A Method For Tracking of Facial Action Points Using Pyramidal Lucas Kanade Algorithm for Expression Recognition

Ashim Saha, Arundhati Das
National Institute of Technology
Agartala, India
ashim.cse@nita.ac.in
arundhati.cse@nita.ac.in

ABSTRACT: *This paper presents a systematic methodology to analyze displacement metrics to be used in recognizing facial expressions with the help of FAP or landmarks on a face particularly on the mouth, nose tip and eye and eye brows. The proposed system tries to automatically perform human face detection, feature point extraction for the further purpose of facial expression recognition stably and robustly over live webcam input of real time sequences of frames and yield promising performance in low-resolution video sequences captured in real-world environments. Thus a sequential three-stage approach is taken, firstly Face Detection, secondly Feature Extraction and finally tracking of the extracted features. In real time, a human face is detected using an improved Haar based algorithm. Then facial features are extracted using a combination of appearance and geometrical approach depending on the ROI extracted by Haar based algorithm. The tracking is done with the help of pyramidal Lucas Kanade algorithm. Several experiments conducted under relatively non-controlled conditions demonstrate the accuracy and robustness of the approach.*

Keywords: FAP (Facial Action Points), Face Detection, Feature Extraction, Facial Expression, ROI (Region Of Interest)

Received: 10 June 2017, Revised 27 July 2017, Accepted 2 August 2017

1. Introduction

A human can detect face and recognize facial expression without any effort, but for a machine and in computer vision it is very difficult [2]. The task of facial feature point detection is intuitive to us and yet the range of tools available to our visual systems is somewhat basic. Our brains don't explicitly compute image transforms or edge maps; our monochromatic dark-adapted vision doesn't use any color information at all [10]. But still we outperform even the best computer vision systems. The visual system of humans allows them to assimilate information from their surroundings. The human visual system (HVS) includes mainly the eye and a portion of the brain as functional parts. The brain is capable of doing the entire complex image processing, while the eye is the biological representative of a camera. The experts in the field of Computer Vision are taking advantage of the HVS model to deal with processes behind the biology and psychology of human facial behaviour [10]. The coming generation computer systems are thought to be sharp enough so that they interact with common users in such a way that echoes face to face communication. The implied and non-verbal gestures expressed through head posture and head body, gesture by hand and facial expressions are remarkably reliable face to face communication means for determining the spoken message in a clear way [23].Facial expressions are specially considered to be one of the strongest and instant means for humans to communicate their true emotions, intention and thinking to each other also this is why much effort has been devoted to their study by cognitive scientists and recently computer vision researchers [3, 23].

2. Related Works

Regarding recognition of different facial muscle behaviors or expressions automatically from static images or video sequences many proposals have been detailed [23]. Almost all the facial expression recognition research is limited to the six primary emotions, i.e. anger, disgust, fear, happiness, sadness, surprise. Displaying of all these six primary emotions are concluded to be universal [1, 2]. Recently, a large number of approaches concentrated in detecting a set of facial muscle movements known as Facial Action Points (FAP), as their combinations may constitute effectively any facial expression. The key idea of detecting FAPs is to first detect and extract features on the face which will contribute a lot of changes in its position under different expressions made by the face. The methods to extract feature points can be geometric, appearance-based and holistic. The geometric feature-based approaches depend on relative positions of facial components which are eyebrows, eyes, nose, mouth etc. Geometric feature based methods [30] takes out information about shapes and locations of facial components to form a feature vector. Appearance based features take out the appearance change of the face (e.g. wrinkles, bulges, furrows) by filters employed to the whole face or a particular portion of the face. Appearance based methods includes choosing appearance information using Gabor wavelets [31]or by using Haar-like features [9]. Lately, PLBP, an approach depending on pyramidal representation of local binary pattern (LBP) has been deployed to find the information related to texture of the face [32]. Other methods can be based on optical flow [33].The very first segment regarding facial expression recognition is begun by Darwin [1]. Afterwards Ekman explained and established six primary emotions which are declared to be universally unique with distinct facial expressions [2]. These six primary emotions are: anger, disgust, fear, happiness, sadness, and surprise. Most of the vision based facial expression studies rely on Ekman’s assumptions about the universal categories of emotions. The Facial Action Coding System (FACS) is a system based on human observations which is developed to ease objective measurement of subtle changes in facial appearance caused by contractions of the facial muscles and each such useful muscle changes is encoded as action units and numbers following a sequence [3].

3. Proposed Methodology

The algorithm is mainly consists of 3 modules. Those are face detection, ROIs localization and extraction and finally tracking of the ROIs as feature points or FAPs.

