Automatic Facial Feature Point Marking & Tracking System

Karthikeya Udupa K M, ,Vinayak Dhar, Pawan Bhadana(*Assistant Professor*)

{ *karthikeyaudupa@gmail.com , vinayak_dhar@yahoo.com* }

B S Anangpuria Institute Of Technology & Management

Abstract

The ability to make computer understand the human face and its ability to express varied range of expressions is a very challenging problem. In this paper we present a technique to make the computer identify the key points of human face and to track it in real-time, the system would use Haar-Like feature [1] to detect the face in the initial frames and then track it using a tuned Continuously Adaptive Mean Shift algorithm [2]. During this phase the Haar-Like features [1] technique would be used to map other facial features like eyes, mouth and nose, these areas would be processed further to extract features and then be marked to be tracked allowing the whole face to be tracked in turn.

Introduction

Human and computer have been in co-existence as two individual entities for long but recent researches in the field of computer vision have allowed the communication between these two to reach a new horizon. We communicate with each other with more humane features like facial expressions and actions, but in the case of computers we are still restricted to keyboard and mouse.

Over the past few years various steps have been made to allow the human's to communicate easier like for example hands, various devices have been made like touch screens. One of the main part of human communication is its face. Human face can express a varied range of expressions and emotions, which if the computer understands can be very useful and save a lot of user's work. In this paper we aim at making the whole process a whole lot simpler by providing a method to map the human face and track the feature points in a live stream. The uses of this method would be various ranging from expression analysis to age & gender detection.

Methodology

The process involves various stages of image processing and calculations to allow it to reach a stage where it would be able to finally track the various points successfully.

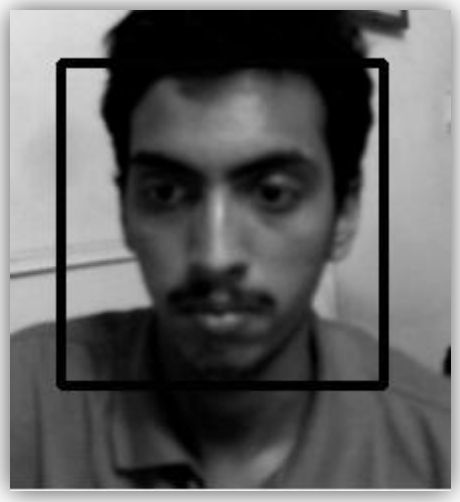
- **Acquiring And Modifying Stream**

The foremost step in this process is to get the image stream which would then be utilized to do all our processing on, A normal web camera was used and the input stream was taken and then the BGR image was converted to greyscale for processing.

- **Face Detection**

As mentioned earlier the first important step is extraction of the initial position of the face in the frame. This process has to be fast and effective so we choose a very famous algorithm devised by Paul Viola [3] and later improved by Rainer Lienhart [4]. The way this algorithm works is as follows.

- Select a data set as sample with a considerable amount of images of human faces (or any particular object in general) these images are said to be positive examples since they contain the face/object.
- Select a set of negative example which does not have a face in it, this might be anything arbitrary. Note that images are to be of the same time.
- This then is used to train classifiers which are themselves a cascade of boosted classifiers and Haar like features, i.e. several classifiers are applied to various regions of interest until either it's rejected of fully accepted. These classifiers then "boost" themselves and complex at every stage. For example AdaBoost.
- Then all these are combined into a simple structure which would be then used to test a given sample.



(First image is the image from stream converted to Greyscale and second one is after application of face detection.)

- **Tracking The Detected Face**

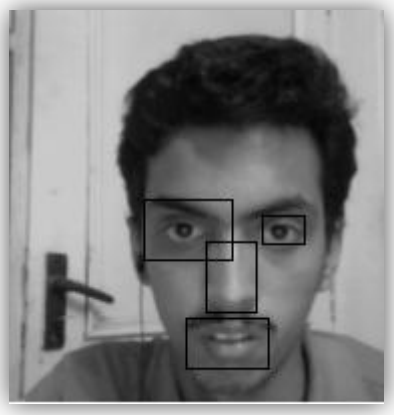
After we detect the initial location of the face what we need is a system to locate its correct location in future without having to do the whole face detection algorithm again.

This can be done by Continuously Adaptive Mean Shift algorithm [2] which is a robust iterative technique for mode of probability distribution. It is based on mean-shift algorithm, it iterates mean-shift but varies the window size, for every iteration it gives a new window size, then it converges and the whole process is applied on the new window and the centre position is located.

- **Extraction Of Various Key Parts Of The Face**

The face constitutes of various parts which has to be identified for us to process ahead. The detected face was made our ROI (Region of interest) and the rest was considered as background and removed. Out of all the various parts our face has we choose the following major parts for our purpose.

- Left Eye
- Right Eye
- Mouth
- Nose



(The image shows the marking of various features of the face.)

