

Laser Tracking

Final results - <https://youtu.be/veeBLcLZFfc>

Introduction

In part of autonomous robotics course in university we had to arrange final project with connection to autonomous machine.

We thought about doing laser tracking on specific faces in order to build some "killing machine" that identify target (our vision was terrorists/wanted people) and shoot towards it accurately bullet.

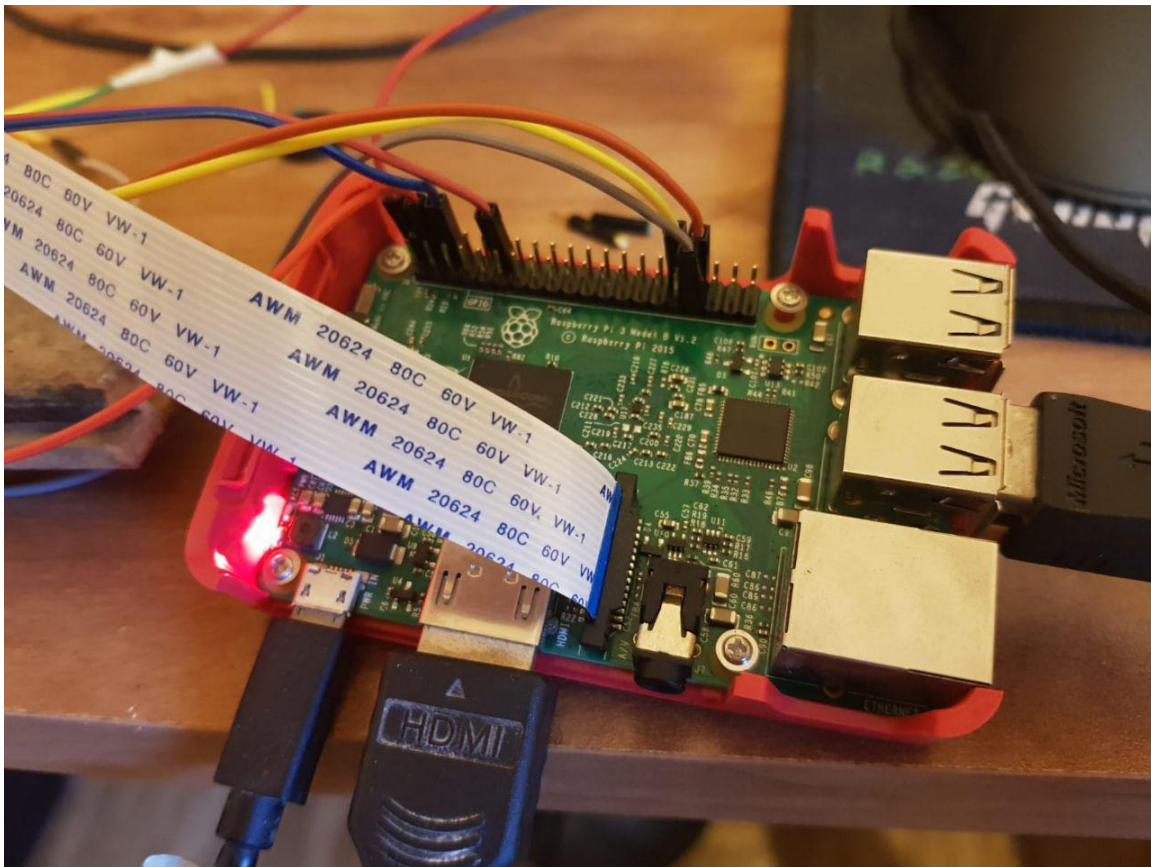
Of course we couldn't build such a machine (real shooting) so we do it with laser beam, our targets would be motion detected so every movement captured in camera will get the laser beam.



