



DESKTOPOGRAPHY

Contents

1	Acknowledgements	2
2	Introduction	3
3	Hardware	4
3.1	Kinect As A Depth Camera	4
3.2	Projector	4
4	Theory - Computer Vision	5
4.1	Finger Segmentation	5
4.2	Alignment of images	6
4.3	Getting the fingertips	7
4.4	Click detection	8
4.5	Gestures	8
5	Theory - Graphical User Interface	9
5.1	Languages we used to create our GUI	9
5.1.1	HTML	9
5.1.2	CSS	10
5.1.3	Javascript	10
5.1.4	JQuery	11
5.1.5	Web-sockets and Python Server sent events	11
6	Future Work	12
7	The Team	13
8	Mentors	13

1 Acknowledgements

We are grateful to the the Electronics club and SnT council IIT Kanpur for providing us with the resources and motivation for this summer project. We thank our mentors and co-ordinaters of Electronics club: Ashok kumar choudhary and Suraj Hinanchal to guide us for our project.

We are thankful to the Hrvoje Benko, Chris Harrison and Andrew D. Wilson from Microsoft who are the actual people behind this idea that we have implemented in this project.

2 Introduction

Our keyboards limit us .Laptops, Desktops ,phones or any other digital devices use only a part of the available space around us. Can you imagine that even the very desks on which we do our daily work have more workspace then these digital devices.



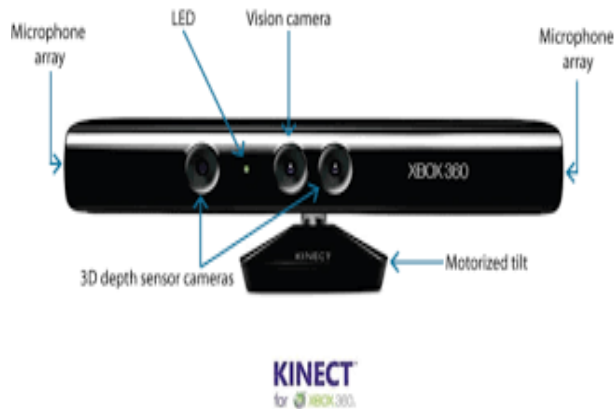
Today's mobile computers provide omnipresent access to information, creation and communication facilities. It is undeniable that they have forever changed the way we work, play and interact. However, mobile interaction is far from solved. Diminutive screens and buttons mar the user experience, and otherwise prevent us from realizing their full potential.

In this paper we explore and prototype a powerful alternative approach to mobile interaction that uses a head on projection onto any surface area the real world provides. For example, the surface area of one hand alone exceeds that of typical smart phone. Tables are often an order of magnitude larger than a tablet computer. If we could appropriate these ad hoc surfaces in an on-demand way, we could retain all of the benefits of mobility while simultaneously expanding the interactive capability.

3 Hardware

3.1 Kinect As A Depth Camera

We have used two devices in our project .A kinect which is a depth camera providing the depth map as well as the RGB video stream. It has an receiver and a sender ,the sender throws out infrared beam , which on reflection from different surfaces is received back into the receiver ,through this technique the kinect determines the relative distances of objects from the kinect and shows the user the output in form of a depth image in real time.



3.2 Projector

The projector is needed to calibrate with the coordinates provided from the kinect and use them to define where to project the interfaces .



