Touch Recognition Technique for Dynamic Touch Pairing System and Tangible Interaction with Real Objects

Using 3D Point Cloud Data to Enable Real Object Tangible Interaction

Unseok Lee and Jiro Tanaka
Department of Computer Science
University of Tsukuba
Tsukuba, Ibaraki, Japan
{leeunseok, jiro}@iplab.cs.tsukuba.ac.jp

***Abstract*— Sensor-based pairing technology between digital objects for interactions is used widely (e.g., smart phone to Bluetooth headset). In addition, research about tangible interactions between daily normal analog objects (e.g., a doll, Lego block) and digital objects has progressed and is also popular. However, such research can only involve interactions with already setup objects. They have to attach sensors to objects for interaction. The paired objects cannot be changed dynamically. In addition, it is difficult to make interactions with various objects simultaneously. The objects with attached sensor(s) for tangible interaction can recognize the touched area of the objects, but cannot recognize touch gesture with a lot of movements. In this paper, we propose a new analog-digital object pairing method and touch recognition technique by intuitive touch interactions using three-dimensional point cloud data. Several touch pairing and touch recognition methods are described in detail. The paired objects are changed dynamically using the proposed method. In addition, tangible interactions between two objects are described after pairing. Finally, we demonstrate the high recognition rate of the proposed method using experiments and describe our system's contribution.**

Keywords-dynamic pairing, point cloud, tangible interaction, 3d gesture, human computer interaction, touch recognition

I. INTRODUCTION

In everyday life, the touching action is natural and common. We touch objects to use them (e.g., a doll or toy to play, open a bottle cap for drinking). Touch interactions with digital devices have also become natural in recent years, because smart devices with touch screens and touch pads are now used widely. Simultaneously, in the field of human–computer interaction (HCI), research on interactions between physical objects and digital devices has progressed rapidly. A physical object is set as an input unit and the digital device is controlled by it. Such interactions are used widely and have become a 'natural' method. However, to use a physical object as an input unit, much effort and time is initially needed to set up sensors [1]. Moreover, it takes time and effort to apply sensors again when using another physical object as the input unit. In addition, the digital object is limited to a particular physical object. The touching objects recognition is also limited. Recognizing the touched area of an object and the touch gesture with 3D objects is difficult by attaching sensors. Thus, there is no 'natural' interaction between various objects. Regarding the input unit, research on methods for making a tangible object for which touch sensing is possible has progressed.

Figure 1. Analog-Digital objects in everyday life

For example, in the bowl project [2], a simple media player in a bowl sits on a living room table and a range of physical objects can be placed within it. When an object is placed in the bowl, related media are played on the TV. The project used radio-frequency identification (RFID) sensors for tangible interactions. However, the system could not provide dynamic pairing and touch gesture recognition between objects. The interactions and possible objects were also limited. The "HandSense" [3] prototype used capacitive sensors for detecting when it was touched or held against a body part. It could determine whether a device was held in the left or right hand by measuring the capacitance on each side. Wimmer [4] presented a method for prototyping grasp-sensitive surfaces using optical fibers. However, all of these examples require attaching sensors to the devices. This is unnatural in the real world. They cannot support multiple object recognition and touch gesture based interactions. Also, the paired object cannot be changed dynamically.

In this paper, we propose a new method for dynamic pairing and touch recognition techniques with tangible interactions between analog objects and digital objects in practical circumstances.

This dynamic pairing is designed through touch interactions. For example, one hand grasps a doll, an analog

2014, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

object. Then, the other hand grasps a smart phone, a digital object. The doll and smart phone are paired and prepared for tangible interactions. The system makes it possible to pair the doll with touch in three dimensions. The smart phone then shows feedback from interactions with the doll. We can change the paired objects dynamically. Figure 1 shows examples of pairing analog and digital objects in everyday life. We also designed touch recognition technique and learned touch gesture based interaction for natural and robust use with existing objects. The natural actions that can be used as gestures, and supporting all objects in everyday life, were considered for tangible interaction. The touch patterns, practical interactions with popular gestures, object size and object hardness were considered as well.

Our proposed methods and techniques are based on three-dimensional(3D) point cloud data using two Kinect units. They capture and calibrate 3D point cloud data. Our system determines touch pairing and tangible interactions of the paired analog object, based on these calibrated data. In this way, the system can readily recognize what objects are touched and trace what objects are paired. Our system also determines touch recognition and learned gesture based tangible interactions of the object. In addition, we can recognize the touched position and movements of the objects. We present the results of tests of recognition rate for pairing and touch recognition rate using the proposed method.

The rest of this paper is organized as follows. Section II introduces related work on depth-based touch sensing and tangible interactions. We describe in detail the principles of the pairing method and the system specifications in Section III. We described touch recognition techniques in detail and tangible interactions in Section VI. We present details on the high recognition rate of our system in Section V. Finally, we describe our contribution and future work in Section VI.

II. RELATED WORK

A. Depth-based Touch Sensing Technologies

In recent years, depth-based cameras and related technology have developed rapidly. Research on obtaining 3D data on objects using depth information has also progressed. The framework for 3D sensing using depth cameras has been improved remarkably [5].

Klompmaker et al. implemented tangible interactions using a depth camera and a 3D sensing framework [12]. They implemented touch detection and object interaction, supporting multi-touch and tangible interactions with arbitrary objects. They used images from a depth camera to determine whether a user's finger touched the object. However, they were unable to support 3D touching and dynamic pairing between objects for tangible interactions.

Wilson et al. used depth-sensing cameras to detect touch on a tabletop [7], using the camera to compare the current input depth image against a model of the touch surface. The interactive surface need not be instrumented in advance for the interaction and this approach allows touch sensing on non-flat surfaces. However, they only supported simple touch recognition and could not address touch in any direction with 3D objects.

B. Tangible Interactions with Analog Objects

"Digital Desktop" by Wellner et al. [8] was used in an early attempt to merge the physical and digital worlds. They implemented a digital working space on a physical desktop where physical paper served as an electronic document. The interaction with papers was by means of bare fingers. "Icon Sticker" [9], based on this idea, is similar. Icon Sticker is a paper representation of digital content. It consists of transferring icons from the computer screen to paper, so they can be handled in the real world and used to access digital content directly. An icon is first converted into a corresponding barcode, which is printed on a sticker. Then the sticker can be attached to a physical object. To access the icon, the user scans the barcode on the sticker with a barcode scanner. "Web Sticker" [10] uses barcodes to represent online information. It is similar to Icon Sticker, but instead of icons it manages Web bookmarks. They use a handheld device with a barcode-reading function to capture the input and display related information.

There were also attempts to improve tagging of physical objects for a more natural tangible interaction. Nishi et al. [11] registered real objects on a user's desktop based on a user indicating a region on the desk by making a snapshot gesture with four fingers. A color histogram was used to model the object and a pointing gesture was used to trigger the recognition. "Enhance Table" also uses a color histogram to model objects. However, the size is predefined and the system is limited to mobile phone recognition.

Although many previous tangible interaction studies have used physical objects for interactions, most of them are token-based approaches and provide only limited use of real objects. They do not support 3D object tracking or pairing for tangible interactions. Thus, to overcome this, we propose a robust 3D object-tracking method that detects touch in three dimensions. The system supports dynamic pairing between analog and digital objects, and makes analog objects accessible to touch anywhere.

