ARForce: A Marker-based Augmented Reality System for Force Distribution Input

Kensei Jo
The University of Tokyo
7-3-1 Hongo, Bunkyo-ku
Tokyo, Japan
Kensei_Jo@ipc.i.u-tokyo.ac.jp

Yasuaki Kakehi
Keio University
5322 Endo, Fujisawa-shi
Kanagawa, Japan
ykakehi@sfc.keio.ac.jp

Kouta Minamizawa
The University of Tokyo
7-3-1 Hongo, Bunkyo-ku
Tokyo, Japan
Kouta_Minamizawa@ipc.i.u-tokyo.ac.jp

Hideaki Nii
The University of Tokyo
7-3-1 Hongo, Bunkyo-ku
Tokyo, Japan
hideaki_nii@ipc.i.u-tokyo.ac.jp

Naoki Kawakami
The University of Tokyo
7-3-1 Hongo, Bunkyo-ku
Tokyo, Japan
kawakami@star.t.u-tokyo.ac.jp

Susumu Tachi
The University of Tokyo
7-3-1 Hongo, Bunkyo-ku
Tokyo, Japan
tachi@star.t.u-tokyo.ac.jp

ABSTRACT

In recent years, the use of augmented reality (AR) systems has become quite common. Many marker-based AR systems can input the positions of physical markers and realize a combination of real-world and computer-generated graphics. However, few systems can recognize other information such as fingertip motions. The objective of our study is to create AR environments in which users can manipulate virtual objects by using natural finger motions. Toward this end, we propose a novel marker-based AR system called “ARForce." ARForce enables users to measure the 3D position of markers as well as also the distribution of force vectors that are applied by a user. Using this system, users can manipulate virtual objects using various finger motions.

Our proposed system comprises a camera and an input device. The input device is an elastic body and it comprises two types of markers. One is a square-shaped marker that enables a user to detect the position of the device. The other markers are small circular-shaped ones that are placed within the elastic body. The positions of the circular markers are moved when a user applies a force to the device. This enables force detection.

Categories and Subject Descriptors

H.5.1 [INFORMATION INTERFACES AND PRESENTATION]: Multimedia Information Systems—Artificial, augmented, and virtual realities; H.5.2 [INFORMATION INTERFACES AND PRESENTATION]: User Interfaces—Haptic I/O, Interaction styles

General Terms

Design, Algorithms, Measurement

Keywords

Augmented Reality, Computer Human Interface

1. INTRODUCTION

In recent years, the use of augmented reality (AR) systems has become quite common. AR involves a combination of real-world and computer-generated graphics. ARToolKit[7] is a prominent example of an AR system; it measures the 3D positions and IDs of physical markers and combines computer-generated graphics with real video images. Users can move or rotate virtual objects by moving their corresponding physical markers. Although ARToolKit can recognize the positions of markers, it is difficult to input other information such as finger motions.

The objective of this research is to develop a system where one can manipulate a virtual object using finger motions such as touching, pushing, twisting, or pinching the virtual object in a 3D space. In order to develop such a system, we require a marker-based AR system that can perform the following three functions: (1) measure the 3D positions of markers, (2) recognize multiple touch points simultaneously, and (3) measure both the magnitude and direction of the force applied using the fingers.

Therefore, we have developed a marker-based AR system called “ARForce." ARForce can detect the distribution of force vectors applied on its surface and measure the 3D position, rotation, and ID of markers by using a vision-based approach. Thus, ARForce enables users to construct AR environments in which they can manipulate virtual objects by using natural finger motions.

In the future, we intend to measure information precisely and control virtual objects in the same manner as real objects. In order to do so, we will develop a novel human-computer interface that can accept a highly natural input.

Advances in Computer Entertainment Technology 2008, Yokohama, Japan.

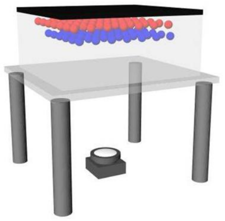

Figure 1: Simplified diagram of GelForce.

Figure 2: ForceTile.

