International Journal of Applied Engineering Research ISSN 0973-4562 Volume 12, Number 15 (2017) pp. 4976-4984

Unified Input Processing Framework for Multi-User Multimodal Interaction on Large Display

[1]Kiyoung Seo and [1*]Kyoung Shin Park

[1]Department of Applied Computer Engineering, Dankook University,
152 Jukjeon-ro, Suji-gu, Yongin-si, Gyeonggi-do, 16890, South Korea.

[*]Corresponding author

Abstract

This paper proposes the unified input processing framework to enable multimodal, multi-user simultaneous interactions on a large display. This framework defines most common interaction schemes and maps input device's functionality to the common input events in the application. It is designed for a transparent and seamlessly transferrable interface mechanism to enable multiple users to interact with the application in the same ways using various input devices. That is, users can use any input device that mostly suits for them based on the spatial and task constraints. This framework currently supports Kinect-based gesture, a multi-touch panel, and a mobile device. The feasibility of this framework is also discussed through the development of collaborative applications that support multiple user interactions with various input devices.

Keywords: Unified Input Processing, Multi-User Interactions, Multiple Input Devices, Public Display

INTRODUCTION

Nowadays the displays tend to be large and high resolution as the high-quality display hardware has dropped in price to an affordable point. With the recent display technology advances, the size and resolution of the display is increasing steadily due to the demands for high-quality and high-resolution image. As technology advances increase the popularity of affordable high-performance computing and various display systems, it has resulted in widespread of public display deployments [1]. However, most of public displays are mainly used for displaying the information to the public without user interaction.

With the increased uses of public displays, they become more interactive and the user interaction issues have received more growing attention in human-computer interaction (HCI) research. The current research and development of user interaction techniques for the public display are widely ranged, such as physical movement[2], laser-pointer[3], multi-touch screen[4], mobile device[5], 3D gyro-mouse[6], motion sensor based interface[7], vision based movement[8] and gesture interface [9]. While these works gave us valuable insights, they have mainly focused on particular input devices or systems.

There have been a number of studies investigating on collaborative user interfaces on the public display, which is large enough to accommodate multiple users. But, many cases are conducted on the table-top display[10] or the tiled display[11], which require specific middleware to build such applications. Therefore it is difficult to directly apply them to other kinds of applications that require various interactions. In this research, its main objective is to support multiple users with different input devices to have the same opportunities to interact with the large public display and work independently without affecting each other.

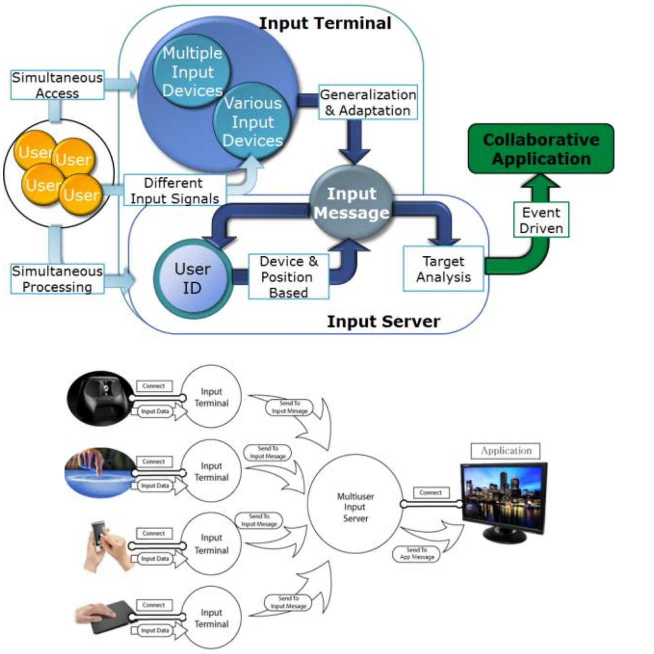

Figure. 1 Diagram of the Unified Input Processing Framework for Multimodal Multi-User Simultaneous Interactions on Large Display

Figure 1 shows the block diagram of the unified input processing framework that enables multiple users to simultaneously and collaboratively interact with the application using various input devices. This framework is composed of the input terminal and the input server. The input terminal is dedicated for each input device. Each input terminal receives input signals from a particular input device and converts the signals to an appropriate input message with the additional information which is then delivered to the input server. The input server manages all input terminals and passes the user's inputs to the target collaborative application. It aggregates all the incoming input messages and translates them to the application commands or events to trigger user interaction. The collaborative application should include the event handler to process multi-user inputs separately.

