Webcam Based Image Control System

Student Name: KONG Fanyu

Advised by: Dr. David Rossiter

CSIT 6910 Independent Project

Fall Semester, 2011

Department of Computer Science and Engineering

The Hong Kong University of Science and Technology

Contents

1 Project Introduction

Nowadays requirements of human-machine friendly system development are increasing, which challenge developers to think about methods to realize more efficient human-computer interactions.

Webcam is widely used by computer users and thought as a common device to approach. So in this project, webcam is applied to realize image control.
To support webcam used image control, popular human body detection software, OpenCV, is used to identify command generated by human body.

This project is solely designed, developed and implemented, under supervision and direction by Dr. David Rossiter.

2 Project Objectives

This Project is aimed to build an image control system by detecting command of finger movement in webcam.

Commands should contain image zoom in/out, image rotation and last/next image browsing. Different controlling options could be chosen in a dropdown list by mouse click, to avoid gesture confusions.

3 Technical Requirements

Technical requirements include both software and hardware aspects. Certain device like webcam is necessary to achieve interactions. A new programming language, processing, which is increasingly used by multimedia developers, is introduced in this project.

3.1 Hardware requirements

This project is developed in a laptop which is equipped with a webcam. It is believed that this system should be flexibly run in any computer with self-equipped or independent cameras.

3.2 Software requirements

- **Operating System**

This project is developed in Windows7. It is believed that it should work in any system which could run Processing software, such as Mac, and GNU/Linux platforms.

- **Programming language**

This project is developed with an open source programming language and environment, Processing.[3] It is developed for people who want to create images, animations, and interactions.

In this project, Processing is chosen because its Java based environment makes it flexible to connect much other software. Processing is a language which is more visualized and readable than Java (*see figure 1*), which can easily use visual context to generate multiple visualizations, and connect other professional tools with importing libraries.

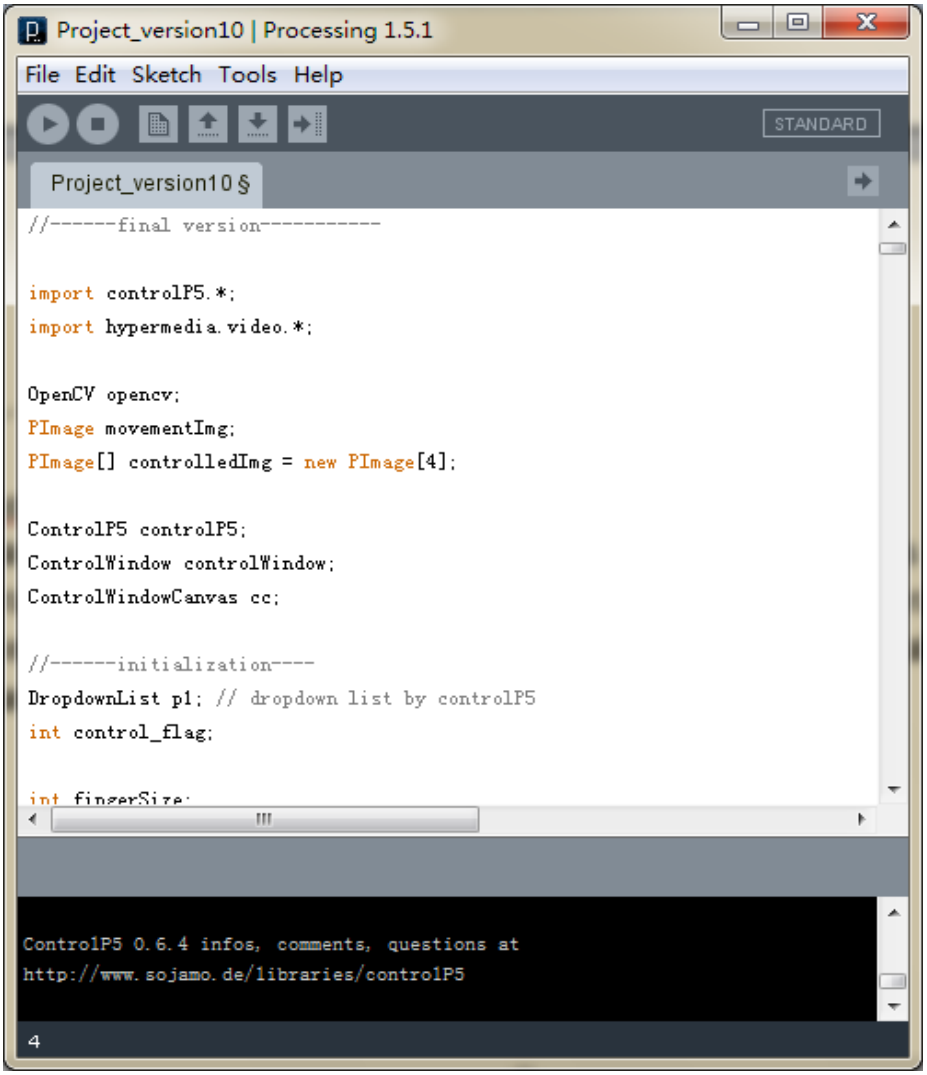

Figure 1 - Processing Programming Interface

- **Contributed Software**

Two libraries are imported to support the webcam based interaction system to realize a second window opening and finger detection in a real-time webcam generated video. They are not included in the Processing software while developed by Processing users communities.

