[Software Capstone Design]

**Smart Home Control Application**

**Using Hand and Projector**

**Academic advisor: Han Tak Don**

Teaching Assistant: Park Yun Jeong

**Team: POINT**

2010147046 Chong Bang Won

2014131003 Gong Eun Young

2014155014 Jeon Su Bin

**| INDEX**

1. Introduction ------------------------------ 02
2. Previous Research ------------------------------ 03
3. Application Proposal ------------------------------ 05
4. Application Implementation ------------------------------ 07
5. Implementation Result ------------------------------ 11
6. References ------------------------------ 17
7. **Introduction**

**1.1 Subject**

Smart home control application using hand and projector.



* 1. **Task**

|  |  |
| --- | --- |
| Control Diverse Devices at Home | Getting device Information |
| - On / Off  - Change volume or channel  - Diverse different functionality  depending on devices | - Current State  (ex. power, channel, temperature)  - Objective State (ex. Temperature)  - Diverse different information  depending on devices |

* 1. **Objective**

To Provide Intuitive, Interactive UI depending on Home Environment

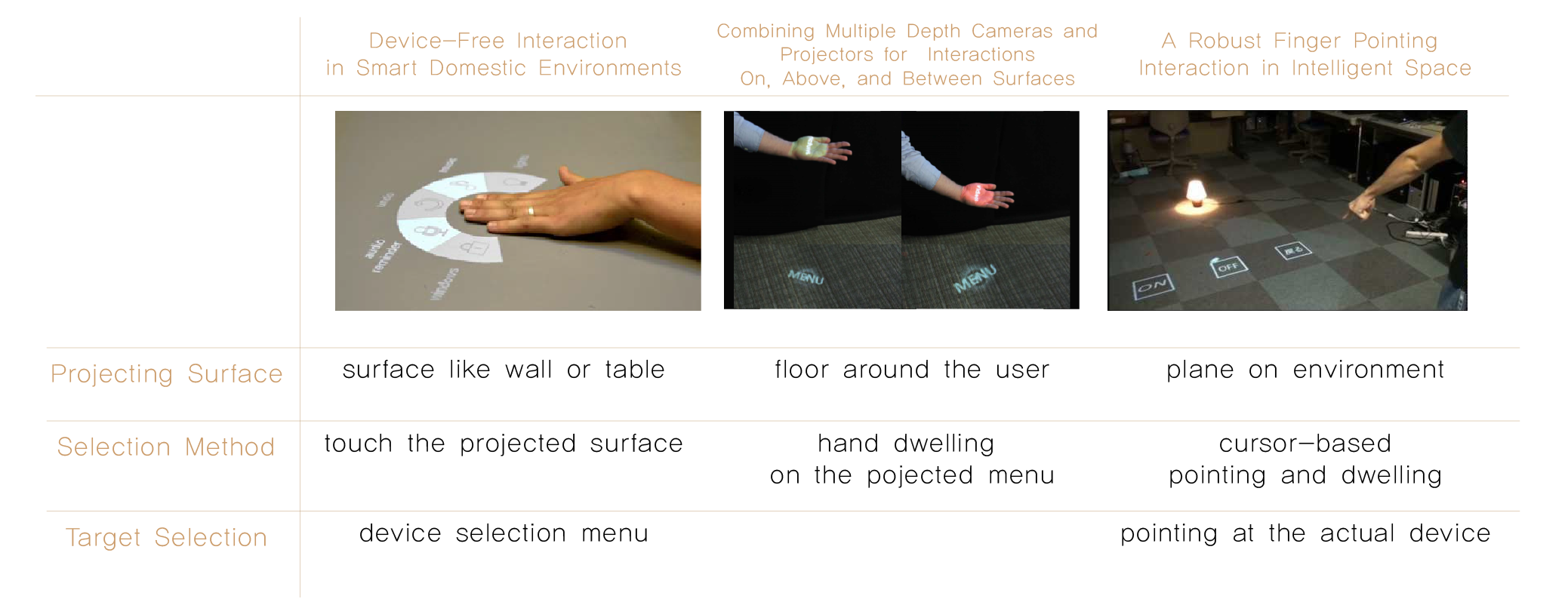
→ 1) Device-free interaction using the hand and natural pose

2) Considering different home environment

1. **Previous Research**

**2.1 Previous Researches**

There are researches regarding to control diverse and different devices at home using hand and projector. Because our objective is device-free, we find researches about combination of hand based interaction and projection display which is possible to give interface to diverse devices. Previous researches are related with our proposal necessity,.

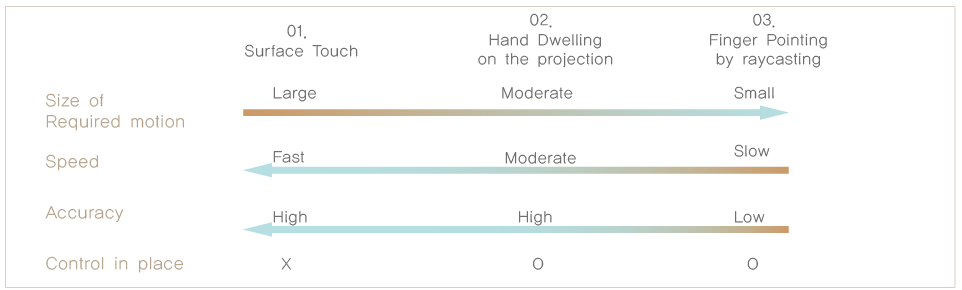


* Device-free interaction in smart domestic environment [1]
* Combining Multiple Depth Cameras and Projectors for Interactions On, Above, and Between Surfaces [2]
* [[1]](#footnote-2)A robust finger pointing interaction in Intelligent Space, System Integration (SII) [3]

Things that we need to focus on are selection method and projecting surface because our main attention is about interaction and projection.