2. BACKGROUND AND RELATED WORK

In the field of human-computer interaction, various types of interactive systems that can capture finger or hand motions in a 3D space have already been proposed. CyberGlove[3] is a well-known example of motion-capture gloves. Soap[2] is a pointing device that uses finger or hand motions in a 3D space. Rekimoto et al. have also proposed a handheld pointing device that uses tilting as the input method[10].

On the other hand, in the field of AR, many systems measure the IDs and positions of physical markers[5, 12]. In addition to systems with such inputs, the development of systems that can recognize richer information about a user's input has also been attempted. Active CyberCode[1] allows a user to issue commands by placing his/her finger on a printed button beside the physical marker. EnhancedDesk[9] recognizes the positions of markers as well as the hand gestures of a user. Haptic Hand[8] uses a special device in addition to markers, and allows user to touch virtual objects.

We propose a novel AR system called "ARForce" that functions as a position marker and as a human-computer interface. This system allows a user to manipulate virtual images using the 3D positions and IDs of markers as well as the distribution of force vectors without using electronic devices. Furthermore, the markers used in our device are invisible to users unlike those used in many other marker-based systems.

In the implementation of ARForce, we adopted a method used in our previous project, "Gelforce"[6, 11], to measure the distribution of force vectors. The sensor used in GelForce comprises a transparent elastic body and two layers of colored markers within the body (Fig. 1). When a force is applied to the body's surface, the internal strain of the body is optically measured in terms of the movement of the markers. Finally, force vectors are calculated from the strain by an elastic theory. Recently, we have also developed a tangible interface called ForceTile[4](Fig. 2) using the same method as that used in GelForce. However, there is an obvious difference in that ForceTile is designed only for a tabletop display, while ARForce can be used in a 3D space.

3. SYSTEM OVERVIEW

Fig. 3 (top) shows the ARForce system. A user controls the tile-shaped input device and applies a force to the bottom of the device using his/her fingers. A camera placed above the user captures an image of the device using infrared (IR) light. Using a computer vision technique and the abovementioned elastic theory, it can measure the 3D positions and IDs of the devices and the distribution of force vectors in real time. Fig. 3 (bottom) shows the measured force vectors and the captured image. The green arrows indicate the force vectors applied using the fingers.

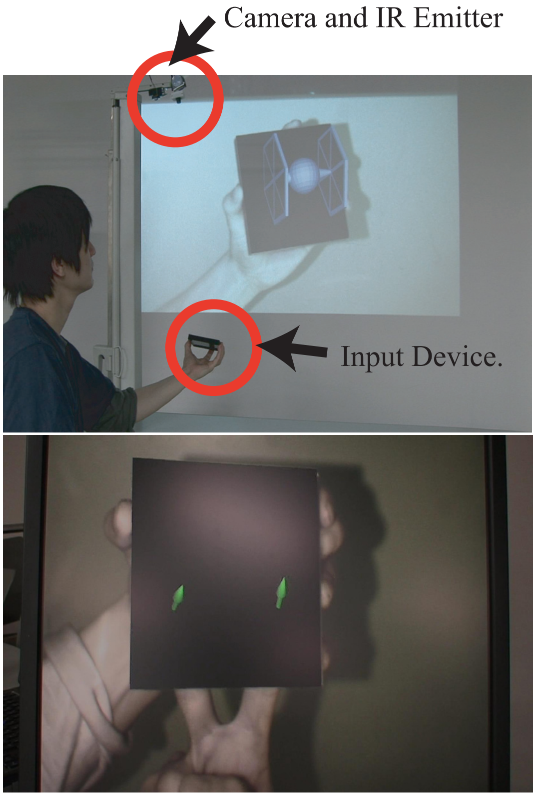

Figure 3: (Top) Appearance of proposed system. (Bottom) Captured image and force vectors. The green arrows indicate the force vectors applied by the user.

3.1 Hardware

Input Device. Fig. 4 shows the structure of the proposed tile-shaped interface. The outer body of the interface is enclosed in an acrylic case. The inner part of the interface contains a transparent elastic body. When the user applies a force to the interface, both the transparent and black elastic bodies are deformed, and the force markers that are placed in the transparent elastic body move.

In order to capture the 3D position of the interface, a square marker is attached to the acrylic case. The force markers and square marker are made of retroreflective materials. Each interface has different patterns of force markers; this enables ID discrimination.