III. TOUCH PAIRING SYSTEM

A. Hardware and Software

Our system consists of two Microsoft Kinect sensors for Xbox 360 with stands.

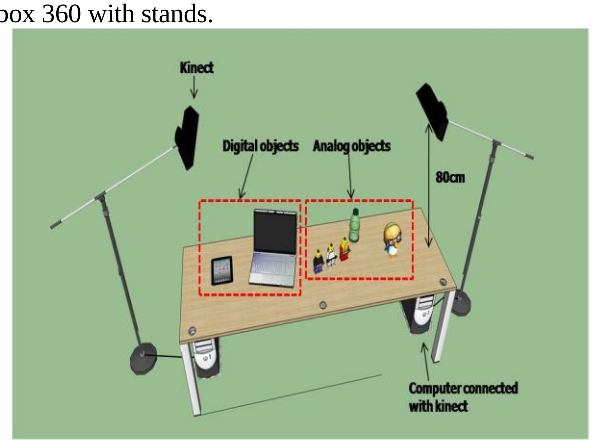

Figure 2. Touch Pairing System Configuration

2014, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

Two computers (Intel i7 2.4Hz, 8GB RAM, and GeForce GTX 660M graphics card) are used to handle the 3D data. A desk and the pairing object for interaction are installed. The analog and digital objects are randomly placed. The Kinect sensors are set at 80 cm from the desk. The computers are connected to each Kinect sensor, and the digital objects have wireless internet or Bluetooth connections with the computer, installed in the bottom of the desk (Figure 2). Our system uses OpenFrameworks OpenNI [6] for the Kinect sensors and a point cloud method that provides example add-ons of frameworks. The system obtains 3D point cloud data and maps the RGB data to the point cloud. The movement of the points is based on pairing recognition. The proposed system was implemented on a Microsoft Windows 7 platform. The pairing recognition module was implemented in Visual Studio 2010 and OpenNI 1.5.4.

B. *Touch Pairing System Architecture*

The entire system consists of three major modules (Figure 3). The input data are obtained by the two Kinect sensors, on the left and right. It calibrates their data from two cameras and processes the data. In the case of using one camera, there are parts that cannot be reconstructed such as both sides of a cup. Because one camera cannot catch all area of objects even if camera sets up top-down direction. It is difficult to find out the touched location and touching gesture itself. It cannot cover all sides of objects using only one camera.

Our system sets up two cameras in proper position that was found out empirically (Figure 2). The system implemented the calibration and 3D reconstruction to obtained data from two cameras. We can recognize touching almost all sides when using these methods. We can find the location of the touch as well. In addition, the difficult side to recognizing is estimated using reconstructed 3D model data. The process is detailed below.

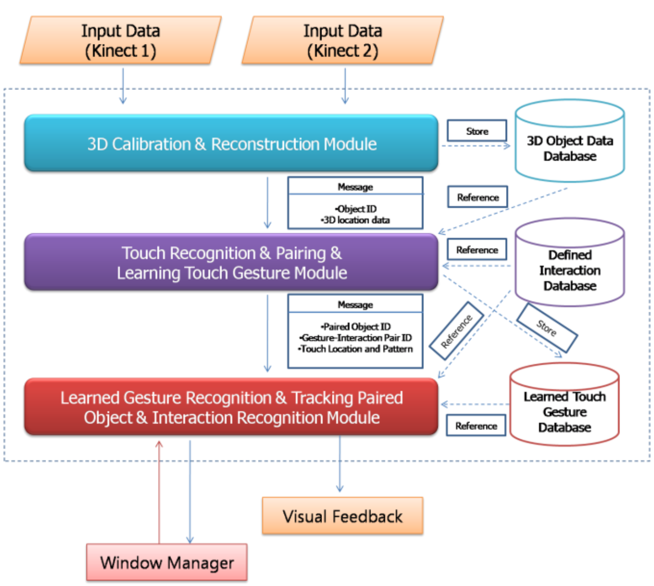

Figure 3. Touch Pairing System Architecture

1) 3D calibration and reconstruction module: In this module, we calibrate depth data for each object, obtained from the two Kinect sensors. The system makes a 3D reconstruction using a point cloud library with calibrated data. The module stores calibrated and reconstructed data in a database, which is then used by the touch-recognition module. After storage, the module sends messages to the touch-recognition and pairing module about object ID and object location using the 3D point cloud data.

2) Touch recognition, pairing, and learning touch gesture module: In this module, we implemented pairing and learning touch gesture with touch recognition method. The process is detailed as follows.

a) Touch recognition and pairing: Touch is recognized in terms of the depth and position of the object and hands using 3D tracking. Using the previous depth information from the 3D reconstruction based on the point cloud, the system determines whether the hand touched the object, and if so, the position of the object. The system recognizes the time of touching between the user's 3D hand point cloud data and the object 3D point cloud data, then determines whether they are paired. A paired analog object's 3D point cloud data are stored and sent to the tracking module with information on the object type. We defined a limited objects database.

b) Learning touch gesture: In this method, the system identificates touched position of hands and fingers with objects. In addtion, our system stores movement of the fingers and gestures to the database. A touched object is stored with its 3D vision image for tracking after paired. Gestures and interaction pair with touched objects can be stored in the database by the system as well. They can be used tangible interactions with learned touch gestures.

3) Learned gesture recognition, tracking paired object, and interaction recognition module: In this module, our proposed system recognizes learned touch gesture and tracks paired object. In addition, a tangible interaction is implemented with touched position and gestures using data from the database. The process is detailed as follows:

a) Learned gesture recognition: The learned touch gesture recognition is implemented using learned touch gesture database that was stored by the learning touch gesture module. The system recognizes touched position and gestures applied to the objects. The paired object and interactions are prepared after recognizing the touch gesture and the touched objects.

b) Tracking paired object and interaction recognition: After pairing, the system tracks the analog object based on saved 3D point cloud data. The paired digital object can be tracked; however, the paired objects do not commonly move. The paired analog object is tangible, based on 3D analog object data, from the 3D calibration and reconstruction module. We can make an interaction with the digital object in this module; the interaction is shown by visual feedback.

2014, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

C. Touch Pairing Method using 3D Point Cloud Data

Our proposed method uses 3D point cloud processing of Kinect depth data. A point cloud itself is a set of data points in a coordinate system. We measured a large number of points on the surface of an object using OpenFramework [6].

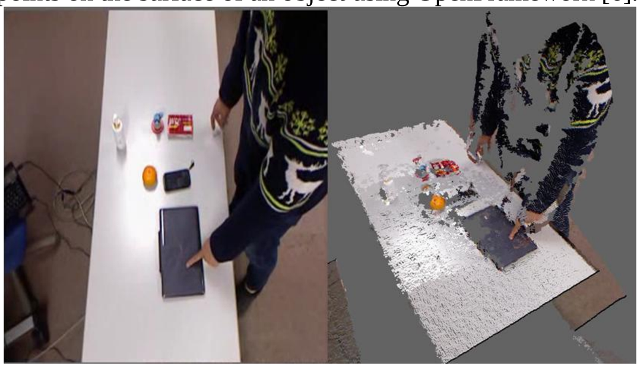

Figure 4. 3D Point Cloud Data

The system obtains RGB data from two Kinect sensors (Figure 4) and assigns them to the depth area. However, not all directions of the object can be reconstructed. Thus, we find the most appropriate location for the Kinects and position them so that they cover most of the experimental space.

