracy, on a scale of -3 to 3 with 0 as neutral. The result is shown in Fig. 7 which clearly demonstrates that finger interactions provide more fun but less accuracy.

Then the users were asked to rate mouse/keyboard interactions vs. finger interactions in terms of comfort and difficulty, similarly on a scale of -3 to 3 with 0 as neutral. The result is shown in Fig. 8 which indicates that finger interactions can provide a bit more comfort but roughly the same difficulty.

In the end, we asked the users to choose the interface that best simulates the Fishing/Jenga game. Our finger interactions were highly preferred as indicated in figure 9.

5.2 Objective Tests

The aim of our objective tests is to measure unbiased performance of the finger-based interface with the mouse/keyboard interface. The time for a user to finish the game is measured. The users' timing result is shown in Fig. 10, with a mean of 53.31 seconds for the mouse/keyboard interface and 73.82 seconds for the finger-based interface. The fastest time is 17 seconds for the mouse/keyboard interface and 36.5 seconds for the finger-based interface. Although the mouse/keyboard is the clear winner, it simplifies the real world problem significantly for most cases.

VI. CONCLUSIONS AND FUTURE WORK

In this paper, we proposed a 3D finger gesture based interface for mixed reality applications. The interface is controlled by the user's bare hands. It is implemented using our enhanced finger detection algorithm in real time, with an extension from 2D to 3D space. The capabilities of the interface were demonstrated through two mixed reality games: finger fishing and Jenga. In these two games, we have designed interactions for the player to manipulate objects in a mixed reality environment. Physics simulation using the Newton Game Dynamics engine enables more realistic behaviors of the virtual objects and user experience. Compared to traditional computer input devices, such as mice and keyboards, the proposed interface provides players a more natural and immersive experience. A user study was conducted and reported to show the effectiveness, accuracy and comfort using our interface.

The finger detection algorithm implemented is only based on the shape characteristics of the fingertips. This can still cause false detection if there are similar finger shape objects in the physical background of the scene. By incorporating more features in the finger detection algorithm, such as skin color, the white background requirement in the system setup can be less stringent. Additionally, the current algorithm have limited accuracy and robustness in the multiple hands tracking. With an improved multiple hands and fingers tracking algorithm, more versatile interactions will be introduced into

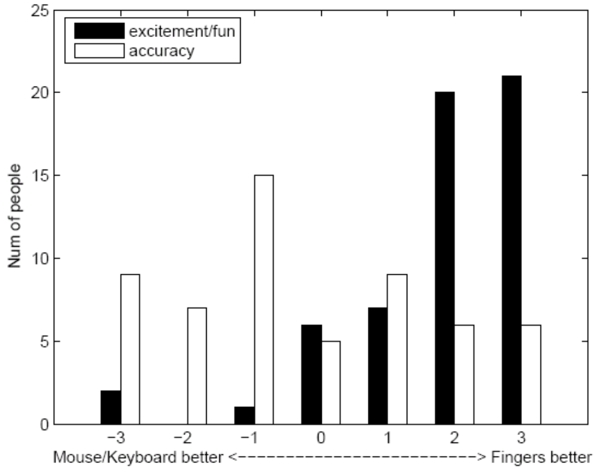


Fig. 7. Test result for mouse Vs. fingers in terms of excitement/fun and accuracy.

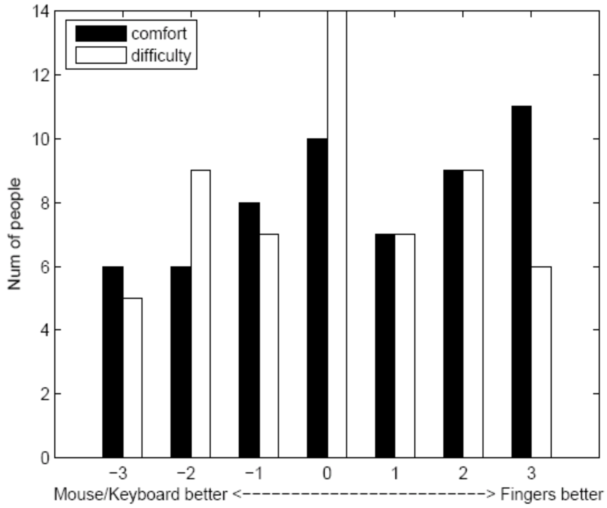


Fig. 8. Test result for mouse Vs. fingers in terms of comfort and difficulty.