OpenCV (import hypermedia.video.*)
OpenCV implementation for Processing includes body detection, face recognition and more.[4] So in this project, to realize webcam activation and finger movement detections, OpenCV is well chosen as an efficient approach.

Before OpenCV library installation, OpenCV 1.0 is necessarily installed to support the library.

controlP5 (import controlP5.*)
controlP5 provides users with custom GUI elements to show/hide and move while the program is running.[5]

Processing is very weak in GUI development itself, while controlP5 could contribute to interface development. In this project, controlP5 contributes to open a new window to show controlled image and a dropdown list to select options.

4 Project Outcomes

4.1 Webcam Active

Webcam activation is achieved by OpenCV library.[6] After capturing real-live video recorded by webcam, video is converted to many static images in every time unit. Processing is good at dealing with static images by examining pixels.

4.2 Finger Detection

Finger detection is realized by OpenCV library as well.

At the beginning of the project, variable finger detection methods have been put forward and seriously compared. An aborted example is to consider color of pixels which is similar to skin color as fingers.[7] Then group these pixels and evaluate the relationship of their movement. [2] This method is the most effective way within Processing without any extra libraries. But disadvantages are too many noise dots. Lights and shadows are

always identified as ‘skin color’, which make evaluation inaccurate and ineffective.

OpenCV is suggested by supervisor of this project. This software is powerful at detecting part of human body, human face and even human skin color. Though OpenCV could distinguish different unique spots of human body, which means it could separate five different fingers of one hand, it is found that by only detecting skin color is smart and efficient enough in this project.

Three spaces are detected instead of scanning the entire image to shorten time. Each space corresponds to one finger generated command. With OpenCV, skin color can be found directly by simply setting “brightness(pixel) > 100”, while 100 is a rough number depending on current light conditions. [1] [6]

4.3 System Interface

The size of the interface is set small to avoid long time in loading pixels for every video shot. However, it causes one problem. Finger movement may affect more than one interactive space.

One solution is, selecting an option from “zoom”, “rotate” and “browse” within a dropdown list by a simple mouse click.

The following two pictures show the interface of this interaction system. The background is a real-time video captured by webcam. The transparent image is the image which will be controlled in the other window. Three yellow points are active points which will detect and identify finger gestures.

A drop down list in *Figure 2* can be moved to anywhere within this window by mouse.

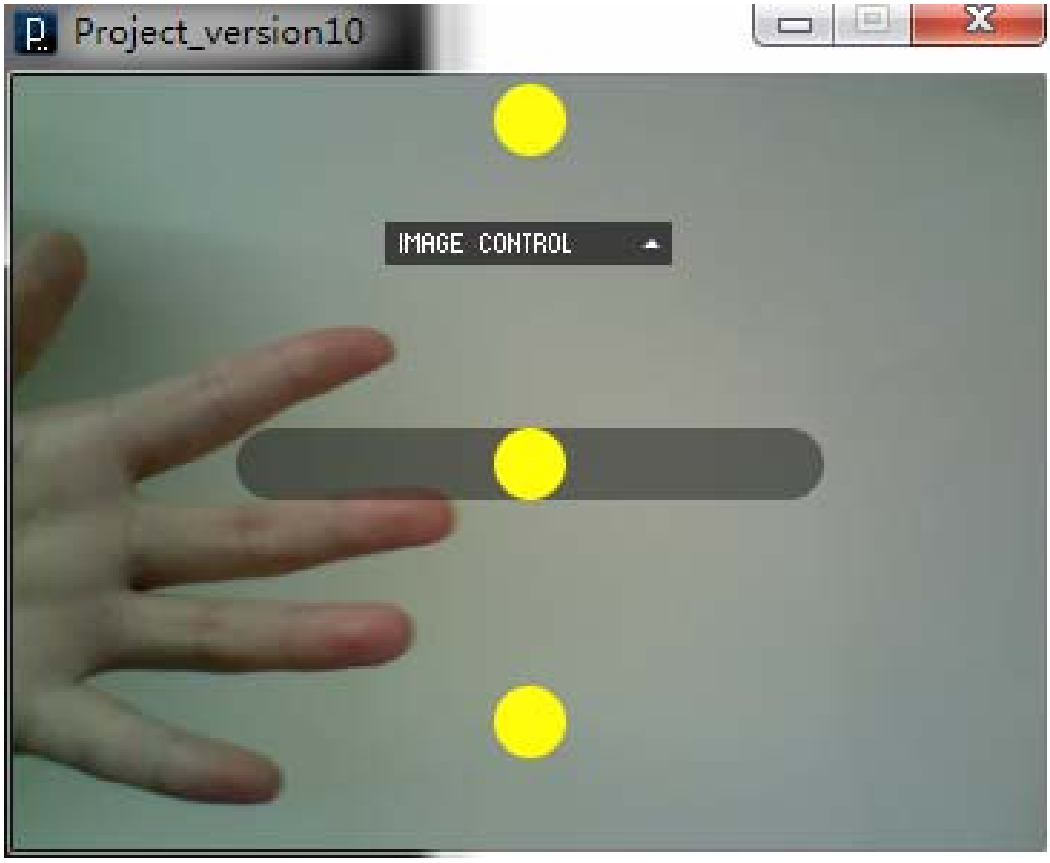

Figure 2 - System Interface

After click once, four options appear as in *Figure 3*. “RESET” button resets the image to the original size and position. “ZOOM” button will active the detection space at the bottom yellow point. “ROTATE” button actives the detection space at the top yellow point. And “BROWSE” button could active the middle detection space.