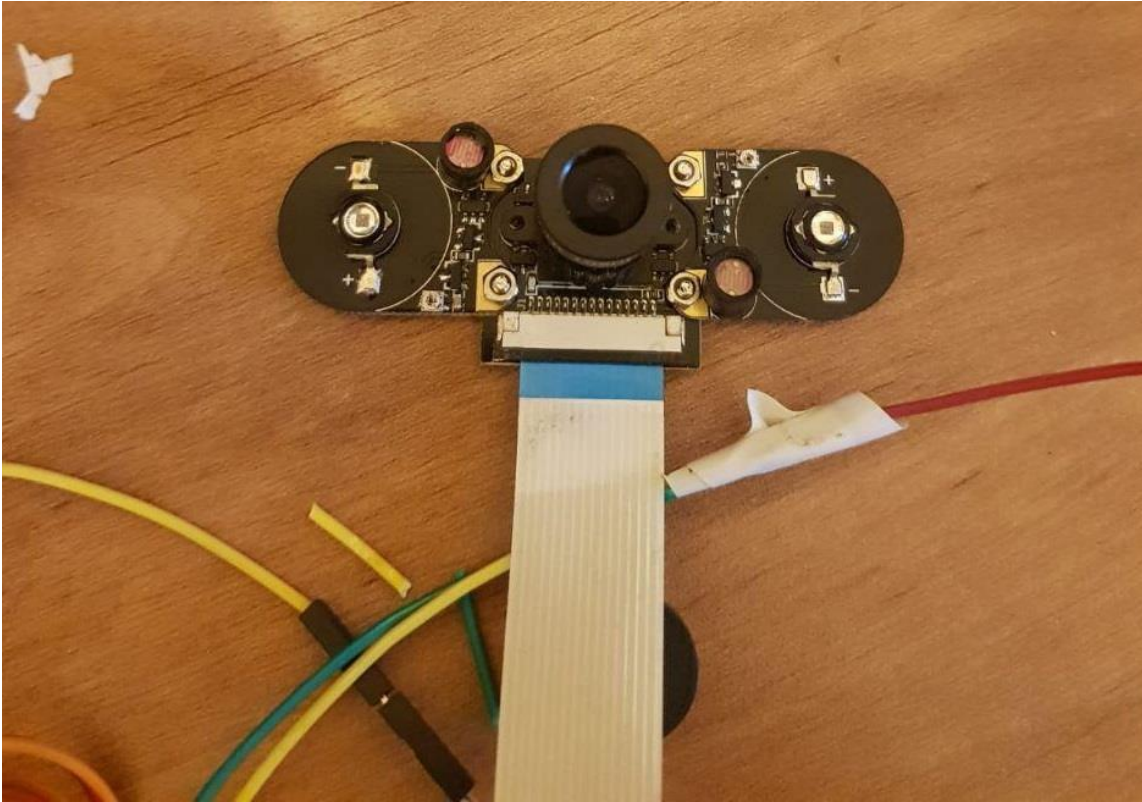
Equipment

With a relatively small budget we achieved a good product.

