The User Interface of Button Type for Stereo Video-See-Through Device

Youngju Choi and Yongduek Seo

Abstract—**This paper proposes a user interface, on see-through system environment which shows the images via two different cameras, that also ordinary user can control the computer systems or other various processes easily. For that, we include an AR technology to synthesize the virtual button to the image which is captured by the camera real-time. And we were looking for the hand position in the image to judge whether the figure selects the button. And the result of judgment visualizes through changing of the button color. The user can easily interact with the system by selecting the virtual button in the screen with watching the screen and moving her fingers at the air**

Index Terms—**AR, HCI(Human Computer Interaction), See-Through, Synthesis, Plane Projective Transformation**

1 Introduction

This paper proposes a user interface of the button type about a see-through device which is receiving an interest as the new computer user interface. This is based on AR(Augmented Reality). The proposed interface is to synthesize the virtual button to a display of the device equipped two cameras. Through this, a user can recognize as if the buttons are at the air and is selecting. As a result a event of the action is executed.

In the paper, we change a color of the button to expresses visually what the button is selected. As well, since the device is equipped with two cameras like human eyes, it can derive a phenomenon which is like using depth information about objects. Consequently, we can receive a more realistic and accurate feeling than one camera. [picture1] is a structure of system. An arrow substitutes a finger.
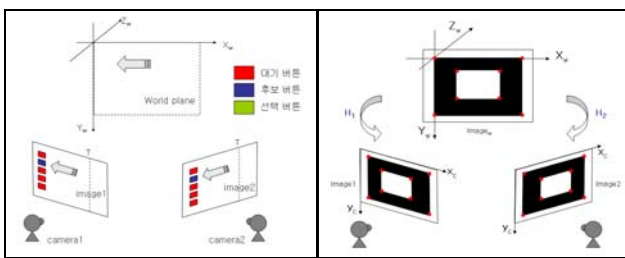
Picture1. Left) The structure of test environment,
Right) Homograph relationship

It gets images (image-i , $i \in \{1, 2\}$) of world coordinate (world plane) by two cameras and draws the virtual red button above the images. If a finger exists in a raw which includes a specific button but the button is not selected, the button become a candidate button and a blue and if a finger is really on the button, the button becomes the selected button and a green. In case that not belong to the above two cases, we look as the ready state and it sets the button with red color. From this, the user oneself can know what kind of button is selected.

2. Main

The overall process is composed with the preprocessing section, the waiting section and the main section.

2.1 Preprocess

A preprocess has two steps. First of all, it analyzes the sample hand image which it takes with the cameras and calculates a average and standard deviation. The values represent the feature of the hand color and these become the standard data for seeking the hand region from the captured image. To receive the effect of the light a little this paper only consider about a hue and saturate on the HSV space, and calculate the mean(hm,sm) and the standard deviation($h\sigma$, $s\sigma$) of the sample image. Under an assumption which the distribution of hand color has a normal distribution, we thinks that the pixel which has the following reliability is the region of the hand color.

$$\left| \frac{hm^i - h^i_{x,y}}{h\sigma^i} \right| < 2.5 \quad \cap \quad \left| \frac{sm^i - s^i_{x,y}}{s\sigma^i} \right| < 2.5 \quad i = \{1,2\}$$

eq1)

Next, we calculate the homography matrix which represents the transformation between the coordinates of the images which are captured by the each camera. The virtual button which is on the world plane is projected to the image which is captured from the each camera by this transformation matrix, so we can synthesize the virtual button to the image to look like that the virtual button is located in the special position of the real world. For this, like the right side image of [Picture1] we built the pattern image (w Image)which is formed the rectangles with black and white and the two cameras. And we got the 2 dimensional points from the image plane and the 3 dimensional points from the world plane.

The relationship of the points is like the equation 2).

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} u \\ v \\ w \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

eq2)

After develop this formula to Ah=0 form, calculate the H matrix through the SVD (singular value decomposition).

2.2 Waiting process

At the waiting process it captures the camera image, analyzes the image and confirms the start sign in real-time. The start sign is that the blob center is located to the inside of the special square. If the sign is confirmed then the main process is started

or else again the next image is captured and the start sign is confirmed.

2.3 Main process

If the main process is started than an image is captured and the hand region is extracted from the image. The extraction process of the hand region has the two steps.