3.1 Algorithmic Steps of the Proposed Face Detection, ROIs Localization and Tracking Method

Face Detection, ROIs localization and Tracking Algorithm

Step 1: Start

Step 2: Declare an object cap of the inbuilt class VideoCapture

Step 3: Using cap, open inbuilt webcam by assigning the index as 0

Step 4: Read and save frame in a Mat object temp_frame

Step 5: Apply flip function on temp_frame

Step 6: Apply Haar cascade and detect face

Step 7: On the detected face region geometrically find the region having mouth and apply Haar cascade and detect mouth

Step 8: On the detected face region geometrically find the area having the left eye and right eye and apply Haar cascade respectively and detect right eye and left eye respectively

Step 9: On the mouth thus detected, geometrically lip corner points are detected and saved as one set of feature point

Step 10: On the left and right eyes next set of feature points are assigned on eye center and upper and lower eye lids

Step 11: Using the eye location eyebrow location is found geometrically and third set of feature points are placed on the eye brows.

Step 12: On the detected face Haar cascade for nose is applied and location of the nose is found and nose tip is saved as another feature point

Step 13: Next, Pyramidal Lukas Kanade optical flow method is applied to track the motion of the feature points

Step 14: Stop

3.2 Face Detection

Viola and Jones face detector [9] is used to extract the face region for all the frames. This is available from the Opencv library [21]. In our algorithm we have used an improved version of our previous paper [21]. Moreover, we have used improvements over that method and the outputs are now more stable and fast. The Haar Functions to search the region of interests (ROIs) are applied using a geometrical approach in the detected face. The positions of the facial features are exploited to get a more stable detection of ROIs. In the method used here, the face is divided mainly into three regions i.e. eyes portion, nose portion, mouth portion and the Haar algorithm is applied only to the regions where the likelihood of obtaining the features are more.

3.3 Features Localization and Extraction

Extracting facial features from the face captured from frame sequences in the live video is essential for successful facial expression recognition. It is the most crucial part of the entire system. Among the mainly used methods, in geometric based methods the chosen geometric features are sensitive to noise and tracking errors, also appearance based features taken in appearance based methods can extract micro patterns in the texture of skin that are crucial for facial expression recognition. Although appearance features do not generalize across subjects as they convert and store only particular appearance information. In the existing methods [29], the user has to manually mark the key feature points in the first frame. The method proposed here is not using manual marking of the feature points in a certain region. A combination of geometrical and appearance based method is used along with optical flow method [25] to track the landmarks. At first, inside the detected face, the ROIs are localized with the simple knowledge of positions of the required features i.e. eyes are localized in the upper most portion of the face, nose in the middle and mouth in the lower most portion of the face. Then, in those respective portions, the Haar Based Algorithm [9] for eyes detection, nose detection and mouth detection is applied respectively. The application of all the Haar based algorithms in all over the face is avoided and thus the detection applied only to a particular portion where probability of finding the features is most. This has added more robustness and stability to the process and particularly for real time environment a fast detection and localization is achieved.

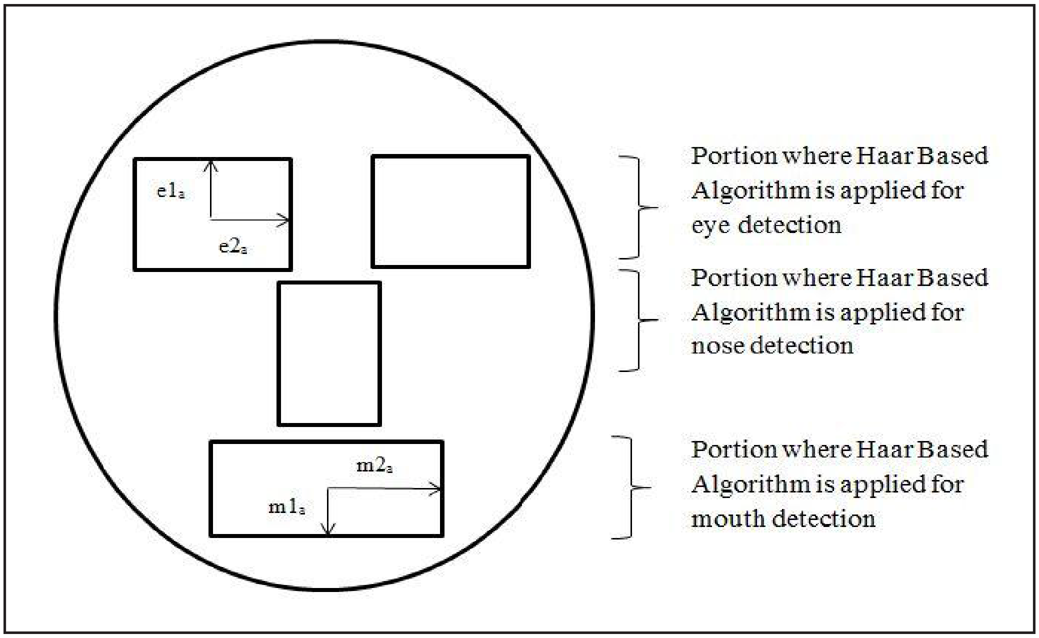

$e1_a$
$e2_a$
$m2_a$
$m1_a$
Portion where Haar Based Algorithm is applied for eye detection
Portion where Haar Based Algorithm is applied for nose detection
Portion where Haar Based Algorithm is applied for mouth detection