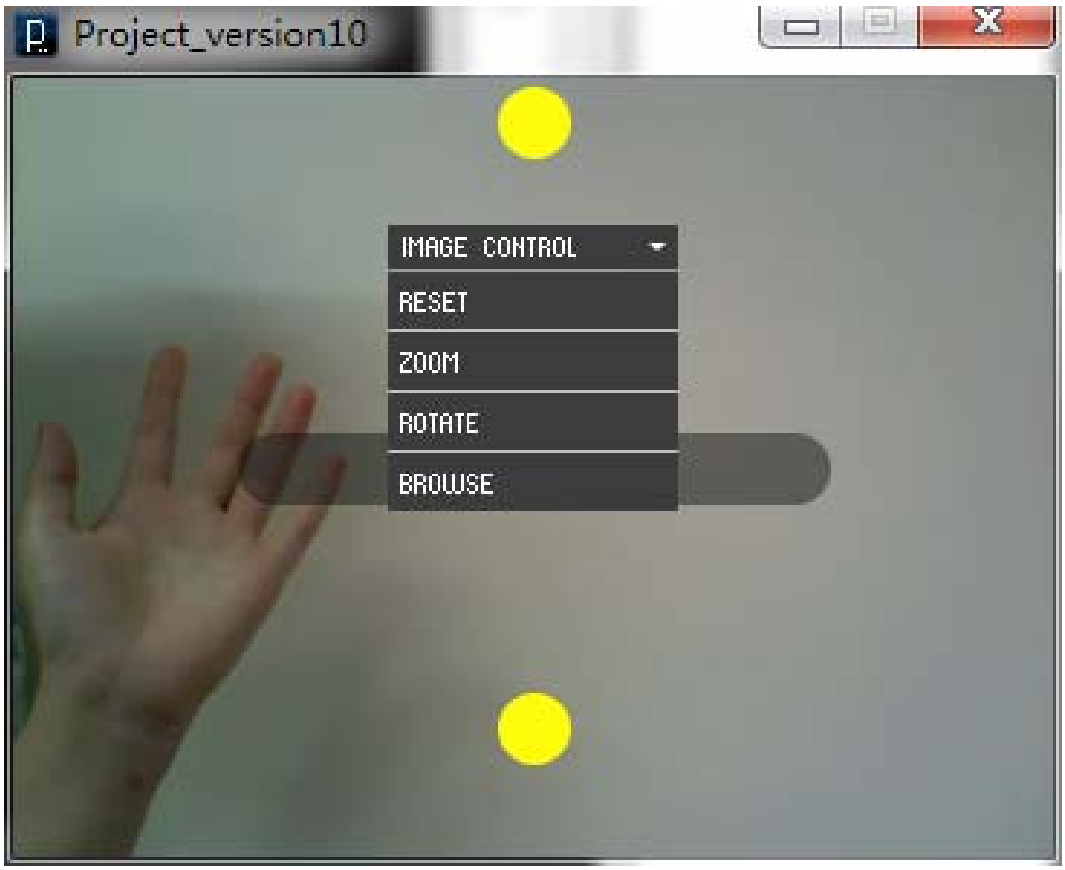

Figure 3 - Dropdown List & Options

The following *Figure 4* is a second window open by controlP5 library, which cannot be generated by Processing itself. *Figure 4* shows the original size and position of a controlled image. After zooming in/out, rotation or browsing, this image will show its change in this window, which will be introduced in detail in the following sections.