These are on the first and second lines of the table.

**2.2 Interaction way Comparison and Limitation**



* Surface Touch is speedy and accurate, however user needs move to the surface
* Hand Dwelling and Finger Pointing can control in place without movement but slow.
* Finger Pointing method use cursor by ray casting, however the more distant the surface is, the more elaborate control is needed → slow and error-prone
* Limitation

: Absence of speedy and accurate way along with in place control probability

**2.3 Projection Comparison and limitation**

* Surface Touch has less relevance between actual object and virtual UI.
* Finger Pointing project the virtual UI just somewhere around the object. No consideration of environment and user location.
* Limitation

: Absence of proper combination of actual object, virtual UI and the environment.

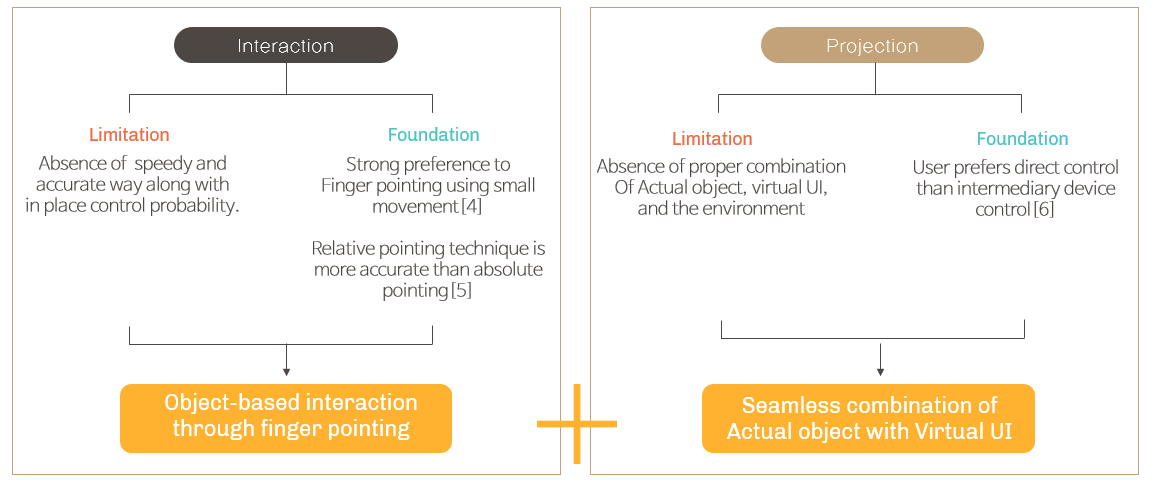
1. **Application Proposal**

**Overview**

We propose new interaction method based on limitation of previous study and study about usability in those situations.

First, in terms of interaction, we have limitation in previous researches. Up the now, there are no speedy and accurate way at distance in projection display. Although finger pointing is error-prone and slow as mentioned above, there are strong preference to finger pointing in projection based augmented reality [4]. It is because finger pointing is most natural way when indicate distant object, and needs small movement. So, we determined to use this method while releasing previous error-prone and slow problem using object based interaction with relative pointing technique. Mikkel verify that relative movement is more accurate than absolute pointing technique especially in distant control situations [5].

Second, in terms of projection, there are no proper combination of Actual object, virtual UI, and the environment. However, user prefer direct control to control using intermediary device according to the user study [6]. Absolutely there are needs but no proper way. So, we propose seamless combination of Actual object with Virtual UI using the position data [interactable device, obstacle, and wall surface]. It is adjustable to diverse environment and environment change.

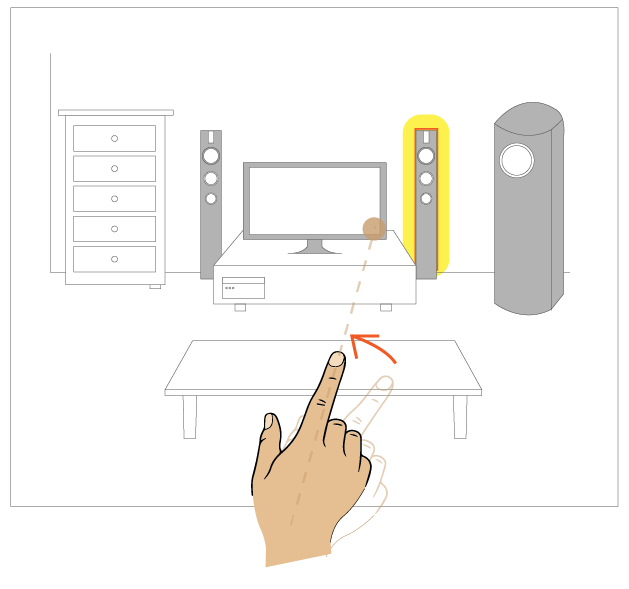
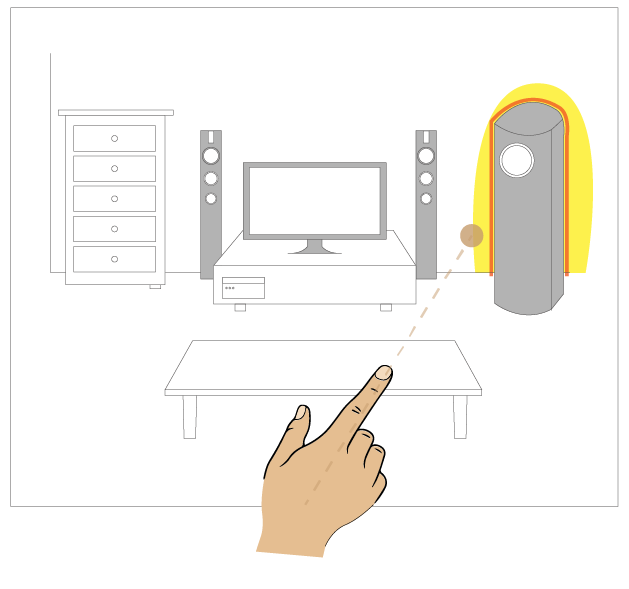


