Animatronic Eye

A
Minor Project Report

Submitted in partial fulfillment of the requirements
for the award of the degree of

Bachelor of Technology
in
Computer Science And Engineering

Submitted by

A. RAJ KUMAR	(17SS1A0504 CSE)
T. VAMSHI	(17SS1A0546 CSE)
V. HARISATHWIK	(17SS1A0550 CSE)
ARYAN KUMAR.N	(17SS1A2905 ME)

Under the guidance of

Ms. Faiza Iram

Department of Computer Science and Engineering
JNTUH COLLEGE OF ENGINEERING, SULTANPUR
Sultanpur (V), Pulkal (M), Sangareddy District, Telangana - 502273.

March 2021.

JNTUH COLLEGE OF ENGINEERING SULTANPUR

Sulthanpur(V),Pulkal(M),Sangareddy-502273 ,Telangana

Department of Computer Science and Engineering

Certificate

This is to certify that the Minor Project report work entitled **"ANIMATRONIC EYE"** is a bonafide work carried out by **A.RAJKUMAR(17SS1A0504 CSE), T.VAMSHI (17SS1A0546 CSE),V.HARISATHWIK(17SS1A0550 CSE),ARYAN KUMAR.N (17SS1A2905 ME)** in partial fulfillment of the requirements of the degree of BACHELOR of TECHNOLOGY in COMPUTER SCIENCE AND ENGINEERING discipline to Jawaharlal Nehru Technological University,Hyderabad during the academic year 2020-21. The results embodied in this report have not been submitted to any other University or Institution for the award of any degree or diploma.

Project Guide	**Head of the Department**
Ms. FAIZA IRAM	**Sri. JOSHI SHRIPAD**
Assistant Professor(C)	**Associate Professor**

PRINCIPAL

Declaration

We hereby declaring that our Minor Project report work is being entitled as **"Animatronic Eye"** is the work done by our teammates A.Rajkumar (17SS1A0504 CSE), T.Vamshi (17SS1A0546 CSE), V.Harisathwik (17SS1A0550 CSE), Aryan Kumar.N (17SS1A2905 ME) and is submitted in the partial fulfillment of the requirements for the award of degree of Bachelor of Technology in Computer Science and Engineering from Jawaharlal Nehru Technological University Hyderabad College of Engineering Sulthanpur. The Results embodied in this report have not been submitted to any other University or Institution for the award of any degree or diploma.

A.RAJKUMAR (17SS1A0504 CSE)
T.VAMSHI (17SS1A0546 CSE)
V.HARISATHWIK (17SS1A0550 CSE)
ARYAN KUMAR.N (17SS1A2905 ME)

Acknowledgement

We wish to take this opportunity to express our deep gratitude to all those who helped, encouraged, motivated, and have extended their co-operation in various ways during out Minor Project Report Work. It is our pleasure to acknowledge the help of all those individuals who were responsible for seeeing the successful completion of Minor Project Report Work.

We express our sincere gratitude **Dr.B.Balu Naik** Principal, JNTUHCES, for his support during the course period.

We express thank **Dr.Venkateshwara Reddy** , Vice Principal, JNTUHCES, for his kind help and co-operation.

We are thankful to **Sri.Joshi Shripad**, Associate Professor and Head of the Department of Computer Science and Engineering of JNTUHCES for his effective suggestion during the course period.

We are thankful to our for their support and effective suggestion during the course period.

We very much thankful to **Sri.S.Ravinder**, Assistant Professor and **Mrs.B.Sangeetha**, Assistant Professor and **Ms.Faiza Iram**, Assistant Professor(C) for their support and guidance through out the course period.

Finally, we express our gratitude with great admiration and respect to our teaching and co-operation staff and non-teaching staff for their moral support and encourgement throught the course.

A.RAJKUMAR (17SS1A0504 CSE)
T.VAMSHI (17SS1A0546 CSE)
V.HARISATHWIK (17SS1A0550 CSE)
ARYAN KUMAR.N (17SS1A2905 ME)

Abstract

Animatronics is the multi-disciplinary field which integrates puppetry, anatomy and mechatronics. It refers to mechatronic puppets. Puppet body, muscles, nerves are made using elastic netting composed of styrene beads. Eyeballs are embedded with cameras to capture the movement and tracking of the objects. Every part of the Animatronic puppet body is implemented by studying the actual behaviour based on the characteristics of holding strength, the flexibility of movement of the parts, depth-sensing of the surroundings, colour intensity, night vision capability, moving capacity of the real-world animals.Implementation of Animatronic Eye is based upon the motors that are attached to the holdings of the vision cameras which are embedded in a plastic eyeball to rotate their vision according to the sensing. When the object moves along the left, the holdings are rotated along the left and moved to right when the object is moved to the right. Similarly, the capturing from both the two camera's embedded eyeballs are merged into one effective final capturing to clearly distinguish the object from its surroundings. From the two cameras, one camera can be used to detect the depth of the object from its surroundings and the second is used to capture the movement. The Higher form of vision cameras is used to implement animatronic puppets with the same capabilities as the real animal. View from the capturing is used for further processing in generating the visual effects for the scene.

List of Figures

Contents

Chapter 1

Introduction

Animatronics refers to the use of robotic devices to emulate a human or an animal or bring lifelike characteristics to an otherwise inanimate object. Animatronic creations include animals (including dinosaurs), plants and even mythical creatures. A robot designed to be a convincing imitation of a human is more specifically labelled as an android. Modern animatronics have found widespread applications in movie special effects and theme parks and have, since their inception, been primarily used as a spectacle of amusement.

Animatronics is a multidisciplinary field which integrates anatomy, robots, mechatronics, and puppetry resulting in lifelike animation. Animatronic figures are often powered by pneumatics, hydraulics, or by electrical means, and can be implemented using both computer control and human control, including teleoperation.

Motion actuators are often used to imitate muscle movements and create realistic motions in limbs. Figures are covered with body shells and flexible skins made of hard and soft plastic materials and finished with details like colours, hair and feathers and other components to make the figure more realistic.

1.1 Objective:

Animatronics is a combination of animation and electronics. What exactly is an animatronic? Basically, an animatronic is a mechanized puppet. It may be pre programmed or remotely controlled. The animatronic may only perform a limited range of movements or it may be incredibly versatile. Animatronics gives a special spirit to the imaginary creatures to make them alive. A virtual creature was implicitly formed on the basis of science and technologies. literature survey.

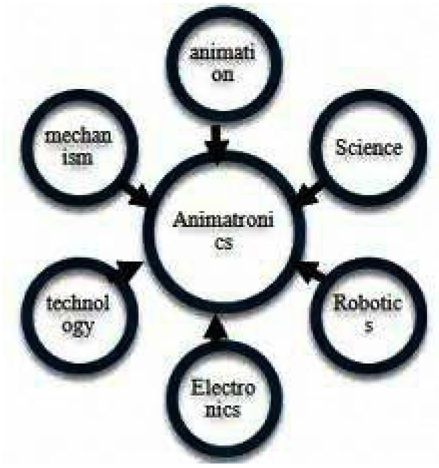


Figure 1.1: Animatronics

Chapter 2

History

Animatronics was developed by Walt Disney in the early sixties. The first use of audio-animatronics was for Walt Disney's enchanted tiki room in Disneyland, which opened in June 1963. Essentially, an animatronic puppet is a figure that is animated by means of electromechanical devices. Early examples were found at the 1964 world fair in the new york hall of presidents and Disneyland. Body language and facial motions were matched to perfection with the recorded speech.

Animatronics was a popular way of entertainment that had proven itself in the theme parks and cinematography industry. An extension of the engineering challenge is to explore the effectiveness of the project's capability to display human emotions and to design the physical mechanisms that display realistic human facial movements. The objective of this effort was to design and build an animatronic robot ssu-1 (savannah state university-1). The issue-1 will be controlled by a pre-programmed embedded microcontroller and will create human-like motions for entertainment purposes.

Figure 2.1: Sparko the Robot Dog (First Modern Animatronics 1940's)

Animatronics gives a special spirit to the imaginary creatures to make them alive. A virtual creature was implicitly formed on the basis of science and technologies. This technology was developed by Walt Disney in the year 1960. It is the creation of machines which seems so animated. The implementation of this system can be made by using a computer or manual control. Three more kinds of powers can be given to the animated figures such as pneumatic, hydraulic, or by electronic means. The specified controls and the programs are done manually by Humans.

Chapter 3

Human Eye Anatomy

3.1 Internal Structure

The human eye is a paired sense organ that reacts to light and allows vision. Rod and cone cells in the retina are photoreceptor cells which are able to detect visible light and convey this information to the brain. Eyes signal information which is used by the brain to elicit the perception of color, shape, depth, movement, and other features. The eye is part of the sensory nervous system.

Similar to the eyes of other mammals, the human eye's non-image-forming photosensitive ganglion cells in the retina receive light signals which affect adjustment of the size of the pupil, regulation and suppression of the hormone melatonin, and enterainment of the circadian rhythm.

Figure 3.1: Human Eye(outer part)

3.2 Structure

Humans have two eyes, situated on the left and the right of the face. The eyes sit in bony cavities called the orbits, in the skull. There are six extraocular muscles that control eye movements. The front visible part of the eye is made up of the whitish sclera, a coloured iris, and the pupil. A thin layer called the conjunctiva sits on top of this. The front part is also called the anterior segment of the eye.

The eye is not shaped like a perfect sphere, rather it is a fused two-piece unit, composed of an anterior (front) segment and the posterior (back) segment. The anterior segment is made up of the cornea, iris and lens. The cornea is transparent and more curved, and is linked to the larger posterior segment, composed of the vitreous, retina, choroid and the outer white shell called the sclera. The cornea is typically about 11.5 mm (0.3 In) in diameter, and 0.5 mm (500 m) in thickness near its center. The posterior chamber constitutes the remaining five-sixths; its diameter is typically about 24 mm. The cornea and sclera are connected by an area termed the limbus. The iris is the pigmented circular structure concentrically surrounding the center of the eye, the pupil, which appears to be black. The size of the pupil, which controls the amount of light entering the eye, is adjusted by the iris' dilator and sphincter muscles.

3.3 Formation of image

Light energy enters the eye through the cornea, through the pupil and then through the lens. The lens shape is changed for near focus (accommodation) and is controlled by the ciliary muscle. Photons of light falling on the light-sensitive cells of the retina (photoreceptor cones and rods) are converted into electrical signals that are transmitted to the brain by the optic nerve and interpreted as sight and vision.

3.4 Movement of Eye

The extraocular muscles execute eye movements and are innervated by three cranial nerves. The muscles are attached to the sclera of the eye at one end and are anchored to the bony orbit of the eye at their opposite ends. Contraction of the muscles produces movement of the eyes within the orbit. The cranial lower motor neurons innervate these muscles and thereby control their contractions.