This paper is organized as follows. First, it describes the related works of the multi-user collaborative interfaces and interactions for the large public display. It then discusses the design issues and implementation of the unified input processing framework. It also describes two collaborative public display applications written based on this unified input processing framework. Finally, the conclusions and future directions are summarized.

RELATED WORK

The large public display allows a number of users to simultaneously observe and interact with the application. It makes the collaborative work more dynamic and intuitive. There have been many collaborative user interfaces studied for the large public display, such as laser-pointers[3], multi-touch screens[4], mobile devices[5], and vision-based gesture interface like Microsoft Kinect[9]. The most commonly used user interface for the public display is a 3D gyro mouse as it enables the user interaction far away from the screen and is easy to use[6]. However, the input data coming from the 3D gyro mouse only supports a single user interaction.

Lumipoint developed by Stanford University supports multi-user interactions with a low-cost laser pointer input device on the Interactive Mural tiled display[3]. In this system, an array of cameras is used to track all laser pointing locations on the display. The tracked data is also sent to a central estimator to produce a consistent set of user input events, such as beginning, continuation or end of a stroke. This system supports laser pointers distinguished by color for the interaction of multiple users for a simple multi-user whiteboard application.

Cyber-Commons developed at UIC Electronic Visualization Laboratory is a 20-foot wide multi-touch tiled display system that allows multiple users to play the Fleet Commander game at the same time[12]. It is a collaborative wall display driven by the SAGE (Scalable Adaptive Graphics Environment) framework[13]. Cyber-Commons is also used with the mobile device, such as iPad, as the paint palette for users to select or mix colors and then draw a painting on the display.

Cooperative gestures is a study of multi-user gesture interactions for co-located groupware on a DiamondTouch tabletop display system[14]. In this research, the CollabDraw application is developed where it attempts to combine the multi-user operations to recognize a single command. For example, two people work together by one's adjusting the thickness of the line while the other's drawing a line directly. However, there are some unexpected flaws found in the user study, such as clearing everything in the entire screen while all users try to perform the erase operation accidentally at the same time.

Arc-Pad is a study of supporting multiple user interactions in the large public display by using the mobile device[15]. It supports the mouse operations on the wall display by using the touch screen of the mobile device. When a user taps a point of a mobile touch screen, it corresponds to a particular point of the wall display, which quickly jumps to move the mouse pointer on the display. The small-step movement operation can be performed by a user's dragging on the mobile touch screen. When a user holds on the touch screen surface, it should not exceed a threshold time in order to properly perform the tab operation. For the drag operation, the drag amount moved by a user's hand on the touch screen has to be over the minimum distance.

CGLXTouch makes it possible to support multi-user touch-based interactions in a large tiled display and a table-top display[16]. It consists of the table-top display and the tiled display configured as display outputs and the mobile device and the table-top display used for input devices. The head node is responsible for processing all users' inputs. The user interaction is processed and delivered to the user's head node via the CGLX input applied to the touch panel. The interaction results are transmitted to the render node of the table-top display and the tiled display, and then the rendered results are also streamed to the display of the mobile device. In conjunction with CGLX, Device Manager allows a large number of users to access the system and enables real-time concurrent user interactions.

Ch'ng investigated the multi-user interactions in the three-dimensional application running on the table-top display[17]. This study found that the users learned through the interaction of the other users. Most users interacted with the object using multiple fingers depending on the size of the object, rather than one finger. Based on this fact, it suggested several user interaction techniques (such as Pinch, Zoom, Swipe, Poke, Drag, and Rotate) to interact with a gesture to the virtual objects in the three-dimensional environment. Interestingly, when the multiple users interacted together on this system, most of them only used one hand.

FlowBlocks is a multi-touch user interface for the interaction of multiple users on a public display[18]. The main focus of this research is to prevent the user errors or conflicts that may arise by multiple users to simultaneously interact through interfaces, in order to make more precise user interactions. Basically, the UI is composed of Widget and Dock where the Widget is inserted into the Dock to make user interaction. It prevents user errors often caused by selecting the incorrect interactions that may occur due to accidental contacts in the public display. In addition, it is possible to form various kinds of controls, such as buttons, links, radio boxes, check boxes, switches, sliders, by combining one or more widgets and docks in a number of ways.