**3.1 Object-based interaction through finger pointing**

Finger pointing is a natural way when user direct distant object and strong preference between other hand-based interaction methods. However Previous pointing method which is based on ray casting and cursor has notorious and widely known problem. The farther distance needs the more elaborate control which is time-consuming and error-prone. In this project, we propose object-based selection according to relative pointing direction change. Previous research has proven that relative direction-based pointing has better usability than absolute direction calculation way.

① Selecting the closest object corresponding with finger pointing direction

② According to relative direction change, changing the selected object (virtual UI or actual device) at relevant direction



* 1. **Seamless combination of Actual object with Virtual UI**

Each home and living room has different environment and device arrangement. Therefore, we give user interface mechanism which is adjustable to different environment. Below images show different home environment and difference of projectable surfaces.

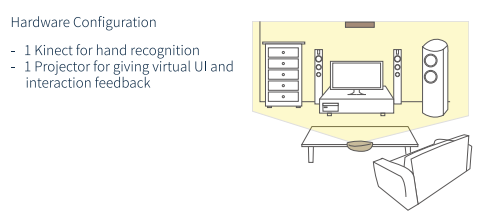


[Envrionment1] [Environment2]

1. Giving virtual UI and feedback about the device using the wall around the device
2. Giving adjustive virtual UI menus depending on diverse home environ. and devices.

|  |  |  |
| --- | --- | --- |
|  | **Environment1** | **Environment2** |
| **Considering Area** |  |  |
| **Corresponding UI** |  |  |