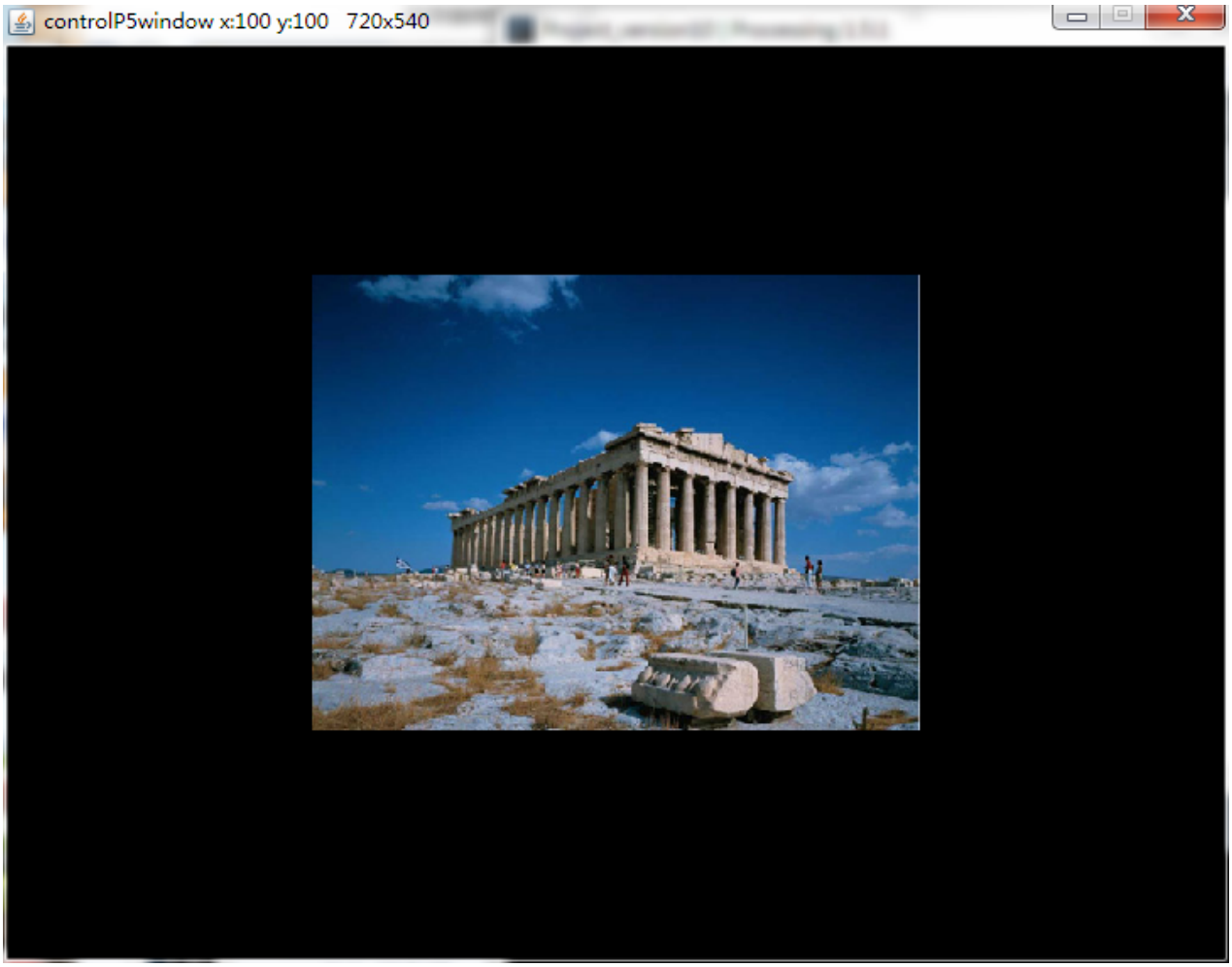

Figure 4 - Image Display Window

4.4 Reset Image to Original Status

By clicking the “RESET” button inside the dropdown list, image is set to the original size and position as in *Figure 5*. It could be applied anytime during the image control.