The previous works showed various perspectives of multi-user interactions with the public display using a variety of input devices. However, most studies depended on a specific input device (such as multi-touch or mobile interface) to enable a number of users to control the application simultaneously in real-time. Some explored multi-modal user interfaces (e.g. multi-touch and mobile devices) to enrich user interaction [16]. Other studies investigated the design issues for problems raised during the collaborative work of multiple users [14].

While these works give us insights to design the collaborative interfaces, there has been no research focused on providing different kinds of input device supports for the multiple users to interact with the public display. The public display implies a number of users to simultaneously engage in interacting with a system. That is, every user can participate and contribute without interfering with the others.

DESIGN ISSUES

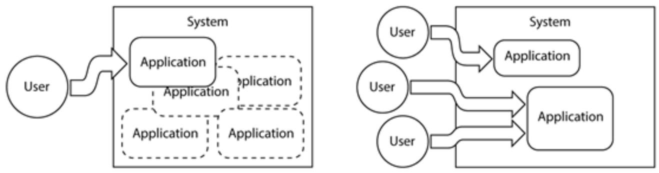

Figure. 2 Single-User Interaction (left) vs. Multi-User Interaction (right)

Figure 2 illustrates the difference between a single-user interaction and multi-user interaction. In a single-user interaction, only one user performs the interaction at a time and the target of interaction is one active application among many applications. In multi-user interaction, not only multiple users perform the interaction but also the target of interactions can be all different applications. Hence, multiple user simultaneous interaction should be processed at once. This section describes the fundamental design aspects that are necessary for input processing of multiple users with different input devices to access various collaborative applications.

1. Generalization of Various Input Devices

There are a variety of applications that meet user's needs and requirements. Such diversity has led to the emergence of a wide variety of input devices that are commercially available today, such as voice, gesture, 3D air mouse, pen digitizers, and haptic devices. However, there are practical limitations with the use of a specific input device alone for multiple users to work together on a large public display. The collaborative application should support more than one input devices accessible at the same time and control in the same way no matter what kinds of input device are used. Hence, the generalization of input device should be considered to adjust different forms of input information from various input devices to eliminate device-specific setting. The generalization maps device-specific function to the common type of input to support the simultaneous use of various input devices.

2. Device-Independent Control

The conventional input devices, such as mouse and keyboard, are designed for single user interaction. The multi-user input devices, like multi-touch panel, are also constrained to the certain number of users due to hardware specifications. The collaborative application should support multiple input devices that suit user's needs rather than bind to a particular input device. In order to enable the use of various input devices, the input processing should support device-independent and mutually exclusive control of the input information. That is, the input from one device should not affect the input from other devices. If it is not possible to rule out the interference between the devices, it is viewed as the same type of input device (such as, two mouse devices connected to a PC).

3. Universal Applicability

The applications are written in various application program interfaces (APIs) that provide the procedures and settings for different input processing. These procedures may have a limit depending on the circumstance. In the multiple-user interaction, however, it is necessary to deliver the input information to the application in the same way regardless of the limitations with the input procedures. Therefore, it is essential to support the input information from various input devices that can be applied to any collaborative application. This means the input processing should provide a similar input mechanism available for different input devices, in order to control the inputs in the same way even if a user interacts with different applications using different input devices.

4. Multiple User Simultaneous Access

The collaborative application should allow multiple users to interact simultaneously with the application. In the collaborative application, the user must be distinguished in order to process the input information of multiple users individually. Since it is not possible to determine what kind of input device is specific to the user, the user identification is determined based on the input device and the location of a user. Also, the user event generated through the input device should be processed by the user individually and processed immediately. Each user interaction can be represented as a cycle from the start to the end of the interaction process. If a user interaction fails to complete the cycle, it cannot achieve the interaction goal. At the same time, if the input information is interfered or altered by other users, it may bring up different results from original desires.