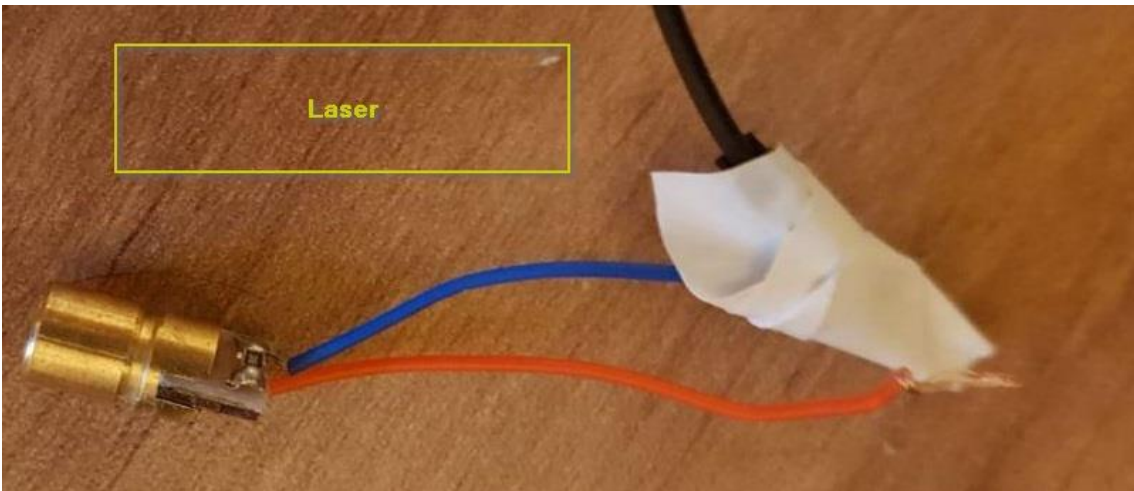
Raspberry pie 3B



Camera



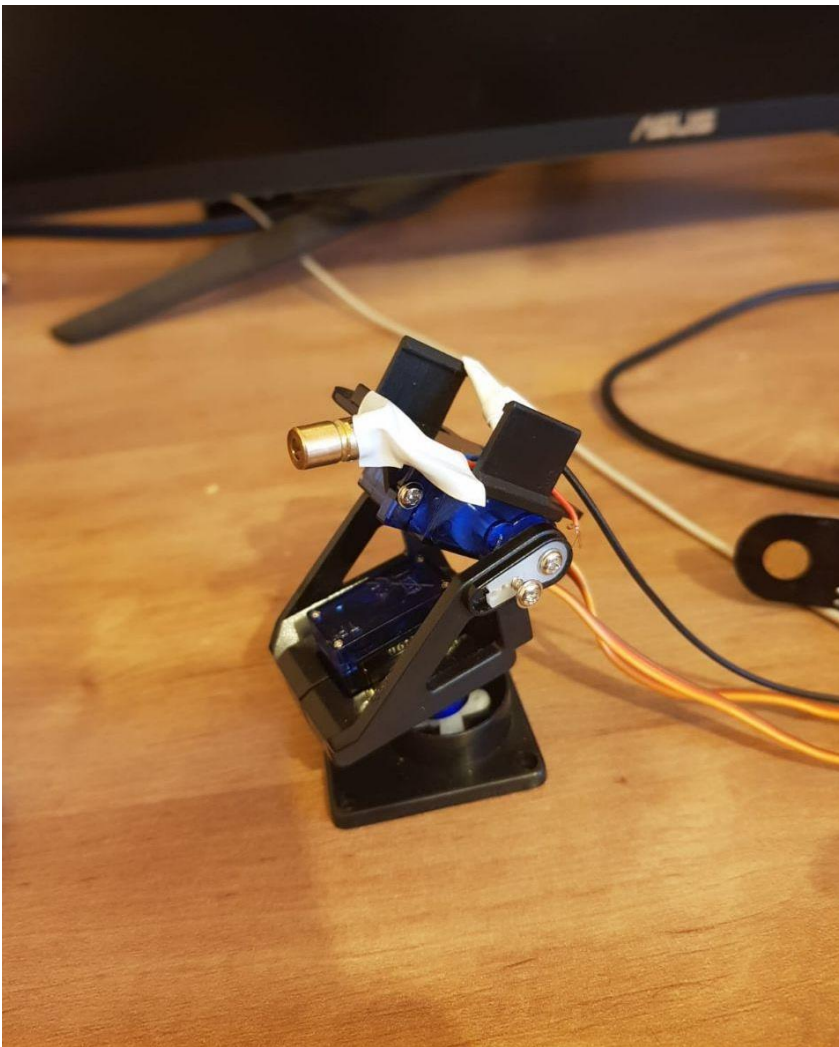
Laser



Servos

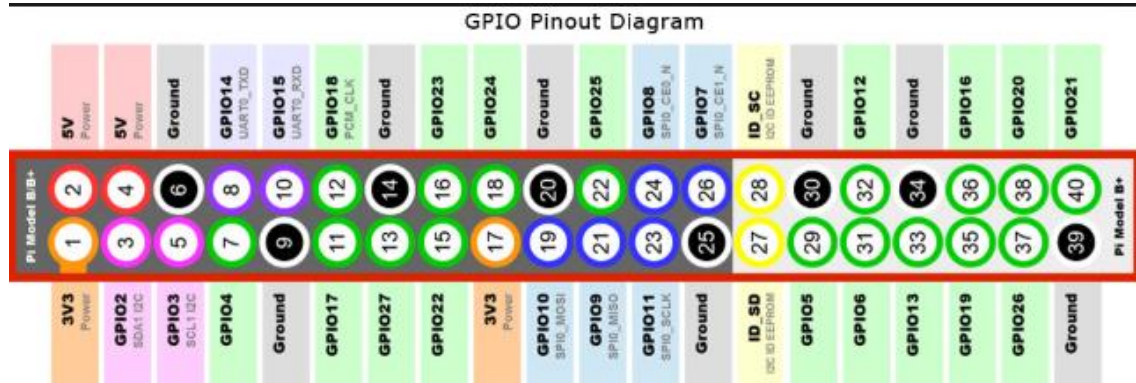


Laser mounted on the 2-axis servos



Connections to raspberry pie

The pie comes with only 2 – 5V ports and 2 – 3v3 ports.