Figure 1. Face ROI Localization-detection and Metrics used for FAP localization

With the help of neutral face images available in JAFFE database, few metrics are measured to have an approximate value from the center of the detected features to the end point features. For example, the four FAPs on the eyes are calculated. Similarly, four FAPs on the mouth are calculated. One FAP on the nose is give as a center point of the nose. For eyes, $e1_a$, an average value is calculated which is a measure of center point of the eye and the upper and lower point of the eye lids; $e2_a$ is the average value of eye center and left or right ends of the eye. Similarly, for mouth, upper and lower FAPs of the lips are taken at a distance of $m1_a$ and left and right FAPs are taken at a distance of $m2_a$ from the center of the mouth. After localization of the FAPs in the first frame of the live video sequence, sparse optical flow method is applied to track those points in the subsequent frames.

3.4 Tracking using Pyramidal Lucas Kanade Algorithm

To produce dense results, the Lucas Kanade algorithm [25] was originally introduced. When this method is applied to a smaller set of points of interest in the input frame it produces sparse results which are very efficient in some applications. The Lukas Kanade algorithm is employed on a sparse context as it is dependent of the local information extracted from small window around the point of interest which is needed to track. This adds a drawback as in large motions of abrupt change cannot retain the point of interest as the point of interest goes outer side of the local window. Pyramidal Lukas Kanade resolves this drawback [8]. In this algorithm, tracking is done from the highest level of an image pyramid (lowest detail) which continues to finer detail in the lower level. Thus sudden motion is tracked over image pyramids. In the proposed method, from the implementation of feature tracking, Pyramidal Lucas Kanade algorithm is proved to be a robust optical flow algorithm. Let us have a point $u = (u_x, u_y)$ on the first frame of the sequences of live video, the aim of tracking points with particular feature is finding the next location $v = u + d$ in next frame J of the sequences of the live video such as $I(u)$ and $J(v)$ are alike. Here, d is the displacement vector. Due to the aperture problem, it is required to establish the notion of alike in a 2D neighborhood sense. Let's assume ω_x and ω_y are two integers. Then d is taken as vector that minimizes the following residual function

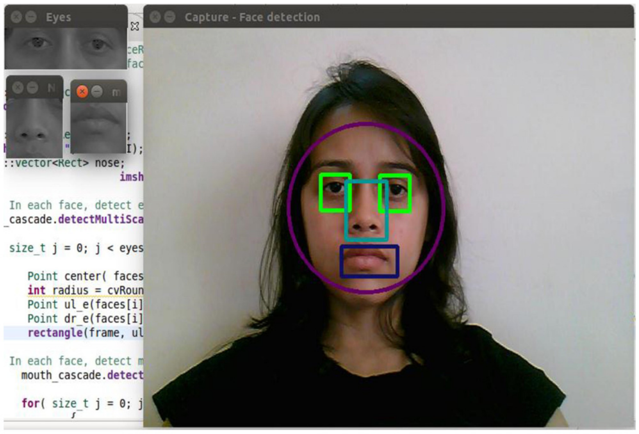
$$\in(d) = \in(d_x, d_y) = \sum_{x=u_x-\omega_x}^{u_z+\omega_x} \sum_{y=u_y-\omega_y}^{u_y+\omega_y} \left(I(x, y) - J(x + d_x, y + d_y)\right)^2$$

The general values for ω_x and ω_y are 2, 3, 4, 5, 6, 7 pixels. Following the definition, the similarity function is measured on one frame of image neighborhood of size $(2\omega_x + 1) \times (2\omega_y + 1)$. This neighborhood is termed as integration window.

4. Implementation & Result

The proposed algorithm is implemented using OpenCV library functions and written in C++language. IDE used is Eclipse 3.8 and the platform is Ubuntu 14.04 LTS 32 bit.

