CamPong

Nofar Carmeli and Rom Herskovitz, supervised by Aaron Wetzler

October 27, 2013
Part I

Introduction

In this project, we introduce the use of a mobile phone, equipped with a camera and a projector, to allow real time hand detection.

We present a demo of an interactive pong game that is controlled by the players’ natural hand movements during the game.

To the best of our knowledge this is the first known use of a commodity cellular phone that uses an inbuilt projector to perform real time structured light projections coupled with real time image processing.

The project is built with very high constraints of both CPU and memory usage which are even more prominent due to the requirement to perform real time image processing in conjunction with screen projecting and in-game logic processing.

This combination between state of the art hardware and software will allow a whole new field of applications based on natural user interface that are ultra-portable and inexpensive.

The game is a modern implementation of the classical pong game, where two players compete in a tennis like game. The game is projected on a flat surface (e.g. a nearby wall) while the camera captures the projection and the players’ movements. Each player controls their in-game paddle simply by moving their hands, and the paddles move with their hands along the sides of the projected screen in real time.

The project is implemented on a “Samsung Galaxy beam” smartphone device.
Part II

General Outline

The project is built for the Samsung Galaxy beam - an android based smartphone that has a built in projector on top and a 5 MP camera. We installed a prism on the projector such that the screen is projected into the cameras’ range of view.

We used Eclipse IDE and OpenCV library in the development process.

The project is made out of 3 modules:

1. Game Engine
2. Image Processing Engine
3. Game View

1 Game Engine

The Game Engine is responsible for handling all the game related logic.

It performs the following tasks:
• Loading the initial game state
• Updating the current ball position
• Updating the location of the paddles according to the changing hand coordinates it receives from the Image Processing Engine
• Keeping track of score
• Playing the game sounds

2 Image Processing Engine

The Image Processing Engine processes the data from the phone’s camera.

It has two main tasks:

2.1 Initialization

Detecting the location of the two stripes on which the paddles can move.

The initialization process is as follows:

1. Project stripes
2. Retrieve live image
3. Perform binary threshold
4. Add opening morphology for noise reduction
5. Create a histogram of the thresholded image
6. Determine stripe boundaries based on the histogram
7. Save stripe images for future use
2.2 Real time hand detection

The real time hand detection process is as follows:

1. Retrieve live image
2. Perform median filtering for noise reduction
3. Extract stripe areas from the image
4. Increase pixel values to further reduce noise
5. Compare the extracted stripes to the saved stripes
6. Create a histogram based on the comparison
7. Find the median of the histogram

The median is used as the new hand location and is passed to the Game engine.

3 Game View

The Game View displays the game to the phone’s screen and the projector. It draws the ball, the paddles and the scoring as they move and change.

Part III

Real Time

Our challenge in real time was to detect the center of the players’ hands using minimum amount of processing so that not to slow the movement of the paddles and the overall game, while still detecting them with the required accuracy.
The most important step we took on our way to this goal was restricting the areas that we consider as input to thin stripes in the middle of each lit side of the screen. We ’cut’ all the images we work with to these small stripes to prevent any overhead originated from working with the original input image. By doing this we created two main advantages:

1. We had fewer pixels to check, which increased speed

2. The pixels we worked with, which were further away from the edges of the stripes had less noise, which increased accuracy.

We had to find the perfect balance – we used the smallest width we could find that wasn’t too sensitive to noise.

In each stripe, we considered the pixels that were significantly darker then the saved background, and used a histogram over the y axis to calculate their median, which was now declared as the new center for the paddle.

We compared this method of pixel comparison against using a threshold, and found it to be much more accurate while not sacrificing performance.

We had to be extremely sensible about the code that runs every time the image updates. We reduced the amount of access to the image pixels to minimum.
Part IV

Screenshots and video

The CamPong movie
Part V

Summary

In conclusion, we showed that real time image processing on smartphones is achievable, and can be used to create a fully functional natural user interface.

We demonstrated the power of combining this technology with screen projection to create a game that is controlled entirely using hand detection, while working at the desired speed and reaction time.

Now we can start to imagine other future uses to this kind of human - mobile device interaction.