Contents

1 Introduction

This project aims to project a screen on a see through material and make it interactive with the help of depth sensing using kinect sensor. We are going to project computer screens on see-through displays which gives an experience as if the image is floating in air. The project will include the integration of two technologies, Heads-Up Display(used in cars) and Desktopography. The HUD works on the principle of reflection through smooth polished surfaces when light falls at an angle with the reflecting surface. As for Desktopography, the idea is to emulate interactions using the data by monitoring the closeness of the user's fingers to the screen. This would lead to the generation of a region in which the touch would be detected. If the user's finger goes in this region, the touch would be detected at that point and thus an appropriate action will be performed.

2 Acknowledgement

We are grateful to the Electronics Club and the Science and Technology Council IITK for presenting us with this opportunity to do the project. We are grateful for all the guidance, material and help extended by them. We would like to thank our project mentors, Shubham Kumar and Shivam Singhal for thier guidance.

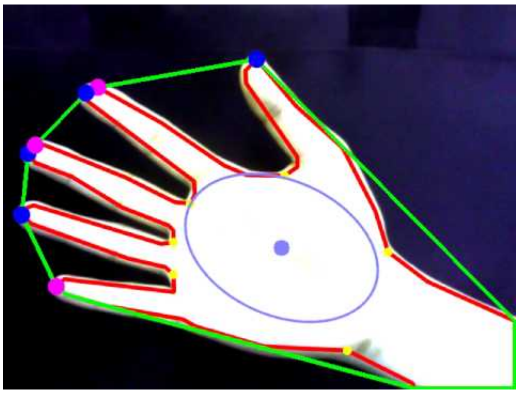
3 Overview

Natural human-computer interaction (N-HCI) is a longstanding important topic in both research and industry due to its wide application and revolution possibility. Keyboards and mice have been major HCI devices for quite a few decades. Finger-based touch-screen HCI technology draws lots of attention and has become one of the fundamental interface components for a flat computer. The recently released Kinect 1 inspires researchers to explore the possibility of device-free HCI by just using the body. In this work, we project our computer screen on a transparent screen, and will operate the screen using a hand detection model and Kinect camera; a See-through display is a new way of HCI. This technology is already used in Head-up displays in cars, and the idea is to integrate this technology with Desktopography to project large screens. First and foremost we obtain RGB video and Depth data using a Kinect sensor after which working on detecting hands in the RGB video has two methods: The first one uses Opencv and mediapipe and the other one uses classical CV methods. By using the Depth Data of the tips of fingers, we find the closeness of the tip with the screen. If this closeness crosses a specific limit value, it is considered as a click. Following this, to display our computer screen on a see-through display, we make use of the working principle of the Heads-Up display, as used in cars to display speedometer and other things on the windshield.

4 Theoretical Background

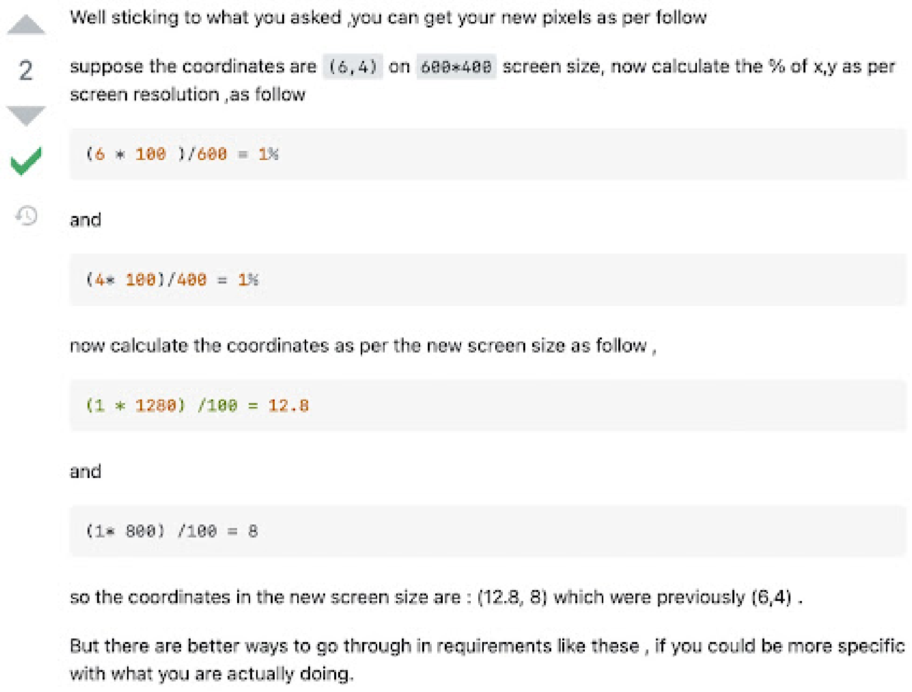
4.1 Hand Detection

There are two methods for hand detection, one uses combination of Opencv and Mediapipe and another one is using classical Opencv. Basically it is the extraction of individual frames from video and to work on it. Mediapipe employs machine learning (ML) to infer 21 3D landmarks of a hand from just a single frame. Whereas current state-of-the-art approaches rely primarily on powerful desktop environments for inference, our method achieves real-time performance on a mobile phone, and even scales to multiple hands.



