Real-time Face Tracking using MATLAB and Arduino

VENKATA SASANK PAMULAPATI[1], YEKULA SUMITH ROHAN[2], VEMULA SAI KIRAN[3], SARANU SANDEEP[4], MARAM SRINIVASA RAO[5]
[1, 2,3,4,5] *Electronics and Communication Engineering, Vasireddy Venkatadri Institute of Technology*

Abstract- ***The proposed method explains the working of Viola-Jones algorithm for face tracking in real time. Face tracking in a real time environment is critical for security surveillance, video coding and Advanced Robotics. It has a potential for wide range of applications by integrating MATLAB and Arduino. The main objective of the project is to detect a human face in every frame of the video coming from a web camera and the captured image is processed using viola-jones algorithm using MATLAB and to detect the faces and send signals to the Arduino board to control the movement of the camera using two servo motors. One servo for horizontal rotation and another for vertical rotation, face shall be tracked actively and maintained in the frame. The MATLAB code may also include some of the built in functions for the effective face recognition and seamless tracking in real time environment. Applications of this concept are wide spread from video coding to advanced robotics.***

Index Terms- ***Cascading Classifiers, Face Tracking, Image Processing, Viola-Jones.***

I. INTRODUCTION

Object detection and tracking are important in many computer vision applications including activity recognition, automotive safety, and surveillance. A simple face tracking system is developed by dividing the tracking problem into three separate problems: Face detection in the frame, Initial facial features used for tracking, Face Tracking. Face detection in MATLAB can be done using many different existing algorithms. These algorithms use different mechanisms to identify the facial features. Some use edge detection methods while some use contrast separation. One of the widely used algorithms is Viola-Jones algorithm. Generally, it is used in object detection applications but, due to its capability of detecting facial features using Haar based feature filters, is extremely easy to use for the face detection applications. The reason behind using MATLAB for image processing is due to its features with inbuilt tools and support for good range of hardware like Arduino and Raspberry Pi. In this project, Arduino Uno was used due to its simplicity and compatibility with the MATLAB. Arduino hardware support package can be added to MATLAB via Add-Ons.

The aim of this paper is to emphasize the use of Viola-Jones object detection framework on human faces in real-time.

II. ALGORITHM

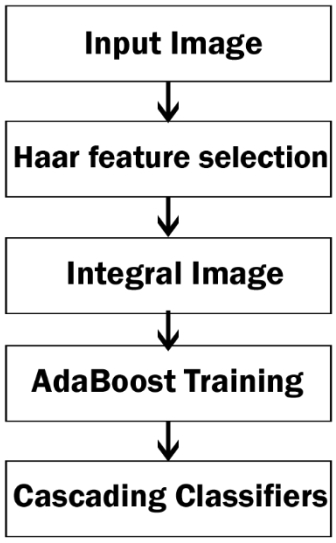
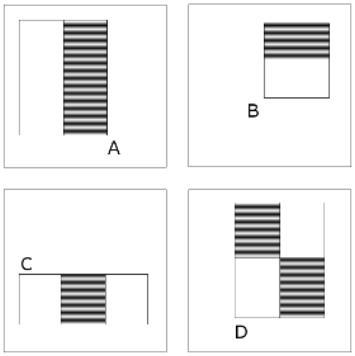


Fig.1: Algorithm flow chart

A. Haar Feature Selection

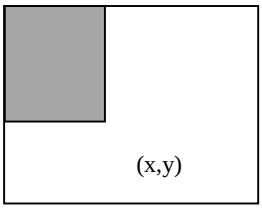
Almost every human face shares similar features and these can be filtered using pre-defined masks. These are nothing but Haar based masks. Haar-like features are digital image features that are widely used in object recognition applications. They got their name from their similarity with Haar wavelets. They were used in the first real-time face detector. Viola and Jones brought in the idea of using Haar wavelets and developed the Haar-like features. A Haar-like feature considers adjacent rectangular regions at a specific locations in a detection window, sums up the pixel

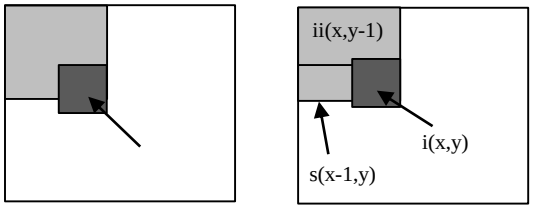
intensities in each region and calculates the difference between the sums. The difference is then used to categorize sub-sections of an image. In the detection phase of the Viola-Jones object detection framework, a window of the target size is moved over the input image, and for each sub suction of the image, the Haar-like feature is calculated. The difference is then compared to a learned threshold that separates non objects from objects. The key advantage of a Haar-like feature over most other features is its calculation speed.



B. Integral Image

The integral image computes a value at each pixel (x,y) that is the sum of the pixel values above and to the left of (x,y) inclusive. This can be quickly computed in one pass through the image.





Cumulative row sum: $s(x,y) = s(x-1,y) + i(x,y)$

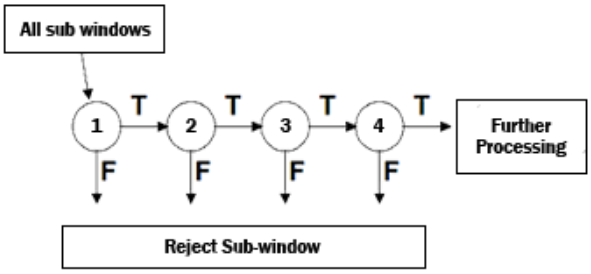
Integral Image: $ii(x,y) = ii(x,y-1) + s(x,y)$

C. AdaBoost Training(Adaptive Boost)

The integral image computes a value at each pixel (x,y) that is the sum of the pixel values above and to the left of (x,y) inclusive. AdaBoost is adaptive by iteratively reducing mis-classifications. The weights of mis-classified descriptors are adjusted for the benefit of generating a combined (weighted) application of the given w weak classifiers. The training set contains face and non-face examples with equal weight. For each round of boosting, it evaluates each rectangular filter on each example and selects the best threshold for each filter and then sets best threshold. After this, the weights are set again.