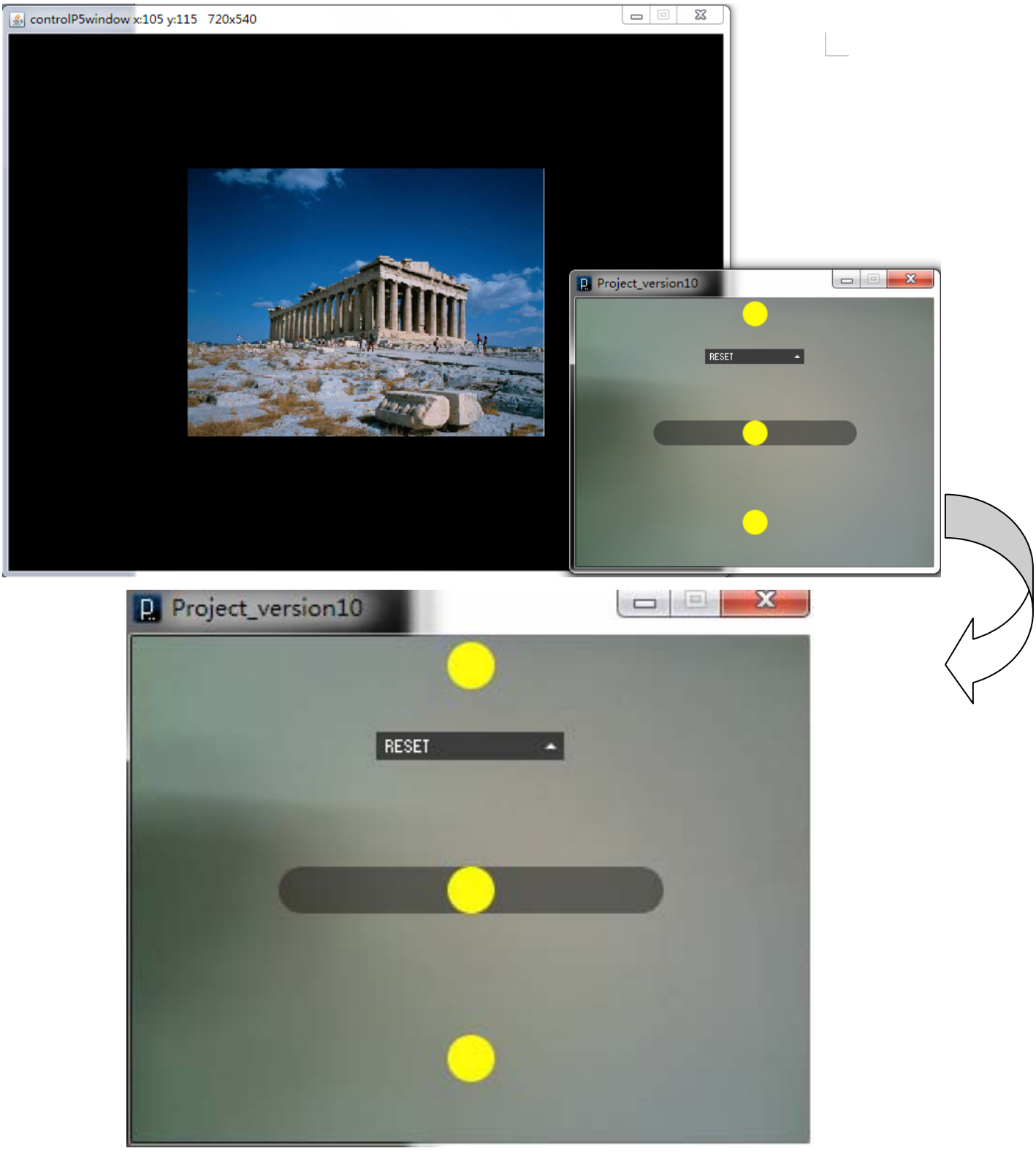

Figure 5 - Image Reset

4.5 Zoom in/out Image

Zoom function can be achieved by measuring distance between two fingers, which equals to distance between two nearest skin color. To make it more similar to human regular behaviors, pixel scanning should be done from bottom to top. Because people always use their fingertips most often as the most flexible parts of fingers.

After selecting “ZOOM”, the bottom space is active. This detection space is set within a certain width and height to avoid interference by other part of body like arm or face. The following *Figure 6* shows how finger movement controls image zoom.

In the program, the minimize distance between middle line to either fingers as the benchmark to control image size. When fingers move out of detection space, the image stays unchanged to wait for next adjustment.