The detected face along with localized ROIs is shown in figure 1. The detection is robust and works properly in low illumination and partially occluded conditions as well. Due to applying Haar algorithm only to those portion where likelihood of obtaining the particular feature is more, the detection and localization is way stable.



```
                                  ceR
                                  fac
  c
                                    e
                                  I);
::vector<Rect> nose;
                                 imsh

 In each face, detect e
_cascade.detectMultiSca

 size_t j = 0; j < eyes

    Point center( faces
    int radius = cvRoun
    Point ul_e(faces[i]
    Point dr_e(faces[i]
    rectangle(frame, ul

 In each face, detect m
   mouth_cascade.detect

   for( size_t j = 0; j
```

Figure 2. Face Detection and ROI localization

In figure 3, 4, 5 and 6, the tracked FAPs with the help of Pyramidal Lucas Kanade are shown. In figure 3, the neutral face in the first frame of the live video sequence is detected, then ROIs are localized according to the method explained in section 3.3 and the FAPs are tagged and saved as feature points. In the next frame where expression of the subject is changed the located FAPs are tracked while the particular saved feature is moved from its original position. Figure 4, 5 and 6 demonstrate the tracked FAPs

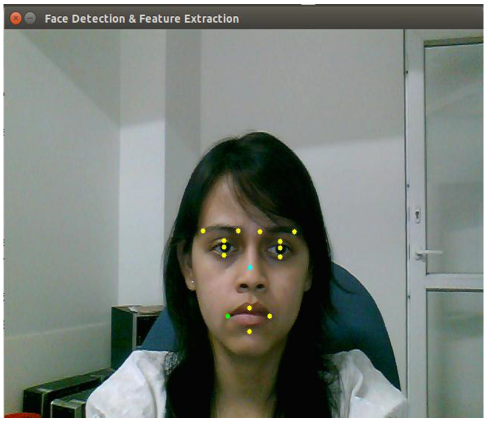


Figure 3. Extracting feature in the first frame of the live video right after ROI localization