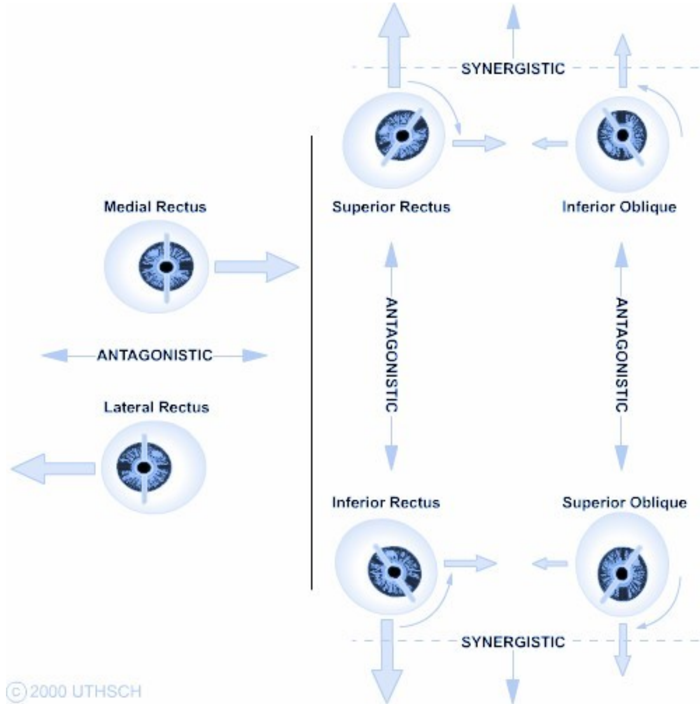


Figure 3.2: Extrocules

3.4.1 The Extraocular Muscles

For each eye, six muscles work together to control eye position and movement. Two extraocular muscles, the medial rectus and lateral rectus, work together to control horizontal eye movements. Contraction of the medial rectus pulls the eye towards the nose (adduction or medial movement). Contraction of the lateral rectus pulls the eye away from the nose (abduction or lateral movement). The actions of these two muscles are antagonistic: one muscle must relax while the other contracts to execute horizontal eye movements. Four other extraocular muscles working together control vertical eye movements and eye rotation around the mid-orbital axis.

The extraocular muscles of the right eye and their actions. Antagonistic actions pull the eye in opposite directions whereas synergistic actions pull the eye in the same direction.

1 . **Superior Rectus** produces

- eye elevation
- minor movements: medial rotation and adduction

2 . **Superior Oblique** produces

- eye depression
- other movements: medial rotation and abduction

3 . **Inferior Rectus** produces

- eye depression
- minor movements: lateral rotation and adduction

4 . **Inferior Oblique** produces

- eye elevation
- other movements: lateral rotation and abduction

To direct the eye upward or downward, two muscles contract synergistically as the two antagonist muscles relax. For example, to elevate the eye while looking straight ahead, the superior rectus and inferior oblique contract together as the inferior rectus and superior oblique relax. The superior rectus and inferior oblique muscles working together pull the eye upward without rotating the eye. To depress the eye while looking straight ahead, the inferior rectus and superior oblique contract together as the superior rectus and inferior oblique relax. The inferior rectus and superior oblique working together pull the eye downward without rotating the eye.

3.5 Formation of Image:

From the outer world light rays from different objects enter through cornea - iris - pupil. And finally fall on the retina. Retina is the place where photoreceptors are located. If you think of the eye as a camera, the retina would be the film. The retina also contains the nerves that tell the brain what the photoreceptors are **seeing**.

There are two types of photoreceptors involved in sight: rods and cones. Rods work at very low levels of light. We use these for night vision because only a few bits of

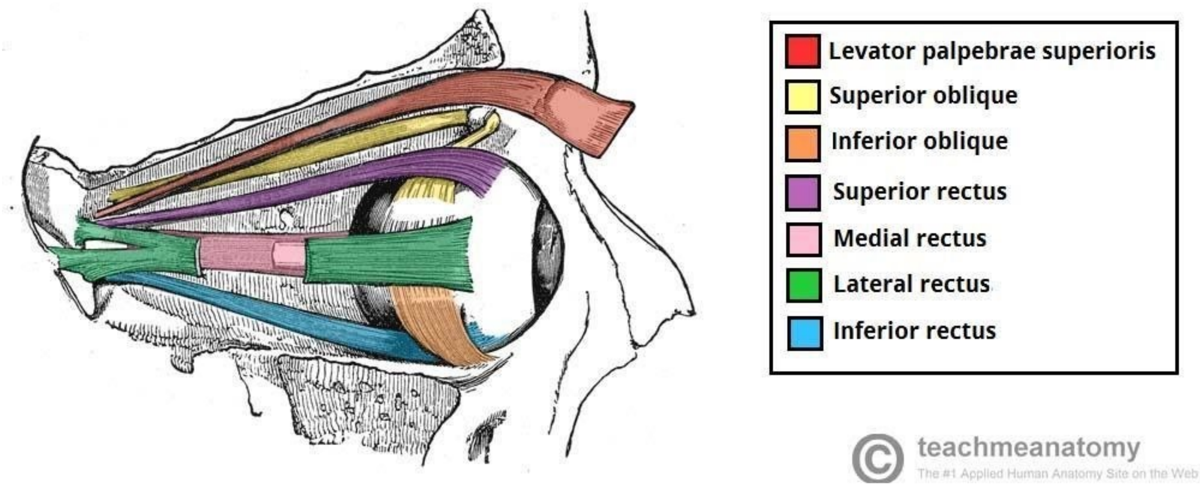


Figure 3.3: Muscle Of Eye

light (photons) can activate a rod. Rods don't help with color vision, which is why at night, we see everything in a gray scale. The human eye has over 100 million rod cells.

Cones require a lot more light and they are used to see color. We have three types of cones: blue, green, and red. The human eye only has about 6 million cones. Many of these are packed into the fovea, a small pit in the back of the eye that helps with the sharpness or detail of images.

Other animals have different numbers of each cell type. Animals that have to see in the dark have many more rods than humans have. Take a close look at the photoreceptors in the drawings above and below. The disks in the outer segments (to the right) are where photoreceptor proteins are held and light is absorbed. Rods have a protein called rhodopsin and cones have photopsins. But wait...these are stuck in the back of the retina. That means that the light is absorbed closer to the outside of the eye.

Light moves through the eye and is absorbed by rods and cones at the back of the eye. The "backwards" organization of rods and cones is helpful for a few different reasons.First of all, the discs containing rhodopsin or photopsin are constantly recycled to keep your visual system healthy. By having the discs right next to the epithelial cells (retinal pigmented epithelium: RPE) at the back of the eye, parts of the old discs can be carried away by cells in the RPE.

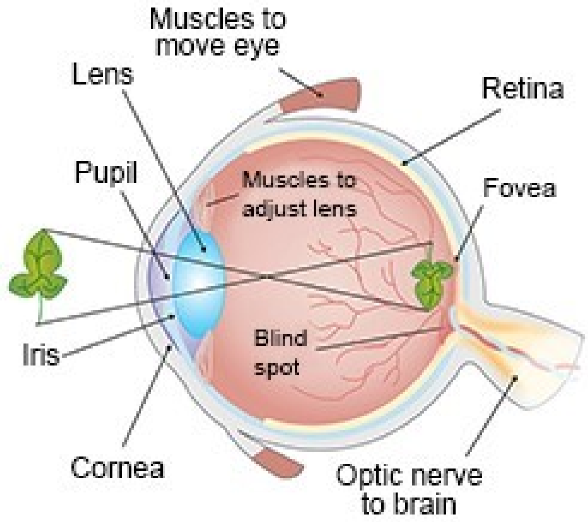


Figure 3.4: Internal structure of Eye

We have three types of cones. If you look at the graph below, you can see each cone is able to detect a range of colors. Even though each cone is most sensitive to a specific color of light (where the line peaks), they also can detect other colors (shown by the stretch of each curve).

Since the three types of cones are commonly labeled by the color at which they are most sensitive (blue, green and red) you might think other colors are not possible. But it is the overlap of the cones and how the brain integrates the signals sent from them that allows us to see millions of colors. For example, the color yellow results from green and red cones being stimulated while the blue cones have no stimulation.

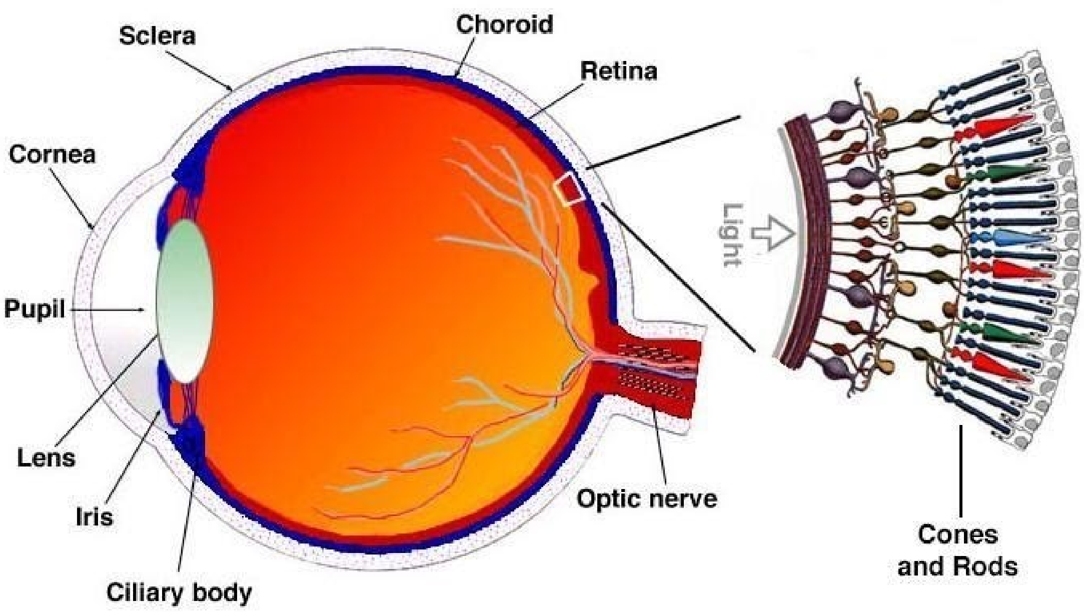


Figure 3.5: Internal Structure Of Eye

Chapter 4

Animatronic Eyes

Many types of robotic or animatronic eyes have been created with a number of actuating mechanisms. To actuate or rotate the eye, a drive or actuating mechanism is provided adjacent to the eye such as in the animatronic figure's head that includes external motors, hydraulic cylinders, gears, belts, pulleys, and other mechanical drive components to drive or move a spherical or eye-shaped orb. As a result, the eye assemblies require a large amount of external space for included moving parts, and space requirements have become a major issue as the eye itself is often dwarfed by the mechanical equipment used to move the eye up and down (e.g., tilt or pitch) and side-to-side (or yaw). The mechanical drive equipment has moving components external to and attached to the eye that needs mounting fixtures and space to freely move. In some cases, existing animatronic eye designs are somewhat unreliable and require significant amounts of maintenance or periodic replacement due, in part, to wear caused by friction of the moving parts including the eye within a socket device. To retrofit an eye assembly, the electromechanical, pneumatic, hydraulic, or other drive or eye-movement systems typically have been completely removed and replaced.