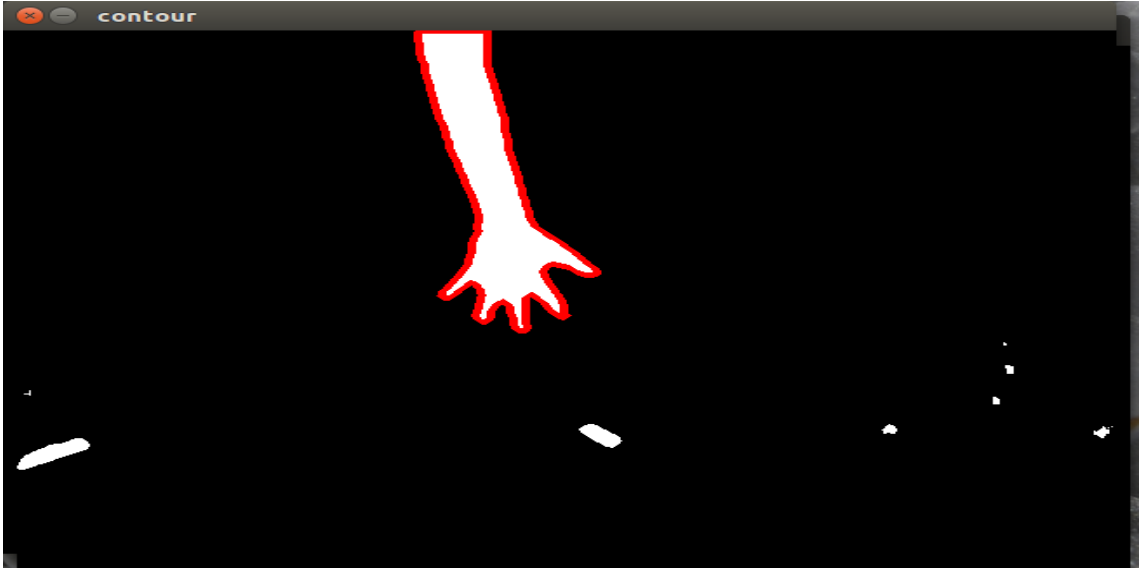
4 Theory - Computer Vision

This section explains the basic functioning of our model in brief. First step involved in the process is to segment the hand from its surroundings ,this can be achieved through various methods like background subtraction, depth segmentation ,etc, but we preferred skin detection technique to achieve this .After getting the hand region through skin detection ,the depth information of each and every point on the hand region is needed ,but the kinect has a problem .The rgb camera has a larger field of view than the infrared camera ,so it is a must to align the depth and bgr images of the kinect. Now after the alignment ,the fingertips of the hand can be accessed easily .This task can be fulfilled through various techniques ,our technique will be explained later. Click detection is the next priority .Click detection involves sending the coordinates of the fingertips as their is a touchdown on the surface. thresholding depth is one way to achieve it.

4.1 Finger Segmentation

Finger segmentation was done through skin detection. Skin detection basically involves firstly selecting a particular color space (we used HSV color space) , after selecting the color space set a max and min color range so that it covers all the skin colors. set a mask on the image through thresholding coloring all the other pixels black and obtaining a white hand. In some cases there will be noises in the hand or in the background these can be eliminated by using proper blurs and filters.