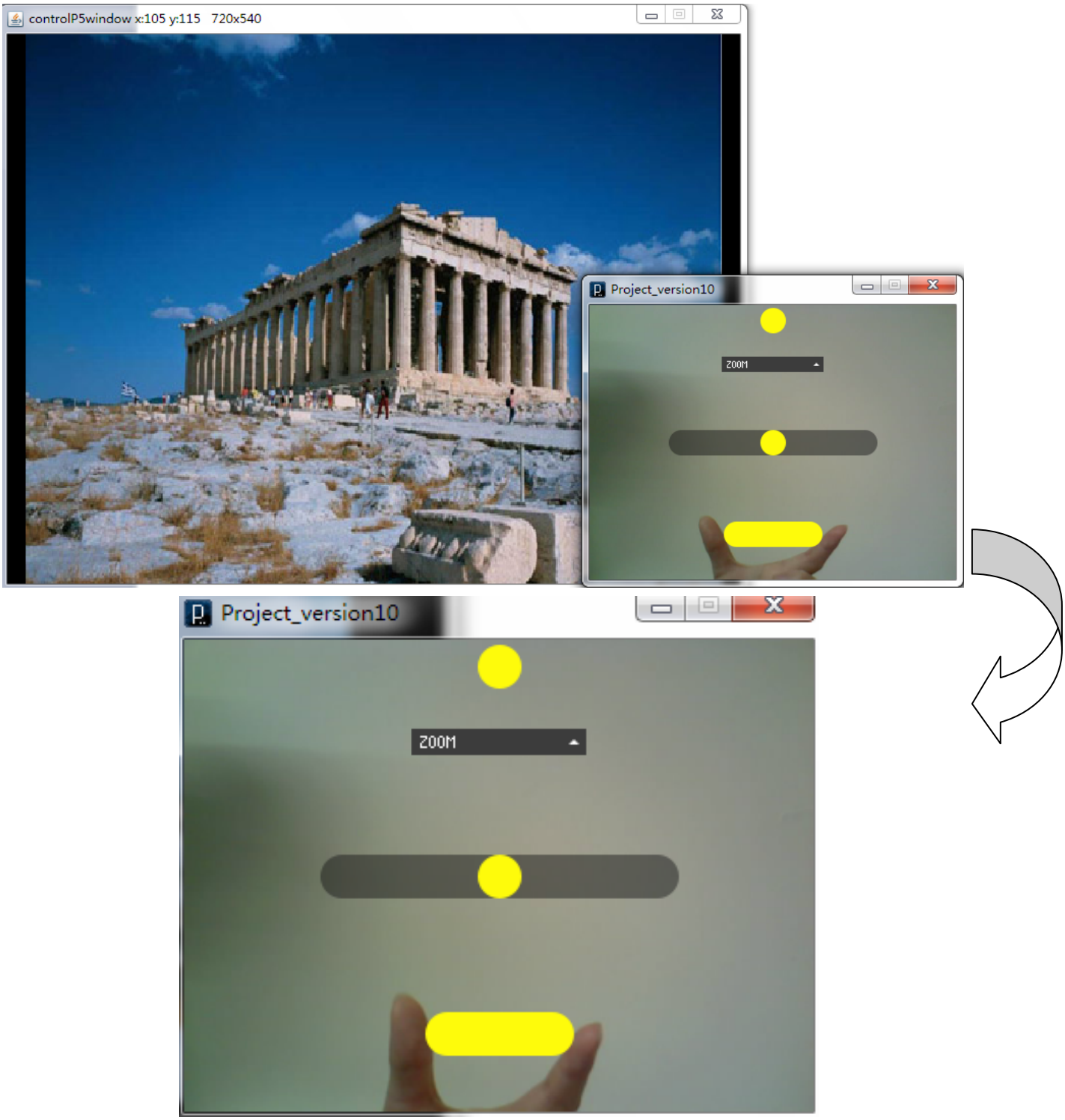

Figure 6 - Zoom in/out

4.6 Image Rotation

Image rotation angle is following the angle between original position and the current finger position. After click the “ROTATION” button, rotation space is active to detect finger. When the system detect skin color within these spaces, it examine the angle distance between the nearest skin color and original position. Then the system rotates the image in the controlled window to that specific angle. When move fingers out of the detection space, the yellow bar and controlled image stay unchanged to wait for later movement.

In the following *Figure 7*, relation between finger gesture and controlled image is clearly displayed. The detection space is 180 degree ranged from left to right, which means image could rotate from -90 degrees to +90 degrees. The reason why not applying 360 degrees is size limitation.

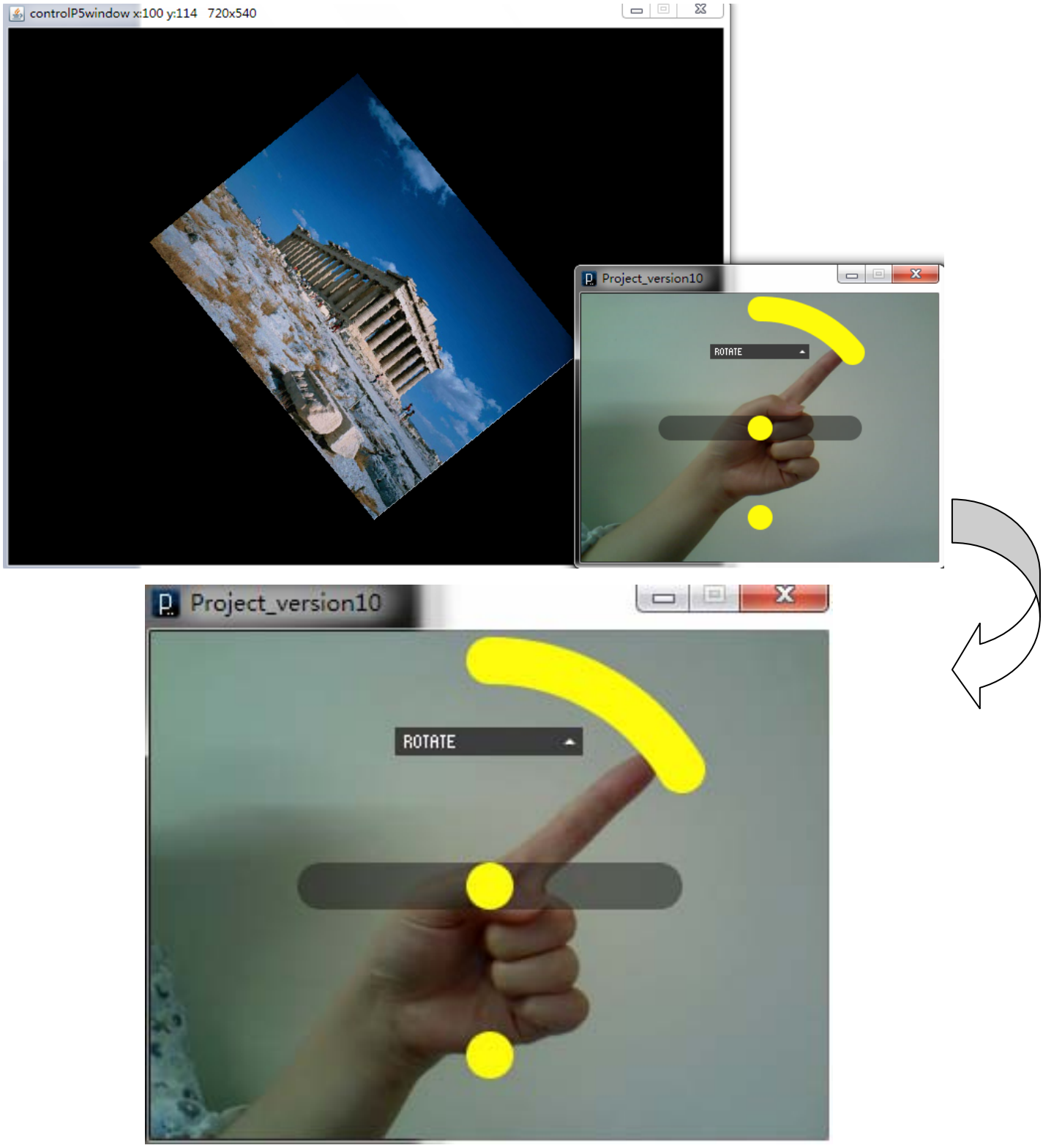

Figure 7 - Image Rotation

4.7 Image Browse

Image browse option could scan all the images inside this Processing file. With the same finger detection method, a certain active space is responded to finger movement.