III. CASCADING CLASSIFIERS

After the training is done, the integral images are matched with the filters and cascaded using the MATLAB objects. The cascade object detector uses the Viola-Jones algorithm to detect people's faces, noses, eyes, mouth, or upper body. We can also use the Image Labeler to train a custom classifier to use with the System objects. The first cascading classifiers is the face detector of viola and jones. The requirement for this classifier was to be fast in order to be implemented on low-power CPUs, such as cameras and phones. Cascades are generally done using the cost-aware adaboost. The sensitivity threshold can be



adjusted so that there is close to 100% true positives and some false positives. After the initial algorithm, it was understood that training the cascade as a whole can be optimized, to achieve a desired true detection with minimal complexity.

D. Cascade Objects in MATLAB

- Cascading of the integral images is done using the inbuilt cascade features of MATLAB. vision.CascadeObjectDetector(); - detects the defined object from the video input.

- imaq.VideoDevice: This object enables MATLAB to interface camera to the system.
- vision.ShapeInserter: This object puts the defined shape at the position if identified Haar features.
- vision.VideoPlayer: This object displays the processing video with the identified features.
- vision.HistogramBasedTracker: This object tracks the movements in the Haar features using the histogram information.

IV. WORKING

A. MATLAB Software

In this proposed method, MATLAB uses the built in vision objects to detect the faces from a live video feed. For the first step, a serial connection to the Arduino hardware opens. Simultaneously, the program reads the video feed from the webcam or external camera. For good efficiency, step is used for extracting frames from the continuous video feed. Using the Viola-Jones detection framework, the frames extracted gets matched with the Haar-features and faces are identified. Nose features gets matched after the faces. Then the boundary boxes will be placed at the identified face locations. Further into the code, it calculates the centroid of the boundary box and checks the part of the frame it falls in. As the frame of the video will be divided into four quadrant, it calculates the distance from the center to the position of centroid of the bounding box on both axes. These are fed to the Arduino serial input. MATLAB does this process continuously until the preset time after there is no face in the frame.

B. Arduino IDE

In this project, Arduino UnoR3 with the ATMega328P micro controller is used to control the tracking motion for the camera on two servo motors (pan-motion & tilt-motion). We used Arduino platform for this model because of its simple interfacing and working with the MATLAB. Adding the hardware support package to the MATLAB through the add-ons is extremely convenient. As the process of face detection happens, serial input at the Arduino receives the distances from the center to that of the bounding box. With this input from the serial port, Arduino code is written such that it compares the input co-ordinates to the conditions in the program and operates the servo motors accordingly. This continuously runs to put the center of the bounding box on the center of the video frame.

C. Physical Connections

Components required for this project includes a web camera, Arduino Uno, a Computer with MATLAB (including Arduino hardware support packages) and Arduino IDE.

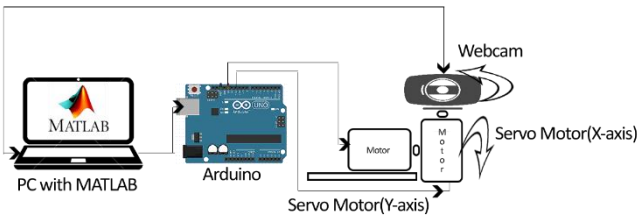


Fig.2: Block diagram

D. Tracking

Arduino actively operates the two servo motors according to the information from the MATLAB. For instance, if the identified face in the upper right corner of the frame of the video, Y-axis servo or the servo for the tilt motion moves to bring the webcam aligned to the center of the Y-axis and the pan-motion servo moves slightly towards the right so that the face is centered to the frame. This process is continuous and tracking is done.

V. CONCLUSION

Finally, the proposed system for tracking faces using Viola-Jones algorithm in MATLAB is better than the existing methods. Usage of adaptive boosting for the algorithm helps in increasing the precision and efficiency of the proposed method. This kind of tracking can be used effectively in the surveillance field. Also, Computer Vision is the major field which holds a vast scope for robotics.

VI. REFERENCES

[1] Viola, P., Jones, M.J.: Robust Real-Time Face Detection. International Journal of Computer Vision 57(2), 137–154 (2004)

[2] Lienhart, R., Maydt, J.: An Extended Set of Haar-like Features for Rapid Object Detection. IEEE Int'l Conf. Image Processing 1, 900–903 (2002)

[3] Huang, Y.C.: A hierarchical face recognition system. Master Thesis, Dept. of Inform. Eng. and Comput. Sci., Feng Chia Univ., Taichung, Taiwan (2003)

[4] Digital Image Processing, Third edition: Rafael C. Gonzalez, Richard E. Woods. Image Processing Algorithms.

[5] Book: Exploring Arduino, tools and techniques for engineering wizardry: Jeremy Blum

[6] Hoang minh Phuong "Extraction of Human Facial Features based on Haar Heature with Adaboost and Image Recognition Techniques" IEEE, 2012.

[7] M.Gopi Krishna, A. Srinivasulu "Face Detection System On AdaBoost Algorithm Using Haar Classifiers" IJMER vol.2, oct 2012