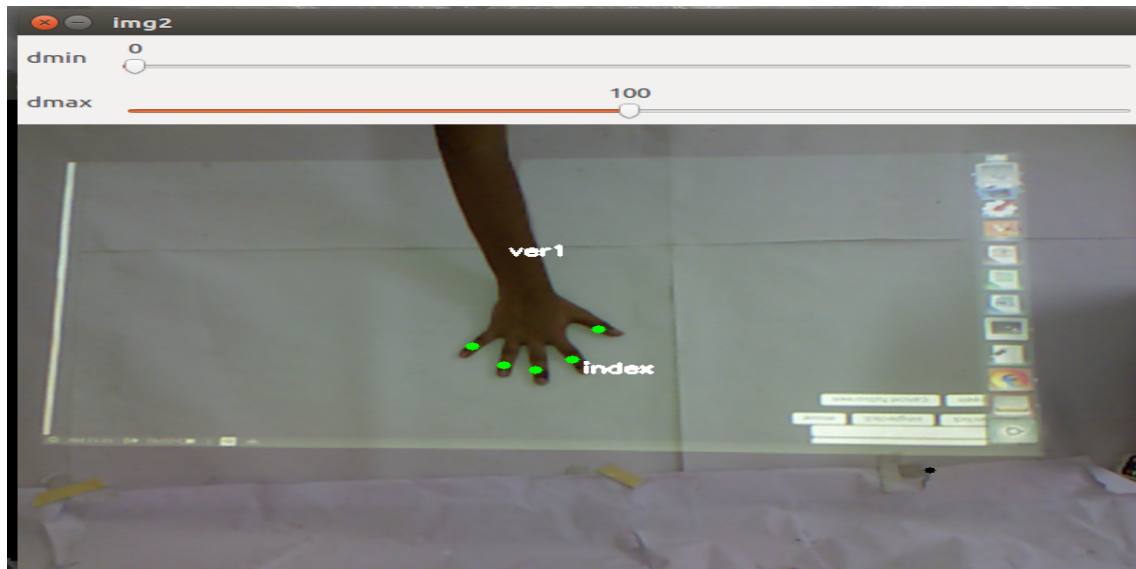




4.2 Alignment of images

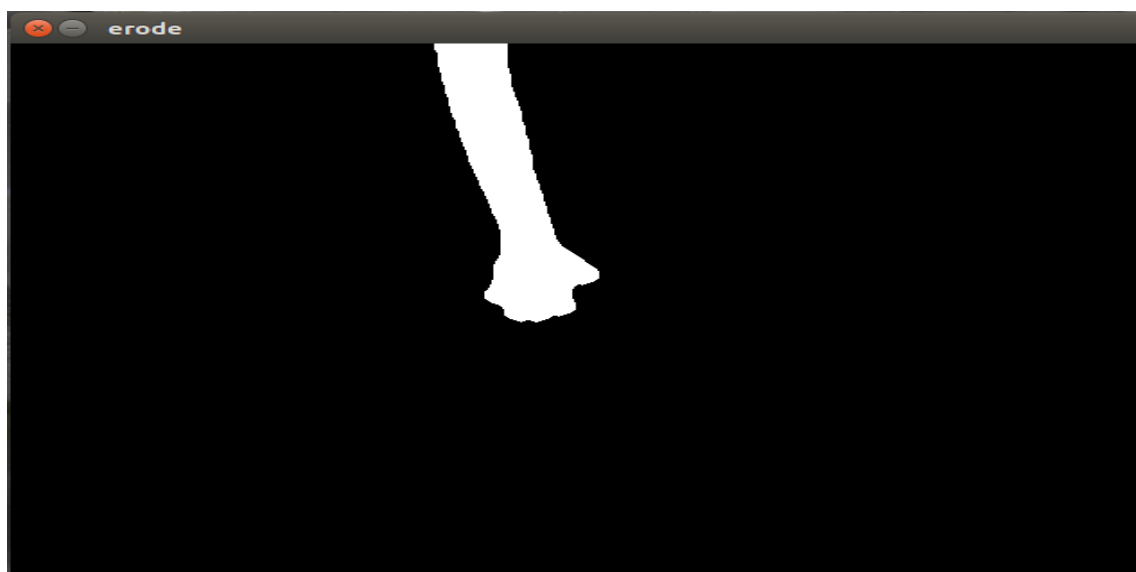
As mentioned above ,the rgb camera and depth camera images need to be aligned . Any alignment task proceeds by obtaining the intrinsics and extrinsics parameters of both the camera .Basic chess board calibration can be done to achieve this .For our need we took 45 images of the chess board in different orientations. Apply the rotation and transformations matrices on the parameters to complete the alignment.

