CURSOR MOVEMENT ON OBJECT MOTION

DIGITAL SIGNAL PROCESSING PRACTICE

MINI-PROJECT

J SARATH CHANDRA- EVD18I015

K BHARATH- ESD18I009

SRI HARSHA- ESD18I013

KRISHNA KISHORE - ESD18I018

Abstract

In some applications of today's world human computer interaction need to without any devices/human touch are needed, so that it would make their tasks more efficient. In today's world of pandemic touch less devices in public places are needed like remote less TV, Air conditioner. These kind of applications may require tracking a moving object like hand or a stick(like stick used in ps4) .Our aim is to control the cursor of your computer, not using any input devices like mouse, trackpad, or stylus, but by tracking movement of an object.

Introduction

Since the computer technology continues to grow up, the importance of human computer interaction is enormously increasing.Creating a virtual human computer interaction device such as mouse application has been designed and implemented using a regular webcam. The moticvation was to create a object tracing application to interact with computer, and develop virtual human computer interaction.

As computer technology continues to develop, people have smaller and smaller electronic devices and want to use them ubiquitously. There is a need for new interfaces designed specifically for use with these smaller devices. Increasingly we are recognizing the importance of human computing interaction (HCI), and in particular vision-based gesture and object recognition. Simple interfaces already exist, such as embedded- keyboard, folder-keyboard and mini-keyboard. However, these interfaces need some amount of space to use and cannot be used while

moving. Touch screens are also a good control interface and nowadays it is used globally in many applications. However, touch screens cannot be applied to desktop systems because of cost and other hardware limitations. By applying vision technology and controlling the mouse by natural hand gestures, we can reduce the work space required.

In this project we are trying to track the the movement of the object and give the input to the mouse pointer and make it move in the same direction.

Brief Literature Survey

Area of virtual human computer interaction is an active field of research in many organisations around the world(GAFA). Researchers have invented many ways to control computer virtually, many of them include artificial neural networks for changing volume of sound track or changing song in a music player, changing tv screen from distance using mobile devices or hand gestures. These applications include using complex machine learning systems and complex hardware to detect the movements and respond accordingly.

Many researchers in the human computer interaction and robotics fields have tried to control mouse movement using video devices. However, all of them used different methods to make a clicking event. One approach, by Erdem et al, used finger tip tracking to control the motion of the mouse. A click of the mouse button was implemented by defining a screen such that a click occurred when a user's hand passed over the region . Another approach was developed by Chu Feng Lien

. He used only the finger-tips to control the mouse cursor and click. His clicking method was based on image density, and required the user to hold the mouse cursor on the desired spot for a short period of time. Paul et al, used still another method to click. They used the motion of the thumb (from a 'thumbs-up' position to a fist) to mark a clicking event thumb. Movement of the hand while making a special hand sign moved the mouse pointer.

A paper of Asanterabi Malima et al, in which they developed a finger counting system to control behaviours of a robot. They proposed a fast algorithm for automatically recognizing a limited set of gestures from hand images for a robot control application.

Some researchers(Zhang Naizhong ,Wen Jing,Wang Jun) tried moving the cursor by tracking movements of mouth. System firstly nonlinearly transformed the input video frame of human head into YCbCr color space, then use the visible chrominance feature of face in this color space to detect human face region. And then for face candidate, use the nearly reversed relationship information between Cb and Cr cluster of face feature to detect mouth position.

There were various methods moving cursor based on laser pointer attached to head to move the cursor. If cursor is at end of the screen laser records the movements of the head.

Methodology

Our method focusses on moving the cursor by movement of the finger tips. But detection of finger tips is difficult instead we are using them colour pointers or any object of definite shape so that we can detect its colour and track its movement. Our colour of choice is red.

After taking the input from the camera, the input frames received are converted into grey scale for further modifications and filtering operations. Filtering of noises is necessary because the image quality is low and often frames at some rate, so input is subjected to some processing noise and other noises because of the camera specifications.

After converting it into grey scale, we will remove the red colour luminance from it and all non red components are set to zero.

We will not be tracing the whole image we are tracing only the centroid of the image, because the centroid is a single point with (x,y) coordinates, which means if these coordinates are helpful in moving the cursor rather than giving the whole image, as it would not able to detect small movements if the whole image is given as the input. for example, lets say if we move the colour pointer by say 1mm, it is undetectable if full image is considered, if only centroid is considered we can move the cursor accordingly in same direction with almost same slope.