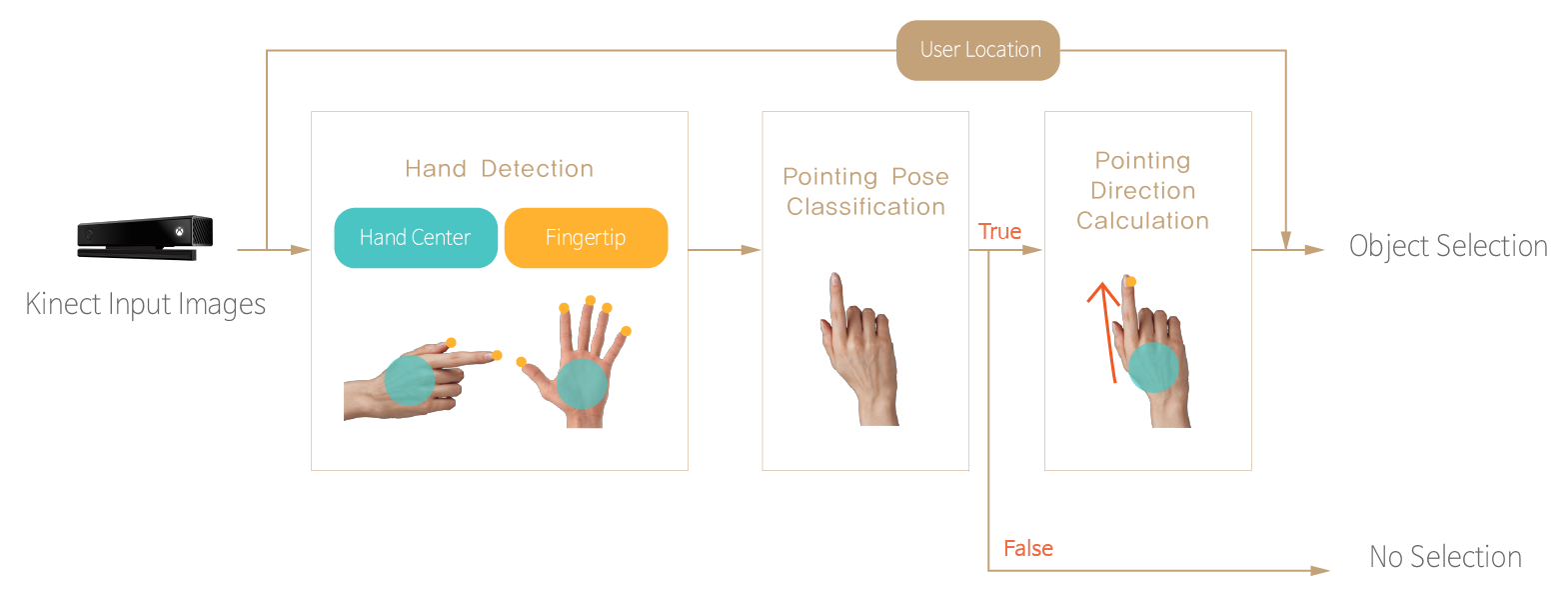
1. **Implementation**

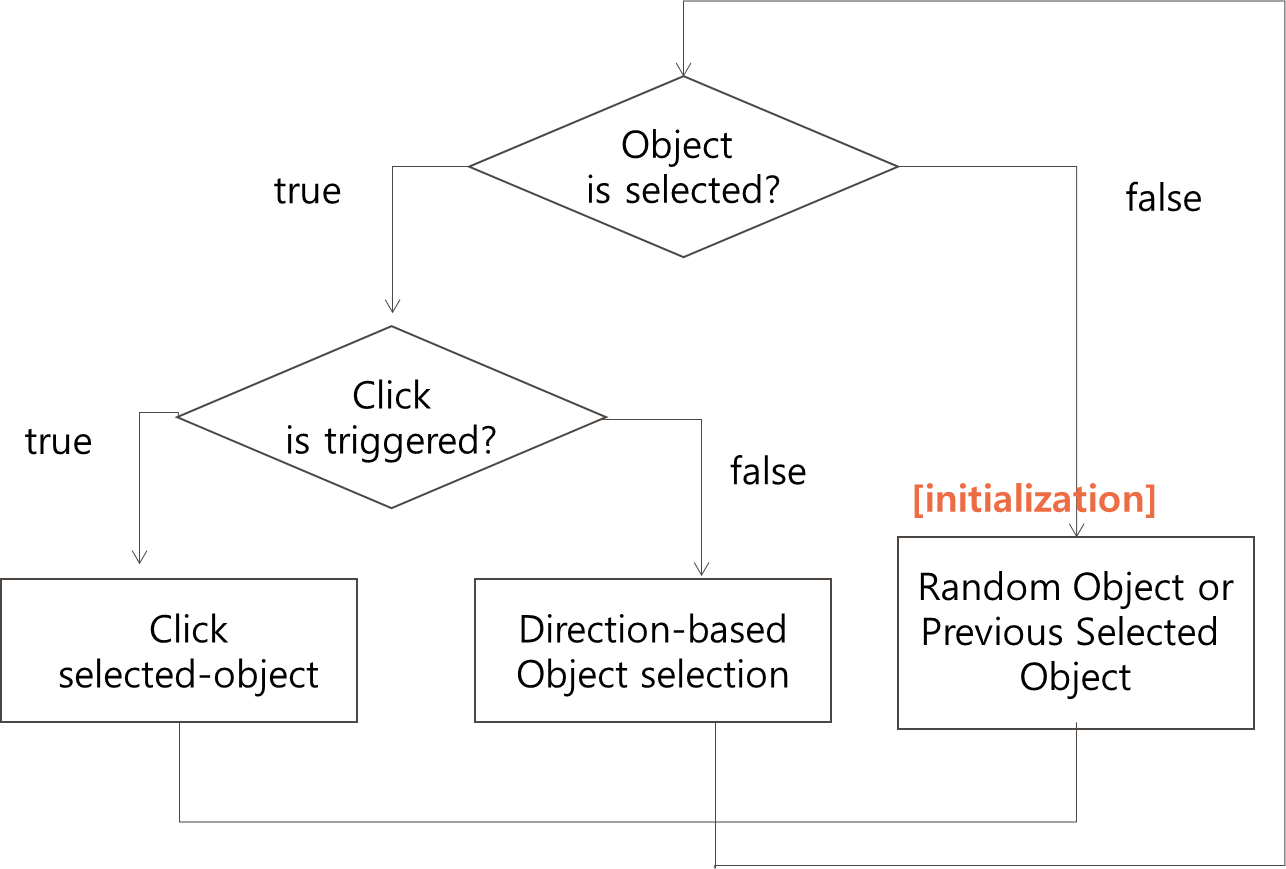
**4.1 Hardware Configuration**

- One Kinect is needed to detect movement. It is for hand recognition. It is positioned near the object.

- One Projector is needed to project virtual UI around devices. It is positioned at ceiling to one wall and around here.

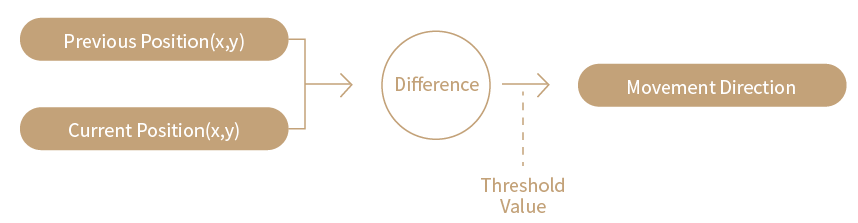
**4.2 Software: Finger Direction Input**