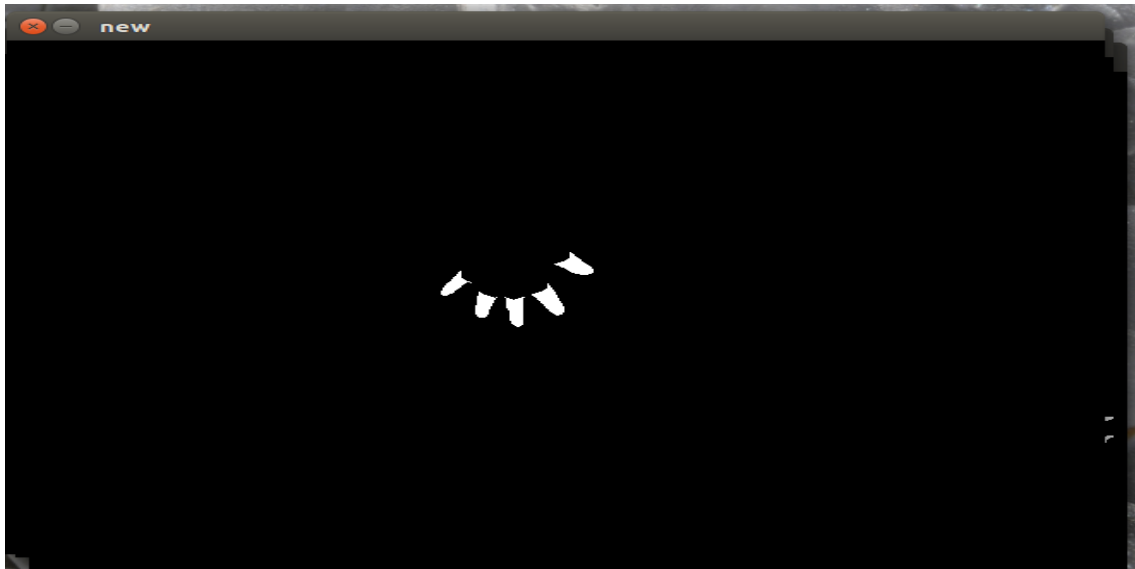




4.3 Getting the fingertips

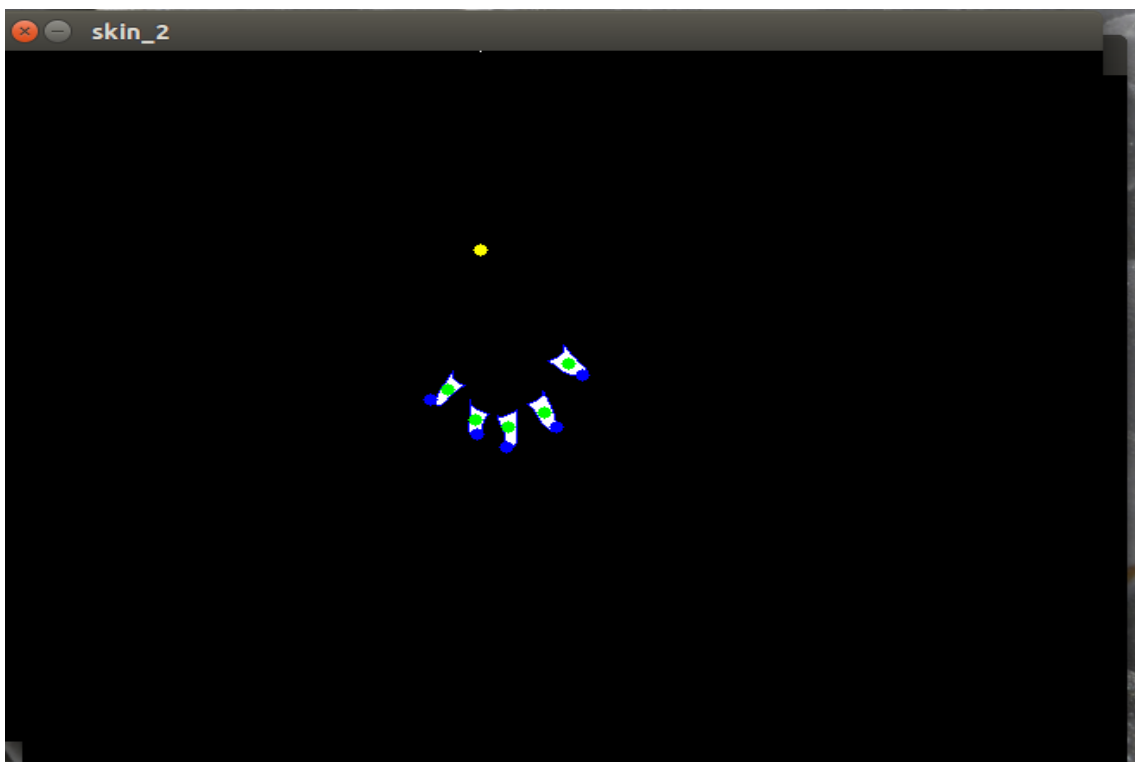
Before proceeding on this step one should ensure a clear mask of the hand ,because now we need to extract the fingertips with negligible noise in our image. Extract out the palm of the hand through the get ellipse method .subtract the two masks to get the clear five fingers .Draw contours on the five fingers and obtain their centroids through the moment function. Opencv also provides the function to fit an ellipse around any contour and get its major axis ,minor axis ,center of the ellipse as well as the angle through which it is rotated. This functionality can be used to get the endpoints of the ellipse. Its easy to understand that one of these endpoints will be the fingertips ,so use the center of the palm and compare the distances of each endpoint to figure out which one corresponds to the fingertip.





