CS-308-2016 Final Report

Tracktor

Team members

Aditya Kumar Akash - 120050046 (Team Leader)
Nishant Kumar singh - 120050043
Anurag Shirolkar - 120050003
JVS Shyam - 120050052

INTRODUCTION

We have built a system having a holder for objects like mobiles and capable of tracking a person. We call the system *'Tracktor'*, from *'Tracker + Rotor'*, as the tracking system contains a rotating motor.

Suppose you are in a video call and want to be able to move around to fetch something or perform a household work, say in kitchen. To keep you hands free from holding the mobile/tablet, you need a camera that tracks you and turns towards you wherever you are.

Tracktor allows people to parallelize their household work with video calls, watching movies etc. It gives you a hands free way to perform video calling or other uses of mobile like watching a cricket match. It can also be used in a skype meeting scenario where a person can demonstrate a project without worrying about the view of the camera.

PROBLEM STATEMENT

The aim of the project is the design a system that tracks the user in present in its view and orients itself towards the user whenever he changes his position. The tracking of person is done based on face detection.

User should also be able to mount his own device, such as mobile phone or tablet, on the system. The users personalize the system to his his/her use.

REQUIREMENTS

1. Functional Requirements

a. The system should take input in the form of video feed or images at regular intervals
b. Using the input it should locate the user.

c. It should allow user to mount his own device, such as a phone, on the system
d. The system should rotate the device towards the user

2. Non-functional Requirements

a. Correctness - the system should correctly track the user and rotate the device
b. Portability - it should be small and lightweight
c. Scalability - it should be able to hold any size of the device ranging from phone to tablets
d. Response time - the system should be quick
e. Price - it should be affordable

3. Hardware Requirements

a. Raspberry Pi
b. USB Camera
c. Servo motor
d. Power source, we have used 5V adapters for both Rpi and the servo motor

4. Software Requirements

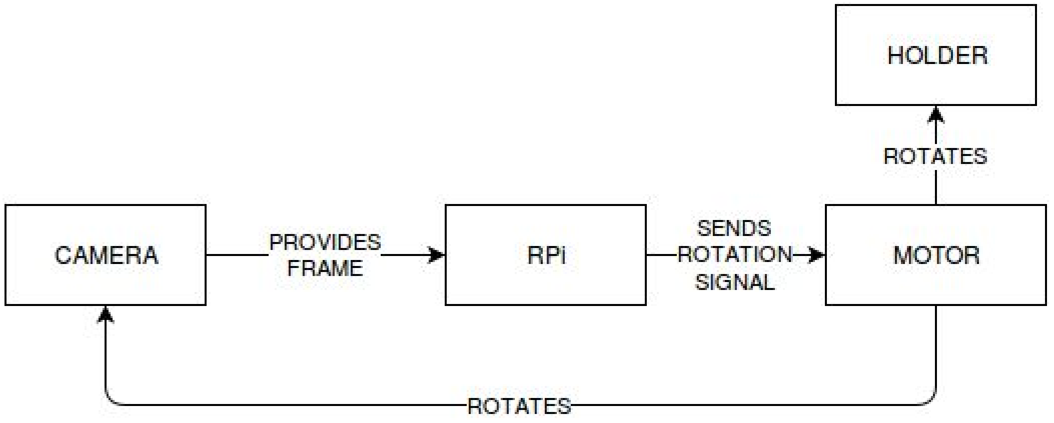
The following softwares need to be installed on Rpi:
a. Raspbian OS
b. Python 2.7
c. OpenCV 2.4.5