In some cases, animatronic eyes cannot perform at the speeds needed to simulate human eye movement. Movements may also differ from smooth human-like action when the drive has discontinuous or step-like movements, which decreases the realism of the eye. Additionally, many animatronic eye assemblies use a closed-loop servo control including a need for a position or other feedback signal such as from optical, magnetic, potentiometer or other position sensing mechanisms. Further, the eye or eyeball's outer surfaces may rub against the seat or socket walls since it is difficult to provide relatively

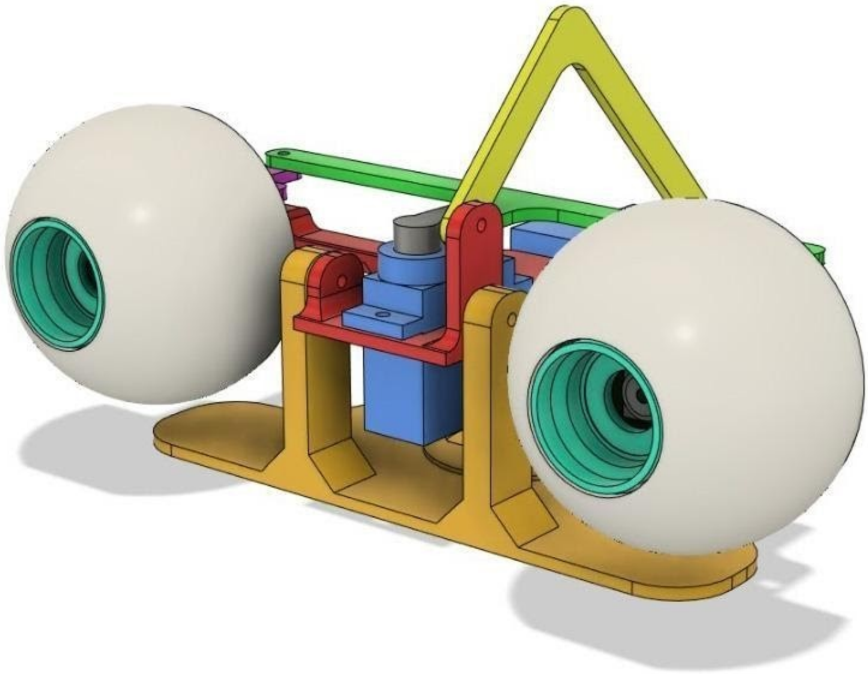


Figure 4.1: Animatronic Eye 3-D design

frictionless support for a rotating sphere or orb, which may further slow its movement, cause wear on painted portions of the eyeball, or even prevent smooth pitch and yaw movements. animatronic eye 3-D design.

More recently, there has been a demand for video capability such as to assist in teleoperation of the animatronics or to provide vision-based interactivity (e.g., track a location of a person or other moving object relative to the animatronic figure and then operate the animatronic figure in response such as by moving the eyes or the whole head).

Some animatronic eye assemblies have been provided with video functionality, with some implementations positioning a tiny video camera within the eyeball itself to move with the eyeball and with it lens at or providing the lens and/or pupil of the eye. However, this creates other problems because the camera power and signal lines may experience wear or be pinched by the movement of the eyeball or interfere with rotation of the eyeball as the movement of the eyeball has to move or drag the cords that extend out the back wall of the eyeball.

Hence, there remains a need for improved designs for animatronic or robotic eye assemblies that better simulate the appearance and movements of the human eye or an animal's eye. Such designs may have a smaller form factor (or use less space for a drive or movement mechanisms) when compared with existing systems, may be designed to better control maintenance demands, and may, at least in some cases, provide video capability.

Chapter 5

Hardware Requirements

5.1 Raspberry pi 4

Raspberry Pi is a series of small single-board computers developed in the United Kingdom by the Raspberry Pi Foundation in association with Broadcom.The Main Aim of the foundation is to promote teaching basic computer science in schools and developing countries. Later, the raspberry pi is used in target markets such as Robotics and now widely in use for weather monitoring for it's low cost and design.

The Raspberry Pi Foundation provides Raspberry Pi OS (formerly called Raspbian), a Debian-based (32-bit) Linux distribution for download, as well as third-party Ubuntu, Windows 10 IoT Core, and specialized distributions for Kodi media centre. It promotes Python and Scratch as the main programming languages. It is the best replacement for desktops with a smaller size. The Size of Raspberry Pi is as of a credit card.

5.2 Specifications of Raspberry Pi:

Raspberry Pi 4 Model B was released in June 2019 with a 1.5 GHz 64-bit quad core ARM Cortex-A72 processor, on-board 802.11ac Wi-Fi, Bluetooth 5, gigabit Ethernet , two USB 2.0 ports, two USB 3.0 ports, and dual-monitor support via a pair of micro HDMI ports for up to 4K resolution. The Raspberry Pi 4 is also powered via a USB-C port, enabling additional power to be provided to downstream peripherals, when used with an appropriate PSU. The initial Raspberry Pi 4 board has a design flaw where third-

Figure 5.1: Raspberry Pi 4 Model B

party e-marked USB cables, such as those used on Apple MacBooks, incorrectly identify it and refuse to provide power. Tom's Hardware tested 14 different cables and found that 11 of them turned on and powered the Pi without issue. The design flaw was fixed in revision 1.2 of the board, released in late 2019.

5.3 Performance :

The Raspberry Pi 4 unlike previous models, which all used a custom interrupt controller poorly suited for virtualisation, the interrupt controller on this SoC is compatible with the ARM Generic Interrupt Controller (GIC) architecture 2.0, providing hardware support for interrupt distribution when using ARM virtualisation capabilities.The Raspberry Pi 4, with a quad-core ARM Cortex-A72 processor, is described as having three times the performance of a Raspberry Pi 3.

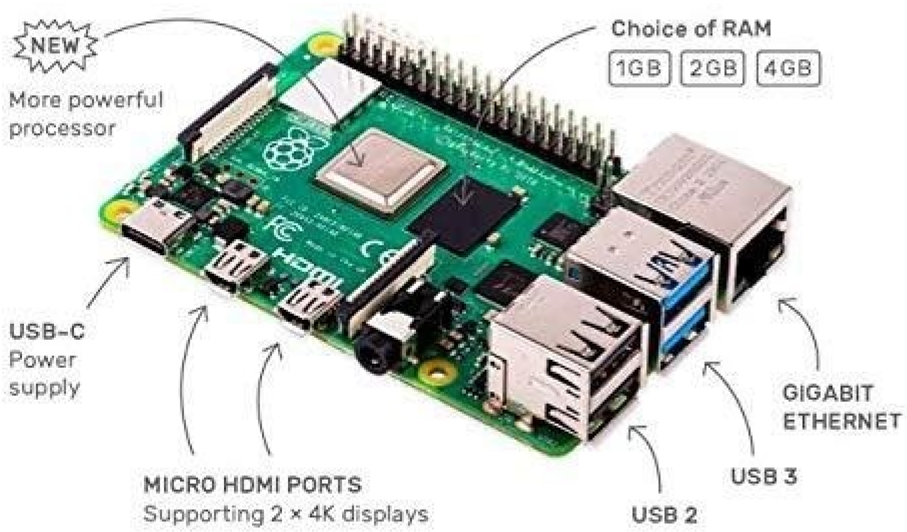


Figure 5.2: Specifications of Raspberry Pi 4

5.4 External Modules

Raspberry Pi also supports external modules that are configurable with GPIO pins for much more functionalities. Few of the modules for Pi are GSM modules, Camera modules, IR modules, Display modules, motors and sensors like Temperature sensor , Moisture sensor , Barometric Sensor etc same as that are supported for Arduino.The Raspberry Pi is one of the best-selling British computers. As of December 2019, more than thirty million boards have been sold. Most Pis are made in a Sony factory in Pencoed, Wales, while others are made in China and Japan.

5.5 Setup Installation

Raspberry Pi OS is the recommended OS for normal use on a Raspberry Pi. Raspberry Pi OS is a free operating system based on Debian, optimised for the Raspberry Pi hardware. Raspberry Pi OS comes with over 35,000 packages: precompiled software bundled in a nice format for easy installation on your Raspberry Pi.

1. Download the latest version of Raspberry Pi Imager and install it.
2. To use Raspberry Pi Imager, you can install it from a terminal using “sudo apt install rpi-imager”.

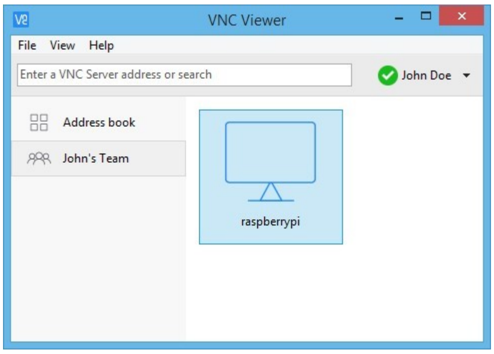


Figure 5.3: VNC window connecting to pi

3. Connect an SD card reader with the SD card inside.
4. Open Raspberry Pi Imager and choose the required OS from the list presented.
5. Choose the SD card you wish and click 'WRITE' to begin writing data to the SD card.

5.6 Remote Access of Pi :

VNC (Virtual Network Computing) is a graphical desktop sharing system that allows you to remotely control the desktop interface of one computer (running VNC server) from another computer or mobile device (running VNC Viewer). VNC viewer transmits the keyboard , mouse , touch events to VNC Server, and receives updates to the screen in return. Enabling VNC Server and establishing the connection . On your Raspberry Pi, boot into the graphical desktop.

. Menu - Preferences - Raspberry Pi Configuration - Interfaces.

. Ensure VNC is Enabled.

. On your Raspberry Pi (using a terminal window or via SSH) run ifconfig to discover your private IP address.

. Enter Raspberry Pi's private IP address and authenticate using Raspberry Pi Credentials.

If you're connecting from the compatible VNC Viewer app from RealVNC, enter the username and password by default, these credentials are pi and raspberry.
To destroy a virtual desktop, run the following command: vncserver -kill :¡ip-address¿.

Figure 5.4: Autodesk

5.7 Fusion 360 :

Fusion 360 is a product of Autodesk, Inc. that offers services for the architecture, engineering, construction, manufacturing, media, education, and entertainment industries. Fusion 360 helps students and educators prepare for the future of design. It's the first 3D CAD, CAM, and CAE tool of its kind, connecting your entire product development process into one cloud-based platform.
Fusion 360 features include :

- Standard design and 3D modeling tools.
- 2 and 3-axis milling, adaptive clearing, turning.
- Water jet, laser cutter, and plasma cutter.
- 3D printing , 2D drawings.
- PCB design with 2 sch. sheets, 2 signal layers, and 80cm2 board area.
- Supported export file types: *.f3z,*.f3d,*.fbx,*.iam,*.ipt,*.obj,*.skp,*.smt,*.step,*.stl,*.stp

Chapter 6

Design And Assembly:

6.1 Design

We'll be using FUSION 360 software for modelling 3d models for the project. There majorly 6 components, the following components are: 1.