First, it separates the pixel to be belonged the region of the hand color and be not from the given image using the mean and standard variable which are got through the sample data. But generally, like the left side image of [Picture2] we can know the fact that is extracted not only the hand region but also other region which has the similar color with the hand at the background.
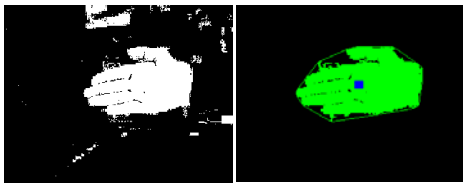
Picture2. Left) the black and white image of the hand color,
Right) the blob of hand obtained by CCL

For to obtain only the hand region, we apply CCL(Connected Component Labeling) to the black and white image and appoint the blob which has the max size from among the blobs to the hand. A color of the obtained hand blob set to a green and a color of the background set to a black.

The next work is to judge a location of the button which is pointed actually by the finger through the extracted hand blob. Generally, since the finger which points the button has the longest length from in the region of the hand blob, it seeks the longest raw and designates an axis of the raw to a location of the hand and sets the button which is belong to the raw to a candidate button. The next step is to confirm the existence yes or no of a green with scanning a region of the candidate button. If a green exists in the button region, it means that a finger is on the button. So we can think that the button is selected and set a color of the button with a green.
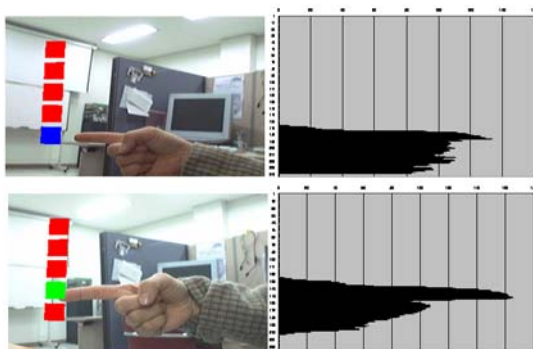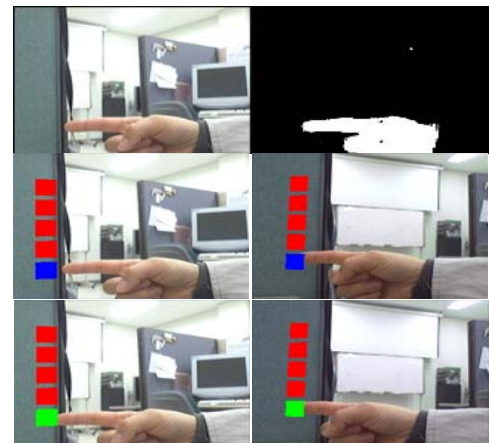

Picture3. Above) State of the candidate button and the histogram of this,
Bottom) State of the selected button and the histogram of this

If the above process is extended to the device with two cameras then seek the candidate button in the images which get with each camera. If the number of the two candidate buttons is same then a color of the buttons becomes a blue. And if a green

exists simultaneously in the region of both candidate buttons with same number then a color of the buttons become a green because the buttons came to be selected actually. Consequently, a color of the buttons is determined by running of a main process and the main process is repeated until the end event occurs.

3 Experiment and Result

Two cameras which are used from the paper are Logitech Quickcam pro5000 and the resolution is 320 x 240. The experiment accomplished from the interior. CCL works used OpenCV libraries.

Picture4. Above) Original image & binary image,
Middle) The state of the candidate button,
Bottom) The state of the selected button

As a result of an experiment we could know that an extraction of a hand region not receive largely an effect about a property of the light and camera and is accomplished well. Also the process came to be ran in real-time without difficulty of a use through the robust and fast process ability. In conclusion, the proposed user interface method decides the selective presence of the virtual button as it grasps the location of the hand using the two cameras. With this user can easily interact with an application system by selecting the virtual button. This method does not demand an additional device for grasping of information of the hand and a study which is many. Therefore the users will be able to use the system easily by intuition.

4 Reference

[1] Richard Hartley and Andrew Zisserman, Multiple View Geometry, p24-64, 2003.
[2] Rafael C. Gonzalez Richard E. Woods, Digital Image Processing, p282-344,2001
[3] Intel Open Computer Vision Library, http://sourceforge.net/projects/**opencv**library