SYSTEM DESIGN

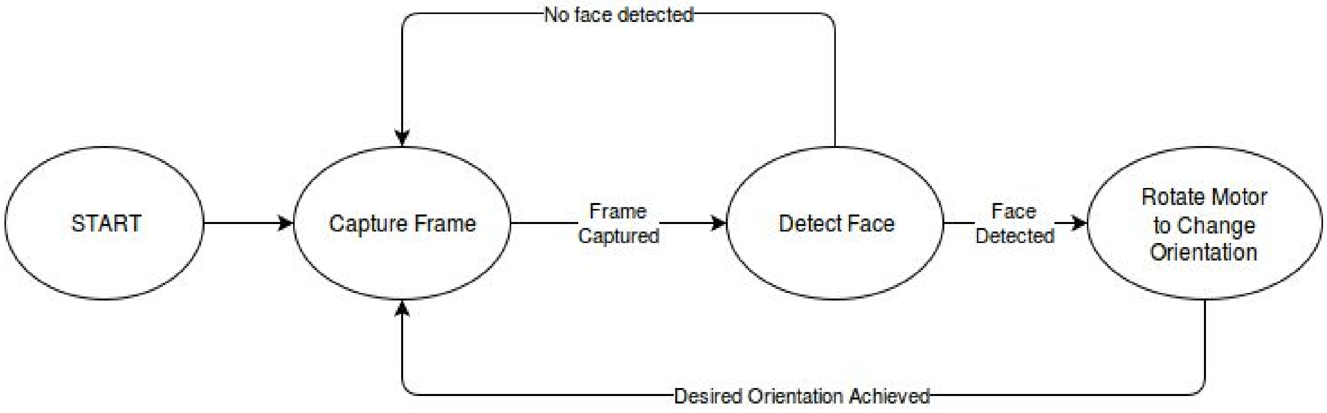
The functional components of the system are following -

1. RPi : Performs face detection and sends signal to the motor to orient itself to face the person in view
2. Camera : Sends continuous frame to the RPi
3. Motor : Rotates the system based on input from the RPi
4. Holder : Allows to keep mobiles firmly attached to the system

The interaction of the components described above can be seen in the following component diagram.



The FSM of the system is shown in the following diagram.



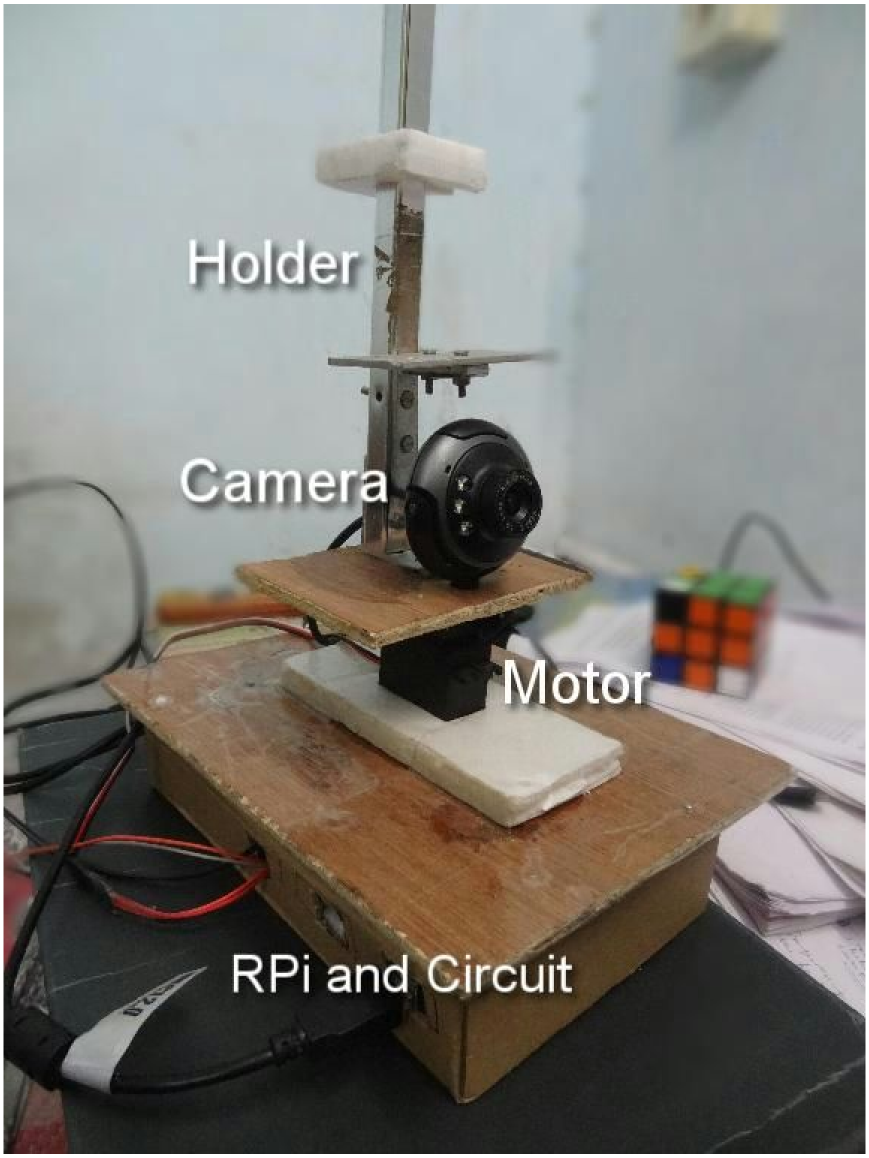
The states of system are :

