International Journal of Scientific Research & Engineering Trends
Volume 5, Issue 3, May-Jun-2019, ISSN (Online): 2395-566X

# Motion Following Camera With Face Tracking

**Ankit Yadav Shubham Mhabdi Rishabh Singh**
Dept. of Electronics and Telecommunication Engineering
Thakur College of Engineering and Technology
Mumbai, India

***Abstract*- There was a day when we faced a problem of adjusting a large area such as banks, offices and even household CCTVs to cover complete area, this CCTVs are fixed and to ensure there is no blindspot multiple CCTVs are used for surveillance. This gave us our problem definition of the project. The plan is to create a Camera base that will rotate the camera in the direction any random person will move. Thus, any movement within its given radius camera basewill rotate in same direction. Our intention is to use Image Processing to detect motions and instruct Arduino to respond accordingly, the Arduino will instruct motors to rotate accordingly and Camera will capture people randomly . The purpose of this project is to design a camera module which when detecting motion starts following moving object and track the face.**

***Keywords* – Camera base, Motors, Arduino, Image Processing.**

## I. INTRODUCTION

The blindspots in static security camera is major issue, multiple camera are used to cover the blindspots which becomes a costly affair. This gave the problem definition for our project.The plan is to create a Camera base that will rotate the camera in the direction any random person or object will move. Thus, any movement within its given radius camera could rotate in 360° angle. Our intention is to use motion sensors. Motion sensors would detect motions and instruct Arduino; in response, the Arduino will instruct servo motor to rotate accordingly and Camera will capture people randomly.

Real time face tracking refers to the task of locating human faces in a video stream and tracking the detected or recognized faces. There are many real world applications of face detection and other image processing techniques. There are many algorithms like Viola Jones ,Kanade-Lucas-Tomasi ,camshift etc. that are quite helpful in face detection.

## II. PROPOSED ALGORITHM

First the camera position is configured by the Arduino to the initial position. Then the camera provides the video input to the MATLAB. The MATLAB detects the face and sends the co-ordinates of the face to the Arduino. Basing on the principle of centroid the Arduino tracks the face. During tracking the face of the person that is detected is captured as an image by the MATLAB and the MATLAB uses face recognition algorithm for recognising the face detected.

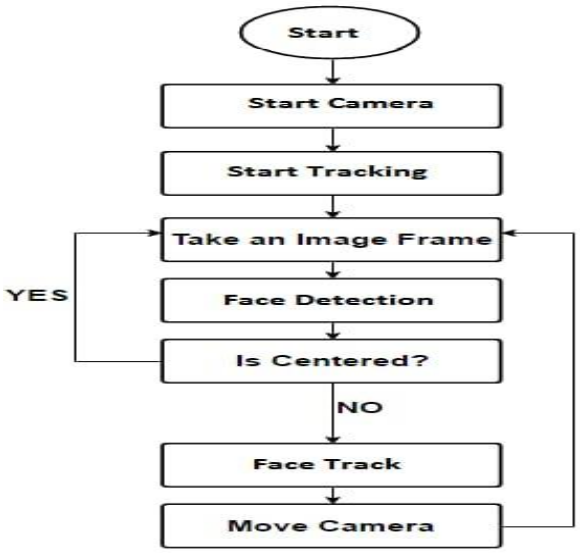


Fig 1 Flowchart of Proposed System.

## III. METHODOLOGY

The algorithm used for human face detection is described below. It consists of four important steps, they are given below

**1. Haar features**

Images are classified mainly on feature based systems because in feature based systems the computations required is less. For detection of face mainly two rectangular features, three rectangular features and four rectangular features are used. These features are called HAAR features and are shown in figure (2). The features are calculated by simple addition and subtraction operation. The features are computed as the difference of sum of pixels in the black portion to the sum of pixels in the white portion. For any rectangular HAAR feature the features are calculated in the above procedure. This

procedure is repeated for different types of HAAR rectangular masks and also by varying the size of the mask.

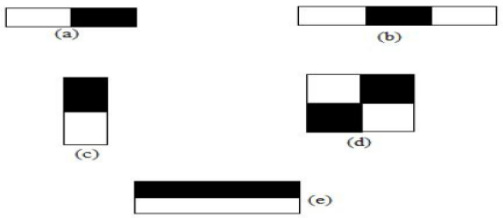


Fig. 2 Haar Features.