6.2 3D Printing

3D printing, or additive manufacturing, is the construction of a three-dimensional object from a CAD model or a digital 3 D model. The term "3D printing" can refer to a variety of processes in which material is deposited, joined or solidified under computer control to create a three-dimensional object, with the material being added together (such as liquids or powder grains being fused together), typically layer by layer.

In the 1980s, 3D printing techniques were considered suitable only for the production of functional or aesthetic prototypes and a more appropriate term for it at the time was rapid prototyping. As of 2019, the precision, repeatability, and material range o f 3D printing have increased to the point that some 3D printing processes are considered viable as an industrial-production technology, whereby t he term additive manufacturing can be used synonymously with 3D printing. One of the key advantages of 3D printing is the ability to produce very complex shapes or geometries that would be otherwise impossible to construct by hand, including hollow parts or parts with internal truss structures to reduce weight. Fused deposition modelling or FDM, is the most common 3D printing process in use as of 2020.

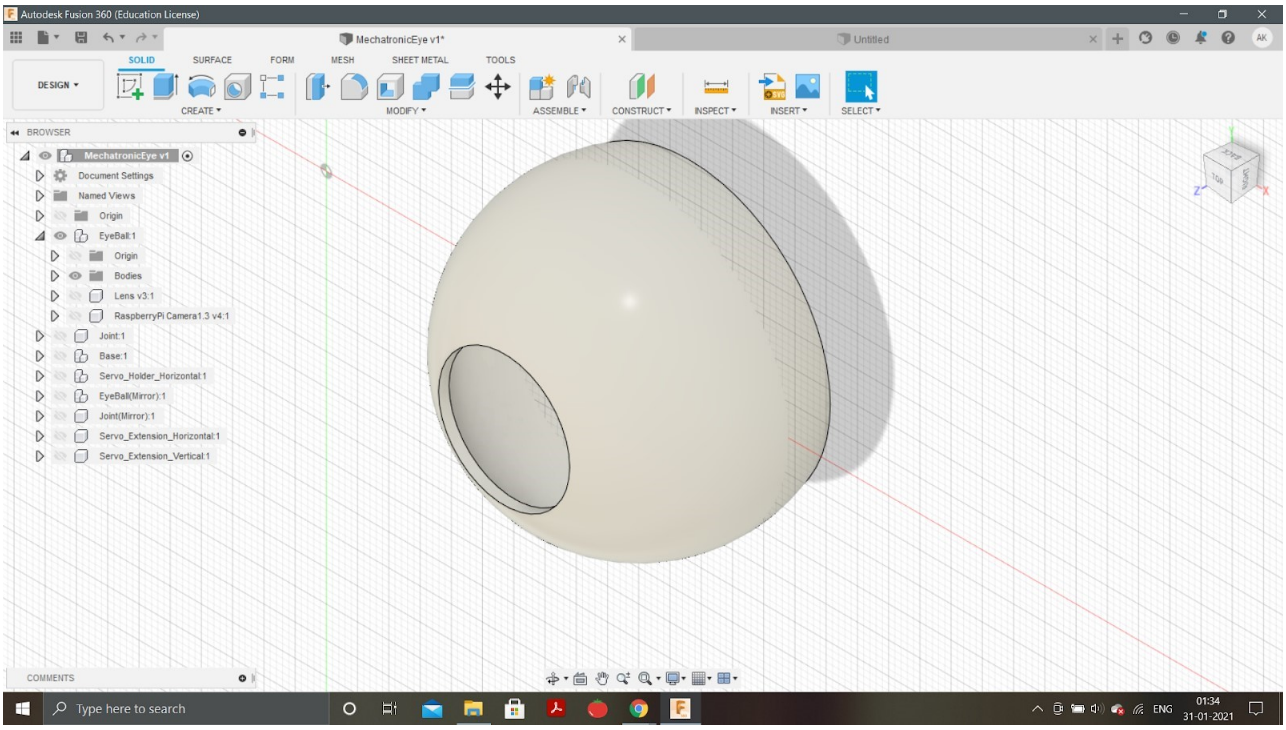


Figure 6.1: EYE

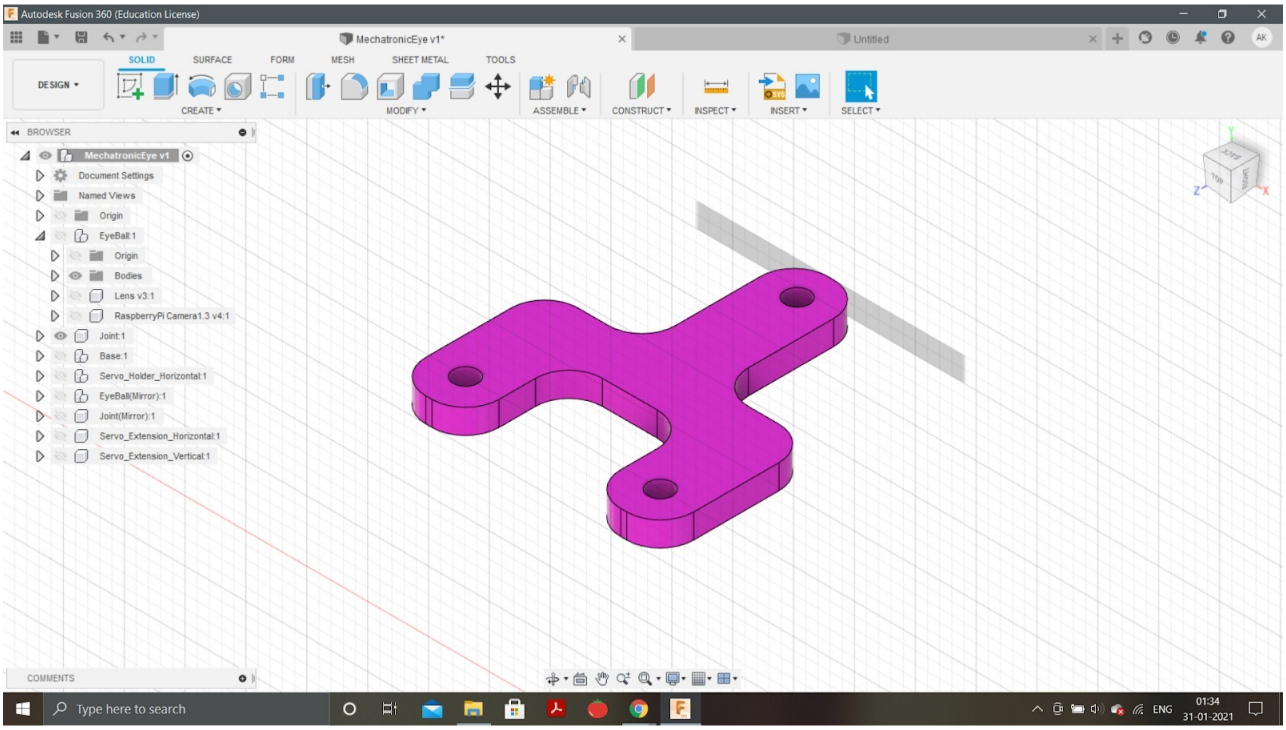


Figure 6.2: Joint

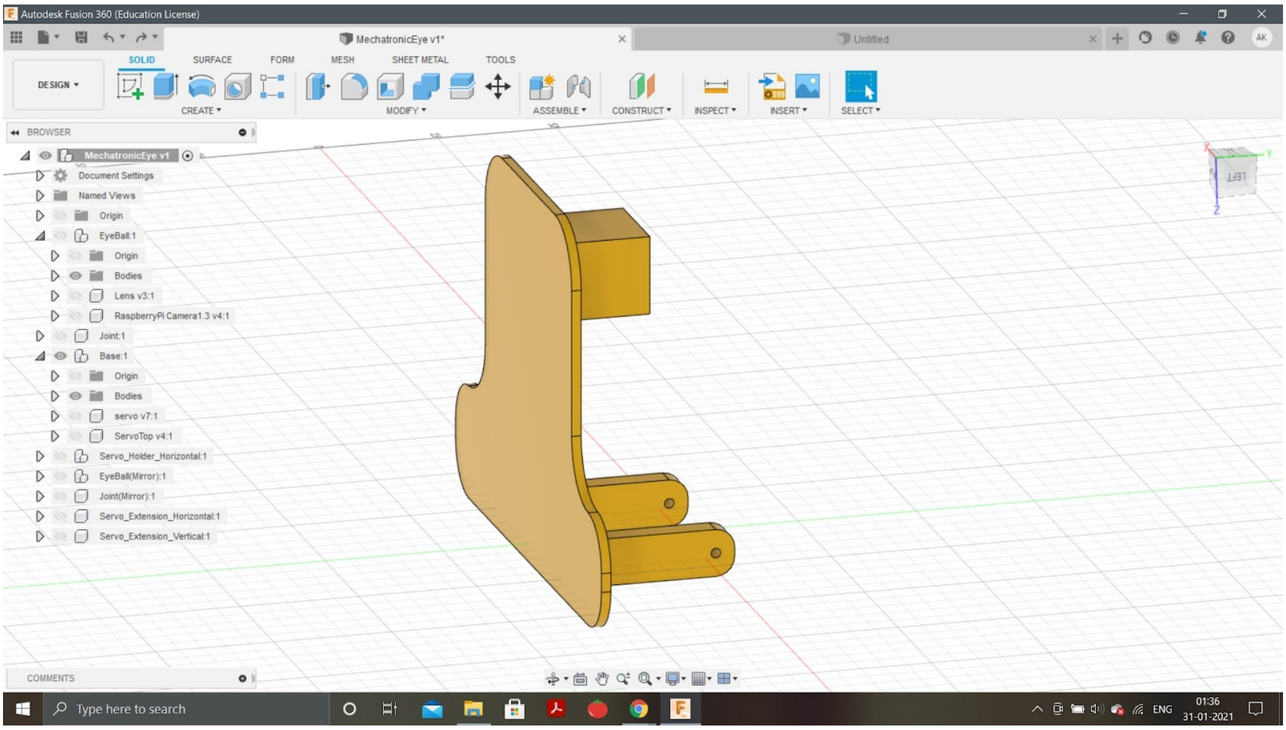


Figure 6.3: Base

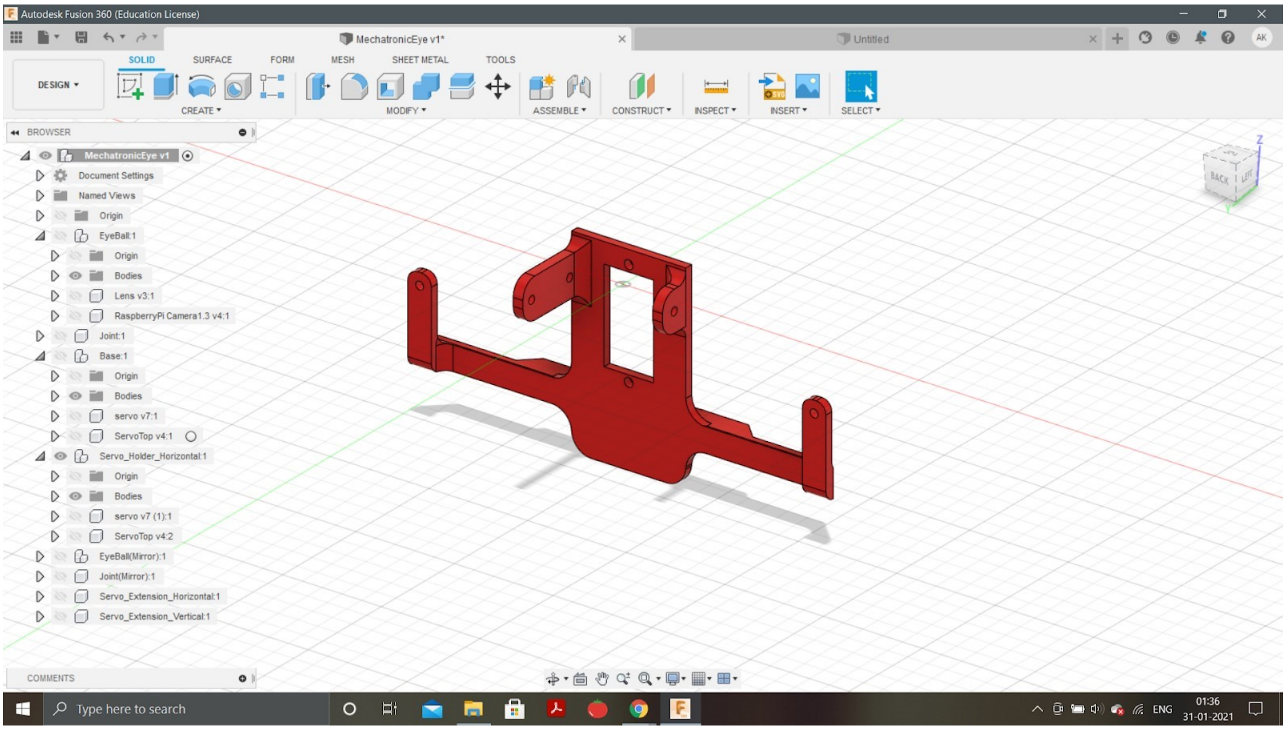


Figure 6.4: Base

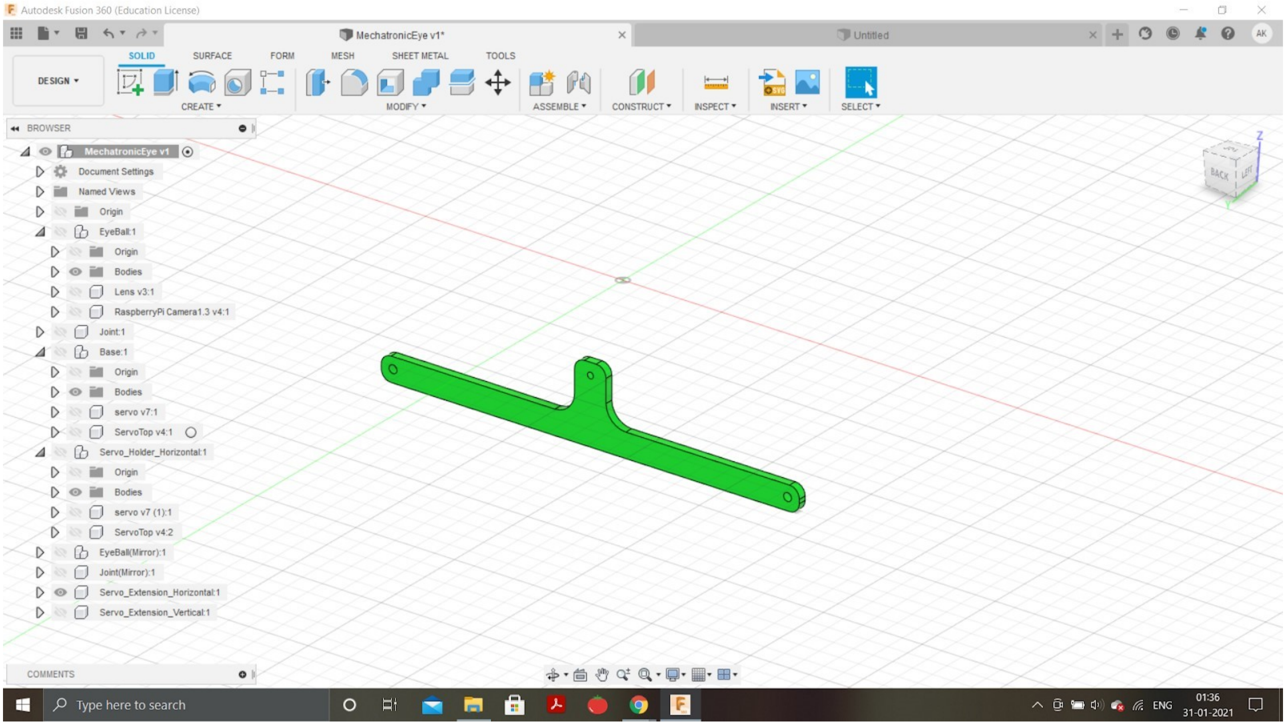


Figure 6.5: Servo Extension Horizontal

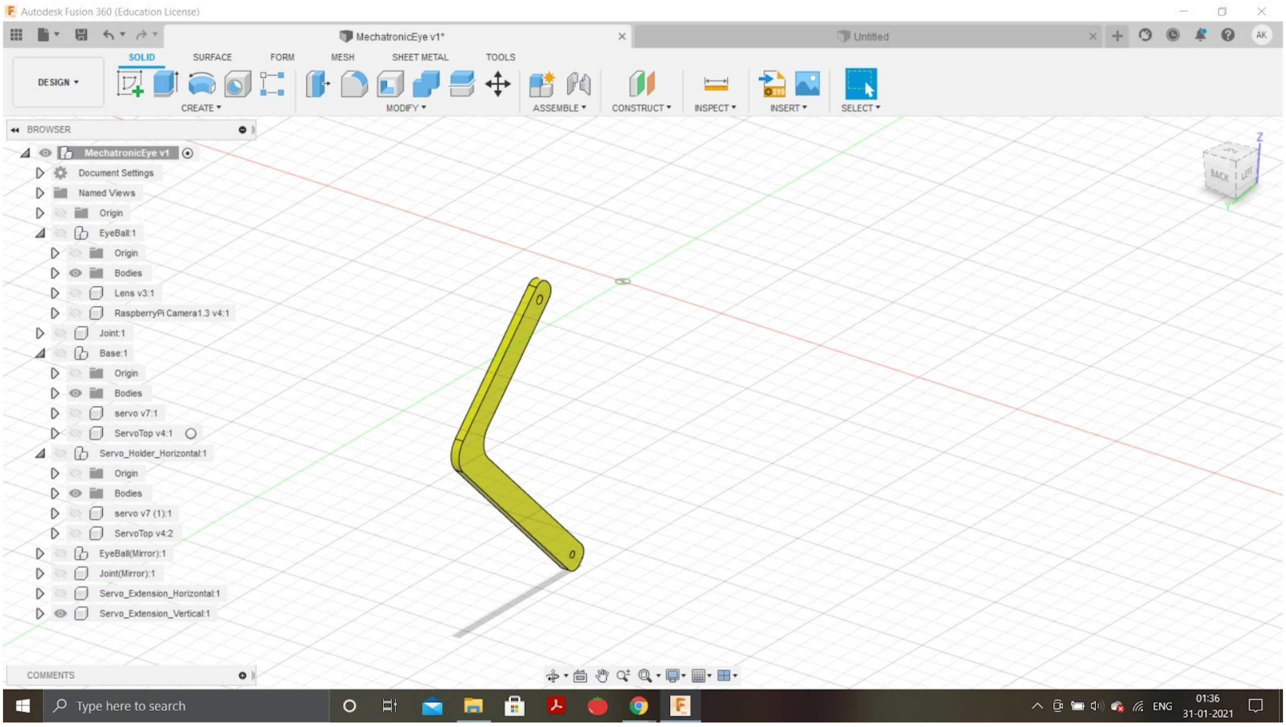


Figure 6.6: Vertical

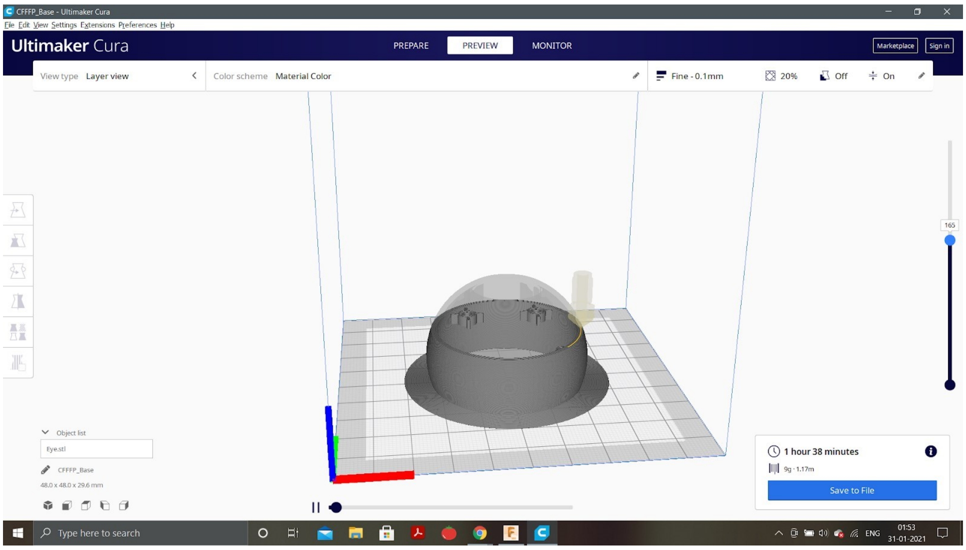


Figure 6.7: CURA SOFTWARE user interface with the eye model preview

6.3 CURA Software

Ultimakers Cura works by slicing t he user's model file into layers and generating a printer-specific g-code. Once finished, the g-code can b e sent to the printer for the manufacture of the physical object. The open-source software, compatible with most desktop 3D printers, can work with files in the most common 3D formats such as STL, OBJ, X3D, 3MF as well as image file formats such as BMP, GIF, JPG and PNG.