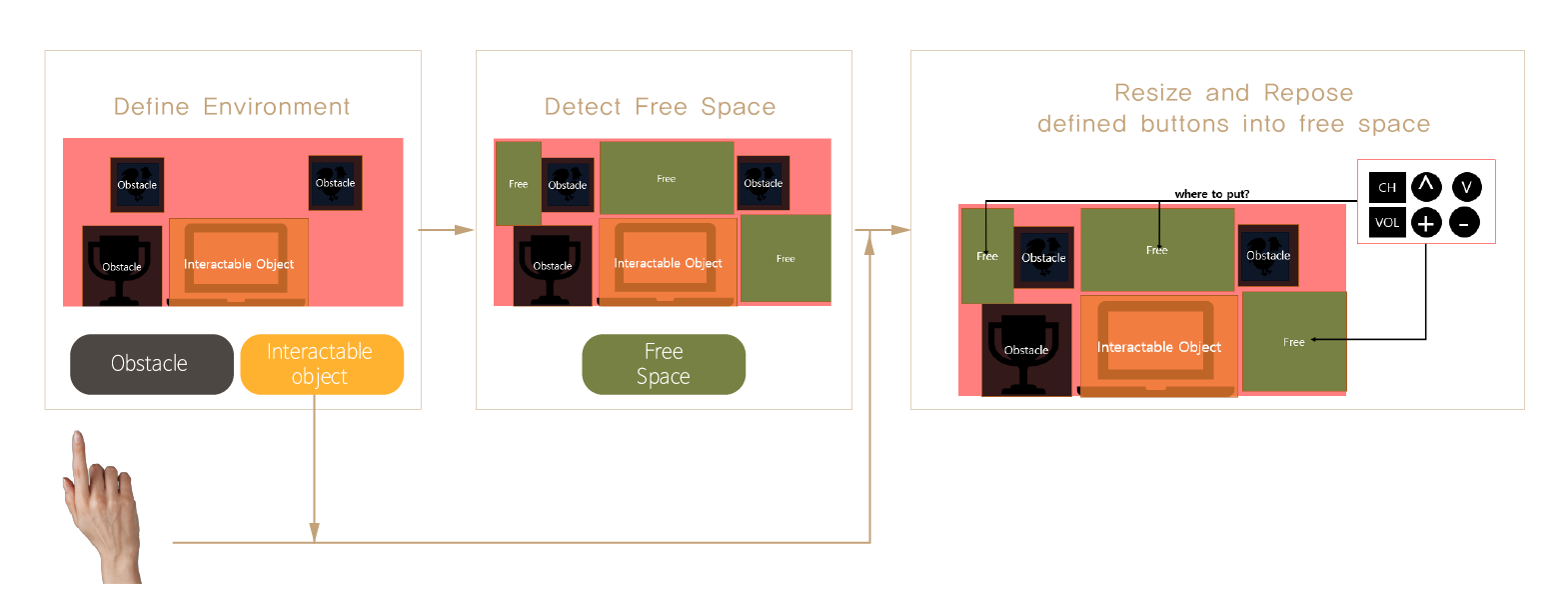
* Hand Detection: Using Kinect SDK, get information of hand center and fingertip.
* Pointing Pose Classification: Given hand state, hand center and number of fingertips, classify hand pose to determine whether it is pointing posture or clicking posture.
* Pointing Direction Calculation: Based on previous hand direction, hand center and fingertips, calculate variation of direction in the case of pointing posture.
* Object Selection: If object exists corresponding direction, select the object. However, over the threshold value, nothing is selected.
* **Detail Algorithm**
* Initialization

: Initial state which is not selected object select nearest object by ray casting

* Object is selected by direction-based object selection
* ‘select’ doesn’t mean ‘click’ and ‘select’ is the state which is object is highlighted
* If the clicked object is end stage, selected object become null
* **Movement Direction**
* We can select the object which is in corresponding direction.
* There is 8-way Direction
* up/ down/ left/ right
* up-left/ up-right/ down-left/ down-right
* Direction calculation will be carried out below.



* 1. **Overall Application including Projection**



- UI Application Development Objective is providing house-oriented UI

- Implementation Method is following.

1) Space Definition: Consider the house environment as follows.

- Interactable Object which can communicate with users.

- NULL Space: NO interaction space which is not allowed UI Display

- Free Space: OTHER space which is allowed UI Display

2) UI Resizing, Repositioning

- Near the selective object, setting up defined objects on free Space

- UI Objects have connections with each other and we will define relative

position based on this relation data.

(ex. volume up / volume down is related so there will be highly displayed near)

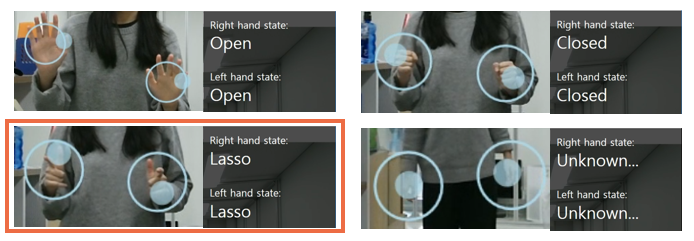
1. **Implementation Result**

We will explain how to implement it in turn.

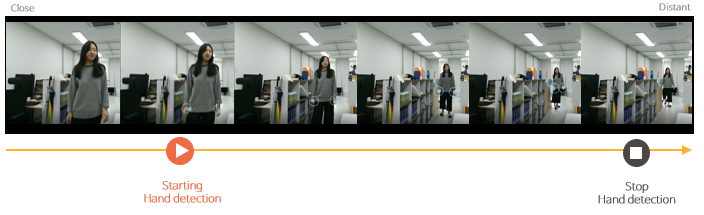
**5.1 Finger Pointing**

We got information about hand center, thumb fingertip, and hand state (open, close, lasso, unknown) using Kinect SDK.

**5.1.1 Hand State**

Kinect SDK provides hand state and according to experiment, in most of pointing case, the state is lasso or unknown. We assume that this information could be used to classify pointing posture.