1) Touch gesture and recognition: Our touch pair system recognizes the touch actions of users' hands based on depth. The system calculates the depth of each point between a user's hand and the object by filtering closer data. The flow of recognition is as follows.

a) Calculation of all depth points: Calculate all points of the analog and digital objects and the user's hand on the table.

b) Determination of finger position: The system calculates the minimum and maximum depths of the finger by defined thresholds because we hold our fingers in specific ways when we touch something.

c) Determination of hand position: The system calculates the minimum and maximum depths of all fingers and the palm; from the front view, the system uses depth information from both sensors simultaneously.

d) Determination of grasping: Using depth data on users' hands and on objects collected from both sensors simultaneously, we found certain threshold values for recognizing the act of grasping.

2) Analog-digital object pairing: Our system performs time calculations between touched analog-digital objects versus touched object-object pairing. Implementation of object-object touching is shown in Figure 5c,d. When the system recognizes the objects that the user wants to pair, the color is changed. The red color refers to the digital object and the analog object is blue. The recognized hand is shown in yellow using 3D point cloud data. The main steps are as follows.

a) Time calculation: For pairing, the user maintains a touching posture for a few seconds after touching is recognized between the objects. The color is changed after the pairing.

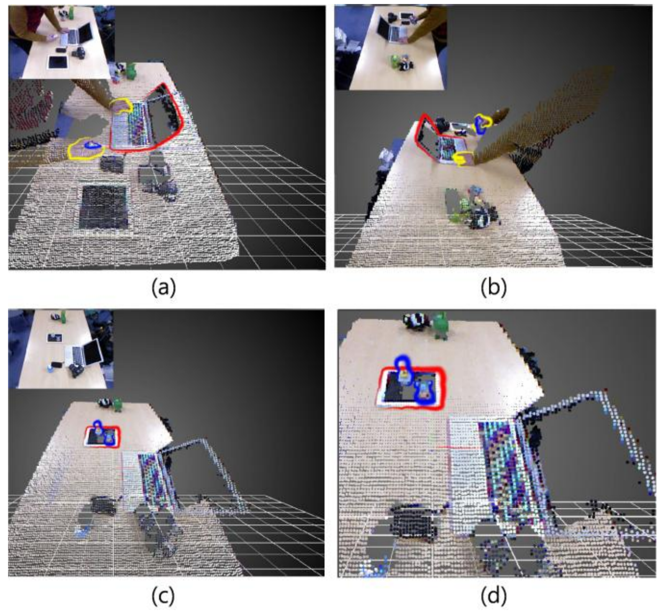

Figure 5. (a) Pairing gesture with hand touching to digital object and grasping analog object from right camera (b) Pairing gesture from left camera (c) Pairing by object-object touching, toy and smart pad are pairing (d) Detailed object-object pairing image

b) Tracking a paired set: To track paired objects, the system calculates 3D point cloud data continuously, which are provided by the real-time reconstruction module (Figure 3). The color of the tracked object is shown.

c) Changing a paired set: To change a paired object set, a pairing gesture is made for some period of time. The user touches what he/she wants to pair. After a few seconds, performing the pairing gesture (Figure 5) will change the pair set as indicated by the color feedback.

IV. TOUCH RECOGNITION TECHNIQUES

In this section, we describe about recognition technology touching 3D object. Our system classified a kind of hand touch patterns. In addition, our system determines whether the object is touched or not in real time. We illustrate about the method recognizing the location of the touched part as well. The system uses RGB-D images from two Kinect cameras and point cloud data for recognizing.

A. Touch patterns of object

The pattern of the touch is various and it is different according to each research. In this paper, we classify patterns into three types of touch in order to use tangible interaction with existing object. Our system classifies into finger touch, hand touch and grasp. Finger and hand touch are top-down touch (Figure 4). Finger touching only or finger and palm touching are distinguished. The above mentioned touch patterns are quite simple touch patterns. On the other hand, grasp touch is more complicated and has various patterns. It

2014, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

happens when we pick the object up or hold it. For example, picking a pencil and grasping a P.E.T bottle are considered grasp touches. Grasp touches can be classified according to object's hardness and size. We touch fingers and palm to the object when we pick the cup up and touch fingers only when we pick the pencil up. A grasp touch can be classified as finger touch or hand touch.

In this research, the system stores user's touched pattern and location value for making tangible objects from existing objects. The touch patterns are classified as described above and become the fundamental initial point for the interaction.

B. Touch Recognition Flow

Our proposed system classifies three parts (Figure 3). The system performs these methods in order. First is 3D calibration and reconstruction. The system performs calibration process on the data from RGB-D camera1 and camera2. The data from RGB-D camera2 rotates 180 degree based on y-axis because camera1 and camera2's data location are opposite.

The system makes points in each data based on 3D model and vision image when the system performs rotating (see red points and blue points in Figure 6b). An object id is generated using 3D shape and vision image data in real time. The system can recognize 3D location using point cloud data. We can store the touched position of the object by using these data. The stored database consists of a structure similar to Table I. The system reconstructs hands model using point cloud data (see Figure 6e) from two cameras. The hand index is assigned to each hand. The mesh and depth value of each finger in hands are stored to database. It calibrates objects using vision and depth image from two cameras for generating the object id. There is 3D value such as mesh, depth as well from each camera. It provides 3D shape, location and vision image of each objects. The system can discriminate objects using these data. In result, the stored 3D object database is used for touch recognition, touched position of object and learning gesture as well. The second is touch recognition and gesture learning. The touch recognition recognizes by using the point cloud and reconstructed 3D model estimation. Our system uses point cloud data when we do simple touch recognition. It is that case of finger touch or hand touch. The system recognizes the location of hand and objects using depth based point cloud data. We defined the depth value of an object surface as $D_{surface}$. It defined the maximum value as D_{max} that is slightly above the $D_{surface}$. It defined upper part of finger joint as D_{min} as well.