6.4 3d printer settings in CURA:

1. Temperature nozzle = 200-degree centigrade
2. Infill Density = 20
3. Infill Pattern = Grid
4. Filament = PLA
5. Filament diameter = 1.75mm
6. Speed = 60 mm/s
7. Bed Size = 100 - 100 mm

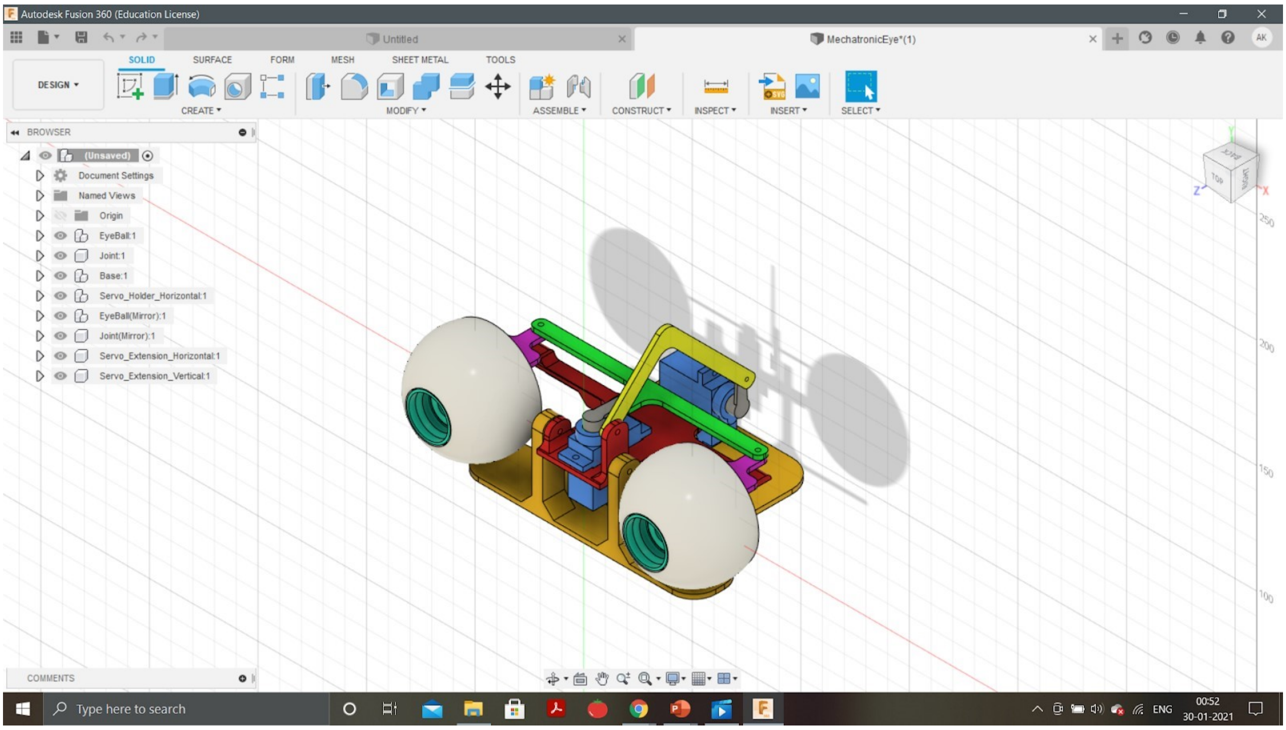


Figure 6.8: Animatronic Eye

6.5 Assembling of the Parts

After 3d printing a ll the components, the c omponents are assembled using M2 bolts and servo motors MG90s. Therefore after assembly, the ANIMATRONIC EYES looks like the following picture.

Chapter 7

Object Tracking

7.1 Introduction

Tracking can be characterized as the problem of assessing the trajectory of an object in the picture or image plane as it moves around a scene. The requirement for high power PCs, the accessibility of high quality and modest camcorders, and the expanding requirement for automated feature analysis has produced a lot of interest for object tracking algorithms. There are three key steps in feature analysis: recognition of target moving objects, tracking of such objects from frame to frame, and analysis of object tracks to perceive their conduct. In its least complex form, tracking can be characterized as the issue of evaluating trajectory of an object in the image plane as it moves around a scene. The main goal of this investigation is to track the real time moving objects in different video frames with the assistance of a proposed algorithm. Median filtering is a non-linear operation which is utilized as a part of image processing to decrease noise. A median filter is more viable than convolution when the objective is at the same time to diminish and save edges.

A variety of issues of current interest in computer vision require the capacity to track moving objects in live streaming for purposes such as inspection, video conferencing, robot navigation, and so on. The difficulties that drive a great part of the exploration in this field are the colossal information data transfer capacity inferred by high resolution frames at high frame rates, and the yearning for real-time intuitive execution. Various innovative routines have been proposed. Nonetheless, the vast majority of these routines use complex models, for example, edges, snakes, splines, formats or computationally expensive Eigen image or condensation algorithms. Despite the fact that these methodologies are expansive in their capacities offering reliable object recognition. In addition to tracking, they are so far not able to run on full video resolution images at high frame rates.

Object tracking means estimating the state of the target object present in the scene from previous information. On a high level of abstraction, there are mainly two levels of object tracking.
- Single Object Tracking(SOT).
- Multiple Object Tracking(MOT).

Proposed method of object tracking is going to use color as it's basis for tracking objects.

7.2 Why Color?

Color has been generally utilized as a real-time tracking framework. It offers a few noteworthy points of interest over geometric signs such as computational simplicity, robustness under partial occlusion, rotation, scale and resolution changes. In the tracking framework, the color blobs are being tracked. The idea of blobs as a representation for image characteristics has a long history in computer vision and has various numerical definitions. It might be a reduced set of pixels that impart a visual property that is definitely not imparted by the surrounding pixels. Body movement analysis is an imperative innovation which combines modern bio mechanics with Computer vision. It is broadly utilized as a part of intelligent control, human machine interaction, movement analysis and different fields. Presently, systems utilized as a part of moving object detection are chiefly the frame subtraction technique, the background subtraction.

7.3 Methodology

A few universally useful algorithms or approaches have been developed for object tracking. Since there is no general answer for the object tracking issue, these systems regularly must be consolidated with domain information so as to adequately tackle an object tracking issue for an issue space. Hence object tracking necessities to be approached from a wide assortment of points of view.

As we have now seen that amid the tracking of the object light brightening goes about as noise. Noise ought to be filtered out through processing, additionally we require that the time needed for the processing of the image or frame ought to be as low as could reasonably be expected, and also we have to see that the movement recognition and tracking of the object ought to be appropriate, in light of the fact that if there is no legitimate movement detection we will not be able to detect and track the target object.

We are bound to track the objects using color feature and motion. Different algorithms or methods have been developed for detecting and tracking objects using color features and motion. A problem inside an object tracking exploration is the quest for a powerful measure of tracking quality. Diverse systems for tracking exist using distinctive attributes e.g., shape, surface, or color, and so on. These strategies perform diversely relying upon the application and are frequently looked at just subjectively.

7.4 Implementation

We are going to use python and especially the opencv module(a computer vision library) to do this object tracking.

7.4.1 Computer Vision (cv2 module in python)

OpenCV (Open Source Computer Vision Library) is an open source computer vision and machine learning software library. OpenCV was built to provide a common infrastructure for computer vision applications and to accelerate the use of machine perception in commercial products. Being a BSD-licensed product, OpenCV makes it easy for businesses to utilize and modify the code. The library has more than 2500 optimized algorithms, which includes a comprehensive set of both classic and state-of-the-art computer vision and machine learning algorithms. These algorithms can be used to detect and recognize faces, identify objects, classify human actions in videos, track camera movements, track moving objects, extract 3D models of objects, produce 3D point clouds from stereo cameras, stitch images together to produce a high resolution image of an entire scene, find similar images from an image database, remove red eyes from images taken using flash, follow eye movements, recognize scenery and establish markers to overlay it with augmented reality, etc. OpenCV has more than 47 thousand people in the user community and an estimated number of downloads exceeding 18 million. The library is used extensively in companies, research groups and by governmental bodies.

Along with well-established companies like Google, Yahoo, Microsoft, Intel,

Figure 7.1: Open CV

IBM, Sony, Honda, Toyota that employ the library, there are many startups such as Applied Minds, VideoSurf, and Zeitera, that make extensive use of OpenCV. OpenCV's deployed uses span the range from stitching street view images together, detecting intrusions in surveillance video in Israel, monitoring mine equipment in China, helping robots navigate and pick up objects at Willow Garage, detection of swimming pool drowning accidents in Europe, running interactive art in Spain and New York, checking runways for debris in Turkey, inspecting labels on products in factories around the world on to rapid face detection in Japan.

It has C++, Python, Java and MATLAB interfaces and supports Windows, Linux, Android and Mac OS. OpenCV leans mostly towards real-time vision applications and takes advantage of MMX and SSE instructions when available. A full-featured CUDAand OpenCL interface are being actively developed right now. There are over 500 algorithms and about 10 times as many functions that compose or support those algorithms. OpenCV is written natively in C++ and has a templated interface that works seamlessly with STL containers.