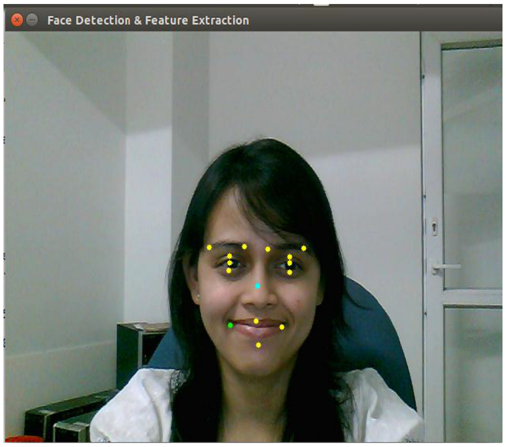


Figure 4. Tracked FAPs in a happy face

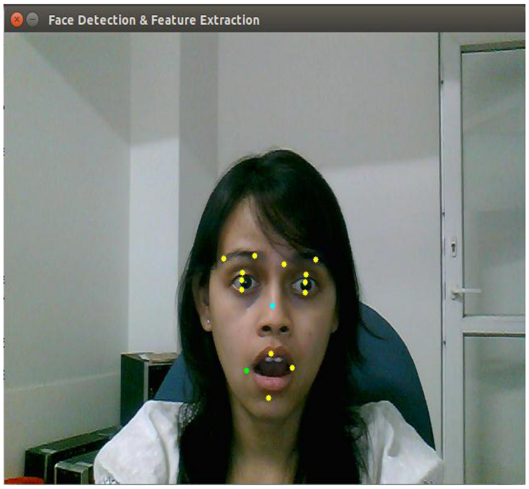


Figure 5. Tracked FAPs in a surprised face

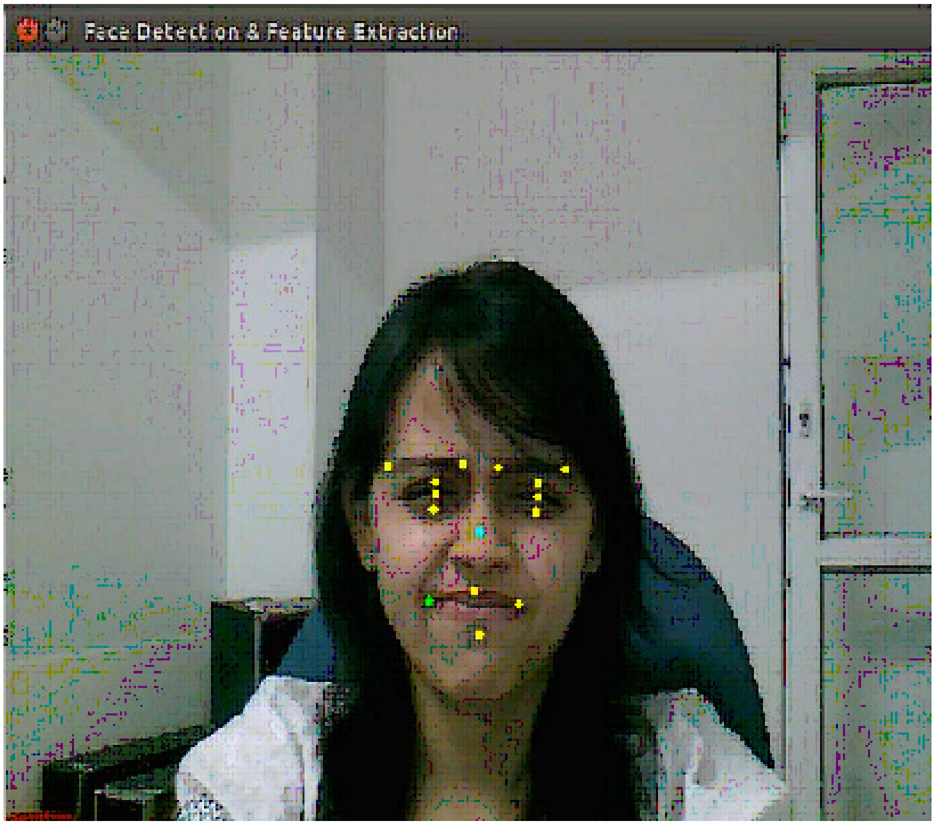


Figure 6. Tracked FAPs in a disgust face

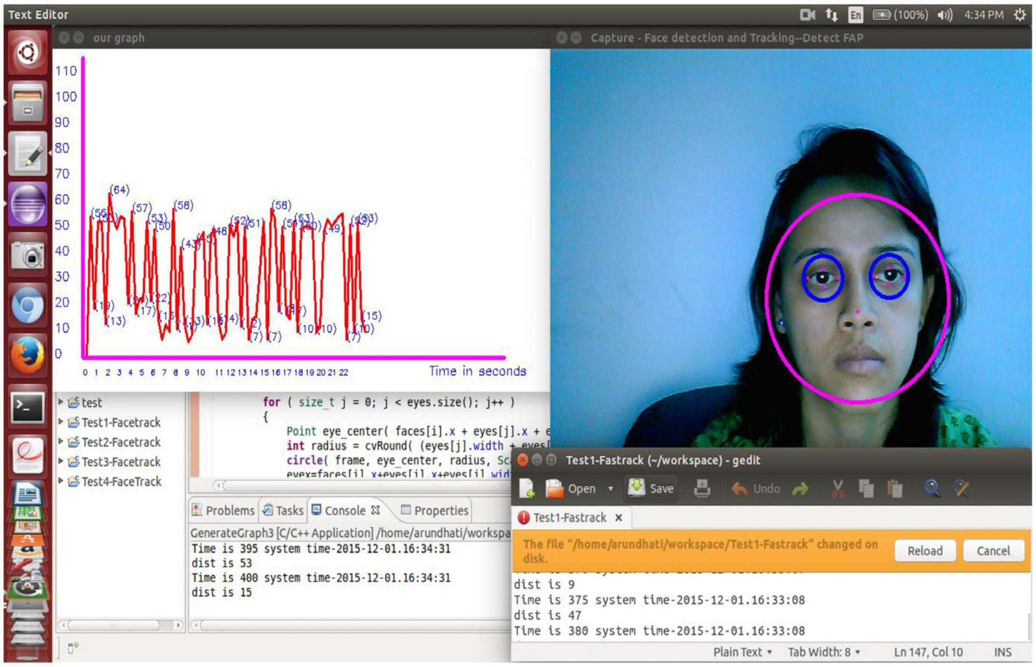
The extracted features in the localized ROIs are saved as feature points. These feature points called FAPs will further be utilized for facial expression recognition. As seen in the figure 3, the FAPs in the mouth region show satisfactory results of tracking in the live testing video. Also, the FAPs in the eyebrow region and in both of the eyes are being tracked well under different facial expressions of the face from neutral to happy, surprise or disgust expressions. As the tracking of the features which help determining one distinct expression from another is done pretty well, the feature points or FAPs may help in classifying different expressions under real time environment.

Three different Euclidian distances(ED) between inner eye corner-nose tip, outer eye corner-nose tip and between eye center and nose tip are measured successfully as shown in figure 7.The Euclidean distance between points p and q is the length of the line segment connecting them. In Cartesian coordinates, if $p= (p_1, p_2,..., p_n)$ and $q = (q_1, q_2,..., q_n)$ are two points in Euclidean n-space, then the distance*d* from *p* to *q*, or from *q* to *p* is given by the Pythagorean formula:

$$d(p,q) = d(q,p) = \sqrt{\sum_{i=1}^{n}(q_i - p_i)^2}$$

Whether the face moves right, left, up or down is tracked and displayed in a message in the console output accordingly. All the data thus extracted from the facial features are directed to a simple log file (text file) for further analysis as shown in figure 8. In addition, a graph as shown in figure 7, depicting Euclidian distance between inner eye corner and nose tip vs. the system time possessing the respective distance has been generated for the analysis of extracted information from the facial changes. The extracted data from the graph may help in further classification of facial expression or facial changes.

