Twinkle: Interacting with Physical Surfaces Using Handheld Projector

Takumi Yoshida* — The University of Tokyo
Yuki Hirobe† — The University of Tokyo
Hideaki Nii‡ — Keio University
Naoki Kawakami§ — The University of Tokyo
Susumu Tachi¶ — Keio University

Abstract

We propose a novel interface called *Twinkle* for interacting with an arbitrary physical surface using a handheld projector and a camera. When a user flashes a projection light on an object, the projected images react as if the user touched the object with the light. The handheld device recognizes the features of the physical environment and displays images and sounds that are generated in real-time according to the user's motion and collisions of projected images with objects. We realize this system by using several image-processing techniques and a collision detection algorithm. We also use an acceleration sensor to compensate for the image processing. In this paper, we explain the principle of interacting with a physical surface. Then, we describe the implementation of the prototype system and some application examples.

Index Terms: H.5.1 [Information Interfaces and Presentation]: Multimedia Information Systems—Artificial, augmented, and virtual realities

1 Introduction

1.1 Interfaces using a handheld projector

Recent years have witnessed the development of many pocket-sized projectors. In the near future, it is expected that such projectors will be installed in portable devices such as cellular phones. Meanwhile, intuitive interfaces that operate according to a user's motions have already become popular, for example, a game controller or a touch screen. Therefore, interfaces that can be used for accessing information using projectors have been widely studied[8][1].

Raskar et al. [7] explored a technique for adaptive projection on non-planar surfaces using conformal texture mapping; they also proposed object augmentation using a handheld projector, including interaction techniques. Forlines et al. [5] presented the design and evaluation of an interaction technique that allows a user to fluidly zoom in on areas of interest and accurately select targets. Cao et al. [2][3] explored the design space by dynamically defining and interacting with virtual information using a passive pen tracked in 3D. However, these interfaces suffer from numerous problems. Some systems require motion-tracking systems in order to measure the position of the projector. Further, the surface where the image is projected is limited to a plain screen such as a white wall. Although handheld devices are quite compact and portable, they are only used under particular circumstances.

In this study, we propose a novel interface called *Twinkle* for interacting with an arbitrary physical flat surface; here, we define interaction as the utilization of a physical surface to perform certain tasks. When a user flashes a projection light from a handheld device, such as a flashlight, onto an object, the projected images react as if the user touched the object with the light. Images and sounds are generated in real-time according to the user's motion and the collisions of the projected images with other objects. Figure 1 shows an application example of *Twinkle*. We define a physical surface as a surface that exists in a physical environment and that is not plain. Examples of such surfaces are a poster on a wall, figures or characters on a whiteboard, and a desk on which objects or drawings are placed (Figure 2).

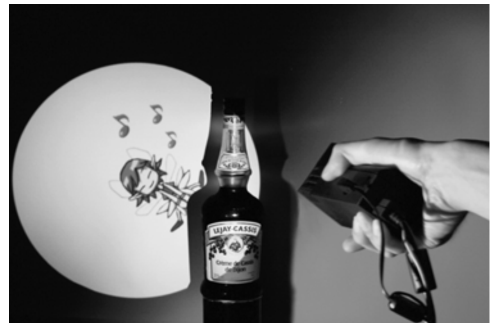

Figure 1: *Twinkle* as a gaming interface. A character projected using a handheld device is superimposed on a real object.

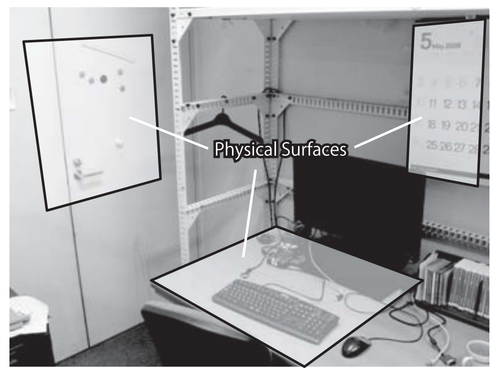

Figure 2: Examples of physical surfaces suitable for *Twinkle* applications. The surface should be nearly-planar, and there should be some objects or drawings on the surface.