In order to properly get the rounded edges of the colour pointer, we add a rectangle(bounding box) which just encircles the object to get a much precise location of the centroid.

These rounded off rectangle values are called bounding values.

The centroid values of the bounding box is given to the cursor using java mouse control operations to move the cursor by tracing the object's motion.

Work done

ESD18I009:

MATLAB Code

EVD18I015:

MATLAB CODE

ESD18I013:

Studying research papers and finding various methods to solve problem statement.

ESD18018

Studying research papers and finding various methods to solve problem statement.

Block Diagram

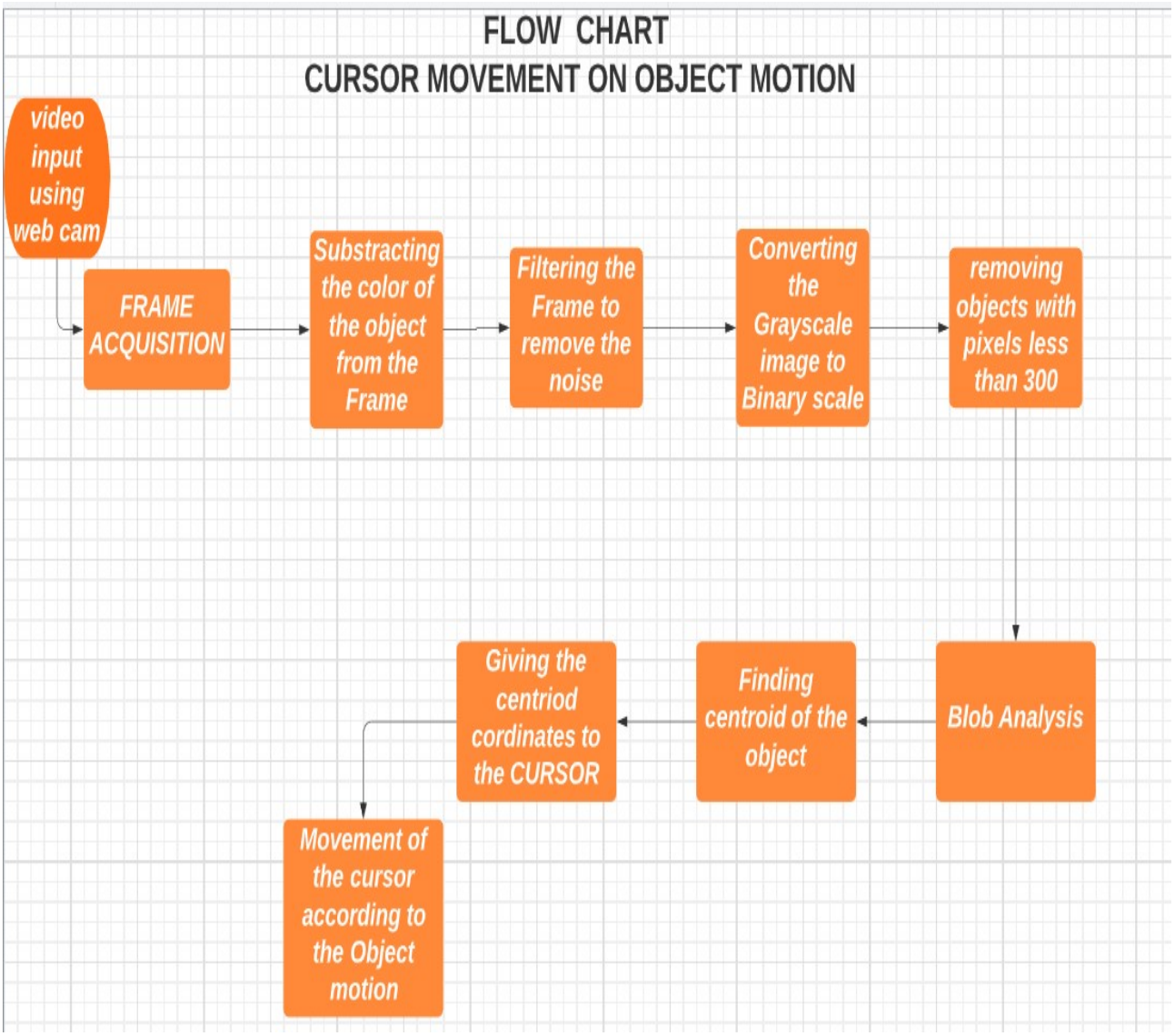

Figure 1: Block Diagram

Execution procedure

1. Video input

Video is being taken as a input from the camera attached to the laptop.