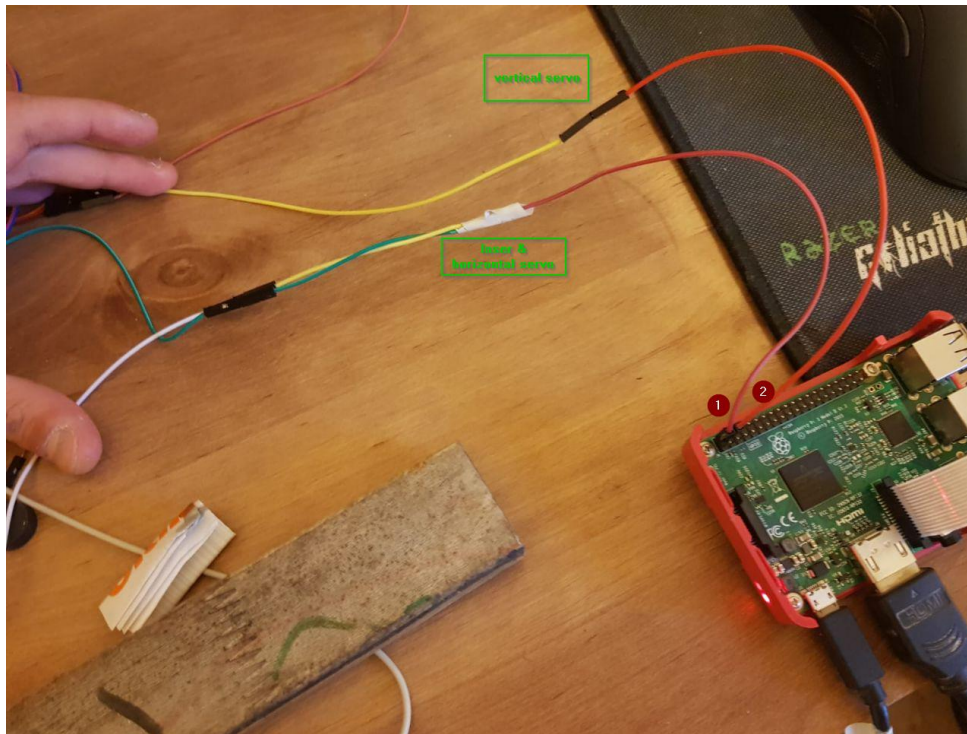


We can see that only 2 and 4 ports are available for 5V input

But we have 3 devices using the 5V:

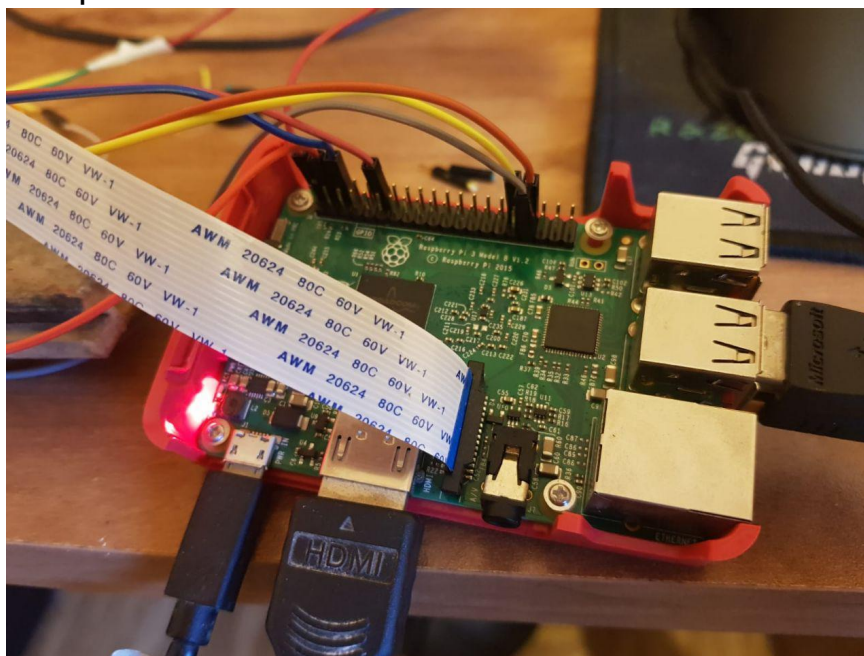
- 1) Laser,
- 2) Horizontal servo
- 3) Vertical servo