1.2 Proposed method

In order to develop this interface without any motion-tracking systems or AR markers, we propose a novel method for recognizing real environments using a handheld projector and a camera. When a user projects light from the handheld projector onto a physical surface, the camera captures the physical surface. By detecting where the projected area exists in the captured image, we can match each pixel of the generated image to each pixel of the captured image. This matching enables us to distinguish the projected image from the real object on the physical surface. As a result, the system can estimate a user's motion and can recognize the features of the real objects. In this manner, we can realize an interaction with real objects by using the projected image as a guide for image processing. Moreover, we propose an algorithm for the collision detection of a

*e-mail: takumi_yoshida@ipc.i.u-tokyo.ac.jp
†e-mail: yuki_hirobe@ipc.i.u-tokyo.ac.jp
‡e-mail: nii@tachilab.org
§e-mail: kawakami@tachilab.org
¶e-mail: tachi@tachilab.org

IEEE Virtual Reality 2010
20 - 24 March, Waltham, Massachusetts, USA
978-1-4244-6236-0/10/$26.00 ©2010 IEEE

projected image and a real object. In addition, we use an acceleration sensor to compensate for the image processing.

The features of our proposed method are as follows. The handheld device comprises a handheld projector, a camera, and an acceleration sensor. The proposed system is quite simple and does not require other motion-tracking systems. Nowadays, many cellular phones are equipped with cameras and acceleration sensors. The proposed interface can therefore be realized using only a cellular phone in the near future. Our system can estimate the relative movements and positions between a projected image and the real objects on a surface. Although it cannot estimate absolute 6-DOF positions and poses unlike conventional methods, the relative information is considered to be sufficient for interacting with a physical surface. However, by using only image processing, the system cannot detect the direction of the gravitational force and it cannot track a rapid movement. Therefore, depending on the application, an acceleration sensor might have to be integrated into the system.

1.3 Applications

The proposed system enables various applications. A few examples of the applications of *Twinkle* are mentioned below. First, we propose a gaming interface. A player operates a character by moving a handheld device. The character interacts with real objects, for example, it walks along a drawing in a poster on the wall. Next, we propose an interface for music composition and musical performance. The pitch of a sound is determined by the size of the object illuminated by the projector. The color of the object and the user's motion respectively determine the tone and the volume, respectively, of the sound. The user can create melodies and rhythms by laying out objects on a surface. This interface enables users to intuitively compose and play music on the basis of their intuition, i.e., without requiring them to possessthey can compose and play music even if they do not have sufficient knowledge of music. Additionally, the proposed interface can be used as an AR annotation system. The system recognizes figures or characters on a surface; correspondingthis information is presented near the relevant objects.

2 Principles

2.1 System overview

Figure 3 shows the system configuration of *Twinkle*. The handheld device comprises a projector, a camera, and an acceleration sensor. The projector illuminates some objects on a physical surface. The contour of the projected area is circular. The camera captures the physical surface that includes the circular projected area.

Twinkle has the following four key functions:

- Pixel matching between the generated image and image captured using a camera.
- Estimation of the user's motion relative to the physical surface.
- Detecting and avoiding collisions between the projected image and the real objects.
- Generating images and sounds according to the user's motion and the features of the real objects.

These functions have several subroutines. Futher, they are implemented by multithreaded parallel processing. The details of each process are described below.

2.2 Pixel matching

Pixel matching between the generated image and the image captured using the camera is carried out by using circle detection. Generally, when a circular image is projected onto a plane, the projected image is elliptical. Now, we assume that the distance between the projector and the camera is sufficiently short as compared to the distance between the projector and the plane. Therefore, we can approximate the shape of the projected image by using a circle. We use a Hough transformation method[4] to detect the circle. Hough transformation provides the position of the center and the radius of the circle. By scaling the detected circle according to the ratio of the radius of the generated circle to that of the detected one, we can realize pixel matching. The detected circular area is extracted for use in other image processing routines (Figure 4). Moreover, we can calculate the distance between the physical surface and the projector by using a triangulation method. The position of the center of the circle corresponds to the distance between the surface and the projection point. This enables us to display an image that has a constant size despite the distance.

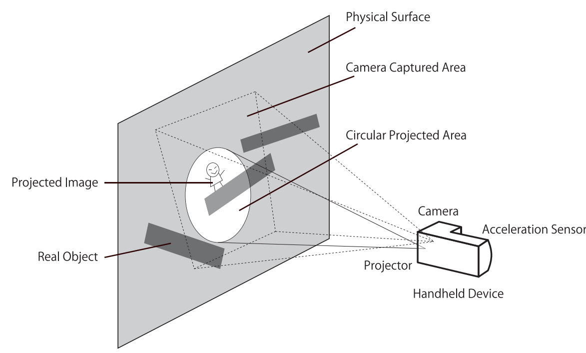

Figure 3: System configuration.