From the implementation we get that the tracking rate of our proposed methodology is better than the tracking rate mentioned by Mu Chun Su et al. in [4]. The tracking rate of testing set is 86.36% while in our method it is 88%-89%. Similarly, for lower face part it is 91% in our proposed methodology which is better than 85.23% in [4].



```
        for ( size_t j = 0; j < eyes.size(); j++ )
        {
            Point eye_center( faces[i].x + eyes[j].x + e
            int radius = cvRound( (eyes[j].width + eyes[
            circle( frame, eye_center, radius, Sc
            eyex=faces[i].x+eyes[i].x+eyes[i].wid
```

GenerateGraph3 [C/C++ Application] /home/arundhati/workspa

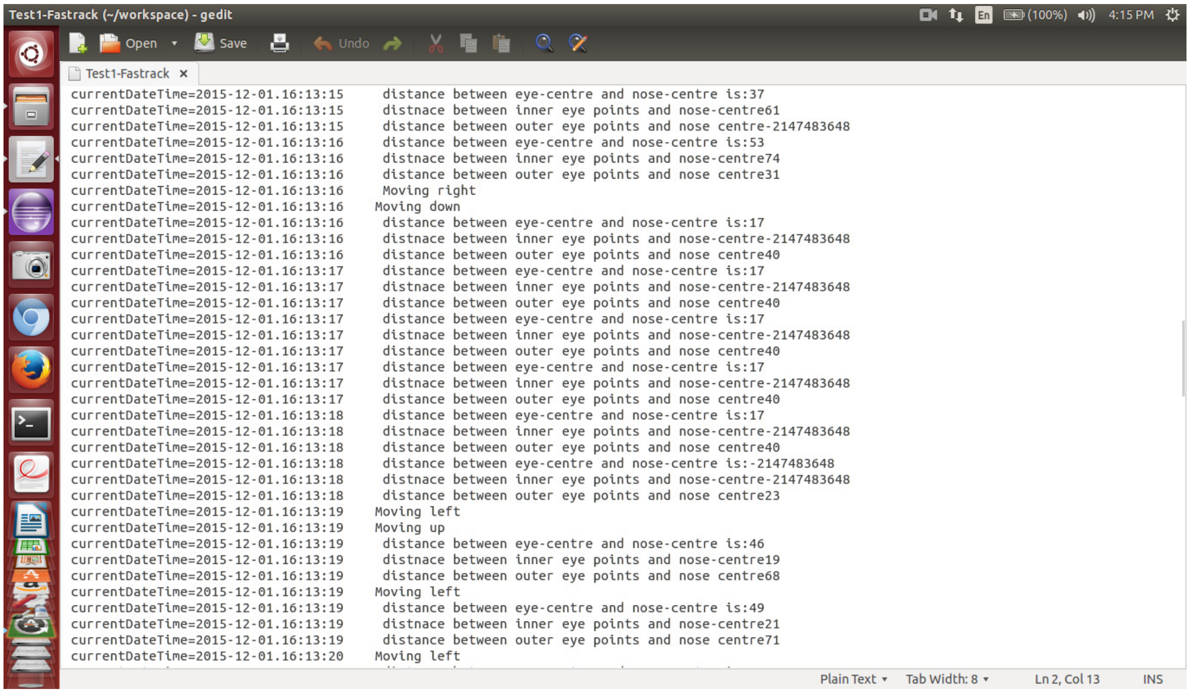
```
Time is 395 system time-2015-12-01.16:34:31
dist is 53
Time is 400 system time-2015-12-01.16:34:31
dist is 15
```

Test1-Fastrack (~/workspace) - gedit

The file "/home/arundhati/workspace/Test1-Fastrack" changed on disk.

```
dist is 9
Time is 375 system time-2015-12-01.16:33:08
dist is 47
Time is 380 system time-2015-12-01.16:33:08
```