2.Frame Acquisition

We are setting frames per trigger as infinity.

Colour space is ”rgb”

frames are being taken at a rate of 5ms

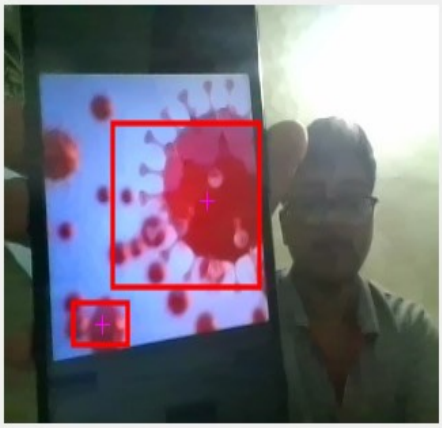

Figure 2: video input

3.Extracting red colour from the data

data(:,:,1) extracts all red colour components from the image.

imsubtract removes red components from grey scale image of the input data.

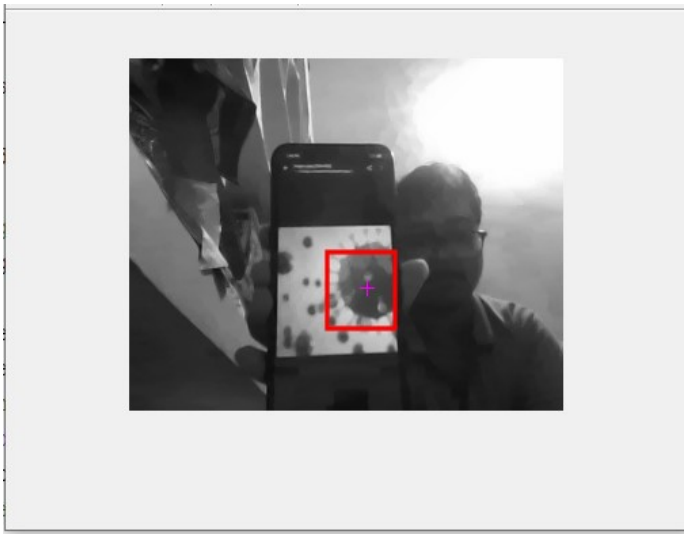

Figure 3: Grey scale

Figure 4: Output-1

4.Filtering

Filtering clears all sorts background noise caused due to various factors such as instability of the camera or system problems while recording the video.

Median filter is used in this case it is help full to reduce noise as well as to preserve edges in the grey scale image.

Figure 5: Output-2

5.Grey scale to Binary

Grey scale is converted into binary. Binary has only two luminance

1. White if the intensity is greater than the specified intensity.
2. Black, if the intensity is less than specified intensity.

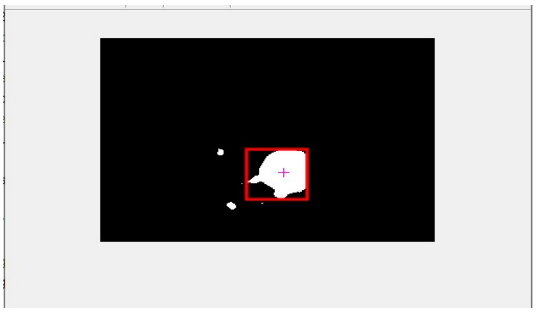

Figure 6: Output-3

6.Removing pixel values < 300

By removing pixel values less than 300 we are left with only red components, which are of considerable illuminance

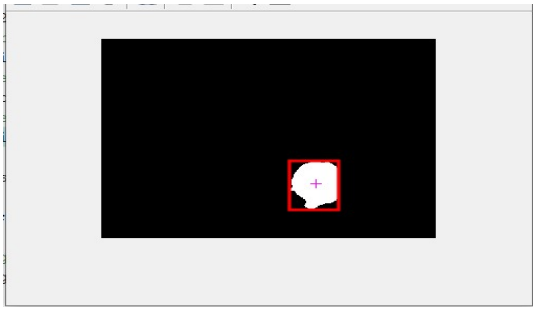

Figure 7: Output-4

7.Blob analysis

The Blob Analysis block calculates statistics for labeled regions in a binary image. The block returns quantities such as the centroid, bounding box, label matrix, and blob count. The Blob Analysis block supports input and output variable-size signals. You can also use the Selector block from Simulink, to select certain blobs based on their statistics.

bwlabel- returns a label matrix that contains labels for the 8 connected objects in binary image.