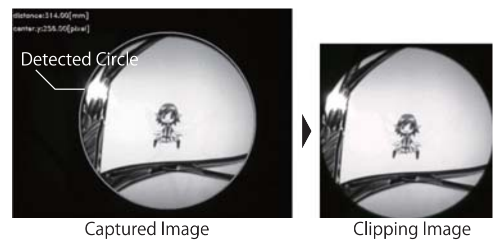

Figure 4: Circle detection. A detected circular area is extracted for use in other image processing routines. The position of the center of the circle corresponds to the distance between the surface and the handheld device.

2.3 Motion estimation

In order to estimate the user's motion, we use a conventional optical flow technique. The camera tracks the objects illuminated by the white area of the projected image. Here, a projected image that is not white and the real objects have to be distinguished, because when a user moves a handheld device, the real objects move in the captured image, but the projected image does not move. Therefore, we generate a mask image that hides the projected image. Tracking points that are included in the mask image area are not used for calculation. By using the optical flow, we can obtain the velocity and the direction of the user's motion in each frame. These values are not absolute but relative to the physical surface.

If the user moves the handheld device very rapidly, the camera cannot track the objects because of the motion blur and frame rate. Therefore, we use the acceleration sensor as an aid in the optical flow. The acceleration sensor helps in detecting shaking movements or rapid changes in the direction of movement. Moreover, it can measure the direction of the gravitational force.

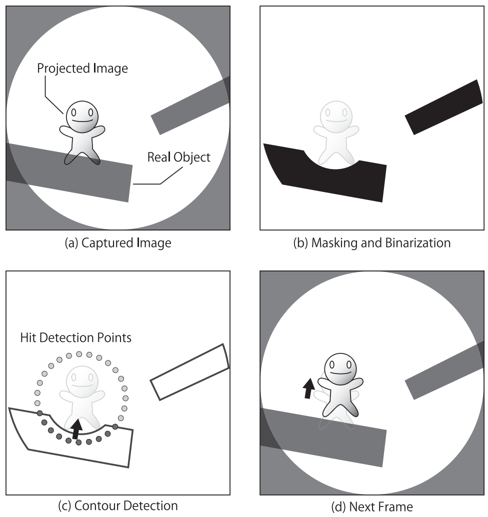

Figure 5: Collision detection algorithm. Hit detection points are arranged in a circular pattern outside the projected image. The displacement of the image in the next frame is determined according to the position and number of points that exist in the object area.

2.4 Collision detection and avoidance algorithm

In this section, we describe how to detect collisions between real objects and a projected image that is not white. Considering the real-time collision avoidance of the projected image in a real environment, a degree of overlap is essentially inevitable because of the time delay. Therefore, we allow a certain level of overlap and propose the following collision avoidance method. Figure 5 shows the process of the collision detection and avoidance algorithm. First, a camera captures the circular projected area that is extracted by pixel matching, as mentioned above. Second, masking and binarization are carried out. The mask hides the projected image. The threshold value of binarization is determined by using Otsu's algorithm[6]. Then, contours of the objects are detected. The objects that have a small area are ignored as noise. Hit detection points are arranged in a circular pattern outside the projected image. Finally, the displacement of the image at the next frame is determined according to the position and number of points that exist in the object area.

2.5 Generating images and sounds

The system obtains some information: pixel matching, user's motion, and collision between projected image and real objects. Images and sounds are generated based on this information depending on the application. For instance, we can make a projected ball image act as a real bouncing ball.

In addition, the system can recognize various features such as color, size, and specific shape by using other image processing techniques. If these features are used, the system can find more applications. In addition, a pattern-matching algorithm helps in estimating the shape of the objects by comparing the object image with previously stored images. By combining character recognition, the following AR annotation system can be realized. When a user aims the handheld device toward a word written in a certain language, the meaning and pronunciation are displayed in another language.