DESIGN AND IMPLEMENTATION

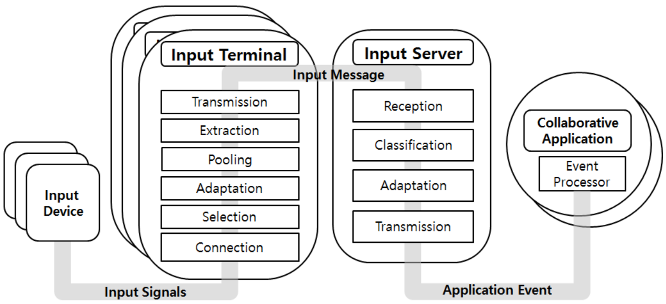

Figure. 3 Overall Architecture of Unified Input Processing Framework

Figure 3 shows the overall architecture and input processing procedure of the unified input processing framework. The input terminal converts different input signals from the device to form the input message. The input server mediates user interactions between the input terminals and the collaborative applications. It consistently separates the input message by each user and transfers it to the appropriate application by identifying the users, overcoming the difference between the input values from different input devices, and accessing the simultaneous input values from multiple input devices.

When multiple users interact with the application using multiple input devices, the input signal delivered to the input terminal is different from one another. However, the multi-user interaction system should embrace all of these incoming input signals. Hence, the unified input message should be formed to support user interactions for different input devices. Also, the input message includes the user identification since users can use the same kind of input device and one input terminal can generate several input messages by multiple users. This user identification places a limit as to grant permission for such interactions to a specific user.

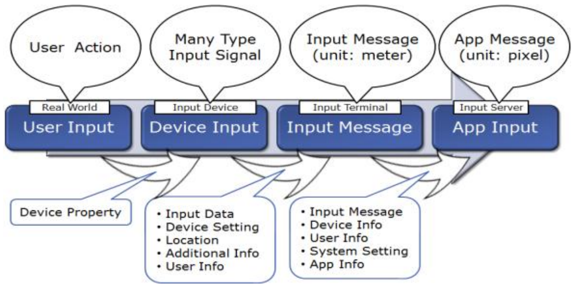

Figure. 4 Information Flows of User Interaction from Input Device to Collaborative Application

1. Input Message

Figure 4 illustrates the information flows of user interaction from the user input to the collaborative application. The input message is a means to deliver user interaction from various input devices to the application by unified input processing. User action is the type of information at the moment of user accessing an input device. As a result of the user input in real world, the input device generates many different types of input signals. The input terminal makes the input message in a generalized form to embrace various input devices. The input server analyses the input message to distinguish the user interaction, applies the user information and system settings, and then sends it to the application.

The input message includes hostname, device name, user ID, position, point count, input points, event, event value, UI ID, and host IP address. Each user's input is separated by user ID since some input devices (such as Kinect and multi-touch panel) can be used by multiple users simultaneously. Position is the actual (x, y, z) value. It is used to identify user to map a user to a target application. It is also used for preventing false recognition of the same user. For example, when employing multiple Kinect devices used for wide-area user tracking, the overlap of the same user can be eliminated by comparing this position.

Point count and input points represent the input status. Point count sets the input target points. Input points are the actual input position (in one meter of the units based on the distance of the real world) and the user state. If input points are not set up properly, user interaction cannot be performed. The input points must be shared and coordinated by all input terminals to ensure consistent user interaction regardless of the type of input device. Event, event value, UI ID, and host IP address are used to convey additional user interaction information.

2. Input Terminal

The input signals are diverse by input devices, and hence the input terminal of the input processing framework makes it easier to adopt new input devices to the multi-user system. The input terminal processes the user inputs from the input device and generates the input message. The input processing procedure on the input terminal is connection, selection, adaptation, pooling, extraction, and transmission phase. The input terminal acquires the input signals from various input devices and it selects valid user input values from the signals. It then makes device-independent unified input data and it accumulates the input data into the buffer or pool to further evaluate user interaction. Finally, it creates the standardized input message and transmits the input message to the input server.

Currently, three types of input terminals are implemented: Kinect, multi-touch panel, and mobile interface. Kinect is the gesture-based multi-user input device. Multi-touch panel is easy to use and recently very popular. The mobile interface can utilize the accessibility based on mobility and various sensors.

Kinect Input Terminal

Kinect consists of microphone, infrared projector and sensor, color camera and voice recognition. The Kinect input terminal utilizes user hand gesture interactions and 3D skeleton tracking. It evaluates user position and gesture states and then selects only validated gesture information and skeleton points. It then adjusts the position and orientation of skeleton values by multiplying the inverse of the 3D transformation matrix followed by subtracting Kinect device's position in order to get the display-based user interaction points (in meter units). These adapted skeleton values are stored in a circular list to further analyze user gesture information. The data stored in the list are then used to create the gesture-based user interaction input message.