4.4 Click detection

Click detection be achieved by setting min and max depth parameters. If the fingertip is above a certain height then no touch is detected ,if below that depth then a touch is detected.



4.5 Gestures

Double clicks ,triple clicks ,single clicks all this were implemented in our model .The time package of python allows you to distinguish these clicks easily by

providing the time lapse between each touchdown.

5 Theory - Graphical User Interface

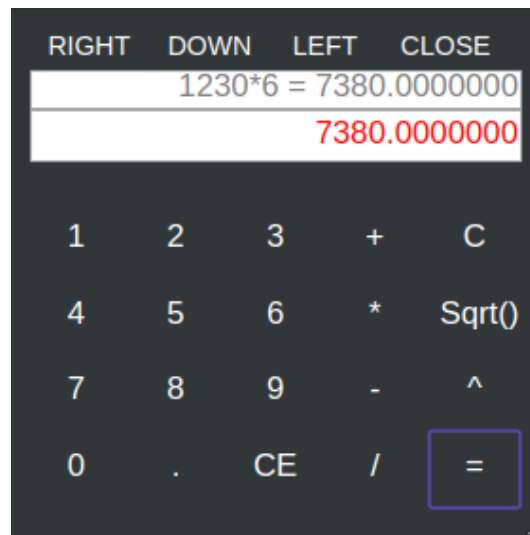
The graphical user interface is a type of user interface that allows users to interact with electronic devices through graphical icons and visual indicators such as secondary notation. For the surface to be interactive there should be something to interact with. so it was very important for us to come up with Graphical User Interface. Designing the visual composition and temporal behavior of a GUI is an important part of software application programming in the area of humancomputer interaction. Our goal was to enhance the efficiency and ease of use for the underlying logical design of a stored program, a design discipline named usability. We chose Javascript, JQuery, HTML5 and CSS as a platform to make our GUI on. We had to decide precisely as to which apps are going to be most important for the user

5.1 Languages we used to create our GUI

We had to decide precisely as to which apps are going to be most important for the user. For making such GUI we chose Javascript, JQuery, HTML5 and CSS as a platform which would create an interface for the interaction between user and the system. Secondly we had to connect the user interface with the camera for which we used sockets and python server sent events.

5.1.1 HTML

Hypertext Markup Language (HTML) is the standard markup language for creating web pages and web applications. Web browsers receive HTML documents from a web server or from local storage and render the documents into multimedia web pages. HTML describes the structure of a web page semantically and originally included cues for the appearance of the document.

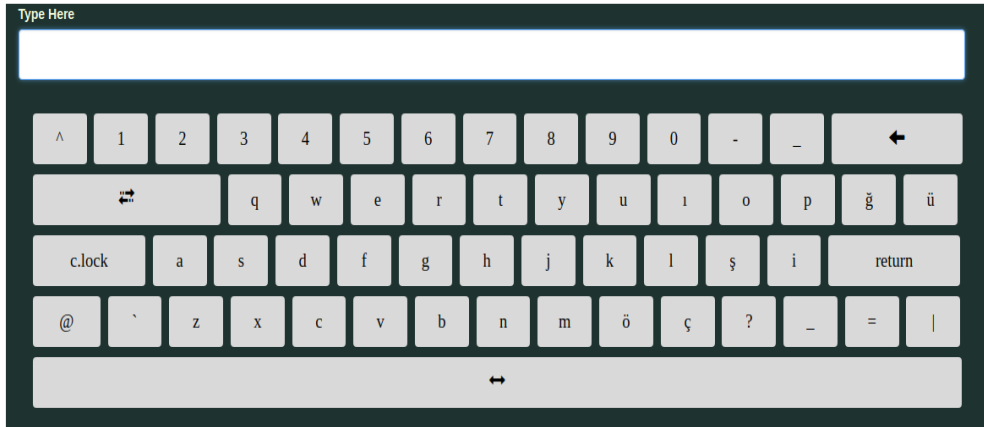


5.1.2 CSS

Cascading Style Sheets (CSS) is a style sheet language used for describing the presentation of a document written in a markup language like HTML. CSS is a cornerstone technology of the World Wide Web, alongside HTML and JavaScript. It was very important for us to make our widgets presentable and cool looking. For this task, CSS proved to be very useful and handy for us. We are now able to change our button's color after clicking using CSS. We have also used CSS for creating designs such as circular edges and colorful buttons.