3 Results

3.1 Implementation

We have developed a handheld device that comprises a small projector (3M, MPro110), an IEEE-1394 camera (Point Gray Research, FireflyMV), and a wireless acceleration sensor (Wireless

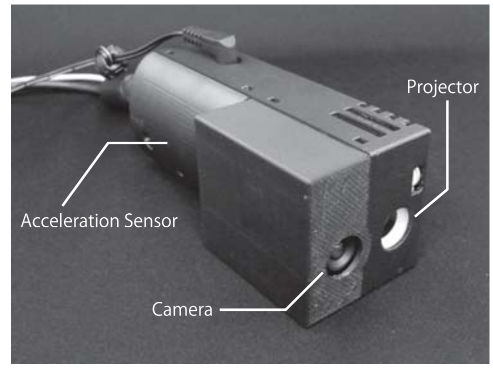

Figure 6: Handheld device.

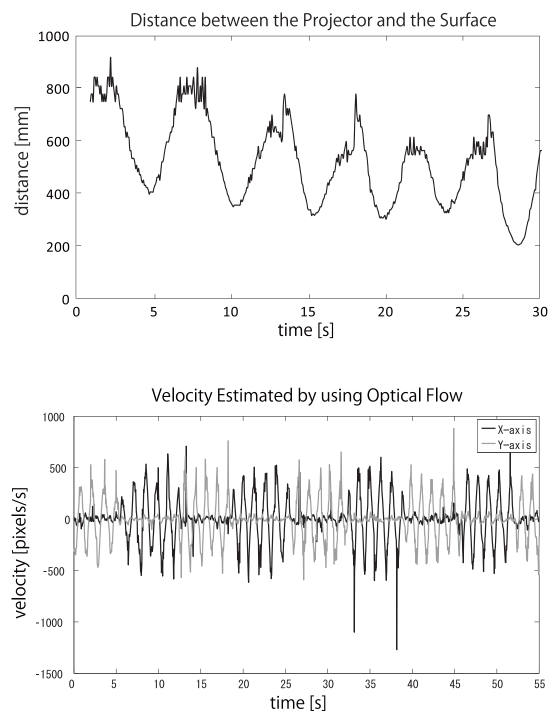

Figure 7: Results of motion estimation.

Technologies, WAA-006). The exterior case was built by a 3D printer (Stratasys, Dimension BST768). Figure 6 shows the handheld device. The device is 14-cm long and weighs 250 g.

The performance specifications of the prototype system are as follows. The resolution of the camera is 640×480 (VGA) and the frame rate is 30 fps. The sampling frequency of the acceleration sensor is 200 Hz. All the processes are executed in parallel on a PC that has a dual-core CPU (2.00 GHz, 1.99 GB RAM). The total refresh rate was approximately 30 fps. Many of the basic image processing routines are implemented using the computer vision library OpenCV.

In order to show the effects of the basic functions, we conducted an evaluation experiment. We placed some figures on a wall. A user aimed the handheld device toward the wall. Then, the user periodically moved the device from side to side and up and down. The system estimated the user's motion by using the optical flow technique. In the same manner, the user periodically moved the device back and forth. The system detected the center of the circular projected area by using the Hough transformation. Then, we esti-

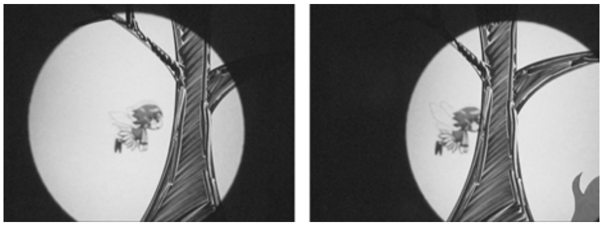

Figure 8: Collision between a projected character and a real object. The entire projected area is moved from left to right; however, the fairy is blocked by an object.

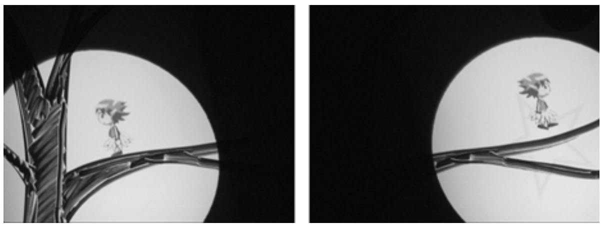

Figure 9: A fairy stands on the ground and walks according to a player's motion.