4.2 Mapping between the Projection and Computer Screen

For mapping between projection and the computer screen, first the coordinates of the point clicked is found out, suppose the coordinates are (6,4), now the screen resolution is calculated by using the screen size. Consider the screen size to be 600*400 in this case, then the resolution will be (6*100)/600=1 and (4*100)/400=1. Using this information, we can now calculate the new coordinates as per the new screen size which is (1*1280)/100=12.8 and (1*800)/100=8. So,the new coordinates in the new screen size are (12.8,8) which were previously (6,4).



Well sticking to what you asked ,you can get your new pixels as per follow

suppose the coordinates are `(6,4)` on `600*400` screen size, now calculate the % of x,y as per screen resolution ,as follow

```
(6 * 100 )/600 = 1%
```

and

```
(4* 100)/400 = 1%
```

now calculate the coordinates as per the new screen size as follow ,

```
(1 * 1280) /100 = 12.8
```

and

```
(1* 800) /100 = 8
```

so the coordinates in the new screen size are : (12.8, 8) which were previously (6,4) .

But there are better ways to go through in requirements like these , if you could be more specific with what you are actually doing.

4.3 Why Intel Real Sense ?

Parallax Issue : A parallax error occurs when you take a picture by looking through the viewfinder of your digital camera , and the picture doesn't look like it did on viewfinder. IntelReal Sense has inbuilt library to manage the parallax effect.

5 Architecture

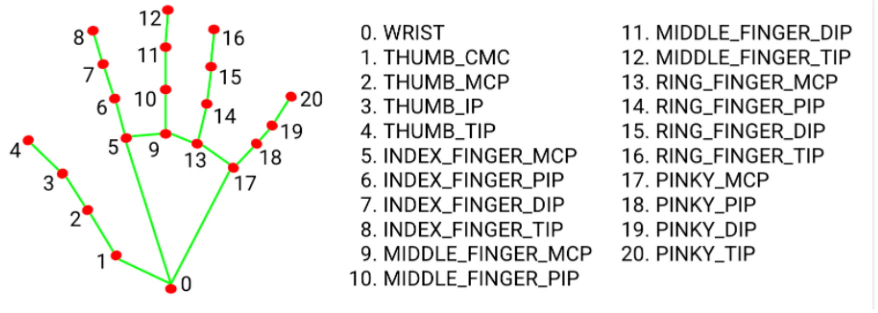
5.1 Python Libraries

5.1.1 Opencv

OpenCV is a cross-platform library using which we can develop real-time computer vision applications. It mainly focuses on image processing, video capture and analysis including features like face detection and object detection.

5.1.2 Mediapipe

Mediapipe is a framework mainly used for building audio, video, or any time series data. With the help of the MediaPipe framework, we can build very impressive pipelines for different media processing functions.We will use it for hand detection.Basically, the MediaPipe uses a single-shot palm detection model and once that is done it performs precise key point localization of 21 3D palm coordinates in the detected hand region.



5.1.3 Pyautogui

PyAutoGUI lets your Python scripts control the mouse and keyboard to automate interactions with other applications. To install with pip, run pip install pyautogui. PyAutoGUI has several features:

- Moving the mouse and clicking in the windows of other applications.
- Sending keystrokes to applications (for example, to fill out forms).

- Take screenshots, give an image (for example, of a button or checkbox), and find it on the screen.
- Locate an application's window, and move, resize, maximize, minimize, or close it (Windows-only, currently).
- Display alert and message boxes.

5.1.4 Libfreenect

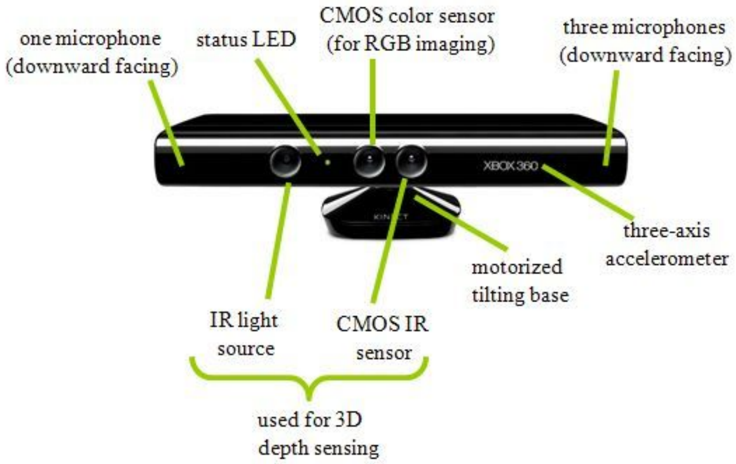
Libfreenect is a userspace driver for the Microsoft Kinect. It runs on Linux, OSX, and Windows and supports RGB and Depth Images, motors, accelerometer .We installed this library to work with RGB and depth images.