Figure 7. Generated graph for analysis of distance metrics calculated from FAPs in the eyes



```
currentDateTime=2015-12-01.16:13:15     distance between eye-centre and nose-centre is:37
currentDateTime=2015-12-01.16:13:15     distnace between inner eye points and nose-centre61
currentDateTime=2015-12-01.16:13:15     distance between outer eye points and nose centre-2147483648
currentDateTime=2015-12-01.16:13:16     distance between eye-centre and nose-centre is:53
currentDateTime=2015-12-01.16:13:16     distnace between inner eye points and nose-centre74
currentDateTime=2015-12-01.16:13:16     distance between outer eye points and nose centre31
currentDateTime=2015-12-01.16:13:16     Moving right
currentDateTime=2015-12-01.16:13:16    Moving down
currentDateTime=2015-12-01.16:13:16     distance between eye-centre and nose-centre is:17
currentDateTime=2015-12-01.16:13:16     distnace between inner eye points and nose-centre-2147483648
currentDateTime=2015-12-01.16:13:16     distance between outer eye points and nose centre40
currentDateTime=2015-12-01.16:13:17     distance between eye-centre and nose-centre is:17
currentDateTime=2015-12-01.16:13:17     distnace between inner eye points and nose-centre-2147483648
currentDateTime=2015-12-01.16:13:17     distance between outer eye points and nose centre40
currentDateTime=2015-12-01.16:13:17     distance between eye-centre and nose-centre is:17
currentDateTime=2015-12-01.16:13:17     distnace between inner eye points and nose-centre-2147483648
currentDateTime=2015-12-01.16:13:17     distance between outer eye points and nose centre40
currentDateTime=2015-12-01.16:13:17     distance between eye-centre and nose-centre is:17
currentDateTime=2015-12-01.16:13:17     distnace between inner eye points and nose-centre-2147483648
currentDateTime=2015-12-01.16:13:17     distance between outer eye points and nose centre40
currentDateTime=2015-12-01.16:13:18     distance between eye-centre and nose-centre is:17
currentDateTime=2015-12-01.16:13:18     distnace between inner eye points and nose-centre-2147483648
currentDateTime=2015-12-01.16:13:18     distance between outer eye points and nose centre40
currentDateTime=2015-12-01.16:13:18     distance between eye-centre and nose-centre is:-2147483648
currentDateTime=2015-12-01.16:13:18     distnace between inner eye points and nose-centre-2147483648
currentDateTime=2015-12-01.16:13:18     distance between outer eye points and nose centre23
currentDateTime=2015-12-01.16:13:19    Moving left
currentDateTime=2015-12-01.16:13:19    Moving up
currentDateTime=2015-12-01.16:13:19     distance between eye-centre and nose-centre is:46
currentDateTime=2015-12-01.16:13:19     distnace between inner eye points and nose-centre19
currentDateTime=2015-12-01.16:13:19     distance between outer eye points and nose centre68
currentDateTime=2015-12-01.16:13:19    Moving left
currentDateTime=2015-12-01.16:13:19     distance between eye-centre and nose-centre is:49
currentDateTime=2015-12-01.16:13:19     distnace between inner eye points and nose-centre21
currentDateTime=2015-12-01.16:13:19     distance between outer eye points and nose centre71
currentDateTime=2015-12-01.16:13:20    Moving left
```

Figure 8. Data extracted and stored in a log file for future analysis

5. Conclusions and Future Work

The primary focus of this proposed system is to detect and track face(s) in real time environment and in a robust way which has been successfully achieved. The detection and tracking is performing pretty well for small occlusions, but for big occlusions, it can be improved for better results. Also, tagging of the facial action points (FAP) in near the both eye lids has been achieved as well as on the nose tip one FAP is tagged which is a non-deformable point irrespective of the any facial muscle activations.

Detection &Tracking done in following Regions	Detection Rate in Laboratory Environment	Tracking Rate in Laboratory environment
Eye Brow	94%	89%
Left Eye	95%	88%
Right Eye	94%	89%
Nose	98%	96%
Mouth	97%	91%

Table 1. Detection and Tracking rate of FAPs

The main work of the method is to recognize the AUs (action units) which constitute the basic expressions like anger, happiness, disgust, fear, sadness, surprise in a live video fed through the webcam connected in a PC. Several methods have been tried to extract features and are being compared to find the best method to proceed further. The method of extracting features with the help of the combination of both appearance and geometrical methods is found to be satisfactory as well as the tracking is accurate and robust. Tagging the region of interests with FAPs, measuring displacements of those points has been achieved partially. We are yet to achieve the final goal, i.e. to classify the facial expression to determine the emotional state of a person. For that more FAPs may be extracted on the forehead region, nose wrinkle region(AU 9), chin region (AU17) as well as on the cheek region (AU6) as they contribute to have more accuracy for determining facial expression [3].

6. Acknowledgement

This research work has been carried out in National Institute of Technology, Agartala, India. The authors thank the team of JAFFE database for permitting to use the databases in this research work.

References:

[1] Darwin, C. (1965). The Expression of Emotions in Man and Animals. University of Chicago Press.

[2] Ekman, P. (1982). Emotion in the Human faces. Cambridge University Press.

[3] Ekman, P., Friesen, W.V. (1978). Facial Action Coding System (FACS). Consulting Psychologists Press.

[4] Su Mu-Chun, Hsieh Yi-Jwuand Huanga De-Yuan. (2007). Simple Approach to Facial Expression Recognition. *In:* Proceedings of the 2007 WSEAS International Conference on Computer Engineering and Applications, Gold Coast, Australia.

[5] Freund, Youv. (2009). A more robust boosting algorithm. *In:* Computer Science and Engineering, UCSD, p 71–78.

[6] Mehrabian, A. (1968). Communication without words. *In Psychology Today*, 7. p. 767–774.

[7] Kaur, M., Vashisht, R., Neeru, N. (2010). Recognition of Facial Expression using Principal Component Analysis and Singular Value Decomposition*. In PhD thesis*, University di Roma La Sapienza.

[8] Bouguet, Jean-Yves. (2000). Pyramidal Implementation of the Lucas Kanade Feature Tracker Description of the algorithm.