When the finger controlled yellow point moves from left to right, the larger window will show the previous image, as shown in *Figure 8*. On the other hand, when slips from left to right, next image will appear. It meets the common human knowledge and expectation. This control system is very similar to ipad or iphone, as a result to be easily learned and accepted.

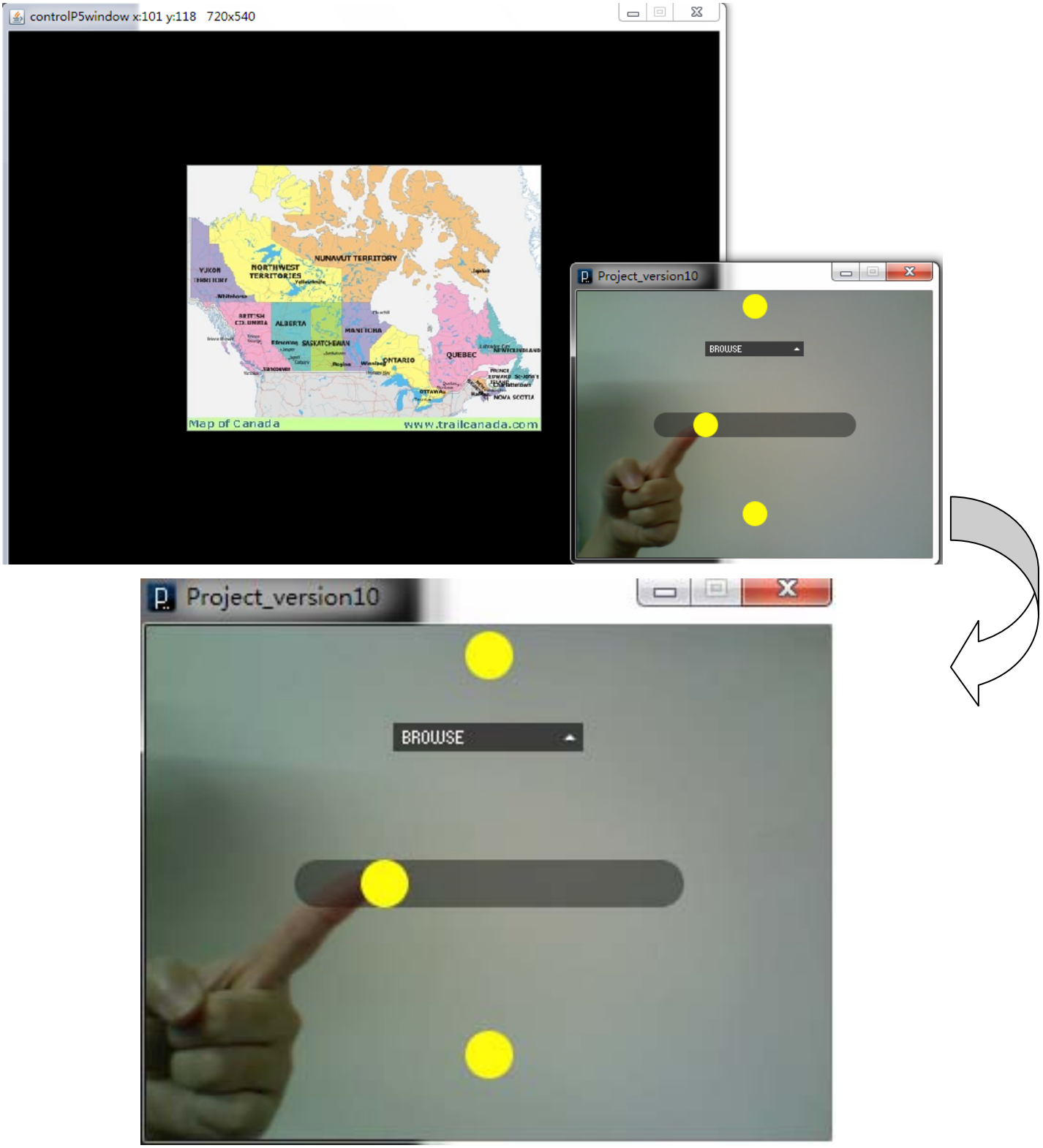

Figure 8 - Image Browse

5 Conclusion

The webcam based image control system is designed and developed based on the traditional image control finger behavior, and traditional webcam. But the combination is new and creative. Interactions between machine and human body through only camera are increasingly popular in games industry. And it is believed to be more welcomed among regular computer users to simplify computer operations.

This project is a creative trial to realize image control by webcam based finger detection. Many creative ideas have been come up with, along with many difficulties. Very few people were trying to connect Processing and OpenCV. Though new methods and new API take long time to research and attempt, but the results are exciting and beyond expected.

In the proposal, this system is expected to be developed to browse image which is connected to Google search. It has been tried during the implementation. But it seems to be very slow to load image from internet when the program is running, because of limitation of Processing itself. So this function is removed to ensure efficiency of this system.