These are very easily visible and play a very important role in identification of human face itself. These parts are also identified using Haar Like Feature [1] classification. The eyes were individually detected since issues were with multiple detection of same eye, this was easily solved by splitting up the image into a desirable ratio depending on the head rotation angle and applying individual eye detection in the parts of the image.

- **Sub Feature Extraction**

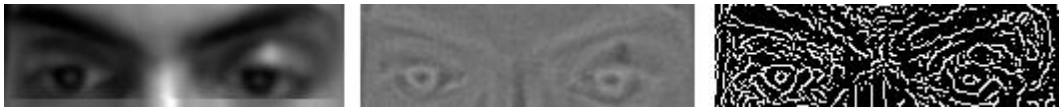
In each of the selected parts there were various points that were to be tracked, each point had to be extracted accurately and quickly for it to be tracked. With each part having a very different structure extracting each of the point in each of the parts was not possible with the same technique, so various techniques were used to do the detections on various parts.

- **Eye**

  The eye area was considered to have 4 major points that were to be tracked. First point was the cornea and the other three points where on the eyebrow with one at each corner and one in the centre.

  The cornea was detected by firstly Gaussian blur to the image then applying Laplace transform to the already cropped image of the eye, this resulted in the extraction of a predominant cornea.

  The eyebrow was detected using canny edge detection algorithm and then marked as per the initial definition.



(First image is the extracted eye area, second one is the Laplace transform to extract cornea, and third is the canny edge detection for eyebrow.)

- **Mouth**

  Mouth too had four predominant points the lip corners, the upper lip and the lower lip. These features were also marked through canny edge detection algorithm which was applied after applying a Gaussian blur.

- **Nose**

  The nose wasn't considered as much important as mouth and eyes even though it's very useful in face position estimation. Its marking had 3 points and it was done with Laplace filtering a smoothened and cropped image of nose.

- **Marking Various Other Features**

Apart from the above mentioned main parts there are a few more points that hold a very important role in face mapping. The points include the cheek area and the chin.

The cheek is an ever moving part of the face and moves almost on every expression so tracking it is very essential, the points are marked by splitting the image into two parts and then deciding on the possible area of cheek and then applying Laplace transform to it and extracting the nearest match.

- **Tracking The Extracted Points**

Once the key points are extracted the main job begins, it is to track the extracted points and make a partial face map. This process is done by Lukas-Kanade Optical Flow Algorithm [5] based point tracking. This would let the computer calculate the facial movements done by the user.

(Eye marked and ready for tracking with optical flow algorithm.)

The Lucas–Kanade method is a two-frame differential method for optical flow estimation developed by Bruce D. Lucas and Takeo Kanade. It introduces an additional term to the optical flow by assuming the flow to be constant in a local neighbourhood around the central pixel under consideration at any given time.

- **Re-initialization And Frequent Checks**

Once all the above processes have been done, it is made sure the face is inside the frame and is being tracked correctly. Since we have activated tracking function earlier it would be used whenever a check is required.

For re-initialization the whole process would have to be called, this would also be done in case a check detects the face isn't present in the frame, it would then be continuously run a face detection function and then process normally as soon as the face is detected inside the frame.

Implementation

(A fully mapped human face with all major features mapped. Green dots represent points being tracked.)

The implementation of the whole process was done in C++, a whole class was built for the process with basic methods to be called from exterior. A live stream from camera was taken as input for the process.

OpenCV Library [6] a open source computer vision library was used for the video handling including stream from the camera and doing some basic operation on the image. The library being well built provided a very firm and robust base for the application since it was very well optimized.

Conclusion

We were able to mark and track the basic features of the human face to an extend on a day to day computer with no high end resources. The end product is a class which can be imported and be used to do multiple things which ranges from emotion recognition to 3D world mapping of the head.

The system still is open for lot of additional improvements which includes increase in the detection speed and the increase in the frame rate. The initial positioning time can be reduced by self learning networks which would match quickly with the points and align the points to it more efficiently. The process still has to be improved to work on completely tilted faces and side turned faces.

Also the quality of the camera is still a very important part of the application so if that could be reduced it would make the process more useable for practical use.

References

[1] Haar Like Features *http://en.wikipedia.org/wiki/Haar-like_features.* Titled *Haar - Like Features.*

[2] Computer Vision Face Tracking For Use in a Perceptual User Interface, Intel Technology Journal, No. Q2. (1998)

[3] Viola and Jones, "Rapid object detection using boosted cascade of simple features", Computer Vision and Pattern Recognition, 2001.

[4] Lienhart, R. and Maydt, J., "An extended set of Haar-like features for rapid object detection",2002.

[5] Lucas B D and Kanade T 1981, An iterative image registration technique with an application to stereo vision. *Proceedings of Imaging understanding workshop*

[6] OpenCV : Open computer vision library. *http://sf.net/projects/opencvlibrary/* .(Intel Research Laboratories.)