TABLE I. 3D OBJECT DATABASE STRUCTURE

	Kinect 1	Kinect 2
Object	Hand Index (Left or Right hand) Hand Value(Mesh, Depth value) Calibrated Object Vision Image and ID	
	Mesh value Depth value Vision image	Mesh value Depth value Vision image

TABLE II. LEARNED TOUCH GESTURE DATABASE STRUCTURE

	Kinect 1	Kinect 2
Object	Hand Index (Left or Right hand)	
	Finger [0][0,1,2](Thumb) Finger [1][0,1,2] (Index) Finger [2][0,1,2](Middle) Finger [3][0,1,2](Ring) Finger [4][0,1,2](Little) Palm value Vision image	Finger [0][0,1,2](Thumb) Finger [1][0,1,2] (Index) Finger [2][0,1,2](Middle) Finger [3][0,1,2](Ring) Finger [4][0,1,2](Little) Palm value Vision image

TABLE III. DEFINED INTERACTION DATABASE

	Object
Interaction	Interaction type Gesture-Function Pair ID Touch Pattern Vision image

The system determines when users touch the object surface when the finger joint is between D_{max} and D_{min} [16]. The touched area is between hand and object point cloud data (see red points in Figure 6d). The hand touch is recognized because all finger joints value and palm value are between each $D_{surface}$ value. These recognized and touched data can be stored when the user wants to make touch gestures. The database that is stored by our system consists of structure similar to Table II for learning touch gestures. The table value is stored when the user makes gesture with touching object. The hand index in the Table II means discriminating touched hand with object. The touched finger value with object are stored as mesh, depth and vision value. The first index of finger's array represents thumb to little finger. The second index of finger's array represents the joint of the finger. Palm value recognizes and stores when the user touches the object with finger and palm. The stored vision images and each value are used for recognizing object as well.

$$\text{Finger }[i][j]-\text{Threshold} \leq \text{Finger }[i][j] \leq \text{Finger }[i][j]+\text{Threshold} (i \neq 0) \quad (1)$$

$$D_{min}[i][j] \leq D_{Finger[i][j]} \times 0.95 \leq D_{max}[i][j] \ (i \geq 0) \quad (2)$$

Finally, the system provides learned gesture recognition and detects defined interactions. These methods refer to the stored database (Table II). The system recognizes touch gesture when the user does the same gesture as the user-defined touch gesture. However, it is difficult to touch exactly the same position as defined. Therefore, we implemented the recognition formula (see formula (1)). The system tracks the touched position from thumb to little finger except thumb. Each touched finger between the thresholds (formula (1)) can be detected. An appropriate value for threshold is 0.3 for each depth value (x,y,z axis value). It had good result in all experiments. After detecting learned touch recognition, the system tracks the user's gesture and records mesh and point cloud motion. The touch gesture motion and interaction can be paired by user. The interaction type is determined as well.

2014, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

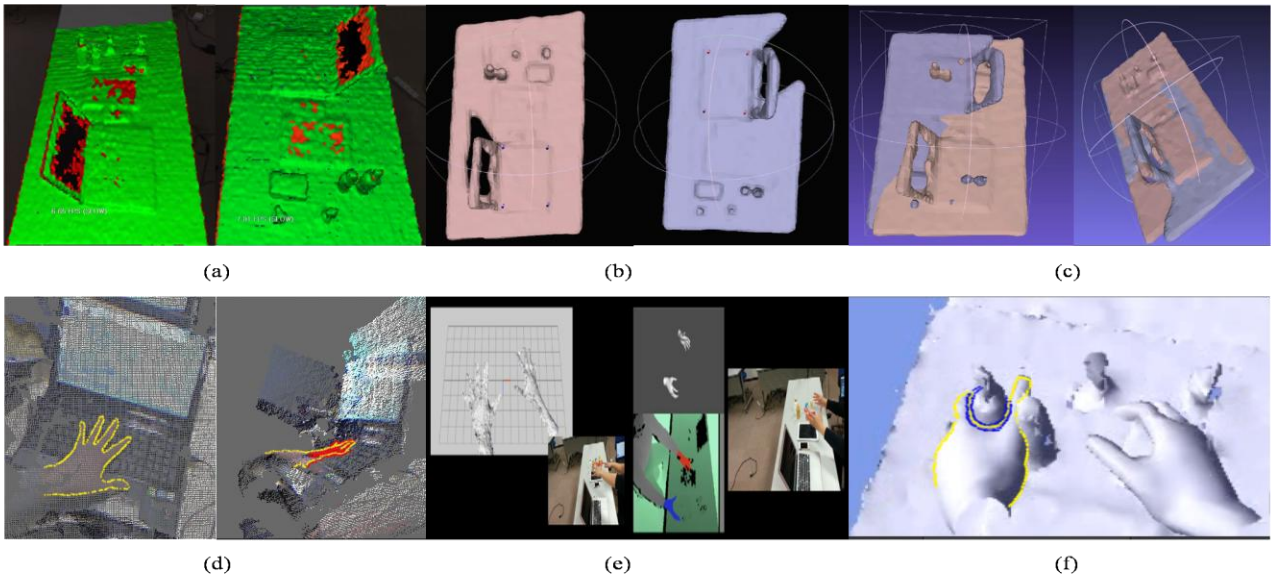

Figure 6. (a) 3D scanning using depth image from RGB-D camera1 and camera2 (b) Point cloud based reconstructing objects from two cameras (c) Calibrating two different direction RGB-Depth data and hole filling processing (d) Hand touch recognition with calibrated point cloud data (e) Hand shape learning and modeling (f) Grasp touch recognition using point cloud based 3D modeling estimation

The interaction returns feedback when the user does touch gesture using these databases.

C. *Touch Recognition Technique*

The touch recognition is conducted using point cloud data and 3D model estimation. Each finger is divided into three parts such as the finger joint for robust touch recognition (see Figure 7b). Each finger joints have 3D point cloud data. Our system tracks the location of the touch between the joint and the object in each joint. It then calculates the number of touched point cloud data and their motion for each joint. Next, it determines touch recognition based on formula that we found empirically (see formula (2)). The system determines whether 95% of the finger joint cloud data are touched in the same position that the user defined or not for each finger joints. Our system can estimate the area that cannot be seen by the camera because the system knows D_{min} and D_{max} value of each finger joint. The finger point cloud data between D_{min} and D_{max} is considered touched even if cameras cannot see the entire area. The system can recognize a 3D touched area with an object using this proposed method. The error rate of recognition is low because each finger joint is managed separately. The system can determine using three finger joints data even if an one finger joint cannot be well recognized.

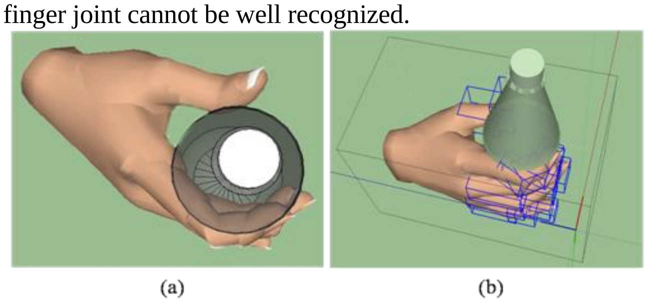

Figure 7. (a) Grasp touch object (b) Dividing finger joint of each fingers

The system uses middle joint of each finger mainly when all the three finger joints touch with object as well. Because the system can estimates the first and third joints point cloud data using middle joints data when all finger joints touched (see Figure 8b).