6 Reference

[1] Gady Agam, January 27, 2006. *Introduction to programming with OpenCV*. [pdf] Department of Computer Science, Illinois Institute of Technology. Available at: <http://www.cs.cornell.edu/courses/cs4670/2010fa/projects/Introduction%20to%20Programming%20With%20OpenCV.pdf> [Accessed 12 October, 2011].

[2] Gunnar Sigurdsson, Andrew Wong, CS223B Winter Quarter 2008. *Finger Tracking for Rapid Cropping Applications.* [pdf] Stanford University. Available at: < https://ccrma.stanford.edu/~wonga/papers/WongSigurdssonFingerTrack.pdf> [Accessed 2 November, 2011]

[3] Processing Official Website, Learning and Forum. <http://processing.org/>

[4] OpenCV, Processing and Java Library. <http://ubaa.net/shared/processing/opencv/>

[5] ControlP5, Processing GUI Library. <http://www.sojamo.de/libraries/controlP5/>

[6] Vadim Pisarevsky. Intel Corporation, Software and Solutions Group. *Introduction to OpenCV.* [pdf] Available at: <http://www.cs.nccu.edu.tw/~whliao/cv2011/opencv_introduction_2007June9.pdf> [Accessed 12 November, 2011]

[7] Sung Kwan Kang, Mi Young Nam, Phill Kyu Rhee, 2008. Convergence and Hybrid Information Technology, 2008. ICHIT '08. *Color Based Hand and Finger Detection Technology for User Interaction.*

7 Appendix (Meeting Minutes)

Minutes of the 1st Project Meeting

Date: 26 Sep. 2011 (Friday)

Time: 10:20 am

Place: Rm. 3512

Attending: KONG Fanyu

Prof. David Rossiter

Absent: None

Recorder: KONG Fanyu

1 Approval of minutes

This is first formal meeting, so there were no minutes to approve.

2 Discussion Items

- Things have done

  1) Detect finger by identifying skin color
  2) Group closed piece of skin color

- Problems

  1) Skin color varies a lot from brightness conditions
  2) Too much noise points from background color

- Things to do

  1) Know more about OpenCV and try to apply it on this project.

3 Meeting adjournment and next meeting

The meeting was adjourned at 10:35 am. The next meeting will be held in 7th October, Friday.

Minutes of the 2nd Project Meeting

Date: 7 Oct. 2011 (Friday)

Time: 10:20 am

Place: Rm. 3512

Attending: KONG Fanyu

Prof. David Rossiter

Absent: None

Recorder: KONG Fanyu

1 Approval of minutes

The minutes of the last meeting were approved without amendment.

2 Discussion Items

- Things have done

    1) Install OpenCV library into Processing sketchbook
    2) Use red points to follow all the fingertips

- Problems

    1) Red points are not stable with too much “tremble”
    2) Running too slowly when window size is big

- Things to do

    1) Set up a separate window to show controlled image.
    2) Make detected finger points more stable with some algorithms like boost.

3 Meeting adjournment and next meeting

The meeting was adjourned at 10:40 am. The next meeting will be held in 21st October, Friday.

Minutes of the 3rd Project Meeting

Date: 21 Oct. 2011 (Friday)

Time: 10:20 am

Place: Rm. 3512

Attending: KONG Fanyu

Prof. David Rossiter

Absent: None

Recorder: KONG Fanyu

1 Approval of minutes

The minutes of the last meeting were approved without amendment.

2 Discussion Items

- Things have done

  1) Zoom in/out image is realized.
  2) Open a new larger window to show controlled image. Original window can be small so that taking less time to scan.

- Problems

  1) Zoom in/out should not be individually selected by mouse click.

- Things to do

  1) Let fingers freely control image zoom in and out.
  2) Add image rotation.

3 Meeting adjournment and next meeting

The meeting was adjourned at 10:40 am. The next meeting will be held in 14th November, Monday.

Minutes of the 4th Project Meeting

Date: 14 Nov. 2011 (Monday)

Time: 10:15 am

Place: Rm. 3512

Attending: KONG Fanyu

Prof. David Rossiter

Absent: None

Recorder: KONG Fanyu

1 Approval of minutes

The minutes of the last meeting were approved without amendment.

2 Discussion Items

- Things have done

    1) Image rotation added.

    2) Fingers can zoom in/out image freely.

- Problems

    1) Detection spaces interrupt each other.

- Things to do

    1) Add image browse, to meet proposal.
    2) Use menu to select command, to avoid interruptions with each other.

3 Meeting adjournment and next meeting

The meeting was adjourned at 10:40 am. The next meeting will be held in 2nd December, Friday.

Minutes of the 5th Project Meeting

Date: 2 Dec. 2011 (Friday)

Time: 10:20 am

Place: Rm. 3512

Attending: KONG Fanyu

Prof. David Rossiter

Absent: None

Recorder: KONG Fanyu

1 Approval of minutes

The minutes of the last meeting were approved without amendment.

2 Discussion Items

- Things have done
    1) Image rotation with any angles with fingers
    2) Project report and display video have been almost finished
- Problems
    1) Improve the appearance of system interface
- Things to do
    1) Remove the unchanged image in the centre of interface
    2) Update project report and presentation video

3 Meeting adjournment and next meeting

This is the final meeting. There is no more meeting for this independent project.