Multi-Touch Input Terminal

A typical input signal from a multi-touch panel is provided in the form of a message in the Microsoft Windows system. However, the default Windows applications are not multi-touch interaction enabled. Hence, the multi-touch application must be registered in the operating system in order to process multi-touch inputs. The multi-touch input terminal is written as an application module because the only registered application can access the multi-touch input values. It registers the application handle to the Windows system to receive multi-touch inputs. The touch input contains the identifier for distinguishing touch points and the horizontal and vertical coordinates relative to the top-left point of the multi-touch panel (in 1/100 pixels). Such system-based multi-touch inputs are converted to the display-based user interaction points (in meter units). Finally, it assigns a user ID based on input points that are bound to within a certain distance.

Mobile Input Terminal

The mobile input terminal is a kind of mobile application. It transforms input signals (e.g. touch-panel and sensors from the mobile device) to the input message for multi-user interactions. This terminal utilizes button interactions, touches and gyro sensors. It receives the input positions and touch gestures. It then converts the horizontal and vertical coordinates (in pixels) of touch input positions to the display-based user interaction points (in meter units) by using the virtual space coordinates with respect to the input positions. The user's dragging on the mobile panel can change amounts of the reference coordinate value on the display.

3. Input Server

When multiple users interact with the application using various input devices, the input messages from all input terminals are sent to the input server simultaneously. The input messages contain different system and the input device information and they do not arrive in a right order. Hence, the input server tags all input messages with the identification in order to appropriately support multi-user synchronous interaction. Also, it takes in charge of specifying the target application for user interaction to be taken based on the user position of the input message.

The input processing procedure on the input server is reception, classification, adaptation, and transmission phase. The input server receives the input message from all input terminals. The input message contains the input values corresponding to the input device and the user interaction environment. The server classifies the input message by the user identification, but in some cases the user identification number given by different input terminals can be redundant. Hence, the input server reassigns the user identifier for each input message independently of each other by using host name, device name, and user ID. It then converts the adjusted input value to the form of application event and transmits it to the target application using the inter-process communication mechanism of Windows system.

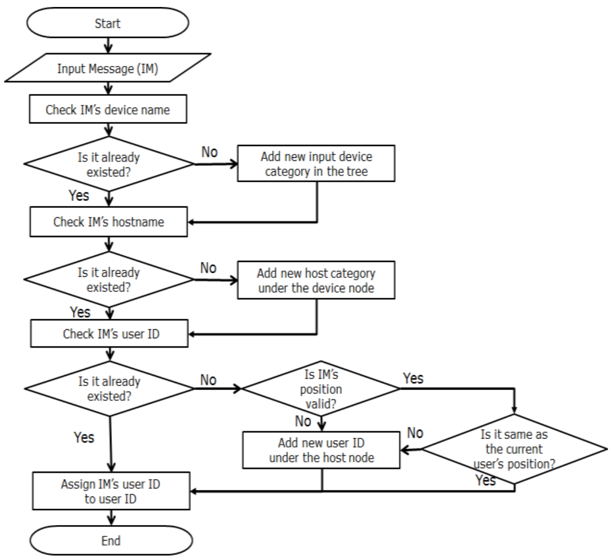

Figure. 5 Flowchart for User Classification on the Input Server

Figure 5 shows the classification of the input message to assign user identification. The input message from different users should be treated mutually exclusive, in order to prevent the interference with the interaction among each other. Once the input message is given a new user identification number, it is continuously processed until the end of the user's interaction. When the user performs a continuous interaction, the input server uses the same user identifier even if the input messages are received from different input terminals (such as, multiple Kinect terminals). Since the input message contains the user information and associated features, it is possible to identify the user and maintain the continuous processing by comparing it before and after. Therefore, the input message is identified as part of a particular user interaction to the application.

The input server adapts the input value and status of the input message to compromise a gap between the numerical value ranges of the user inputs (in meter units) by the input terminal and the screen coordinates used by the Windows system (in pixel units). If the application client window is located differently from the origin of the screen coordinates, it must be adjusted to provide valid user inputs to the application window. The input value from all input messages becomes within a certain range of the same pixel unit (regardless of the type of input devices) in the application window.

APPLICATIONS

This section describes two collaborative applications utilizing the unified input processing framework to support multi-user collaborative interactions: the iTILE ImageWorld tiled display application and the Unity3D cultural heritage building layout and navigation application.