Bounding Box:returns the x and y bounds of the smallest rectangle enclosing a polyshape (Any regular or irregular shape can used).

Synax:

$[xlim, ylim] = boundingbox(polyin)$returns the x and y bounds of the smallest rectangle enclosing a polyshape. xlim and ylim are two-element row vectors whose first elements correspond to the lower x and y bounds, and whose second elements correspond to the upper x and y bounds.

When polyin is an array of polyshape objects, xlim and ylim describe the bounding box enclosing all polyshape elements of polyin.

regionprops:

$stats = regionprops(BW, properties)$ returns measurements for the set of properties for each 8-connected component (object) in the binary image(output received from labelling the connected objects of the image), BW. You can use regionprops on contiguous regions and discontiguous regions.

Centroid:

$[x, y] = centroid(polyin)$returns the x-coordinates and the y-coordinates of the centroid of a polyshape.

rectangle:$rectangle('Position', pos,'Curvature', cur)$ adds curvature to the sides of the rectangle. For different curvatures along the horizontal and vertical sides, specify cur as a two-element vector of the form [horizontal vertical]. For the same length of curvature along all sides, specify cur as a scalar value. Specify values between 0 (no curvature) and 1 (maximum curvature).

Importing Java Classes

public class Robot

extends Object

This class is used to generate native system input events for the purposes of test automation, self-running demos, and other applications where control of the mouse and keyboard is needed. The primary purpose of Robot is to facilitate automated testing of Java platform implementations.

Using the class to generate input events differs from posting events to the AWT event queue or AWT components in that the events are generated in the platform's native input queue. For example, Robot.mouseMove will actually move the mouse cursor instead of just generating mouse move events.

Clearing memory

All memory is cleared at the end of the program.

using *flushdata* command. It deletes all the frames currently stored in memory or only those frames associated with execution of trigger.

Relevant code

```
%video input from web cam ,format winvideo,aspect ratio 320x240
vid=videoinput('winvideo',1,'YUY2_320x240');
%setting frames per trigger as infinity
set(vid, 'FramesPerTrigger',Inf);
%specifying the color space as rgb
set(vid,'ReturnedColorspace','rgb');
%to take frames every 5ms
vid.FrameGrabInterval=5;
%command to start acquiring video
start(vid);
while(vid.FramesAcquired<=200)
   %storing the extracted frames into data
   data=getsnapshot(vid);
   %for tracing a red object we have to substarct the red component from image
   diff_im1=imsubtract(data(:,:,1),rgb2gray(data));
   %using median filter to remove the noise
   diff_im2=medfilt2(diff_im1,[3,3]);
   %converting the resulting grayscale image into binary image
   diff_im3=im2bw(diff_im2,0.18);
   %removing all the objects which are in less than 300 pixels
   diff_im4=bwareaopen(diff_im3,300);
   %labeling all the connected objects in the image
   bw=bwlabel(diff_im4,8);
   %blob analysis
```

```
    stats=regionprops(bw,'BoundingBox','centroid');
    %showing image
    imshow(data);
    hold on
    %loop for rectangular bounding boxes
    for object=1:length(stats)
        %finding the centroid and boundary values
        bb=stats(object).BoundingBox;
        bc=stats(object).Centroid;
        %using rectangle function to draw rectangles with data
        rectangle('position',bb,'EdgeColor','r','LineWidth',2)
        plot(bc(1),bc(2),'-m+')
%processing completed
%importing java files for mouse control operations
import java.awt.Robot;
vm=Robot;
%giving the coordinate of the centriod of the object to the cursor
vm.mouseMove(bc(1),bc(2));
    end
    %closing all loops and deleting memory
    hold off
end
stop(vid);
flushdata(vid);
```

Hardware setup

Our hardware setup is a laptop with a webcam:

Figure 8: Hardware setup

Simulations and Discussions

Output

The output of the executed procedure is the movement of the cursor by tracing the object that is in movement before the camera.

Outputs from initial stages of execution.