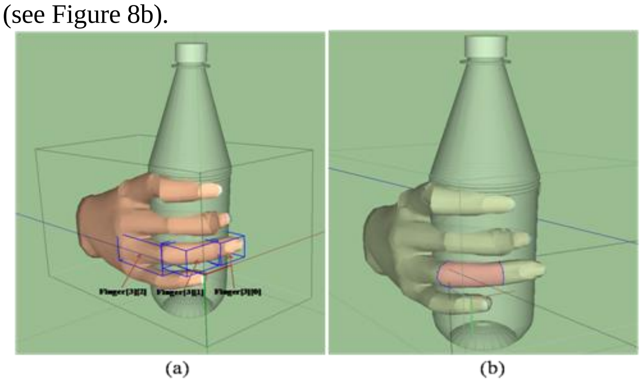

Figure 8. (a) Finger joint of ring finger (b) Point cloud data of second joint in ring finger

D. *Tangible Interactions*

After pairing objects, the system can be used in various tangible interactions. We described tangible interactions using the proposed touch recognition technique. We paired the interaction functions to the learned touch gesture using touch pairing. We implemented five types of interactions; flight game, map control, music control, instrument and drawing using existing object touching gestures. We describe in detail their functions and gestures.

1) Flight game and map control interaction: We designed a flight joystick using analog objects. Such a joystick can be used when playing a flight or shooting game. Our system made a pairing between a notebook PC and a lotion bottle with a cap as a game controller. The game can

2014, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

be controlled by pushing the lotion bottle's cap (see Figure 9b) and moving it. The movements can be recognized based on the cloud data for the object and the hand. We also designed a map controller with paired objects (see Figures 5d and 9c). Our system tracks the two toys and the touched position; the map is moved to provide feedback.

2) Music player interaction: We implemented a simple interaction: a volume controller using toy. The system stored the toy object by calibrating and reconstructing. The user performed clockwise or counterclockwise rotations for learning gesture. The system stores touched location of object and hand motion of the gesture to the database. The user makes connection between the gesture and volume up or down interactions. After that, the music player's volume is changed when the user does a clockwise or a counter clockwise touch gesture (see Figure 9d).

3) Instrument interaction: We implemented an instrument controller using existing objects. The objects represent parts of drum instruments in this interaction. The multiple objects touch recognitions are conducted for doing learning gesture. These recognitions are used not only depth and mesh values of objects but also vision data of objects, because it has to discriminate objects and multiple touch recognition with those objects. There are eight instruments in the drums on the tablet (see Figure 9e). Each existing object is defined to control two instruments. The user assigns touch gesture to the object. This interaction can control the volume and play styles by using learned gesture.

4) Drawing interaction: A drawing interaction is provided. The interaction is performed on the existing pen object as input device. Our system tracks the pen's movement to draw the interaction feedback. The drawing is performed when the user does writing action on the desk. We can write the character by making a writing action. The proposed interaction can change the pen color and control the thickness by touching the pen (see Figure 9f). The system obtains color changing feedback when the user draws after touching a user-defined location on the pen. We can extend to adding more function using touch recognition technique. It can be used as real tablet pen.

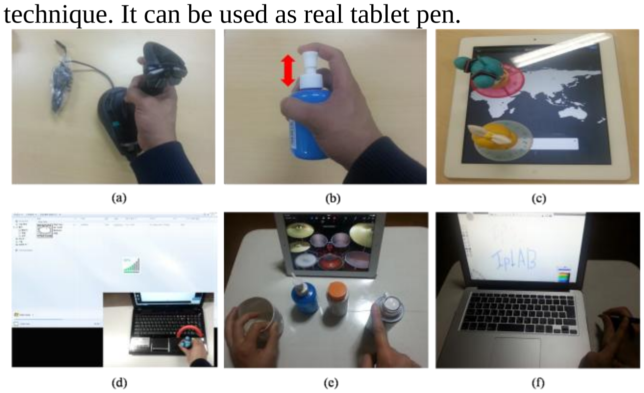

Figure 9. (a)Flight game joystick (b) Paired object controller as flight game joystic (c) Map control (d) Music player interaction (e) Drum instrument interaction (f) Drawing interaction

V. EVALUATION

We evaluated the touch pair system focusing on recognition accuracy. We evaluated touch recognition while changing the analog and digital objects alternately. We also evaluated 3D object recognition and tracking to demonstrate the system's usability and robustness.

A. *Pairing Recognition Accuracy Experiment*

Four analog objects and three digital objects were used. We evaluated finger touching, hand touching, grasping, and object-object touching for each object. The experiments were performed on computers (Intel Core i7 CPU, 2.5 GHz, and 8.0 Gb RAM) using two Microsoft Kinect sensors for Xbox.

We performed the experiments with 10 volunteers. We explained each touch pairing method. Then every volunteer performed four touching gestures to each object 100 times. We defined three second as the period for completing pairing via touching. When the system recognized a pairing, it showed color feedback. Touch recognition success alone was not counted. The participants were allowed to touch analog objects only with their fingers and digital objects only with their hand. Table IV shows the average pairing recognition success for the 10 volunteers.

Our proposed system showed >90% pairing recognition with the objects provided. We found that the average finger-based pairing recognition rate was higher than hand-based pairing. Finger pairing was recognized best when one or two fingers were used. Hand pairing requires checking whether the palm is touching. Thus, hand-based pairing recognition was less accurate than finger-based pairing. In addition, the success rate for touching a smartphone was lower than that for touching the other objects. This may be due to the size of the object. Most adult hands are bigger than most smart phones. Thus, it becomes difficult for the system to find the positions of the fingers and palm.

TABLE IV. FINGER AND HAND TOUCH PAIRING RECOGNITION RESULT

Digital / **Analog**	**Note PC**		**Tablet**		**SmartPhone**	
	finger	***Hand***	***finger***	***Hand***	***finger***	***hand***
Toy	98.3	95.4	99.1	95.3	95.3	93.2
Black Doll	95.7	93.3	96.2	94	92.2	90.1
Green Cube	98.4	96.7	99.3	95.7	96	94
Pet Bottle	96.7	95.3	97.1	95.4	93	91

percentage of recognition rate(%)

Table V shows the results for other pairing methods, such as grasping and object-object pairing recognition. The experiments were performed in the same way as described in Table IV. Grasping-based pairing recognition accuracy was >85% with the objects provided. This method uses point data from many directions. Generally, the front, side, and back surfaces of an object are touched when holding something with the hand. Thus, grasping has to be determined by analyzing the data from many directions.

2014, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

TABLE V. GRASPING AND OBJECT-OBJECT TOUCH PAIRING RECOGNITION RESULT

Digital / Analog	Note PC		Tablet		SmartPhone	
	grasp	*object*	*grasp*	*object*	*grasp*	*object*
Toy	91.4	94.3	89.1	98.1	87.3	94.2
Black Doll	85.4	95.4	85.1	97	85.2	91.1
Green Cube	91.2	96.2	90.1	96.7	87	96
Pet Bottle	90.1	97.1	90.2	97.4	88	94.3

percentage of recognition rate(%)

Thus, as a whole, the recognition rate was lower than Table IV. We also found that the recognition rate differed by object size and hardness. The plastic toy, green cube, and bottle are relatively hard. However, the doll is very soft. When the user touches or grasps a softer object, the system encounters some difficulties in determining the touch depth. Thus, recognition accuracy was lowest for the doll among all objects provided. However, generally, the touch recognition rate was high enough to be uses for touch pairing system and tangible interactions.