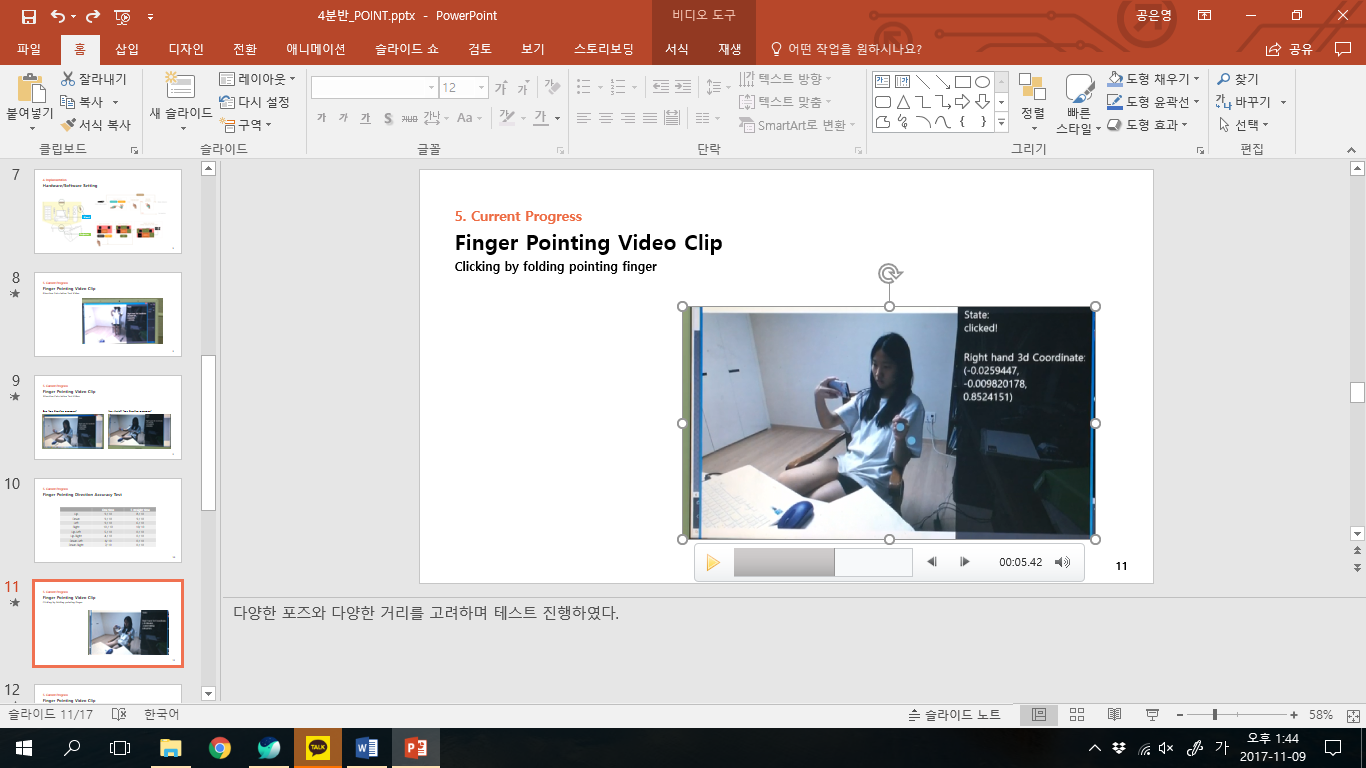
**5.1.2 Distance effect**

We hand the Kinect on two walls. So, it is important to know distance threshold. We experimented how farther it could detect user’s hand.

**5.1.3 Pose variance**

 Moreover, we assume space is home which is comfortable environment. So, we test various pose like sitting, bending knees, and side, and back not just considering stiffen front standing pose. Fortunately, in diverse pose hand is detected well.

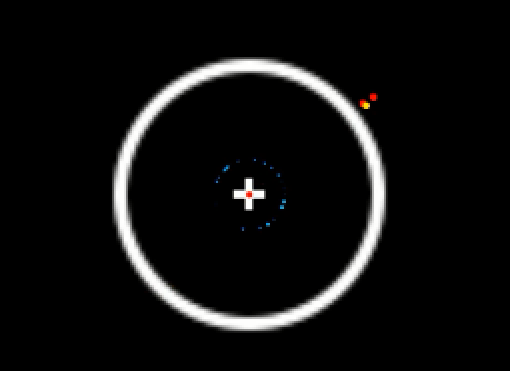
**5.1.4 Pointing Pose Classification**

Distinct clicking action.

**5.1.5 Input Data Processing**

Processing data for input follows this procedure

1. Receive delta position of pointing every frame.
2. Check if received delta position is larger than threshold. This process is to prevent
   1. (yes) strict delta pos to threshold.
   2. (no) use received data
3. Accumulate processed delta position data.
4. If accumulated data exceeds threshold it creates directional event.
5. Accumulated data is reduced by factor. (accum = accum \* factor);



Display of accumulated data. Red dot shows the current accumulated data.

White ring represents the threshold to create directional event.

* 1. **Projection UI**

**5.2.1 Defining Environment**

* + Pre-defined environment: interactable object, obstacle
  + Implementation of Making Free Space

: making free space which is surface and could project virtual UI.

* + Tested on virtual world. Environment is composed of Lamp, TV, Radio.
  + TV function: Channel Up/Down. Volume Up/Down. Power On/Off
  + Radio Function: Music Next/Prev. Volume Up/Down. Play/Pause.
  + Lamp Function: Light On/Off.