Figure. 6 Multi-User Interaction on iTILE Image World using Kinect Gestures: Pointing and Selection (left) and Image Rotation (right)

1. iTILE Image World

Figure 6 shows users selecting and manipulating the images using Kinect-based gestures on the iTILE ImageWorld. The iTILE ImageWorld is an interactive tiled display application written in the OSG(Open Scene Graph) 3D graphics library and iTILE framework. This application helps multiple users to view the images on the large tiled display and select and manipulate the images collaboratively by gestures or the mobile interfaces. This application loads the images of the user-predefined directory and presents thumbnails vertically on the left of the display. The users can scroll up/down the thumbnails to explore more images displayed other than in the thumbnails. They can freely select the thumbnail image using a pointing action (i.e., holding their pointer on a particular thumbnail long enough) and drag the selected thumbnail to the center of the display and release it to appear the original-sized image. Then, they can move or zoom in or out and rotate the image.

If multiple copies of the images are superimposed together, users can select a thumbnail image to bring up the corresponding original-sized image in front of the others. The users can remove the image by moving it over the edge of the outer side of the display or the thumbnails. The user interaction is constrained to one user per image, to minimize the difference between the interactions by input devices and to simplify the input processing. Multiple users can simultaneously interact and manipulate the different images. However, when multiple users attempt to manipulate the same image, the image taken by one user is protected so that it cannot be accessed by the others. This is needed to ensure the independent operation of each other.

This application supports multi-user interactions with Kinect and mobile interfaces. When using Kinect, the user's right hand movements are tracked to follow the pointer. The users can select or release the image by right hand grasp or release gesture. Then they can move the selected image by right hand movements and zoom or rotate by both hand's stretching in or out and moving up or down respectively. The users can also move the pointers by the dragging gesture on the mobile touch panel. They can select or release the image by double-tabbing touch gesture. They can move, zoom, or rotate the selected image by one-point dragging, by two-point stretching, or two-point moving up or down interaction respectively.

In this application, the multi-user interactions are delivered to the OSG event handler and processed by the user identifier to generate the input object. Then, the event value is used to determine the type of user interaction, such as select, move, zoom, or rotate. The input position in the event is used to select the target image. If there is one input point, it is used for selecting or moving the image. If there are more than two input points, the first two input points are used for zooming (with the changes in distance between two points) or rotating (with the directions of interaction of the two points).

2. Unity3D Cultural Heritage Building Layout and Navigation

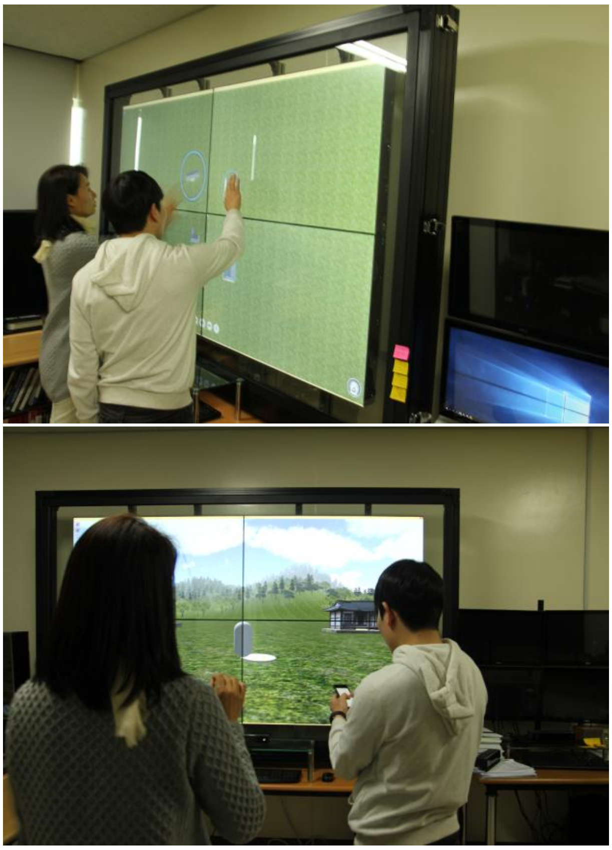

Figure. 7 Unity3D Cultural Heritage Building Layout using Multi-touch (left) and Navigation using Kinect and Mobile Interface (right)

Figure 7 shows two users interacting with the Unity3D-based cultural heritage building layout and navigation application using Kinect-based gesture, multi-touch, and mobile interface. This application provides two modes: layout (left) and navigation (right). When the new user is joining in this application, the icon representing this user (with different input device and color) is appeared on the top left of the screen. In addition, the cursor with the same user icon is appeared on the center of the screen that follows the user's cursor movements. The users can toggle between the layout and navigation mode by clicking the mode icon.