B. *3D Object Recognition Accuracy*

1) Single Object Recognition: In this experiment, we present the results for 3D object recognition with an 3D object database. Eighteen existing objects were used for evaluation. We divide by hardness, into 2 group. Each group was divided into three group again by object size. The size based group includes three objects in total. The hardness value 1 means soft objects and value 2 means hard objects. The size value 1 means small size such as pen, eraser and card and value 2 means normal size such as beverage can, cup and plastic lotion bottle. 3 means big size such as cap, shoes, books, etc. We do not need to consider very big object because our system considered object that can be put on a desk (see Figure 1). The experiments were performed on a computer with two Intel Core i7 CPU 2.5GHz and 8.0 GB RAM computers and two Microsoft Kinect for Xbox 360. We performed the experiments with ten volunteers. We explained each touch method thoroughly. We touch the object by each touch method and stored 3D object database to all object. The place of object where volunteers touched is indicated by a sticker. The sticker shows the place to be touched by the volunteers in recognition experiment. A volunteer performed 10 times three touching methods to each object. We checked whether system recognizes touching or not. The graph shows the average of the number of successful recognition of every object from ten volunteers.

In result, we can obtain good recognition result in Figure 10. We can see that the result is lower in size 1 in comparison with another size. The grasp touch is lower in each size as well. This is due to the fact that a small sized object cannot be seen by the camera from all the angles. The area that cannot be seen is estimated. The grasp touch recognition is more complicated compared with other touch recognition as well, because the system has to consider more than two sides when grasp touch is occurring.

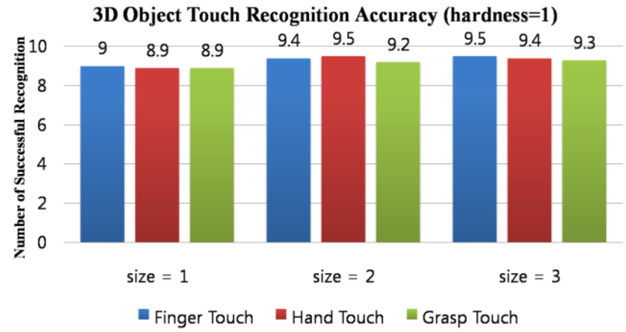

Figure 10. Touch Recognition Accuracy in hardness one

We obtained a higher result in Figure 11 than Figure 10. All size and touch methods scored over 90% successful recognition results, as seen in Figure 11. The recognition rate of hard objects is higher than soft objects, because the depth value was a little changed when the volunteer touched soft objects. As a result, the hard and big size object were obtained almost highest recognition rate.

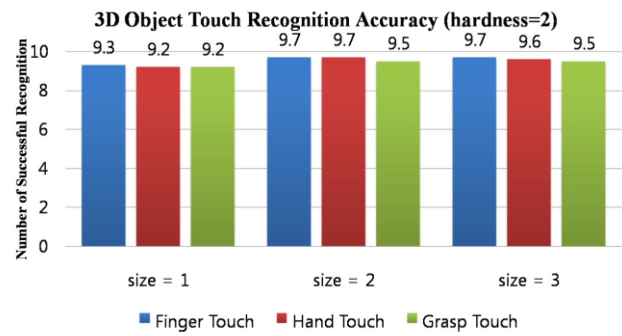

Figure 11. Touch Recognition Accuracy in hardness two

2) Multiple Object Recognition: The volunteers choose the objects that they want to use. The object size and hardness was not considered for natural situation, because the size and hardness of objects are not considered so much when we use existing objects in real life. In this experiment, we have already stored all objects in 3D object database and position that are indicated by a small sticker. They choose two objects at first. After that, they touch one object of them. The system checked whether it recognizes touching or not and the correct object id (see Table I). Next, they choose two object that they want to use. After that, they do same way. The objects number are increased from three to five and repeated the same experiment. The experiments were done ten times for each group of object. The result graph shows the average number of successes with different touch method from ten volunteers. The obtained result was over 90% for all situations (see Figure 12). The result shows that the number of objects did not have an effect in the recognition accuracy. We limit the number of objects to

2014, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

five because of the size of the desk in this experiment. However, the recognition rate does not get lowered so much even if the number of object is increased. The object size and hardness have an effect to the recognition accuracy more than object number.

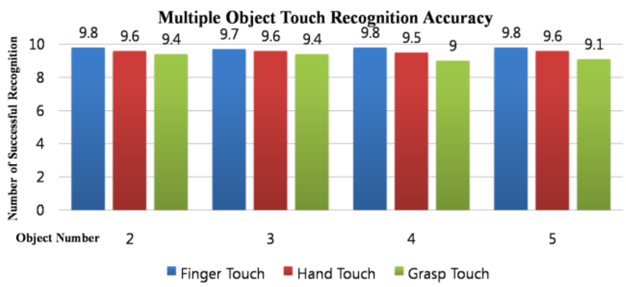

Figure 12. Multiple Object Touch Recognition Accuracy

3) Real-time 3D Object Tracking Accuracy: We evaluated object tracking after pairing. We moved analog objects during a 10 minutes experimental period (e.g., left-right, front-back).

Figure 13 shows the recognition accuracy for these tests. We obtained recognition rates of >80% for all objects. Since we used two Kinect sensors for real-time 3D object reconstruction and data comparison, we obtained low error rates.

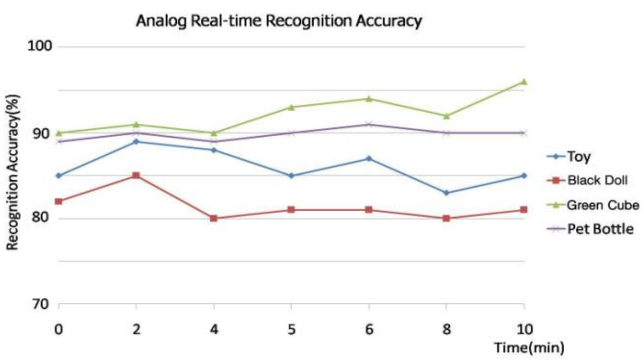

Figure 13. Analog Objects Tracking Accuracy

Figure 14 shows the recognition accuracy for three digital objects over 10 minutes. Smart phone recognition was <80% in around 4 minutes and 8 minutes after tracking.

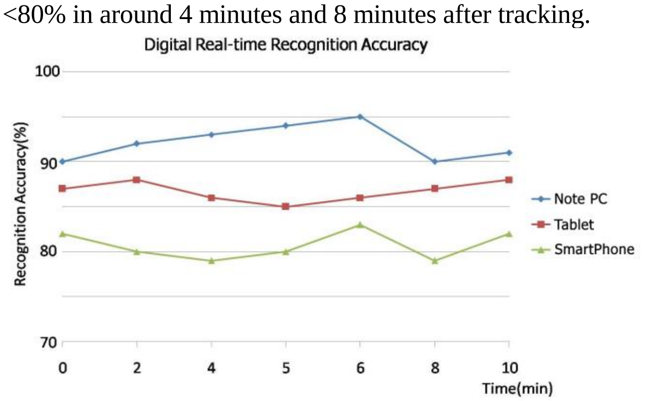

Figure 14. Digital Objects Tracking Accuracy

This is because the user was holding the smart phone with his/her hand. In particular, when we moved the smart phone with a front-back motion, the recognition rate decreased.