The acrylic case acts as an IR filter that blocks the visible light spectrum and only allows the passage of IR light. Therefore, the interface appears black, and it is impossible to observe the inner part of the interface with the naked eye.

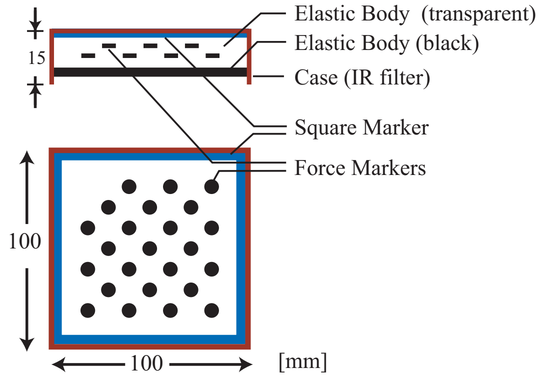

Figure 4: Structure of proposed interface.

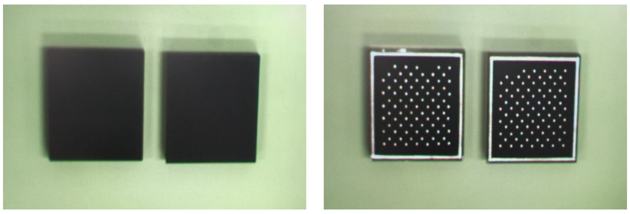

Figure 5: (left) Camera image without IR light. (right) Camera image under IR light. The inner part of the device is invisible to the naked eye; however, the use of the camera and IR light enables us to capture a view of the same. Each interface has different dot patterns, which enables ID detection.

Camera. It is difficult to capture the force markers using normal cameras because many CCD cameras have optical filters that block IR light. Therefore, we removed the filter from our camera. An IR emitter is placed near the camera because the force markers and the square marker are made of retroreflective materials. Fig. 5 shows photographs of the input device.

3.2 Software

This system calculates the distribution of force vectors using the same method as that used in GelForce. GelForce captures the displacement of the force markers that is caused by the force applied by the user.

However, in our proposed system, the positions of the force markers depend on the force vectors as well as on the position of the device. Therefore, it is difficult to measure the displacement of the force markers. Hence, we use an algorithm comprising the following steps.

1. Register the positions of force markers and square marker beforehand in each device coordinate system.
2. Search for the square marker in the captured image and recognize the ID from the pattern of the force markers.
3. Calculate the 3D position of the input device by measuring the corner position of the square marker in the picture coordinate system.
4. Estimate the positions of the force markers in the picture coordinate system from the 3D position of the input device.
5. Measure the real position of force markers in the picture and calculate the relative displacement. This displacement is caused by the force applied by the user's finger.
6. Convert the displacement from the picture coordinate system to the device coordinate system. Further, calculate the force vectors using the same method as that used in GelForce.

4. EVALUATION EXPERIMENT

The proposed method enables a user to measure the force vectors that are applied on the input device. The device is not fixed; instead, it is moved in 3D space, and the positional relationship between the camera and the device varies with time. Therefore, this experiment was performed to confirm whether the ARForce system will function accurately.

4.1 Experimental Procedure

In this experiment, we measured the force vectors using the ARForce system for several relative positions of the input device and the camera. Fig. 6 shows a simplified diagram of the experimental arrangement. Fig. 7 shows a photograph of the experimental arrangement.

The experimental procedure was as follows. First, we fixed the input device and placed a camera in front of it. D and θ represent the distance and angle between the input device and the camera, respectively. Next, we applied pressure using a tension gauge and recorded the estimated force using the ARForce system. We performed the tests under two experimental conditions.

1. D was varied between 0.4 to 1.0 [m] for fixed $\theta = 0$ [deg].
2. θ was varied between 0 to 30 [deg] for fixed $D = 0.6$ [m].

A tension gauge was attached to a linear stage and the force direction was perpendicular to the input surface (+Z direction). The force magnitude was varied between 0 to 10 [N]. We did not perform measurements for a force greater than 10 [N] because we assume that ARForce inputs the force applied by the fingertip. Furthermore, we placed a rubber ball at the end of the tension gauge because the contact face between the fingertip and the elastic body has a large area. The diameter of the ball was 21 [mm].