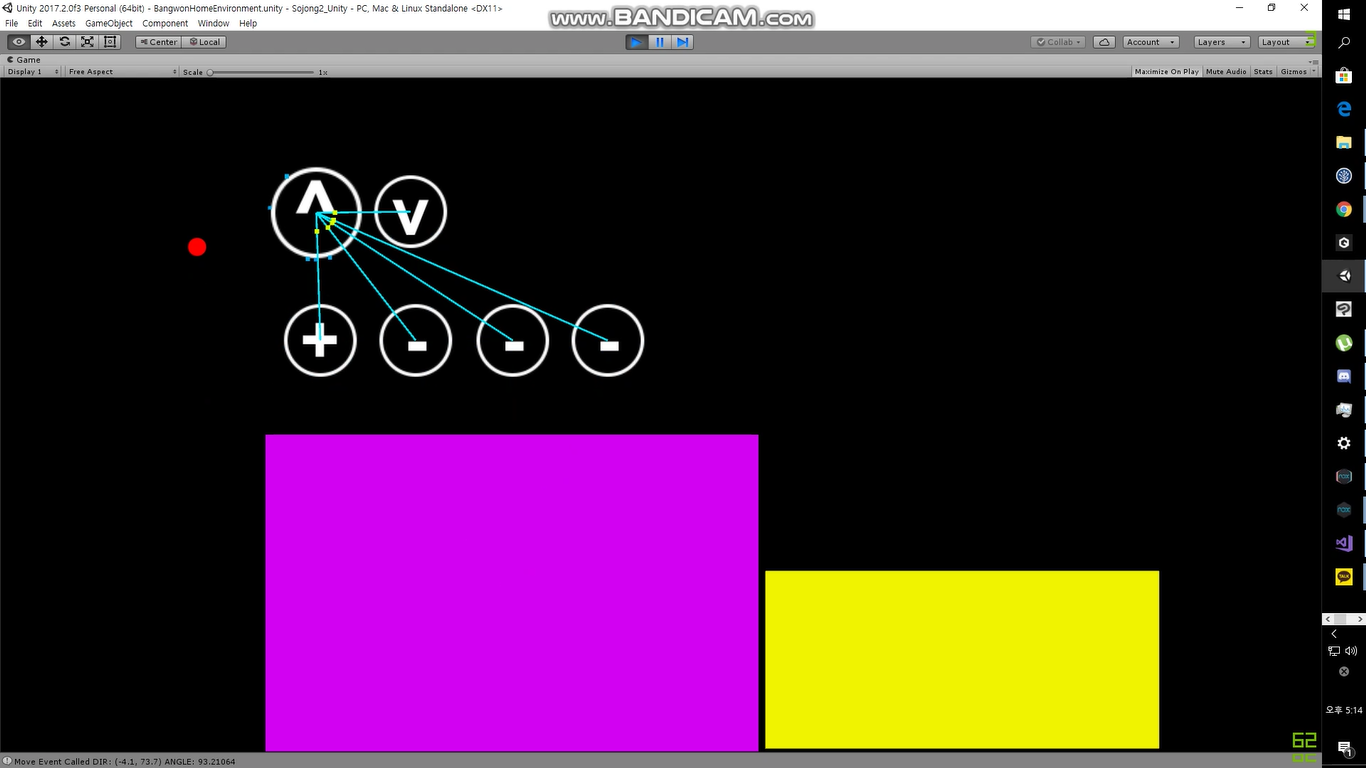


Virtual environment composed of TV(Middle), Radio(Bottom), Lamp(Right)

**5.2.2 Interaction with object**

* Selection of interactable object are done by clicking at focused object.
* Controlling selected object are done by clicking focused UI object.

**5.2.3 Changing focus of UI Objects.**

* + Changing focus of UI objects are done by receiving directional event.
  + If received directional event, focus of UI object will be changed to the closest object which has closest direction from event direction.
  + Focus of UI can only be changed to linked UI objects which are displayed as connected line with currently focus UI object.

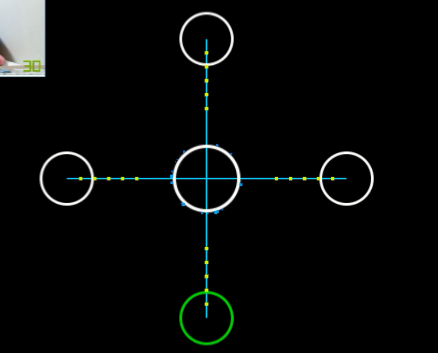
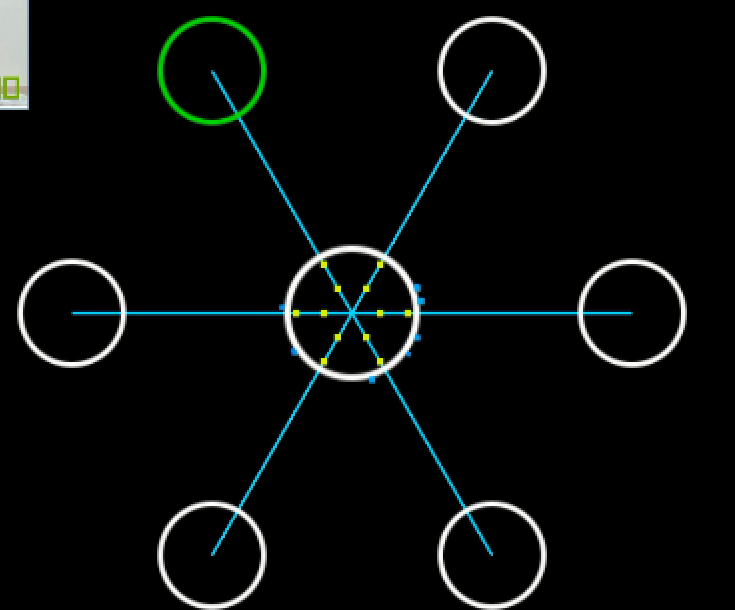
**5.2.4 Real Home Environment**