C. Learned Gesture and Tangible Interaction Recognition

In this experiment, we evaluated touch gesture recognition accuracy for tangible interaction using learned touch gesture database (see Table II). We defined the touch gesture and the system learned the touch gesture to store it in the database. We implemented five interactions such as flight game, map controller, music controller, instrument controller and drawing controller. We stored several gestures to each interaction. The flight game interaction was push gesture as shooting missile, moving gesture as moving flight. The music controller was clockwise as volume up, counterclockwise as volume down and double tap as play or pause. The map controller was same gesture with music player controller as moving map, zoom in or zoom out. The instrument controller was tap as play each instrument in the drum set, grip gesture as changing sound effect and swipe top from the bottom edge or down from the top edge as volume control. The drawing interaction was grip pen gesture in drawing mode, swipe down from top as pen's thickness changing and double tap for color changing. The defined gestures are evaluated by ten volunteers. They performed each gesture 10 times in the interaction. We checked whether the system recognizes the gestures or not. After that, we checked whether the correct feedback happened or not. The graph shows the average number of successes for each gesture in the interaction from ten volunteers. We obtained over 90% successful result (see Figure 15). The tap and double tap gestures to object are perfectly recognized. The drawing interaction and instrument interaction have similar gestures. The drawing interaction obtained lower rate than instrument interaction. The result was due to the object size, because it was difficult to capture all touched locations and motion of the pen object in drawing interaction.

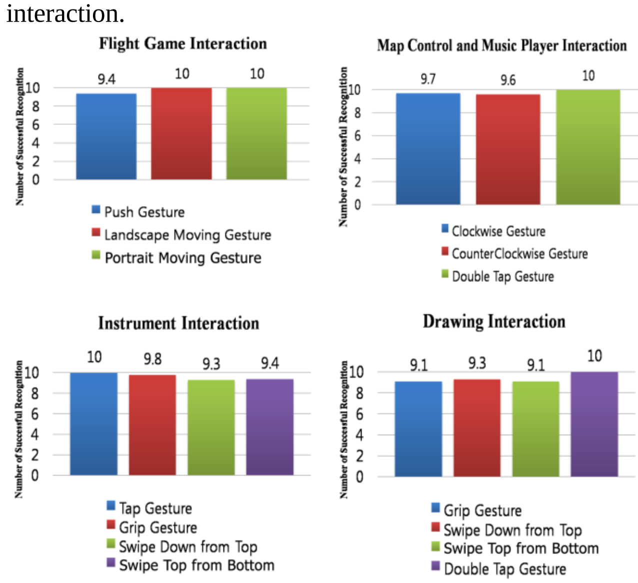

Figure 15. Recognition Accuracy of Each Interaction Gestures

2014, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

We found that the object size and hardness are important for touch and gesture recognition in our system.

D. Discussion

In our experiments, we found out that the touch and gesture recognition with 3D objects can have good recognition accuracy rate. Our proposed method recognized 3D object touched location and learned gesture using RGB-D Kinect cameras without attaching additional sensors. We obtained good results over 90% successful recognition in all experiments. We found that the size and hardness of objects have an effect on the experiment results. The number of objects had no effect on the experiment results. The big sized and hard object's area can be captured for better touch recognition. The recognition rate is high because it is not an estimation. The estimated area of object for touch recognition is increased when the object is small or soft. In this case, the recognition rate is lower than the recognition rate for a big or hard object. However, the results between each experiment do not need to consider so much for using natural tangible interactions. The proposed system can cover their interactions and gestures.

In real life, we expected our natural actions to be used as interactions in general. For instance, a feedback happens when we grip the cup. The interaction occurs when we tap objects once or twice. A feedback is generated when we do clockwise or counterclockwise gestures with objects as real controller. It is interesting and useful when actions that used in real life are used.

The proposed interactions are designed by taking these points into consideration. The natural gestures are connected with interactions. The music controller, instrument controller and drawing interaction are functions that are frequently used. The designed gestures are friendly and frequently used as well. Therefore, the volunteers who participated to evaluation practiced well and became experts in using the proposed gesture and functions in learned gesture experiments. As a result, we obtain good results and proved the system's usability and robustness.

VI. CONCLUSION AND FUTURE WORK

In this paper, we described a new touch recognition techniques and dynamic pairing method using two Kinect camera based on 3D point cloud data. We described the touch pattern, touch system architecture and databases for recognition. We implemented tangible interactions using learned gestures that are stored by users with the proposed techniques. We did eleven types of experiments in finger, hand and grasping touch to evaluate our proposed system usability and prove the recognition robustness.

The accuracies were >90% for finger and hand touching and >85% for grasping and object-object touching. Almost the same results were obtained when we changed the locations of pairs dynamically. We obtained good real-time 3D object tracking results as well, despite using objects of different size, shape, and hardness. The accuracies were >89% for single object recognition in hardness value 1 and >92% in hardness value 2. We obtained the accuracies were >90% for multiple object recognition and >91% for learned gesture recognition. Especially, we obtained good real-time tangible interaction with good gesture recognition results in many situations as well, despite using objects of different size, shape and hardness. In addition, a remarkable result is the ability to implement tangible interactions with functions and frequently used gesture without installing any additional sensors to the multiple object. The contributions of the present study can be summarized as follows.

First, we have provided a new pairing method. Using this method, we can dynamically pair analog and digital objects, and various tangible interactions can be achieved.

Second, we presented a new touch recognition technique. There are many researches that recognize touching and motion gesture. However, they need to attach additional sensor to the objects and cannot recognize well touched location of object from users. They cannot recognize natural gesture with touching existing object as well. By proposing this recognition method between objects, the various tangible interactions can be extended. Third, it can recognize multiple object touching and touching gestures. We can dynamically change the object that we want to be tangible object in real time. It means the proposed technique is natural and robust when we use it in real life, because we used multiple objects and gestures that we are used to in everyday life. We do not need to learn gestures or make sensor based object for tangible interaction. Our actions and existing objects are enough, and can be used for tangible interaction simply using proposed techniques.

Finally, in the experiments, we obtained good 3D object recognition rate and learned gesture recognition results. We evaluated three pattern of touch and object type such as by size or hardness for natural using and robustness in real time recognition. The learned touch gestures were evaluated with practical interactions such as flight game, map controller, music player, instrument controller and drawing. The users can expect practical and natural use of existing object as tangible objects because we obtained good results in recognition experiments. In near future work, we can expect various practical tangible interactions by using our proposed touch recognition system. The augmented graphical support interface and interaction can be implemented with existing objects. The existing objects can be remote controllers to control devices that are located at home using touch recognition system as well. For instance, the cup can be television controller, and then changed to light controller. The combination of interface with head mounted display is widely used; natural and interesting touch interaction can be expected.

REFERENCES

[1] U. Lee and J. Tanaka, “TouchPair : Dynamic Analog-Digital Object Pairing for Tangible Interaction using 3D Point Cloud Data,” in ACHI’14 The Seventh International Conference on Advances in Computer-Human Interactions, Mar. 2014, pp. 166-171.