We required the Young's modulus of the elastic body in the input device to measure the force vectors. However, obtaining the Young's modulus was difficult because the elastic body was made of urethane resin and its value strongly depended on the ratio of hardener. The Young's modulus determines the constant of proportion of the force magnitude. Therefore, we calibrated the constant of proportion using an actual measurement value under conditions of $D = 0.6$ [m] and $\theta = 0$ [deg].

We used a Point Grey Research Firefly MV camera that can capture monochrome images at a resolution of 640 × 480 pixels; this camera has a horizontal angle of view of approximately 31 [deg]. The input device has 83 force markers, and it can measure 21 force vectors. In this experiment, we used a transparent acrylic case instead of an IR filter.

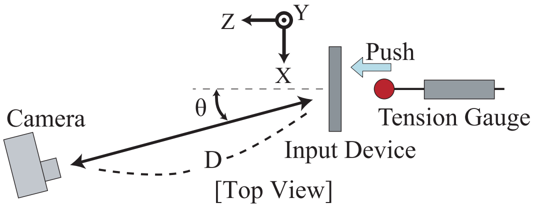

Figure 6: Simplified diagram of experimental arrangement.

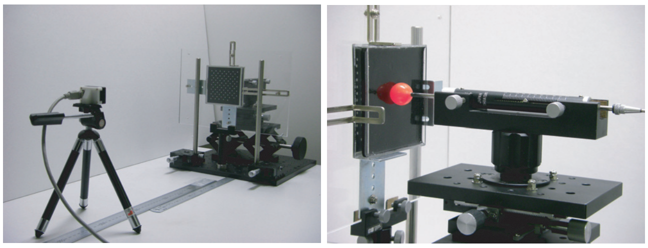

Figure 7: Experimental Arrangement.

4.2 Result and Discussion

Distance. Fig. 8 (left) shows a plot of the experimental result. It is observed that the estimated force F_z increases with the input force. The graphs of $D = 0.4$, 0.6, and 0.8 [m] are quite similar. However, when $D = 1.0$ [m], the estimated force is inaccurate. This is because in such cases, i.e., when the distance is long, the size of the input device in the camera image is too small to recognize the force markers. Therefore, it is difficult to measure the force vectors, and this yields the results shown in the figure. Furthermore, in such cases, a single pixel corresponds to a large length in the real world. Thus, a slight error in force marker detection will cause a significant error in the estimated force vectors. Fig. 8 (right) shows the estimated force vectors when the input force is zero. In ideal conditions, no force vectors must exist. When D ranges from 0.4 to 0.8, small force vectors were measured. However, when $D = 1.0$, there exist many large force vectors. In conclusion, this system cannot be used when $D > 0.8$ [m].

Angle. The estimated force for several values of θ are plotted in Fig. 9. When θ ranges from 0 to 20 [deg], the graphs have almost similar values. This implies that the system can measure force vectors irrespective of the value of θ if it is not greater than 20 [deg]. However, when $\theta = 30$ [deg] and the input force is greater than 7 [N], the estimated force is inaccurate because the system cannot recognize the force markers.