So we had to plug in 2 wires of the 5V into 1 pin and the one remaining to the second 5V pin:



(At first, we tried to plug our laser to a 3v3 pin but the beam strength was very weak. The same case happened with the servo when plugged to the 3v3 pin)

Our pie final connections would seem like this:

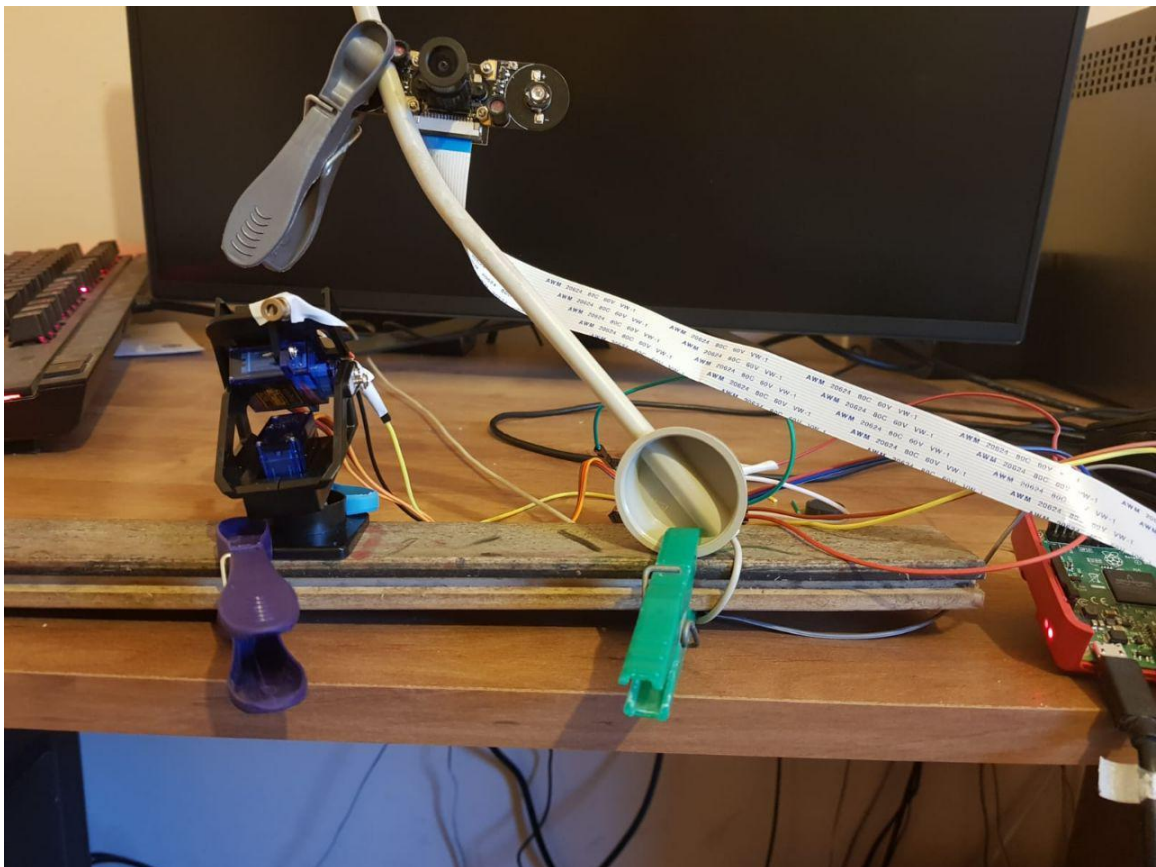


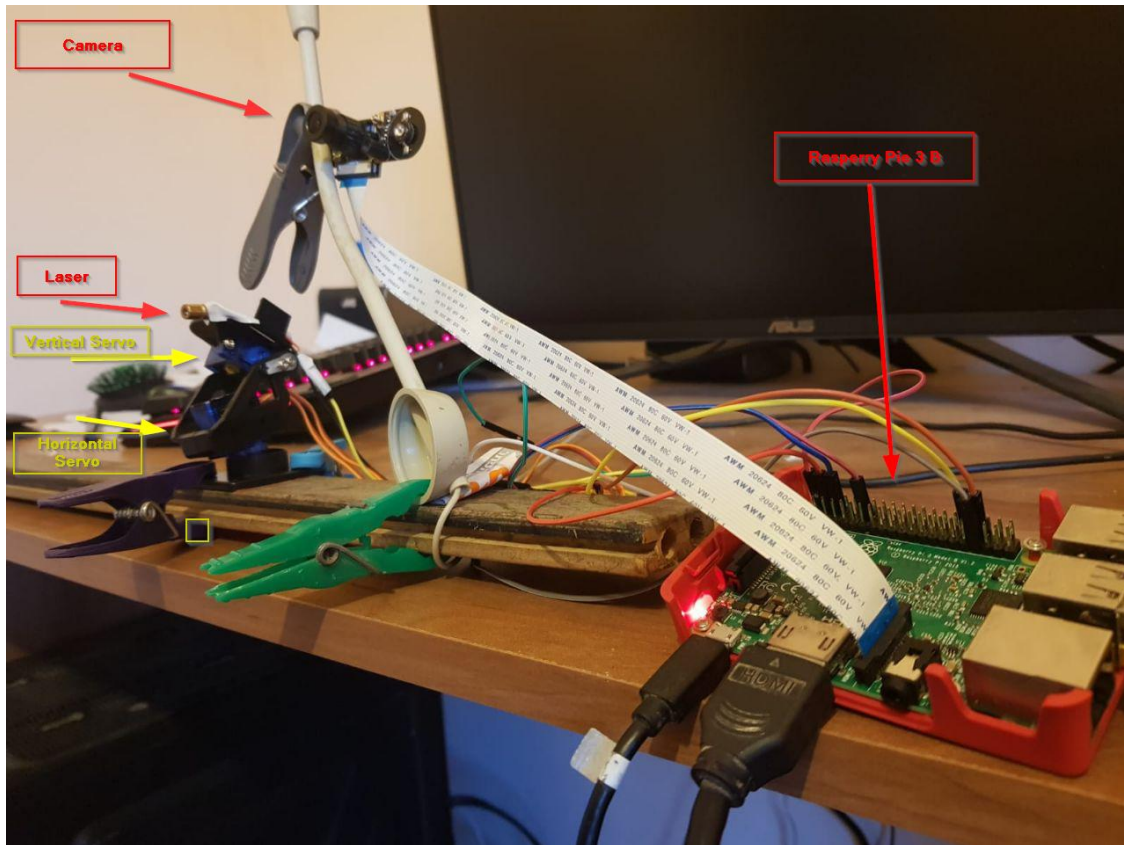
Model Architecture

First we need to locate the camera in the closest place from the laser which will give us the best precision.

We figured the best results made by locating the camera slightly above the servo, this gave us nice field of view and well sync between those two.

With our cheap budget we manage to improvise the following structure:





We used what we could find and improvise the "camera holder" which is microphone stand, a piece of wood and some ties to hold it properly enough.

This is critical part, locating the camera and servo as we did, because we scale the image and try to give good angle to servo to actually hit our desired target.

Performance

All calculations made in raspberry pi, although his average stats –

- Quad Core 1.2GHz Broadcom BCM2837 64bit CPU
- 1GB RAM

We faced difficulties in performance specially if we open new thread or even for loop it may cause struggles on frames per second.

We didn't quite understand why starting new thread given that raspberry pi processor is quad core effects heavily on performance and decrease fps rate – which is not good.

The solution we made in our code is the least impact we could do to performance, we tried a lot of things – recognize the laser/ red light, calibration function to improve laser aiming but unfortunately it cause bad performance and we decided to remove it from our code.

(Some functions is still in the code but not in use)

However, the final results was good and fps rate is kind of acceptable and the speed of the servo and laser according to the frames we get were fine.