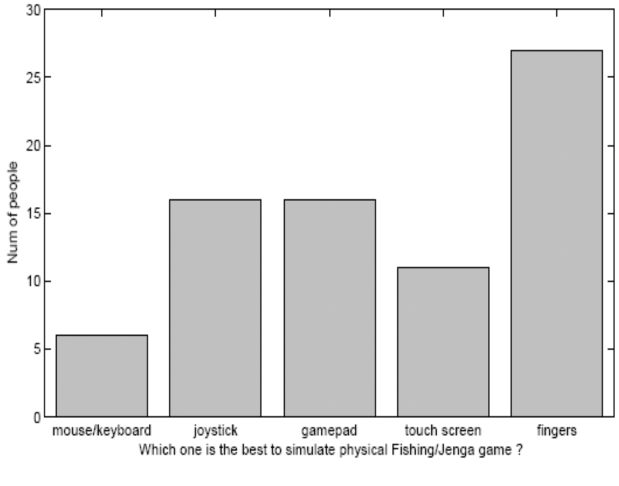


Fig. 9. Test result: which one of the following do you think best simulates the Fishing/Jenga Game?

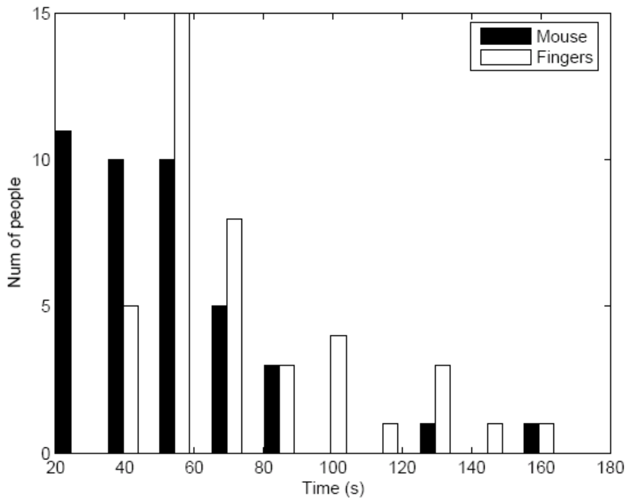


Fig. 10. User’s Timing Performance Result

our interface, such as multi-hand cooperation in Mikado [24] or multi-player competition.

REFERENCES

[1] S. Audet, M. Bedrosian, C. ClementL and M. Dinculescu, MulTetris: A test of graspable user interfaces in collaborative games. *Course project, McGill University, Canada*, 2006

[2] M. Billinghurst, H. Kato, and I. Poupyrev. 2001. The MagicBook: Moving seamlessly between reality and virtuality. *IEEE Comput. Graph. Appl.* 21, 3, 6–8,2001

[3] Bullet. Bullet continuous collision detection and physics library. http://www.continuousphysics.com/Bullet/.

[4] A. D. Cheok, K. H. Goh, W. Liu, F. Farbiz, S. W. Fong, S. L. Teo,Y. Li and X.Yang. Human pacman: a mobile, wide-area entertainment system based on physical, social, and ubiquitous computing. *Personal Ubiquitous Comput*. 8, 2, 71–81,2004

[5] J. Crowley, F. B ́erard and J. Coutaz. Finger tracking as an input device for augmented reality. In Proc. *International Conference on Automatic Face and Gesture Recognition*,1995

[6] C. Hardenberg and F. Brard. Bare-hand human computer interaction. *Proc. Perceptual User Interfaces*,2001

[7] Havok. Havok physics. http://www.havok.com/.

[8] H. Ishii,and B.Ullmer. Tangible bits: Towards seamless interfaces between people, bits and atoms. In Proceedings of CHI’97, 234–241,1997. LAPTEV, I., AND LINDEBERG, T. 2000.

[9] Tracking of multi-state hand models using particle filtering and a hierarchy of multiscale image features. In *Technical Report ISRN KTH/NA/P- 00/12-SE,* The Royal Institute of Technology (KTH),*1997*

[10] C. Lien and C. Huang. Model-based articulated hand motion tracking for gesture recognition. *Image and Vision Computing* 16, 2, 121–134,1998

[11] Newton Physics Engine. A free win32 physics engine. http://www.physicsengine.com/.