1. *Start* - This is the default off state of the system. From any other state, upon removing the power source we reach this state.
2. *Capture Frame* - In this state the RPi interacts with camera and obtains the image frame. Next the system moves into ‘Detect Face’ state.
3. *Detect Face* - In this state, system perform face detection to detect for faces in the image captured. In case no image is present the system moves to ‘Capture Frame’ state. In case one person is present the system computes the angle required to orient the camera towards the person and moves next state of rotation of motor. In case more than one person is present, the

system breaks tie arbitrary and computes the required angle of rotation to orient the system and moves to next state.

4. *Rotate Motor* - In this state the motor performs rotation as per the signal from received from the RPi.

The snapshot of mechanical part of the system -



WORKING OF THE SYSTEM AND TEST RESULTS

Working

The device has a USB Camera mounted on the rotating platter. The camera captures images continuously and processes the image to find faces in it. Adaboost algorithm has been used to detect face in the frames. It uses local binary patterns feature extraction method for classification purpose. Once the face is detected, the x-coordinate of the face center is calculated. Using this information duty cycle of the servo motor is found out. The angle that the motor is required to turn through in order to align the mounted device towards the user is given by the duty cycle. PWM signal with the calculated duty cycle is sent to the servo motor which then rotates the stand by the desired angle. This process keeps on repeating.

Testing

1. To test whether the system takes the video input correctly, we display the video frames on a computer.
2. In the video shown on the computer, we highlight the user’s location by drawing a green rectangle. To ensure whether the system correctly recognizes the face, we tested it with different users and with different backgrounds. The system correctly draws rectangle around the user’s face.
3. We have tried to mount a number of different sized phones on the system. The system works without the devices falling.
4. The system correctly orients the device towards the user whenever the user moves without letting the device fall or without having the weight of the device slow it down.

DISCUSSION OF SYSTEM

a. What all components of your project worked as per plan?
    1. Camera
    2. Raspberry pi
    3. Servo motor

b. What we added more than discussed in SRS? - Nothing

c. Changes made in plan from SRS
    i. The plan was to use the motion detector to detect movements and use the camera only when there is motion. Currently the system doesn't make use of the motion detector and instead uses a continuous feed from camera. The reason being that it was complicated to interface both the camera and the motion detector and also the system was quick and good enough with only the camera.
    ii. The system can only cover ~225° due to limited movement of servo motor

FUTURE WORK

1. Interfacing multiple cameras in order to capture the entire 360^0 view of the scene. Currently the camera can capture only about 53^0 field of view. If the user is beyond that, it won't detect them.
2. Incorporating motion sensor to save power. By using motion sensor we can turn off video capture if there is no one in the scene to save power.
3. Tracking the subject can also be done using motion sensor. The idea is to only capture images at regular intervals of say 1 minute and meanwhile in between use the motion sensor to track the movement of the user and turn the rotor accordingly. It would be faster than current implementation since image processing won't be required for every frame.

CONCLUSION

We were able to built tracktor which has following benefits

- Tracktor helps you parallelize household works with video calling or watching movies etc.
- It is a system that tracks a user orients itself towards them
- It can hold device with varying sizes and rotate them

The project provided a great opportunity to work with embedded system. We worked together as a team to tackle the challenges faced in completing the project.

The working demo of the project can be seen at : https://youtu.be/ywYAUtYPsEM

Screencast showing setup and working of system :
https://www.youtube.com/watch?v=qWQz7CYZ5Fg

REFERENCES

- OpenCV
- Raspberry Pi
- http://docs.opencv.org/2.4/doc/user_guide/ug_traincascade.html
- http://computers.tutsplus.com/tutorials/controlling-dc-motors-using-python-with-a-raspberry-pi--cms-20051