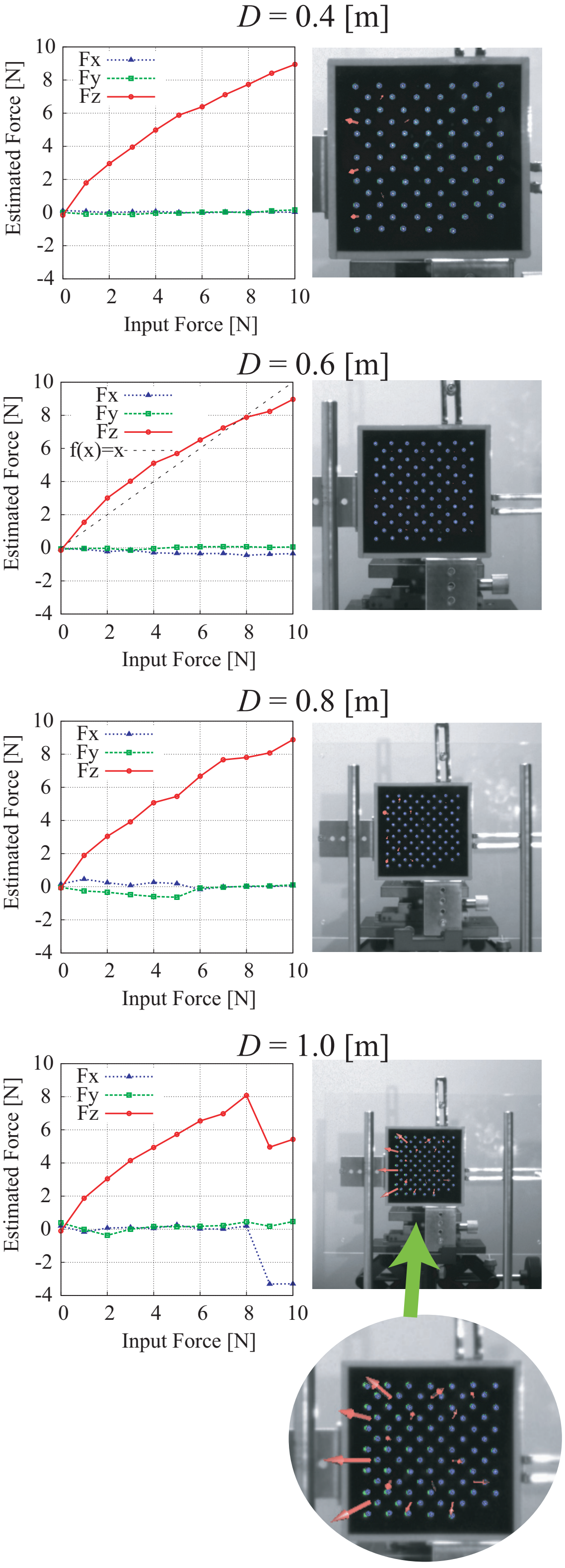

Figure 8: (left) Relationship between input force and estimated force. Each dot indicates the average value of 10 measurements. (right) The red arrows in the pictures indicate the estimated force vectors when the input force is zero. These arrows indicate the mean error in the force estimation.

In this implementation, we did not consider the refractive index. Therefore, when θ is large, it is difficult to measure the precise displacement of the force markers. Fig. 10 shows the estimated force vectors when the input force is zero, $D = 0.6$ [m], and $\theta = 30$ [deg]. The blue dots indicate the estimated positions of the force markers that are calculated from the position of the square marker, and the green dots indicate their actual positions. In an ideal case, the blue and green dots must coincide; however, this does not occur because of the refractive index. Therefore, there exist many errors in the force vectors even when the input force is zero.

This experiment suggests that ARForce will function when D is 0.8 [m] or less and θ is 20 [deg] or less. Note that this value may depend on the camera's angle of view and the size of the input device. Furthermore, in this experiment, the input force was applied only along the Z direction, and we did not examine other force directions. In the future, we will conduct more precise experiments.

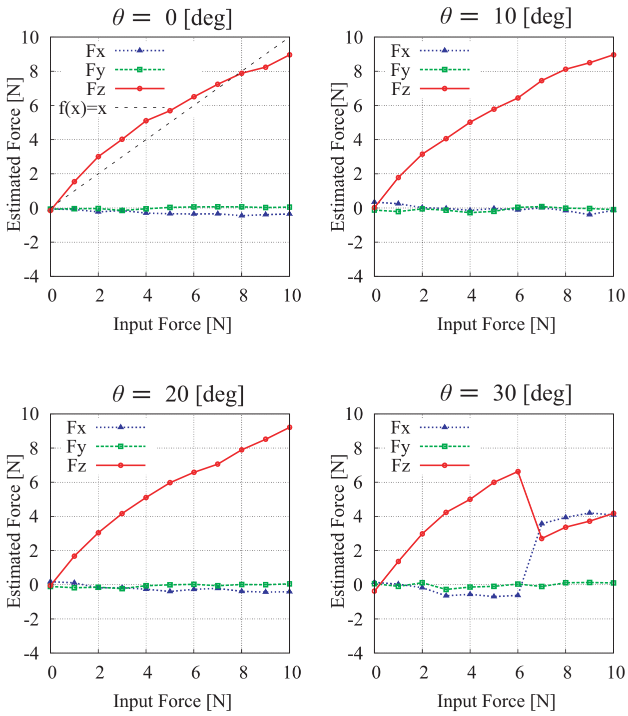

Figure 9: Relationship between input force and estimated force. ($D = 0.6$ [m], $\theta = 0 \sim 30$ [deg])

5. APPLICATIONS

In this section, we introduce several prototype applications that use the ARForce system.