5.1.5 Pyrealsense

Pyrealsense is a module that can store the depth scale in the user's home directory. This can later be loaded and used to project depth data.

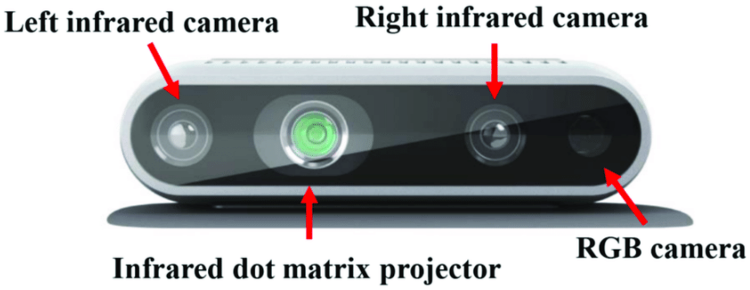
5.2 Kinect Sensor

Kinect is a line of motion sensing input device produced by Microsoft and first released in 2010. The devices generally contain RGB cameras, and infrared projectors and detectors that map depth through either structured light or time of flight calculations, which can, in turn, be used to perform real-time gesture recognition and body skeletal detection, among other capabilities. They also contain microphones that can be used for speech recognition and voice control. The User's hand is recognized with the help of a Kinect Version 2 by comparing its depth. The entire system is portable, i.e., it can be projected on any transparent surface..



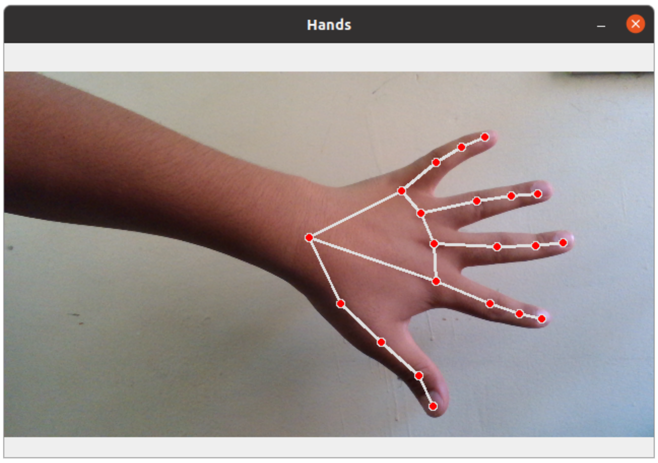
5.3 Intel RealSense

Intel RealSense Technology is a product range of depth and tracking technologies designed to give machines and devices depth perception capabilities.The RealSense product is made of Vision Processors, Depth and Tracking Modules, and Depth Cameras, supported by an open source, cross-platform SDK, simplifying supporting cameras for third party software developers, system integrators, ODMs and OEMs. The Intel RealSense Depth Camera D435 is ideal for capturing stereo depth in a variety of applications that help perceive the world in 3D. The depth data of the user's hand is taken to calculate if it is within the threshold limit. The camera stores the depth data.

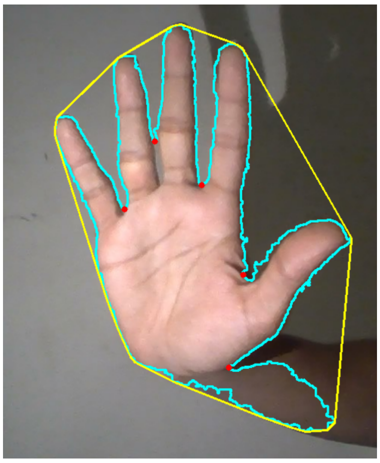


6 Result

We worked on two techniques for hand detection. We used hand detection using the MediaPipe library available on python. It is a high-fidelity hand and finger tracking solution which uses ML to mark different points on the hand and detect touch when a particular set of points coincide. Next, we tried on using OpenCV Python Hand Detection which detects hand from an image by masking it. We are constantly working on masking to get better results.



Hand detection with mediapipe



Hand detection with open CV

7 GitHub Repository

`https://github.com/eclubiitk/desktopography.git`