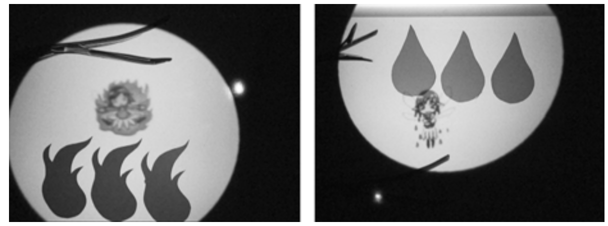

Figure 10: Interactions according to color detection.

Figure 11: Interface for music composition and musical performance.

mated the distance between the physical surface and the handheld device by using the triangulation method. Figure 7 shows the results of motion estimation. These results show that the system can estimate the user's motion relative to the physical surface without any external tracking systems.

3.2 Application examples

In this section, we introduce two application examples that we developed using this system.

3.2.1 Gaming Interface

The first example is a gaming interface. This is an action game that regards real objects and drawings on a wall as obstacles or ground. A player can customize a game field . The rules of this game are as follows. A player operates a flying fairy that is displayed at the center of the projected area. Usually, the fairy follows the movement of the projection light; however, if there exists an obstacle in the direction of movement, she cannot move, as shown in Figure 8. This function is realized by using the collision detection and avoidance technique described in section 2.4. When there is ground under her feet, she lands on the ground and walks according to the player's motion (Figure 9). At this time, the system recognizes the tilt of the ground and determines the fairy's posture. Moreover, an acceleration sensor detects the direction of gravitational force in order to ensure that the fairy always walks erect. Color detection has also been implemented. When the fairy touches a red object, she catches fire. When she touches a blue object, she gets wet (Figure 10).

3.2.2 Music Interface

Second, we developed an interface for music composition and musical performance. A pointer was displayed on the center of the projected area. When the pointer was located on a dark object on a whiteboard, a sound was produced and ripples were generated. The system measured the size of the dark area by using a contour detection algorithm. The pitch of a sound and the color of ripples were determined by the size of the object. The sound volume was changed according to the velocity of the pointer. The trajectory of the pointer's motion was also displayed by using optical flow. As a result, a user was able to compose and play music by flashing the projector onto the whiteboard (Figure 11).

4 Conclusions

In this paper, we proposed an interface called *Twinkle* for interacting with physical surfaces by using a handheld projector. The system helps in estimating the user's motion and in estimating the position relative to the physical surface by using projected light as a guide for image processing. Then, we developed a prototype system and implemented two application examples. These examples showed that the system worked effectively and it can be used for various applications.

We plan to develop more sophisticated applications by using this system. In the future, two or more people should be able to use the interface under an interconnected environment.

References

[1] F. Berard. The magic table: Computer-vision based augmentation of a whiteboard for creative meetings. *IEEE International Conference in Computer Vision, Workshop on Projector-Camera Systems (PROCAMS'03)*, 2003.

[2] X. Cao and R. Balakrishnan. Interacting with dynamically defined information spaces using a handheld projector and a pen. *Proceedings of the 19th annual ACM symposium on User interface Software and Technology*, pages 225–234, 2006.

[3] X. Cao, C. Forlines, and R. Balakrishnan. Multi-user interaction using handheld projectors. *In Proceedings of the 20th Annual ACM Symposium on User interface Software and Technology*, pages 43–52, 2007.

[4] R. O. Duda and P. E. Hart. Use of the hough transformation to detect lines and curves in pictures. *Comm. ACM*, 15:11–15, 1972.

[5] C. Forlines, R. Balakrishnan, P. Beardsley, J. v. Baar, and R. Raskar. Zoom-and-pick: facilitating visual zooming and precision pointing with interactive handheld projectors. *In Proceedings of the 18th annual ACM symposium on User interface software and technology*, pages 73–82, 2005.

[6] N. Otsu. A threshold selection method from gray-level histograms. *IEEE Trans. Sys., Man., Cyber.*, 9:62–66, 1979.

[7] R. Raskar, J. van Baar, P. Beardsley, T. Willwacher, S. Rao, and C. Forlines. ilamps: Geometrically aware and self-configuring projectors. *ACM Transactions on Graphics*, 22(3), 2003.

[8] J. Underkoffler and H. Ishii. Illuminating light: an optical design tool with a luminous-tangible interface. In *CHI '98: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 542–549, 1998.