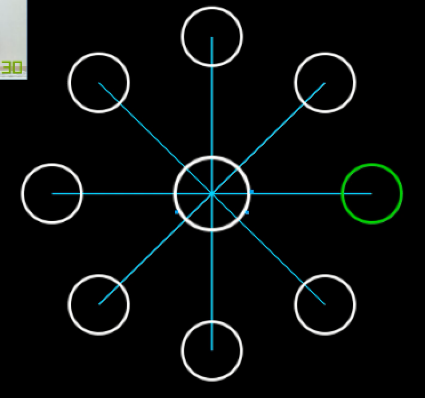
* + Real home environment was composed of two interactable object: TV, Radio
  + TV is left monitor
  + Radio is right monitor.

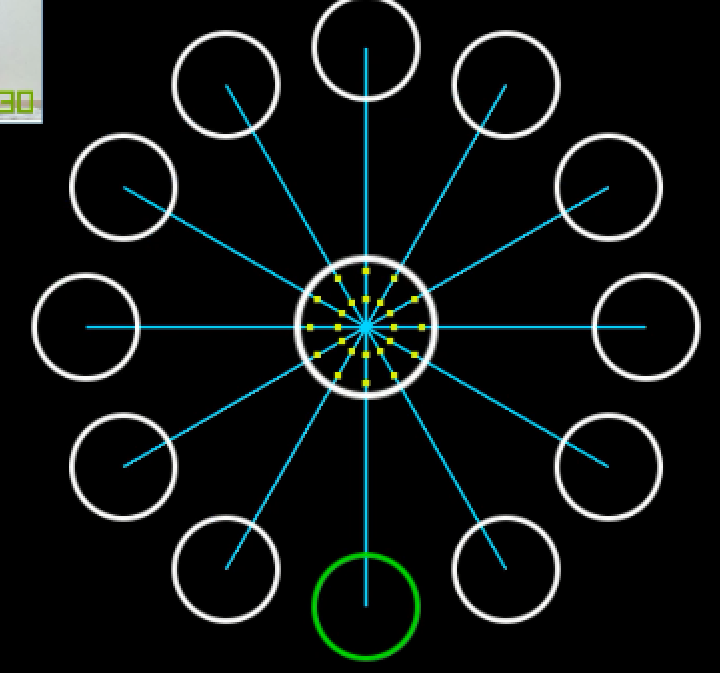


**5.2.5 Accuracy Test**

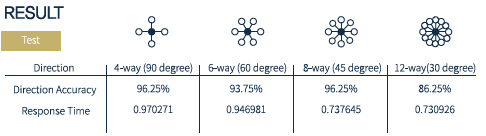
* This test is to measure the accuracy of creating directional input event under cases where UI are placed every 90, 60, 45, 30 degrees.
* Repeat 100 times for each test case

 **90 degrees 60 degrees**



**45 degrees 30 degrees**



* Result of test is shown in upper table
* Result shows that 30 degrees are much harder than rest of angles. This result help us to create this rule. No two or more UI object should be in 45 degrees. When user is trying to move from one UI object to another, if there are UI objects that are closer than 45 degrees, it has high ratio of failing to move to wanted UI object.

**6. Reference**

[1] Felix Heidrich, Ivan Golod, Peter Russell, Martina Ziefle, Device-free interaction in smart domestic environment, Proceedings of the 4th Augmented Human International Conference (2013), 65-68, DOI: 10.1145/2459236.2459248

[2] Andy Wilson, Hrvoje Benko, Combining Multiple Depth Cameras and Projectors for Interactions On, Above, and Between Surfaces, Proceedings of the 23nd annual ACM symposium on User interface software and technology (2010), 273-282, DOI: 10.1145/1866029.1866073

[3] Yusuke Koizumi, Dinh Tuan Tran, Joo-Ho Lee, A robust finger pointing interaction in Intelligent Space, System Integration (SII), IEEE/SICE International Symposium (2011), 230-235, DOI: 10.1109/SII.2011.6147451

[4] Stephen Voida, Mark Podlaseck, Rick Kjeldsen, Claudio Pinhanez, A study on the manipulation of 2D objects in a projector / camera-based augmented reality environment, Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (2005), 611-620, DOI: 10.1145/1054972.1055056

[5] Mikkel R. Jakobsen Yvonne Jansen, Sebastian Boring, Kasper Hornbæk, Should I Stay or Should I Go? Selecting Between Touch and Mid-Air Gestures for Large-Display Interaction, Human-Computer Interaction? INTERACT 2015, 455-473, DOI: 10.1007/978-3-319-22698-9\_31

[6] Michael Beigl. 1999. Point & Click—Interaction in Smart Environments. In Handheld and Ubiquitous Computing (Lecture Notes in Computer Science), Hans. Gellersen (Ed.), Vol. 1707. Springer Berlin Heidelberg, 311–313. DOI: 10.1007/3-540-48157-5\_31

1. <https://www.youtube.com/watch?v=7JP_1dPcuJM> [↑](#footnote-ref-2)