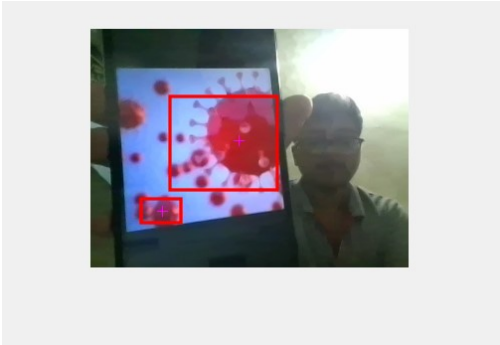

Figure 9: Input data

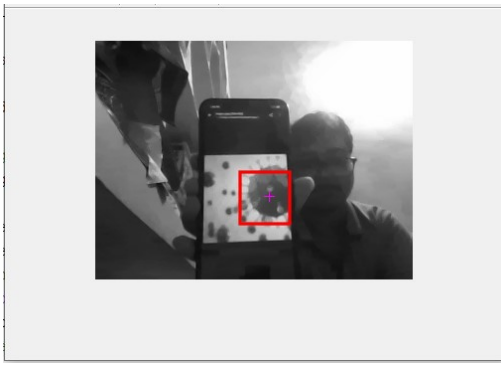

Figure 10: Grey scale

Figure 11: After red components from the frame

Figure 12: Median filter

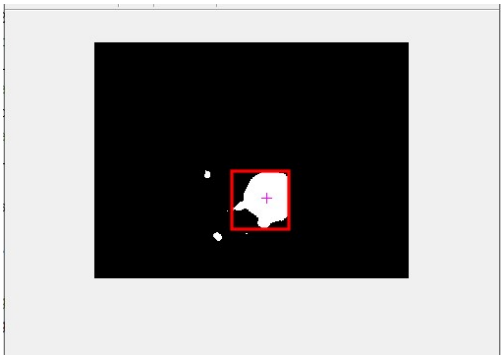

Figure 13: Binary Image

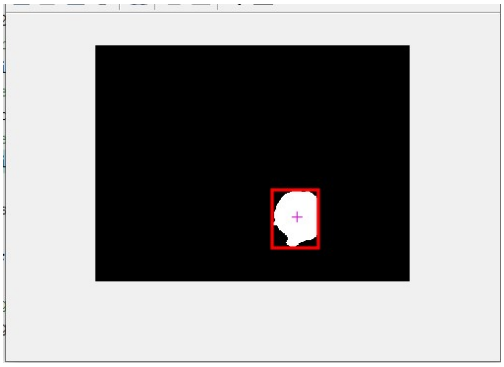

Figure 14: After removing pixels <300

Each of the above images are the output of the initial stages of operation.

First two images are the input and the grey scale converted image

Second set of images indicate the removal of red components from the grey scaled image and filtering of the image.

Third set of images indicate the binary converted image from grey scale and removal of pixels less then 300.

These are initial outputs which are analysed, the final output is the movement of the cursor which is the response to the movement of the object and cannot be captured in a image, it is presented as a video, that will be attached with this document at the time of submission.

Conclusion

In this study, an object tracking based virtual mouse application has been developed and implemented using a webcam. The system has been implemented in MATLAB environment using MATLAB Image Processing Toolbox. As an object a red colour object is used to make the detection easy and fast. Object detection and motion tracking worked very well. Using the pointer moving the cursor.

If there are multiple red coloured object movements on the screen the system follows the object whose pixel values is greater than 300, this helps in avoiding the problem of which object to follow.

Moving object detection has many applications like video surveillance, activity recognition, road condition monitoring, airport safety, monitoring of protection along marine border.

References

1- M. E., Erdem, I. A., Atalay, V., Cetin, A. E. (2003). ”Computer vision based unistroke keyboard system and mouse for the handicapped” 2003 International Conference on Multimedia and Expo. ICME ’03. Proceedings .

2-Chu-Feng Lien, “Portable Vision-Based HCI – A Real-time Hand Mouse System on Handheld Devices”, National Taiwan University, Computer Science and Information Engineering Department

3-Asanterabi Malima, Erol Özgür, and Müjdat Çetin,”A Fast Algorithm for vision-based hand gesture recognition for robot control”,Faculty of Engineering and Natural Sciences, Sabancı University, Tuzla, İstanbul, Turkey.

4-Zhang Naizhong ,Wen Jing, Wang Jun, ”Hand-Free Head Mouse Control Based on Mouth Tracking”,Dalian Polytechnic University,China.