**2. Integral Image**

Rectangular features of an image are best described by its intermediate form which we call it as an integral image.
This can be explained by taking the below image In the above diagram, point 1 describes the total sum of pixels that are present in region A. For point 2, gives the sum of pixels of the regions A and B. Similarly, for points 3 and 4 sum of pixels in the region (A, C) and (A, B, C, D). Where qq(a,b) is an integral image and q(a',b') is the original image. Using the following formula, Where p(a,b) is the cumulative row sum p(a,-1) = 0 and qq(-1,b)= 0

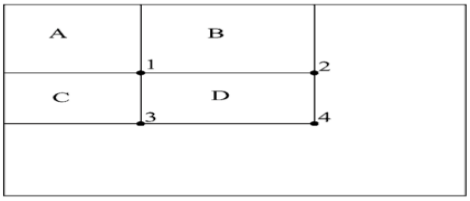


Fig. 3 Integral image.

**3. Ada Boost**

During the extraction of the features from the image, the features obtained are nearly 160,000 in an image. It is very complex and expensive to check all the features to find the face features from the image, for this purpose AdaBoost technique is used. Out of all the features obtained some number of features is selected to make an efficient classifier, for this purpose the AdaBoost adapts threshold levels such that to make a strong classifier from all the weak classifiers obtained.

Let $c_j(x)$, $f_j$, phij and $p_j$ denotes the classifier, feature, threshold and parity respectively indicating the direction of inequality where x is a 24x24 pixel sub-window of an image. For initial classifiers, they have a less error rate of 0.1 to 0.3 and also less complex, further the complexity increases and has an error rate of 0.4 to 0.5. AdaBoost runs for rounds starting from t=0,1,2…T, for every round it generates and calls the classifiers and distribution of weights wt is done which determines the importance of image in the data set for classification. For each round the weight of the incorrect classified image is increased and weight of the classified image is decreased such that the next new classifier focuses on the incorrectly classified images. The procedure for boosting is given below,

- Given example images (x1, y1) ……. (xn, yn) where yi = 0 for negative and yi = 1 for positive examples.
- Initialize weights w (1, i) =1/2u,1/2v for yi= 0,1 respectively where u and v are the number of negatives and positive
- Respectively.
- For t=1, ……., T:
- Normalize the weights, so that wt is a probability distribution.
- For each feature, j, train a classifier tj which is restricted to using a single feature. The error is evaluated with respect to wt,
- Choose the classifier, tt, with the lowest error
- Update the weights:
- where ei = 0 if example xi is classified correctly, ei = 1 otherwise, and formula
- The final strong classifier is:

**4. Cascading**

The positive and negative parts of an image are segregates using the classifier i.e., simple classifier. Weak classifiers are nothing but simple classifiers. Complex classifiers are strong classifiers in which it will take the output from the previous classifier and it will take as present input in which it may reject or accept that particular window. To achieve the accurate output for the detection this process goes on. The sub band need to pass all the intermediate classifiers in order to get the detected output otherwise the sub band window will be rejected in the path. This overall form of detection as a degenerate tree is known as cascading.

Using AdaBoost technique, stages in the cascade using training classifiers are constructed. To get the minimum false negatives default AdaBoost threshold values are designed and to yield low error rates. But at lower threshold it gets higher detection as well aa higher false positive rates. The accuracy in classifiers decreases if we reduce the computation time. So there is a tradeoff between accuracy and speed. To limit this one a two-feature strong classifier is constructed which is known as first stage classifier. In this we can detect 100 percent of the faces with a false positive rate of 40 percent by adjusting the threshold.

The second classifier goes through more difficult task then the first. The complete idea is to remove the unnecessary areas and to increase the probability of face detection in the starting stages only. So that there will be a gradual increase in the probability in further stages. The cascading is shown below in the figure (4).

International Journal of Scientific Research & Engineering Trends
Volume 5, Issue 3, May-Jun-2019, ISSN (Online): 2395-566X

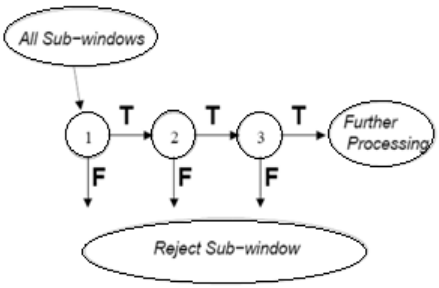


Fig 4 Cascading.

**5. Tracking of face**

Initially the camera position is set and considered it as origin. Now after running the Viola Jones Algorithm the face is detected in the real-time video which is taken from the camera and sends the coordinates of the face for tracking. Now initially the video input is given to the MATLAB for face detection and inserts a bounding box around the face.

The algorithm detects the face from the video and calculates the coordinates of the centroid of the box. These coordinates from the face are sent as input to the Arduino Uno board. The camera is connected to two servos for pan and tilt of the camera. These servos are connected to the digital outputs of the Arduino board and coordinates are given as input. The camera set position is taken as origin and four quadrants are formed. If the coordinates are in a quadrant the Arduino board sends signals to the servos such the coordinates are again come back to the origin position. If the coordinates are on positive x-direction then the servo pans to the negative x-direction makes the coordinates to then origin. Similarly, for tilting also, if the coordinates have varied x and y values then both pan and tilt servos work simultaneously to get the face in range. Thus, the tracking of image takes place by using the coordinates of the face sent by the MATLAB..