Virtual Objects Viewer. In this application, we use ARForce as a virtual object viewer to demonstrate the recognition of finger gestures. This application involves the overlay of virtual images on a live image such as ARToolKit. Furthermore, when the user expands or twists the input surface of the device, the virtual images are deformed according to the gestures. For example, when the user expands the surface rapidly, the graphics enlarge rapidly. Fig. 11 shows a photograph of this application.

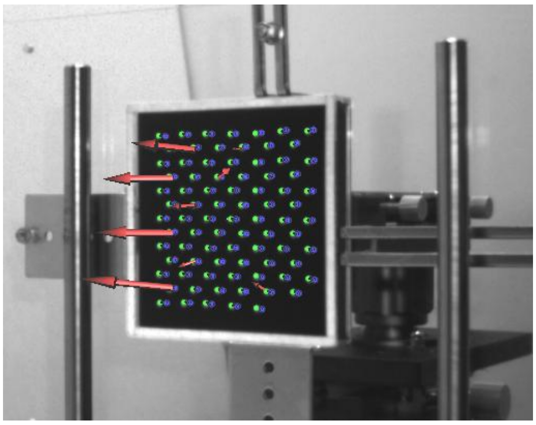

Figure 10: Error in force vectors when input force is zero. ($D = 0.6$ [m] and $\theta = 30$ [deg])

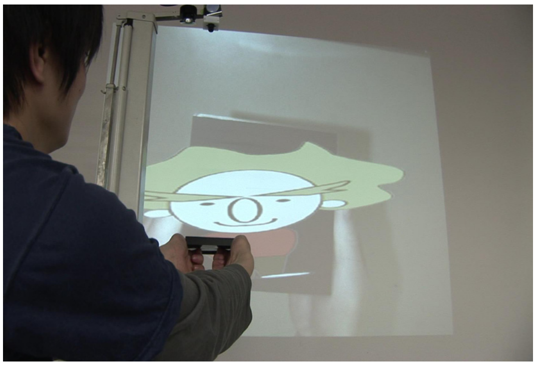

Figure 11: Application of Virtual Objects Viewer.

Movie Player. Fig. 12 shows a photograph of the movie player. In this application, users can control the time parameter of the movies displayed on the interface by means of twisting gestures. When the user twists his/her hand in a clockwise direction, the movie plays in the forward direction; a counterclockwise twist results in the movie being rewound. In addition, the playback speed can be changed according to the twisting power.

Shooting Game. We have also implemented a shooting game as an entertainment application for multiple users. Fig. 13 shows a photograph of this application. Users can operate a virtual spaceship using the interface. The position and tilt of the spaceship corresponds to the interface's position and posture. In addition, users can shoot beams by pushing the input surface of the ARForce interface. The direction of the beam changes according to the direction of force applied by the user.

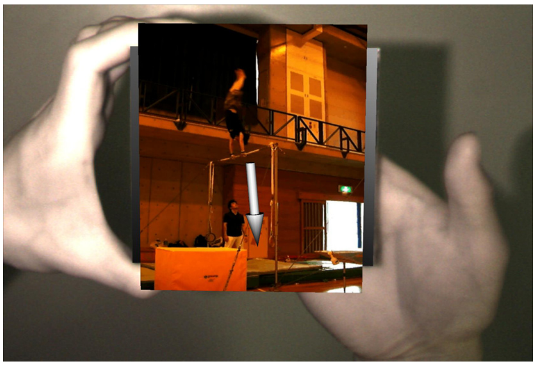

Figure 12: Movie Player.