[9] Viola, P., Jones, M. (2001). Robust Real-time Object Detection. *In:* 2nd international workshop on statistical and computational theories of vision - modeling, learning, computing, and sampling, p. 1-25.

[10] Gonzalez, R., C., Woods, R., E (2008). *Digital Image Processing*, Third Edition.

[11] Shin, Jongju, Kim, Daijin. (2014). Hybrid Approach for Facial Feature Detection and Tracking under Occlusion. *IEEE Signal Processing Letters*, p . 403–410.

[12] Papageorgiou, C., Oren, M., Poggio, T. (1998). A general framework for object detection. *In:* International Conference on Computer Vision, 21, p. 555- 562.

[13] Chibelushi, C., C., Bourel, F. (2002). Facial Expression Recognition: A Brief Tutorial Overview. *In:* Proceedings of the 2nd Annual International Conference on Mobile Computing and Networking ACM, New York, p 155–163.

[14] Craw, I., Ellis, H., Lishman, J. R. (1987). Automatic extraction of face features. *Pattern Recognition Lett,* 183-187.

[15] Valentin, D., Abdi, H., Otoole, A.J. (1994). Connectionist models of face processing, A survey. *Pattern Recognition Lett*, 1209-1230.

[16] Valstar, Michel, F., Pantic, Maja. (2012). Fully Automatic Recognition of the Temporal Phases of Facial Actions. *IEEE Transaction on Systems, Man and Cybernetics* 42 (1).

[17] Froba, B., Ernst, A. (2004). Face detection with the modified census transform. *In:* Sixth IEEE International Conference on Automatic Face and Gesture Recognition, p. 91-96.

[18] Valstar, Michel, F., Pantic, Maja. (2006). Fully Automatic Facial Action Unit Detection and Temporal Analysis. *In:* IEEE Computer Vision and Pattern Recognition Workshop, CVPRW'06.

[19] Zhang, Zhengyou. (1998). Feature-Based Facial Expression Recognition: Sensitivity Analysis and Experiments With a Multi-Layer Perceptron. *International Journal of Pattern Recognition and Artificial Intelligence*.

[20] Lienhart, Maydt (2002). An Extended set of Haar-like Features for Rapid Object Detection. *In:* IEEE , p. 1522-4880.

[21] OpenCV Library, http://www.opencv.org

[22] Facial Expression Analysis Article, http://www.scholarpedia.org/article/Facialexpressionanalysis

[23] Filareti, T., Sotiris, M (2009). Real time 2D+3D facial action and expression recognition. *Journal of Pattern Recognition, Pattern Recognition,* 43. 1763-1775.

[24] Pantic, Maja, Leon, J. M (2000). Automatic Analysis of Facial Expression: The state of the Art. *IEEE Transaction on Pattern Analysis and Machine Intelligence,* 22 (12).

[25] Lucas, Bruce, D., Kanade, Takeo. (1981). An Iterative Image Registration Technique with an Application to Stereo Vision. *In:* Proceedings of Imaging Understanding Workshop, p. 121-130.

[26] Lewis, Michael, Jeannette, M., Barrett, Lisa, F. (2008). *Handbook of Emotions*. Third Edition.

[27] Kotsia, Irene, Pitas, Ioannis. (2007). Facial Expression Recognition in Image Sequences Using Geometric Deformation Features and Support Vector Machines. *IEEE Transaction On Image Processing,* 16 (1).

[28] Das, Arundhati, P., Mameeta, Saha, Ashim. (2015). Real-Time Robust Face Detection and Tracking using extended Haar functions and improved Boosting algorithm. *In: P*roceedings of ICGCIoT 2015, IEEE.

[29] Jeffrey, F., Cohn, Adena ,J., Zlochower, James, J., Lien, Kanade, Takeo. (1998). Feature-Point Tracking by Optical Flow Discriminates Subtle Differences in Facial Expression.

[30] Pantic, M., Patras (2006). Dynamics of Facial Expression: Recognition of Facial Actions and Their Temporal Segments From Face Profile Image Sequences. *In IEEE transactions on Systems, Man, and Cybernetics—PART B: CYBERNETICS,* 36 (2).

[31] Littlewort., et al. (2006). Dynamics of facial expression extracted automatically from video. *Image and Vision Computing* 24. 615–625.

[32] Khan, Rizwan., Ahmed, Meyer., Alexandre, Konik., Hubert, Bouakaz Saïda (2013). Framework for reliable, real-time facial expression recognition for low resolution images. *Pattern Recognition Letters,* 34. 1159–1168.

[33] Hsieh, Chao-Kuei., Lai, Shang-Hong., Chen, Yung-Chang (2010). An Optical Flow-Based Approach to Robust Face Recognition Under Expression Variations. *IEEE Transactions on Image Processing,* 19 (1).