We think with much better processor the performance can easily raise and make the fps limit higher, image processing would be smoother and adding more features will be more possible without interrupting performance.

Also, we had some struggles in voltage performance – 2 pins for 5V and we need 3 spots of voltage and the solution required some cutting of wires which is fine but I saw some changes in performance according to this connection and I am not sure this is the right solution for this, but eventually it worked.

Code Review

Libraries required:

- Imutils
- Cv2
- RPi.GPIO – should be already in pi
- Numpy

RPi.GPIO brief guide

In order to get all mechanical objects to work such as laser and servo connections we have to control them via python – so the way we doing that shortly explained here.

First we need to set mode of GPIO to Board status –

```
GPIO.setmode(GPIO.BOARD)
```

Now we choose which index of pin we want to activate –

```
GPIO.setup(33, GPIO.OUT) // In this case 33 index
```

Then we want to determine how much power we want it to receive –

```
laser=GPIO.PWM(33, 50)
```

Start it –

```
laser.start(0) // 0 means permanent
```

Stop it –

```
laser.stop()
```

We have the function of moving the degree of vertical/horizontal servo, using GPIO function –

We figured that the best results of rotating the servo's constantly was by this formula –

$(\text{vertical_angle} / 18) + 2$

We don't really understand why it worked but after some tests and asking couple of people this is it.

The same goes to horizontal angle.

After calculation of desired degree we enabling the output to this pin by

```
GPIO.output(12, True)
```

```
vertical.ChangeDutyCycle(duty)
```

Sleep is required after this method, we did try without or even decrease the sleep rate but the current rate made the best results.

If no sleep the servo will freaking out.

For more information of this part you can go into the code and explore it!

Laser to Point function –

Basically we have the output from motion tracker model which is rectangle – we allow only one box each frame of motion.

Each frame we have that output – (x,y,width,height) so the center is $(x+w/2, y+h/2)$

This is our desired point to aim at.

We scaled the image by made some tests of right,left,up,down bounds.

This parameters stored in the beginning of the code.

After we know our boundaries we can compute the degree towards that point by simple calculation –

$$\text{vertical_angle} = \text{vertical_max} - ((y - \text{diff_y}) / 375) * \text{vertical_range}$$

$$\text{horizontal_angle} = \text{horizontal_max} - ((x - \text{diff_x}) / \text{frame_width}) * \text{horizontal_range}$$

- Diff_y (should be our calibration fix)
- Diff_x (should be our calibration fix)

Then we just rotating to this angles.

Update diff function

We wanted to improve our system by detecting the laser in the image and calculate the differences between real hit against desired hit.

We did it by filtering red color from our frame – this gave us only red color (which is the laser color) then we calculate the most point that closer to our target (gave us indication of the laser) and calculating the differences between those two.

In reality it gave us poor results specially in fps rate, may work with much better equipment.

Is Laser function

Function that check if laser found in frame.

In reality also effects performance and also wasn't necessary.

The rest of the code is kind simple –

We used opencv to detect motion, briefly we compare each frame with his last and detect the things had changed and mark it.

The code is pretty standard and many explanations can be found on the internet, also notes inside the code.

The main thing is we detect each frame one motion, we calculate the (x,y,w,h) values and aim towards his center by activate laser to point function – this will give us new degrees to move to and we execute it.

This code can work on many neural network models as face recognition, object detection and so on...

Just need to give the rectangle that you want to aim to and that's all.

Important! The calibration process is very critical, good calibration may cause very good results! You may think doing it automatically but with our equipment we couldn't really do much about it and did the calibration manually.

Summary

The results was good according to our equipment – we didn't actually use some static camera holder and had to improvised a lot, I believe with good preparation of these things like the holder and placing the servo in good balance from the camera may lead to perfect system.

Also processing issue is heavily noticeable in raspberry pi.

Calibration is very important step to make this all work, we had to scale the boundaries of the image with the servo to understand the degrees, also because our camera holder was with old microphone and some times camera moved a little and the whole calibration out of range and we had to do it again.

We had some struggles on plugin the laser and servo because we didn't know how to properly install it but all this things we learned on the way.

To conclusion, this work was very fun to built, dealing with embedded tool and some electricity things, also the idea of the project is extremely cool and may improve in the future.

Aviv Harazi

Maor Bakshi