[12] R. Nielsen, T.F. Delman and T. Lossing. A mixed reality game for urban planning. In *Proc Computers in Urban Planning and Urban Management*,2005.

[13] ODE. Open dynamics engine. http://www.ode.org/.

[14] J. Rehg and T. Kanade. Digiteyes: Vision-based 3D human hand tracking. In *Technical Report CMU-CS-93-220*,1993

[15] Y. Sato, Y. Kobayashi and H. Koike. Fast tracking of hands and fingertips in infrared images for augmented desk interface. In Proc. *International Conference on Automatic Face and Gesture Recognition*,2000

[16] J. Segen. Gesture VR: Vision-based 3D hand interface for spatial interaction. In *Proc. ACM Multimedia Conference*,1998

[17] P. Song, S. Winkler, S. Gilani and Z. Zhou. Vision-based projected tabletop interface for finger interactions. *In IEEE International Workshop on Human Computer Interaction* (HCI) 2007, 49–58,2007

[18] B. Thonmas, B. Close, J. Donoghue, J. Squires, P.D. Bondi and W. Piekarski. First person indoor/outdoor augmented reality application: *Arquake. Personal and Ubiquitous Computing* 6, 1, 75–86,2000

[19] Tokamak Game Physics. http://www.tokamakphysics.com/.

[20] J. Trisch and C. Malsburg. Robust classification of hand postures against complex background. *In Proc. International Conference On Automatic Face and Gesture Recognition*,1996

[21] P. Uray, D.T. Kienzl, and D.U. Marsche. MRI: a mixed reality interface for the masses. *In ACM SIGGRAPH Emerging technologies*,2006

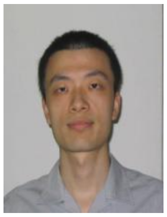
[22] I. Wanderley, J. Kelner, N. Costa and V. Teichrieb. A survey of interaction in mixed reality systems. *In Symposium on Virtual Reality,* 1–4,2006

[23] S. Winkler, H. Yu and Z.Y. Zhou. Tangible mixed reality desktop for digital media management. *In SPIE Engineering Reality of Virtual Reality, vol. 6490B*,2007

[24] S.Y. Lee, I.J. Kim and S.C. Ahn. Real-time 3d video avatar for tangible interface. *TSI workshop*,2006

[25] MIKADO GAME. http://www.allwag.co.uk/detail861700 Mikado-Game.aspx.

Peng Song received his B.Eng and B.A. degrees in Computer Engineering and English Language respectively, from Tianjin University, China in 2002, and his Ph.D degree in Computer Engineering from Nanyang Technological University in 2007. His research interests are in computer vision, graphics, human-computer interaction, and projector-camera systems. He has been working as a Research Fellow in National University of Singapore, and now is working as a Research Fellow in the Institute for Infocomm Research, Singapore. He was awarded the Best Paper Award in the 3rd IEEE Workshop on Projector-Camera Systems in 2005.



Hang Yu received the B.Eng. and M.Eng. degrees in refrigeration and cryogenics engineering from Shanghai Jiao Tong University, Shanghai, China, in 1998 and 2001, respectively, and the Ph.D degree from National University of Singapore in 2007. His research topics are on image and media processing, volume graphics. Currently, he is a Research Fellow with the Institute for Infocomm Research, Singapore. His research interest is on computer graphics.

Stefan Winkler holds an M.Sc. degree in Electrical Engineering from the University of Technology in Vienna, Austria, and a Ph.D. degree from the Ecole Polytechnique F ́ed ́erale de Lausanne (EPFL), Switzerland. Dr. Winkler is currently Principal Technologist for Symmetricom’s QoE Assurance Division. Prior to that, he was Chief Scientist of Genista Corporation, which he co-founded in 2001. He has also held assistant professor positions at the National University of Singapore (NUS) and the University of Lausanne, Switzerland. Dr. Winkler has published more than 50 papers and is the author of the book “Digital Video Quality.” His interests include visual perception, media quality, computer vision, and human-computer interaction.