5.1.3 Javascript

JavaScript, often abbreviated as JS, is a high-level, interpreted programming language. It is a language which is also characterized as dynamic, weakly typed, prototype-based, and multi-paradigm. Alongside HTML and CSS, JavaScript is one of the three core technologies of the World Wide Web. JavaScript enables interactive web pages and thus is an essential part of web applications. The vast majority of websites use it, and all major web browsers have a dedicated JavaScript engine to execute it. JavaScript was very important for our GUI to make it functional. For example, if we make a calculator, then to make it able to compute equations, JavaScript was very important. Or similarly, operations such as closing any widgets.



5.1.4 JQuery

jQuery's syntax is designed to make it easier to navigate a document, select DOM elements, create animations, handle events, and develop Ajax applications. jQuery also provides capabilities for developers to create plug-ins on top of the JavaScript library. We used it to create animations of dynamic clock etc. Some of special properties that we achieved to create- Rotation-We can now rotate our widget by some angle according to our convenience. Dragging of some widget-We are able to drag some widget from one place to another so as to create some space for our work.



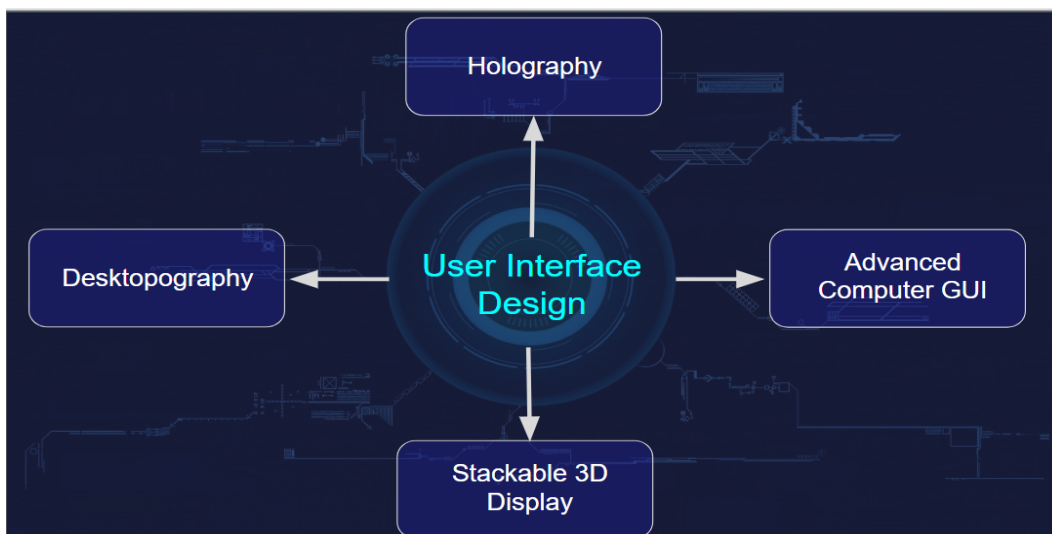
5.1.5 Web-sockets and Python Server sent events

Now we have the camera that gives us output as coordinates of the index finger and the number of fingers touching the projected surface and the GUI which

the user faces to interact with the system. To set up this connection we used UDP server-client system, Flask a web framework which acts as a web server to host our html,css and javascript pages, a web client to display the frontend to the user and server sent events which sends the events to the web client. Initially the UDP server gets the camera data and sends to the UDP client and then the Flask web server hosts the user interface. Now Server sent events sends the event stream to the web client where the ui is displayed. We have added different gestures like four finger touch two finger drag, singer finger touch to click in our GUI. We can also add many other gestures in our GUI without actually using the mouse cursor.

6 Future Work

Desktopography is just a part of the idea that we wish to implement ,i.e the HUMAN COMPUTER INTERFACE .Our's is a long term project and desktopography was just the beginning .In the ongoing year we further wish to create more such devices which fulfill our dream of Virtual Reality.



7 The Team

1. Karan Vaish
2. Shobhit Jagga
3. Naveen Balaji
4. Soumya Ranjan Dash
5. Jay Mundra
6. Prajwal H.G
7. Shivam kumar

8 Mentors

Ashok Kumar Choudhary and Suraj Hanchinal