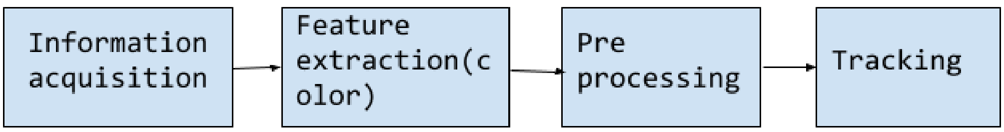


Figure 7.2: Block Diagram

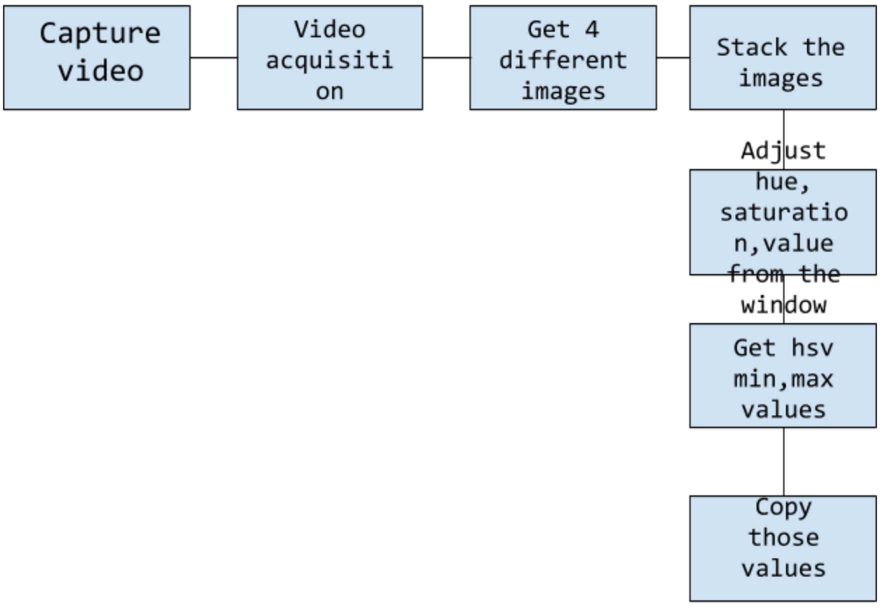


Figure 7.3: Stage implementation

7.5 Working of colorpicker.py program:

```
import cv2
import numpy as np

cap = cv2.VideoCapture(0)

def stackImages(scale, imgArray):
    rows = len(imgArray)
    cols = len(imgArray[0])
    rowsAvailable = isinstance(imgArray[0], list)
    width = imgArray[0][0].shape[1]
    height = imgArray[0][0].shape[0]
    if rowsAvailable:
        for x in range ( 0, rows):
            for y in range(0, cols):
                if imgArray[x][y].shape[:2] ==
                        imgArray[0][0].shape [:2]:
                    imgArray[x][y] =
```

```
                    cv2.resize(imgArray[x][y],
                        (0, 0), None, scale, scale)
            else:
                imgArray[x][y] =
                    cv2.resize(imgArray[x][y],
                        (imgArray[0][0].shape[1],
                            imgArray[0][0].shape[0]),
                        None, scale, scale)

            if len(imgArray[x][y].shape) == 2:
                imgArray[x][y] =
                cv2.cvtColor(imgArray[x][y],
                    cv2.COLOR_GRAY2BGR)

    imageBlank = np.zeros((height, width, 3), np.uint8)
    hor = [imageBlank]*rows
    hor_con = [imageBlank]*rows

    for x in range(0, rows):
        hor[x] = np.hstack(imgArray[x])
    ver = np.vstack(hor)
else:
    for x in range(0, rows):
        if imgArray[x].shape[:2] == imgArray[0].shape[:2]:
            imgArray[x] =
            cv2.resize(imgArray[x], (0, 0),
                None, scale, scale)

        else:
            imgArray[x] =
            cv2.resize(imgArray[x], (imgArray[0].shape[1],
                imgArray[0].shape[0]), None,scale, scale)
        if len(imgArray[x].shape) == 2:
            imgArray[x] =
            cv2.cvtColor(imgArray[x], cv2.COLOR_GRAY2BGR)
```

```
        hor= np.hstack(imgArray)
        ver = hor
    return ver

cv2.namedWindow("TrackBars")
cv2.resizeWindow("TrackBars",120,320)
cv2.createTrackbar("Hue Min","TrackBars",0,179,empty)
cv2.createTrackbar("Hue Max","TrackBars",179,179,empty)
cv2.createTrackbar("Sat Min","TrackBars",0,255,empty)
cv2.createTrackbar("Sat Max","TrackBars",255,255,empty)
cv2.createTrackbar("Val Min","TrackBars",0,255,empty)
cv2.createTrackbar("Val Max","TrackBars",255,255,empty)

while True:
    _, img = cap.read()
    img = cv2.resize(img, (640,480))

    imgHSV = cv2.cvtColor(img,cv2.COLOR_BGR2HSV)
    h_min = cv2.getTrackbarPos("Hue Min","TrackBars")
    h_max = cv2.getTrackbarPos("Hue Max", "TrackBars")
    s_min = cv2.getTrackbarPos("Sat Min", "TrackBars")
    s_max = cv2.getTrackbarPos("Sat Max", "TrackBars")
    v_min = cv2.getTrackbarPos("Val Min", "TrackBars")
    v_max = cv2.getTrackbarPos("Val Max", "TrackBars")

    print(f'{h_min},{h_max},{s_min},{s_max},{v_min},{v_max}')

    lower = np.array([h_min,s_min,v_min])
    upper = np.array([h_max,s_max,v_max])
    mask = cv2.inRange(imgHSV,lower,upper)
    imgResult = cv2.bitwise_and(img,img,mask=mask)

    imgStack = stackImages(0.6,([img,imgHSV],[mask,imgResult]))
    cv2.imshow("Stacked Images", imgStack)
```

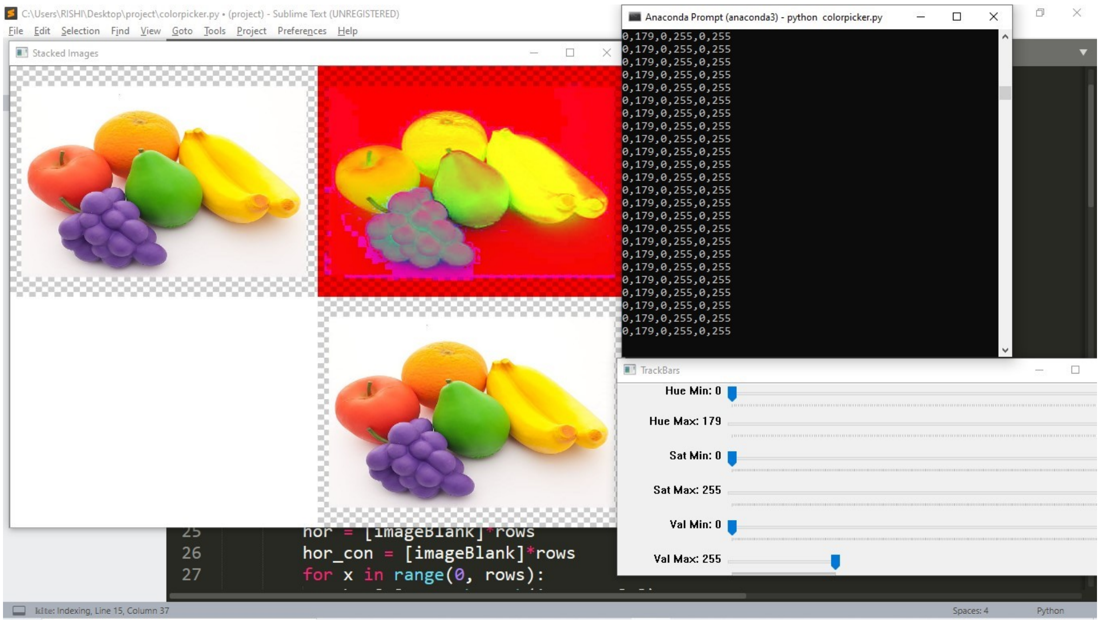
```
    cv2.waitKey(1)
```

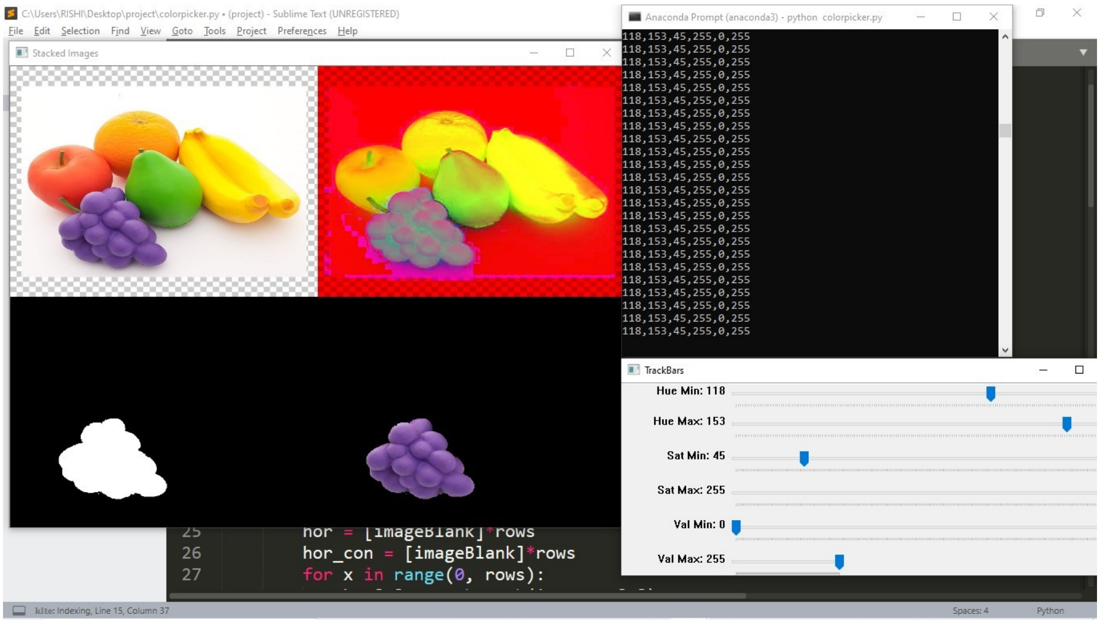
7.6 Working of main.py program:

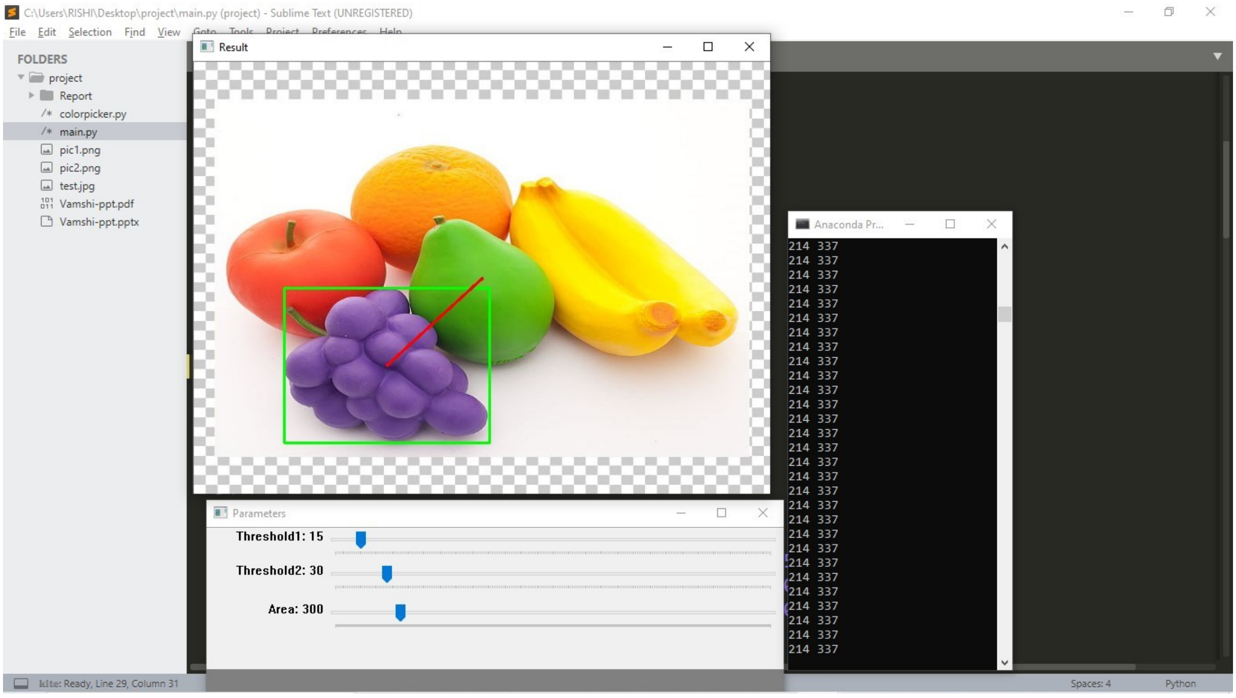
```
import cv2
import numpy as np
from time import sleep
import pigpio
pi = pigpio.pi()

servoPinVer = 32
servoPinHor = 33

frameWidth = 480
frameHeight = 320
cap = cv2.VideoCapture(0)

posX = 0
posY = 1700

hsvVal = [93,114,117,255,0,255]


def empty(a):
    pass

cv2.namedWindow("Parameters")
cv2.resizeWindow("Parameters",640,240)
cv2.createTrackbar("Threshold1","Parameters",15,255,empty)
cv2.createTrackbar("Threshold2","Parameters",30,255,empty)
cv2.createTrackbar("Area","Parameters",300,2000,empty)


def getContours(img,imgContour):
    cx, cy = 0,0
```

```
    contours, hierarchy =
        cv2.findContours(img, cv2.RETR_EXTERNAL,
            cv2.CHAIN_APPROX_NONE)
    for cnt in contours:
        area = cv2.contourArea(cnt)
        areaMin = cv2.getTrackbarPos("Area", "Parameters")
        if area > areaMin:
            peri = cv2.arcLength(cnt, True)
            approx = cv2.approxPolyDP(cnt, 0.02 * peri, True)
            x , y , w, h = cv2.boundingRect(approx)
            cv2.rectangle(imgContour, (x , y ),
                (x + w , y + h ), (0, 255, 0), 2)

            cx = int(x + (w / 2))
            cy = int(y + (h / 2))

            cv2.line(imgContour, (int(frameWidth/2),
                int(frameHeight/2)), (cx,cy),
                    (0, 0, 255), 2)
    return cx,cy

def conRange(oldMin,oldMax,newMin,newMax,oldValue):
    return (((oldValue -oldMin)*(newMax-newMin))/
        (oldMax-oldMin))+ newMin



while True:

    #Image processing
    _, img = cap.read()
    img = cv2.resize(img,(frameWidth,frameHeight))
    imgContour = img.copy()
    imgHsv = cv2.cvtColor(img,cv2.COLOR_BGR2HSV)

    lower = np.array([hsvVal[0],hsvVal[2],hsvVal[4]])
```

```
upper = np.array([hsvVal[1],hsvVal[3],hsvVal[5]])
mask = cv2.inRange(imgHsv,lower,upper)
result = cv2.bitwise_and(img,img, mask = mask)
mask = cv2.cvtColor(mask, cv2.COLOR_GRAY2BGR)

imgBlur = cv2.GaussianBlur(mask, (7, 7), 1)
imgGray = cv2.cvtColor(imgBlur, cv2.COLOR_BGR2GRAY)
threshold1 = cv2.getTrackbarPos("Threshold1", "Parameters")
threshold2 = cv2.getTrackbarPos("Threshold2", "Parameters")
imgCanny = cv2.Canny(imgGray, threshold1, threshold2)
kernel = np.ones((5, 5))
imgDil = cv2.dilate(imgCanny, kernel, iterations=1)
cx,cy = getContours(imgDil, imgContour)
print(cx,cy)



# Moving motors
#vertical
if cy!=0:
    centerY = frameHeight//2
    if cy>centerY+20:
        posY = int(np.clip(posY+10,1400,2500))
    elif cy<centerY-20:
        posY =  int(np.clip(posY-10,1400,2500))
    pi.set_servo_pulsewidth(17,posY)

 #### Horizontal
if cx!=0:
   posX = int(conRange(frameWidth,0,2000,2500,cx))
   pi.set_servo_pulsewidth(18,posX)

cv2.imshow('Result', imgContour)
if cv2.waitKey(1) & 0xFF == ord('q'):
    break
```







Output Console

Chapter 8

Applications

8.1 Film And Television

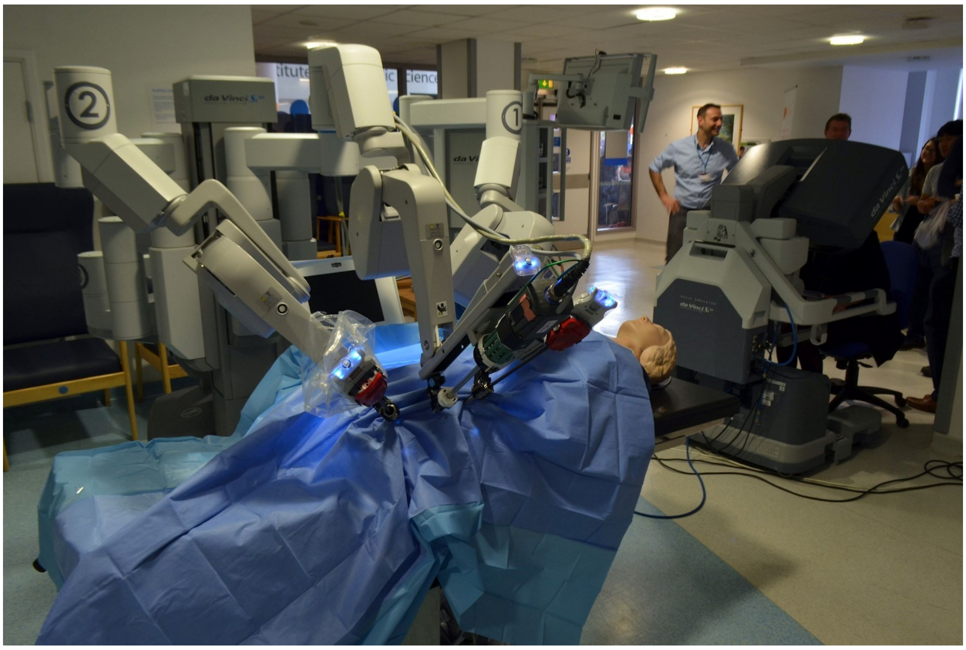
The film industry has been a driving force revolutionizing the technology used to develop animatronics. Animatronics are used in situations where a creature does not exist, the action is too risky or costly to use real actors or animals, or the action could never be obtained with a living person or animal. Its main advantage over CGI and stop motion is that the simulated creature has a physical presence moving in front of the camera in real-time.

T-rex animatronics in Jurassic park (1993)

The technology behind animatronics has become more advance and sophisticated o ver the years, making the puppets even more realistic and lifelike. Animatronics were first introduced by Disney in the 1964 film Mary Poppins which featured an animatronic bird. Since then, animatronics have been used extensively in such movies as Jaws and E.T. the Extra-Terrestrial, which relied heavily on animatronics. Directors such a s Steven Spielberg and Jim Henson have been pioneers in using animatronics in the film industry.

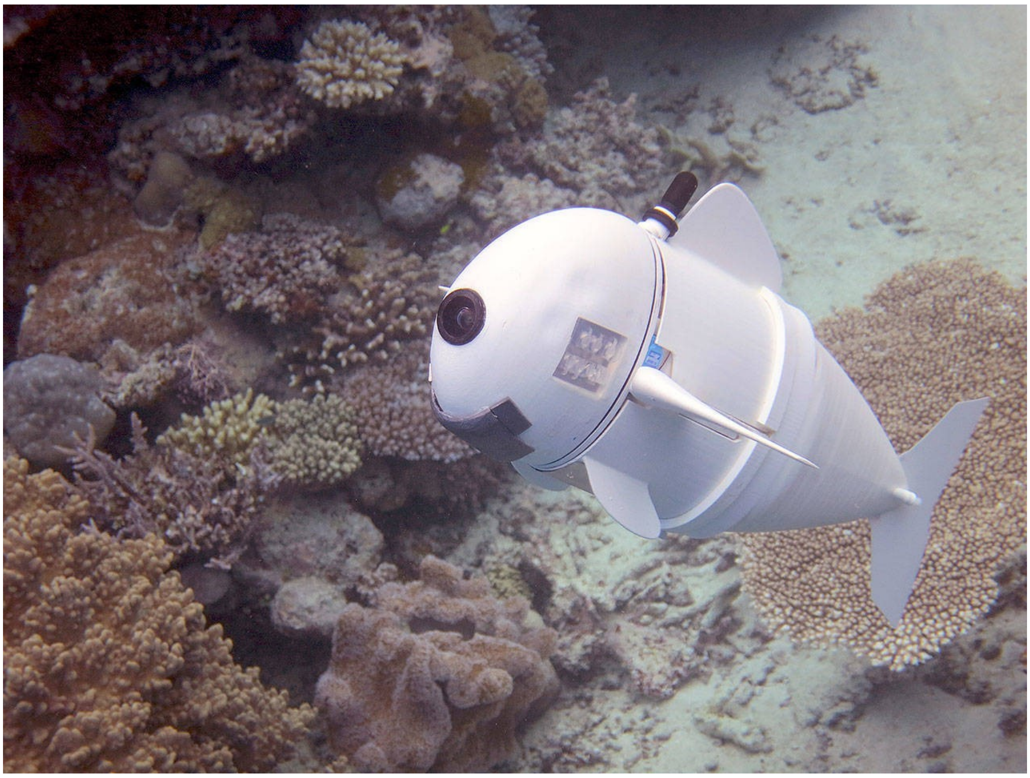
8.2 Medical

Animatronics could be employed in the field of medicine to simulate real-time surgeries on live patients. They could also be used for creating artificial limbs that are controlled by microcontrollers, which requires a significant advancement and contribution in the field ANN's .



8.3 Ichthyology

Animatronics could also be employed in the field of ICHTHYOLOGY, in which an artificial fish is designed that is visually similar to the fish under study. Using underwater propellers and thrusters or a flexible body and tail lines with servomotors, this artificial fish could easily mingle with t he corresponding species and attaching a small camera inside of it will enable corresponding species and attaching a small camera inside of it will enable the researchers to study understand the behaviour of fish underwater.



MIT Soft Robotics Fish

7.4. DEFENCE SYSTEM: Animatronics could b e employed in the field of surveillance by creating an animatronic animal, which could be equipped with a camera or sensor eyes for monitoring certain hotspots. For Military use in bomb diffusing.

Boston Dynamics SPOT

Aerium RoBird Ornithopter drone V3

Chapter 9

Conclusion And Future Scope

We have presented animatronic eyes and its mechanism in detail. We have also presented how opencv can be used to implement object tracking facilities in this project. In future, most of the researchers from different backgrounds like from computer science and electrical departments are invited to implement more image recognition techniques and multiple object tracking mechanisms. Design engineers from several departments are invited to improve the design of this animatronic eye in more efficient way. Machine learning researchers may apply several efficient algorithms to improve its performance.

Chapter 10

References

https://www.murtazahassan.com/

https://en.wikipedia.org/wiki/Human_eye

https://nba.uth.tmc.edu/neuroscience/m/s3/chapter08.html/

https://opencv.org/

https://medium.com/visionwizard/object-tracking-675d7a33e687

Real-time object detection and tracking using colour feature- a research paper from BBVL Deepak, Tanjot sethi, Meta devprasad murthy.

https://en.wikipedia.org/wiki/Raspberry_Pi

https://www.raspberrypi.org/products/raspberry-pi-4-model-b/

https://www.raspberrypi.org/documentation/

https://www.autodesk.in/products/fusion-360/personal