In the layout mode, multiple users can simultaneously layout the buildings (i.e., place new buildings into desired location and direction) with the top-down view. They can also move, resize, or change the directions of the placed buildings. If they want to remove the building, they just need to drag and drop the building into the trashcan icon. In the navigation mode, the users can walk through the virtual environment with the first-person view, but only one user at a time can move around the environment which affects the others' view. Hence, the users must collaborate with each other to perform the seamless navigational interaction.

When the new user is joined, the marker is appeared on the screen which also followed by the user's movement, such as Kinect's right hand, one-point touch and dragging on multi-touch panel, and double-tab and dragging action on the mobile interface. When using Kinect, the users can select and move the building by right-hand grasped dragging, change the size or orientation of the building by both-hand grasped stretching or rotating gesture. In the navigation mode, the user can move forward or backward the environment by right-hand grasped moving up or down gesture and pan the direction by right-hand grasped moving left or right gesture.

When using the multi-touch panel, the round-shaped marker is appeared on the point where the user touched. The users can select the building with the one-point touch. They can move the building by one-point touch dragging. The size or the direction of the building can be changed by the user's two-point touches. In the navigation mode, the user can move around the environment by dragging forward and backward or left and right on the touch panel.

When using the mobile interface, the user can select the building by the user's dragging action. Similar to multi-touch, the size or the direction of the building can be adjusted by the user's pinch-zooming action. In the navigation mode, the user can move forward or backward by enlarging or narrowing the gap between two finger points on the mobile touch panel. The user can change the navigational direction by clock-wise or counter clock-wise rotating between two finger points. If the user tabs and holds the selected building for more than three seconds, the detail building information is also popped in a separate window.

In this application, the event manager processes the input messages to separate different user's interaction. It generates the input object for each user, which is based on the user identifier. Each input object distinguishes whether more than one or more input points exist. If point count is one, the user interaction is represented by event and input point. Event indicates the type of user interaction and input point contains the location and the start/end time of the user interaction. If point count is more than two, the user interaction will take advantage of the first two input points. The type of user interaction varies depending on whether or not applying the following input point. The control of movement or UI is performed by applying the only one input point. The resizing and rotation interaction is performed by applying two input points.

CONCLUSION

The advances in technology lead to the large-size and high-resolution display. Nowadays the size of home appliance displays such as TV and computer monitor is also getting bigger. Moreover, the full HD resolution broadcasting service has started a few years ago, and it has already discussed the 4K UHD broadcasting service technology. This form of public display has been utilized in various places such as stations, airports, museums and public institutions. However most cases use one-way information display or shows the advertisement. There are few studies done on multiple user interaction techniques on public display. Moreover, prior works on user interaction have mainly focuses on 3D navigation and object manipulations using 3D input devices.

This research aims at designing and developing the unified input processing framework to support multiple users with various input devices to collaboratively work together on the large public display. In particular, it enables simultaneous user interactions to take advantage of the screen sharing and collaborative work between multiple users. The unified input processing framework provides the procedure of the input information to find a universal means applied to different input devices and to pass multi-user interactions to a wide range of collaborative applications. The iTILE ImageWorld tiled display application and the Unity3D cultural heritage building layout and navigation application revealed that the proposed unified input processing framework ensures multiple users to collaboratively use multiple and different kinds of input devices, such as Kinect gesture interaction, 12-point multi-touch screen, and mobile interfaces, to interact with the application.

This framework aims at supporting the functionalities that can be passed to each of the user inputs to the application even if the users use different input devices at the same time. It also provides the procedure related to the type and method for multi-user interaction. In particular, the input terminal modules support the existing input devices to be used in the multi-user interaction. The results of this work promote the wide spreads of the public display and it can be utilized in a variety of fields, such as scientific visualization and big data processing. Future research direction will be improving this framework to provide more intuitive visual feedback to support fluid group interaction. Furthermore, it is necessary to evaluate a number of collaborative user interactions on this framework and analyze with respect to the time that can be utilized to share the application.