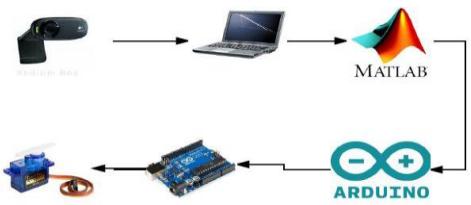


Fig 5 Tracking of Face

## IV. SIMULATION AND RESULTS

Figures and tables must be centered in the columns. Large figures and tables may span across both columns. Any table or figure that takes up more than 1 column width must be positioned either at the top or at the bottom of the page. The Figures 6, 7 shows the detection and tracking of the face using MATLAB and Arduino

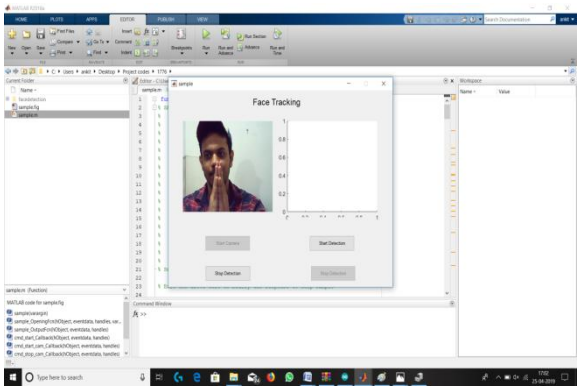


Fig.6 Face Detection using Matlab.

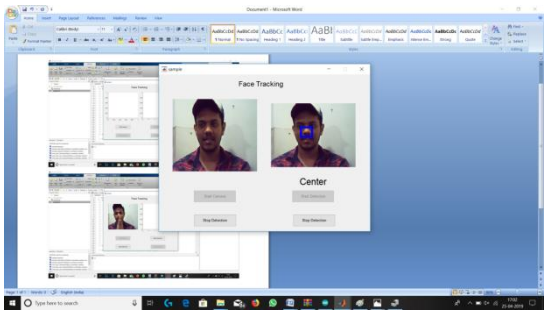


Fig 7 Nose at Centre of Frame.

Fig. 8 Hardware Setup

## V. CONCLUSIONS

Our aim was to build a device that rotates itself so as to track the person while in motion. A device is successfully built, Which is 'Motion Following Motorized Camera Base'. we first used servo motors, its response was not quick enough and was unstable, so decided to go with dc motor. For detecting motion previously using PIR Motion sensors was the intension , then we finally came at outcome of face detecting by using image processing. The entire project is specifically useful in Home/Office based security system which can be great when security is required for large area.

## REFERENCES

1. Maneesh Ayi, Ajay Kamal Ganti, Maheswari Adimulam, Badiganti Karthik, Manisha Banam, G Vimala Kumari: "Real-Time Face Tracking Using MATLAB And Arduino." Student, Assistant Professor, Department Of Electronics And Communication Engineering, MVGR College Of Engineering. In December 2017 International Journal For Research In Applied Science &Engineering Technology (IJRASET).
2. Kiran R, M Lohith, Yogesh E, Sai Kumar A, Anitha J, "Interfacing Of MATLAB With Arduino For Face Recognition"In 2017 Recent Advances In Technology And Engineering (RATE-2017),International Journal Of Science, Engineering And Technology.
3. Venkata Sasank Pamulapati ,Yekula Sumith Rohan, Vemula Sai Kiran, Saranu Sandeep, Maram Srinivasa Rao. "Real-Time Face Tracking Using Matlab And Arduino".Electronics And Communication Engineering, Vasireddy Venkatadri Institute Of Technology, In FEB 2018 IRE Journals.
4. Http://Www.Ipol.Im/Pub/Art/2014/104/Article.Pdf
5. Https://Www.Cs.Cmu.Edu/~Efros/Courses/LBMV07/Papers/Viola-Cvpr-01.Pdf
6. https://create.arduino.cc/projecthub/lindsi8784/motion-following-motorized-camera-base 61afeb?ref=platform & ref_id=424_trending &offset=19
7. http://www.instructables.com/id/Real-Time-Face-Tracking-Robot-With-Arduino-and-Mat/?ALLSTEPS
8. http://www.instructables.com/id/PIR-Motion-Sensor-Tutorial/?ALLSTEPS
9. https://en.wikipedia.org/wiki/Viola%E2%80%93Jones object_detection_framework
10. https://in.mathworks.com/help/matlab/learn_matlab/product-description.html