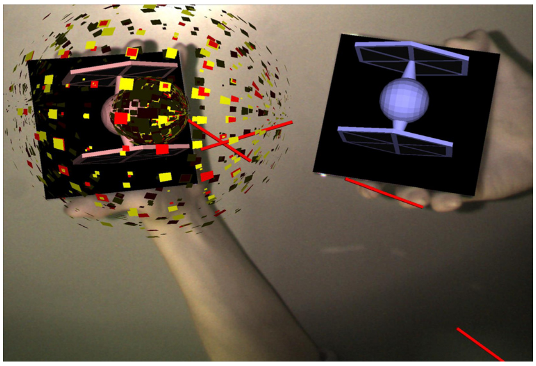

Figure 13: Shooting Game.

6. CONCLUSIONS AND FUTURE WORKS

In this paper, we have proposed a novel marker-based AR system that can measure the 3D positions and IDs of markers as well as the distribution of the magnitude and direction of force vectors applied on the system. The experiment revealed that ARForce can be used to measure force vectors irrespective of the distance or angle between the device and camera if the device is located in a certain region. Using this system, we developed simple applications that enable users to manipulate virtual objects using finger motions such as pushing and twisting.

We intend to do the following as future works. First, we will continue to improve the method and algorithm used in our system in order to measure force vectors more precisely. In particular, we plan to solve the problem of the refraction factor that complicates the recognition of force markers. Second, we will examine the performance of ARForce in greater detail. In this paper, we performed experiments wherein the force was applied only along one direction. Third, we will implement a series of new AR applications including games, simulations, media art, and scientific visualizations.

7. REFERENCES

[1] Y. Ayatsuka and J. Rekimoto. Active CyberCode: A Directly Controllable 2D Codes. In CHI 2006 Extended Abstract, pages 490–495, 2006.

[2] P. Baudisch, M. Sinclair, and A. Wilson. Soap: a Pointing Device that Works in Mid-Air. In UIST 2006 (Technote), pages 43 – 46, Montreux, Switzerland, 2006.

[3] CyberGlove. http://www.immersion.com/3d/products/cyber_glove.php.

[4] Y. Kakehi, K. Jo, K. Sato, K. Minamizawa, H. Nii, N. Kawakami, T. Naemura, and S. Tachi. ForceTile: Tabletop Tangible Interface with Vision-based Force Distribution. In ACM SIGGRAPH 2008 New Tech Demo, 2008.

[5] Y. Kakehi, T. Naemura, and M. Matsushita. Tablescape Plus: Interactive Small-sized Displays Upstanding on Tabletop Display. In 2nd Annual IEEE International Workshop on Horizontal Interactive Human-Computer Systems (Tabletop 2007), pages 155 – 162, 2007.

[6] K. Kamiyama, K. Vlack, T. Mizota, H. Kajimoto, N. Kawakami, and S. Tachi. Vision-based sensor for real-time measuring of surface traction fields. IEEE Computer Graphics and Applications, pages 68 – 75, 2005.

[7] H. Kato and M. Billinghurst. Marker tracking and HMD calibration for a video-based augmented reality conferencing system. In Proceedings 2nd IEEE and ACM International Workshop on Augmented Reality (IWAR'99), pages 85 – 94, San Francisco, CA, USA, 1999.

[8] L. Kohli and M. Whitton. The haptic hand: providing user interface feedback with the non-dominant hand in virtual environments. In Proceedings of Graphics Interface 2005, pages 1 – 8, 2005.

[9] H. Koike, Y. Sato, Y. Kobayashi, H. Tobita, and M. Kobayashi. Interactive Textbook and Interactive Venn Diagram: Natural and Intuitive Interface on Augmented Desk System. In Conference Extended Abstracts on Human Factors in Computer Systems (ACM CHI 2000), pages 121–128, April 2000.

[10] J. Rekimoto. Tilting Operations for Small Screen Interfaces. In User Interface and Software Technology (UIST'96), pages 167 – 168, 1996.

[11] K. Sato, K. Minamizawa, N. Kawakami, and S. Tachi. Haptic Telexistence. In 34th Int. Conf. On Computer Graphics and Interactive Techniques (ACM SIGGRAPH 2007), Emerging Technologies, 2007.

[12] E. Woods, P. Mason, and M. Billinghurst. MagicMouse: an Inexpensive 6-Degree-of-Freedom Mouse. In Proceedings of Graphite 2003, pages 285 – 286, 2003.