REFERENCES

[1] J. Leigh, A. Johnson, L. Renambot, T. Peterka, B. Jeong, D. J. Sandin, J. Talandis, R. Jagodic, S. Nam, H. Hur and Y. Sun, Scalable Resolution Display Walls, Proceedings of the IEEE, 101(1), 2013, pp. 115-129.

[2] R. Ball and C. North, Realizing Embodied Interaction for Visual Analytics through Large Displays, Computers & Graphics, 31(3), 2007, pp. 380-400.

[3] J. Davis and X. Chen, Lumipoint: Multi-user laser-based interaction on large tiled displays, Displays, 23(5), 2002, pp. 205-211.

[4] P. Peltonen, E. Kurvinen, A. Salovaara, G. Jacucci, T. Ilmonen, J. Evans, A. Oulasvirta and P. Saarikko, It's Mine, Don't Touch!: Interactions at a Large Multi-touch Display in a City Centre, in: Proc. of the Conference on Human Factors in Computing Systems, ACM, New York, NY, USA, 2008, pp. 1285-1294.

[5] Y. Cho, M. Kim and K. S. Park, LOTUS: Composing a Multi-User Interactive Tiled Display Virtual Environment, Visual Computer, 28(1), 2012, pp. 99-109.

[6] K. Ponto, K. Doerr and F. Kuester, Giga-stack: a method for visualizing giga-pixel layered imagery on massively tiled display, Future Generation Computer Systems, 26(5), 2010, pp. 693-700.

[7] S. Kim, M. Kim, Y. Cho and K. S. Park, iTILE Framework for Constructing Interactive Tiled Display Applications, in: Proc. of International Conference on Computer Graphics Theory and Applications, 2009, pp. 367-372.

[8] D. Vogel and R. Balakrishnan, Distant freehand pointing and clicking on very large, high resolution displays, in: Proc. of the 18th Annual Symposium on User Interface Software and Technology, ACM, New York, NY, USA, 2005, pp. 33-42.

[9] K. Y. Seo, S. Kim, Y. Cho, S. Park and K. S. Park, A Design on Gestural User Interaction Techniques for Tiled Displays using Kinects, Communications in Computer and Information Science, 373, 2013, pp. 376-379.

[10] P. Dietz and D. Leigh, DiamondTouch: a multi-user touch technology, in: Proc. of the 14th Annual Symposium on User Interface Software and Technology, ACM, New York, NY, USA, 2001, pp. 219-226.

[11] R. Jagodic, L. Renambot, A. Johnson, J. Leigh and S. Deshpande, “Enabling multi-user interaction in large high-resolution distributed environments,” Future Generation Computer Systems, 27(7), 2011, pp. 914-923.

[12] Cyber-Commons http://www.sagecommons.org/tag/cyber commons/

[13] B. Jeong, L. Renambot, R. Jagodic, R. Singh, J. Aguilera, A. Johnson and J. Leigh, High-performance dynamic graphics streaming for scalable adaptive graphics environment, in: Proc. of the ACM/IEEE Conference on Supercomputing, ACM, New York, NY, USA, 2006, pp. 24-24.

[14] M. R. Morris, A. Huang, A. Paepcke, and T. Winograd, Cooperative Gestures: multi-user gestural interactions for co-located groupware, in: Proc. of the Conference on Human Factors in Computing Systems, ACM, New York, NY, USA, 2006, pp. 1201-1210.

[15] D. C. McCallum and P. Irani, Arc-Pad:Absolute+Relative Cursor Positioning for Large Displays with a Mobile Touchscreen, Proc. of the 22nd Annual Symposium on User Interface Software and Technology, ACM, New York, NY, USA, 2009, pp.153-156.

[16] K. Ponto, K. Doerr, T. Wypych, J. Kooker and F. Kuester, CGLXTouch: A multi user multi-touch approach for ultra high-resolution collaborative workspaces, Future Generation Computer Systems, 27(6), 2011, pp. 649-656.

[17] E. Ch'ng, The Mirror Between Two Worlds: Multitouch-Multiuser Interaction for 3D Digital Objects and Environments, in Harris, D. (ed.), Digital Media and Technologies for Virtual Artistic Space, 2013.

[18] F. Block, D. Wigdor, B. C. Philips, M. S. Horn and C. Shen, FlowBlocks: A Multi-Touch UI for Crowd Interaction. Proc. of UIST, 2012, pp. 497-507.