[2] E. S. Martinusseon, J. Knutsen, and T. Arnall, “Bowl: token-based media for children,” in DUX '07 Proceedings of the 2007 conference on Designing for User eXperiences, Nov. 2007, Article No. 17.

2014, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

[3] R. Wimmer and S. Boring, “HandSense: discriminating different ways of grasping and holding a tangible user interace,” in TEI '09 Proceedings of the 3rd International Conference on Tangible and Embedded Interaction, Feb. 2009, pp. 359-362.

[4] R. Wimmer, “FlyEye: Grasp-Sensitive Surfaces Using Optical Fiber,” in TEI '10 Proceedings of the fourth international conference on Tangible, embedded, and embodied interaction, Jan. 2010, pp. 245-248.

[5] OpenNI 3D sensing framework : Last visited on Nov, 2013. http://www.openni.org.

[6] openFrameworks : Last visited on Aug, 2013. http://www.openframeworks.cc.

[7] A. D. Wilson, “Using a depth camera as a touch sensor,” in ITS '10 ACM International Conference on Interactive Tabletops and Surfaces, Nov. 2010, pp. 69-72.

[8] P. Wellner, ‘’The digitaldesk calculator: Tangible manipulation on a desk top display,” in 4th annual ACM symposium on User interface software and technology UIST, Nov. 1991, pp. 27-33.

[9] I. Siio and Y. Mima, “Iconstickers: converting computer icons into real paper icon,” in 8th International Conference on Human-Computer Interaction: Ergonomics and User Interfaces, 1999, pp. 271-275.

[10] P. Ljungstrand, J. Redstrom, and L. E. Holmquist, “Webstickers: using physical tokens to access, manage and share bookmarks to the web,” in Designingaugmented reality environments DARE 2000, Apr. 2000, pp. 23-31.

[11] T. Nishi, Y. Sato, and H. Koike, “Interactive object registration and recognition for augmented desk interface,” in IFIP conference on human-computer interface Interact 2001, Mar. 2001, pp. 240-246.

[12] F. Klompmaker, K. Nebe, and A. Fast, “dSensingNI: a framework for advanced tangible interaction using a depth camera,” in TEI '12 Proceedings of the Sixth International Conference on Tangible, Embedded and Embodied Interaction, Feb. 2012, pp. 217-224.

[13] S. Azenkot, and S. Zhai, "Touch behavior with different postures on soft smartphone keyboards," in MobileHCI ’12, Sep. 2012, pp. 251-260.

[14] L. Cheng, F. Hsiao, Y. Liu, and M. Chen, "iRotate grasp: automatic screen rotation based on grasp of mobile devices," in UIST ’12 Adjunct, Oct. 2012, pp. 15-16.

[15] M. Goel, J. Wobbrock, and S. Patel, "GripSense: using built-in sensors to detect hand posture and pressure on commodity mobile phones," in UIST ’12, Oct. 2012, pp. 545-554.

[16] A. D. Wilson, "Depth sensing video cameras for 3D tangible tabletop interaction," in Proc. IEEE Tabletop 2007, Oct. 2007, pp. 201-204.

[17] A. D. Wilson, "Using a Depth Camera as a Touch Sensor," in ITS '10, Nov. 2010, pp. 69-72.

[18] A. D. Wilson, "PlayAnywhere: a compact interactive tabletop projection-vision system," in Proc. ACM UIST 2005, Oct. 2005, pp. 83-92.

[19] O. Hilliges, S. Izadi, A. D. Wilson, S. Hodges, A. Garcia-Mendoza, and A. Butz, "Interactions in the air: adding further depth to interactive tabletops," in Proc. ACM UIST2009, Oct. 2009, pp. 139-148.

[20] P. Dietz, and D. Leigh, "DiamondTouch: a multi-user touch technology," in Proc. ACM UIST 2001, Nov. 2001, pp. 219-226.

[21] C. T. Dang, M. Straub, and E. André, "Hand distinction for multi-touch tabletop interaction," in Proc. ACM ITS 2009, Nov. 2009, pp. 101-108.

[22] H. Benko, and A. D. Wilson, "DepthTouch: using depth sensing camera to enable freehand interactions on and above the interactive surface," in Poster Presentation at the IEEE on Tabletops and Interactive Surfaces '08, Mar. 2008.

[23] A. Agarwal, S. Izadi, M. Chandraker, and A. Blake, "High precision multi-touch sensing on surfaces using overhead cameras," in Proc. IEEE Tabletop 2007, Oct. 2007, pp. 197-200.

[24] A. Butler, S. Izadi, and S. Hodges, "Sidesight: multi-touch interaction around small devices," in Proceedings of UIST ’08, Oct. 2008, pp. 201-204.

[25] K. Kim, W. Chang, S. Cho, J. Shim, H. Lee, J. Park, Y. Lee, and S. Kim, "Hand Grip Pattern Recognition for Mobile User Interfaces," in Proceedings of the National Conference on Artificial Intelligence, 2006, pp. 1789-1794.

[26] P. G. Kry, and D. K. Pai, "Grasp recognition and manipulation with the tango," in 10th International Symposium on Experimental Robotics 2006, Jul. 2006, pp. 551-559.

[27] H. Ishii, and B. Ullmer, "Tangible bits: towards seamless interfaces between people, bits and atoms," in SIGCHI conference on Human factors in computing systems, Mar. 1997, pp. 234-241.

[28] P. Mistry, T. Kuroki, and C. Chang, "Tapuma: tangible public map for information acquirement through the things we carry," in Proceedings of the 1st international conference on Ambient media and systems, Ambi-Sys '08, Feb. 2008, pp. 12:1-12:5.

[29] H. Song, H. Benko, F. Guimbreti`ere, S. Izadi, X. Cao, and K. Hinckley, "Grips and gestures on a multi-touch pen," in CHI'11, May. 2011, pp. 1323-1332.

[30] S. Hwang, A. Bianchi, and K. Wohn, "MicPen:pressure-sensitive pen interaction using microphone with standard touchscreen," in CHI EA ’12, May. 2012, pp. 1847-1852.

[31] S. Izadi, D. Kim, O. Hilliges, D. Molyneaux, R. Newcombe, P. Kohli, J. Shotton, S. Hodges, D. Freeman, A. Davison, and A. Fitzgibbon, "KinectFusion: real-time 3D reconstruction and interaction using a moving depth camera," in UIST ’11, Oct. 2011, pp. 559-568.

[32] P. Lopes, R. Jota, and J.A. Jorge, "Augmenting touch interaction through acoustic sensing," in ITS ’11, Nov. 2011, pp. 53-56.

[33] A. Wilson, "TouchLight: An Imaging Touch Screen and Display for Gesture-Based Interaction," in ICMI 2004, Oct. 2004, pp. 69-76.

[34] A. Wilson, "Robust Computer Vision-Based Detection of Pinching for One and Two-Handed Gesture Input," in UIST 2006, Oct. 2006, pp. 255-258.

[35] B. Piper, C. Ratti, and H. Ishii, "Illuminating Clay: A 3-D Tangible Interface for Land-scape Analysis," in CHI 2002, Apr. 2002, pp. 355-362.

2014, © Copyright by authors, Published under agreement with IARIA - www.iaria.org