ISSN: 0976-9102 (ONLINE)
DOI: 10.21917/ijivp.2018.0271

SELECTED SINGLE FACE TRACKING IN TECHNICALLY CHALLENGING DIFFERENT BACKGROUND VIDEO SEQUENCES USING COMBINED FEATURES

S. Ranganatha[1], Y.P. Gowramma[2], G.N. Karthik[3] and A.S. Sharan[4]

[1,4]*Department of Computer Science and Engineering, Government Engineering College, Hassan, India*
[2]*Department of Computer Science and Engineering, Kalpataru Institute of Technology, India*
[3]*Department of Computer Science and Engineering, Rajeev Institute of Technology, India*

Abstract

The commonly identified limitations of video face trackers are, the inability to track human face in different background video sequences with the conditions like occlusion, low quality, abrupt motions and failing to track single face when it contain multiple faces. In this paper, we propose a novel algorithm to track human face in different background video sequences with the conditions listed above. The proposed algorithm describes an improved KLT tracker. We collect Eigen, FAST as well as HOG features and combine them together. The combined features are given to the tracker to track the face. The algorithm being proposed is tested on challenging datasets videos and measured for performance using the standard metrics.

Keywords:

Track Human Face, Different Background, Video Sequences, KLT, Combined Features

1. INTRODUCTION

A lot of advancement in technology is leading to go deeper into the current hot topics of research fields and image processing is one of them. Video processing is part of image processing, which processes the frames/images. Face tracking is an interesting research area in video processing that tracks face(s) under various technical challenges [1], [2]. In face tracking, face is the region of interest (ROI) and face detection is the first activity to take place; and in order to detect it perfectly, we need a robust detector.

An important approach in the pursuit of face detection is Viola Jones algorithm [3], [4]. The main drawback of this approach is, it is trained to detect only the frontal posed faces; and it cannot detect the faces that are tilted, occluded and having various expressions. Hence, the detector needs to be trained with various images of different subjects. But, even after proper training, the algorithm fails to detect the faces under conditions like illumination. The technical problems of face detection being discussed cannot be solved easily using the detectors currently available. But, the proposed algorithm is capable of solving such problems robustly.

Human eye is the best detector compared to machine detectors. Hence, instead of using an available face detector, our proposed algorithm allows the user to draw the ROI location of a human face to track it efficiently. Feature selection is the second activity that follows after locating the ROI. As the features selected directly affect both space and time complexities, we need to consider few searching features. Face tracking is the third and last activity that searches for the known features in the remaining frames of the video sequence.

Video sequences will be having different backgrounds if they are captured under the conditions like, 1. Static sensor and moving face(s), 2. Moving sensor and static face(s), and 3. Moving sensor and moving face(s). Most of the present algorithms fail to track the face in all the three different background conditions. But, the proposed algorithm can track any selected single face in different background video sequences.

The forthcoming sections of this paper are arranged as follows. Section 2 houses the related works that are referred. Section 3 includes design and implementation of our proposed algorithm. Section 4 i.e. experimental evaluation consist of datasets, performance metrics, results and analysis and time analysis as sub-sections. Finally, the paper halts with conclusion and future work in section 5.

2. LITERATURE REVIEW

Feature based trackers extract the elements of interest from ROI; Eigen [5], FAST (Features from Accelerated Segment Test) [6] and HOG (Histograms of Oriented Gradients) [7] features are few among them. Every matrix has n unique characteristic elements called Eigen values. Shi and Tomasi [5] have suggested Eigen points as relevant feature values for tracking. These points will remain static during rotation and translation transformations; and dynamic under projective condition. Rosten et al. [6] have presented the FAST feature detector. It allows fast on-line activity of the tracking setup. The time (ms) taken by FAST to perform feature detection is sufficiently smaller compared to SUSAN [8] and Harris [9] detectors. Two important approaches for human face detection are, sub-window and part based. Sub-window based approaches can be formulated by fusing different types of features, such as HOG [7]. HOG features are easily trainable; and image information such as edge and texture can be detected effortlessly.

Comaniciu and Meer [10] have proposed a mean shift [11], [12] based nonparametric method for analysing complex feature space and delineated unusually formed clusters in it. Mean shift discovers the maxima of a density function, the familiarly known mode seeking technique. It can be used for object tracking; and the application sphere covers cluster analysis in image processing and computer vision. But, improper window size can force the modes to be federated. By incorporating suitable modifications to mean shift, Bradski [13] proposed an algorithm called CAMSHIFT. Color based approaches make use of this algorithm for visual tracking. But, this algorithm fails i). When the objects are having multiple hues, and ii). While updating the ROI information to the next frames of the video sequence. The contributions of kanade et al. [5], [14], [15] have ended in an algorithm called KLT. It is one of the robust point tracking algorithm available till date. But, it has the issues of window size; i.e. small window is susceptible to noise and may miss quick motion. The solutions to these problems are addressed in the proposed algorithm.

Ranganatha and Gowramma [16] have proposed a novel face tracking algorithm by fusing corner detection algorithm [9], centroid [17] and KLT. Their algorithm is able to track single face, only in the videos of static sensor and moving face(s) category. Based on CAMSHIFT, Ranganatha and Gowramma [18] have proposed an algorithm for rapid face tracking. Again, this algorithm is capable of tracking moving single face in the videos which are captured using a static camera. Ranganatha and Gowramma have proposed three efficient algorithms [19], [20], [21] for tracking multiple faces in all (three) categories of different background video sequences. These algorithms are developed to detect and track all the faces in the input video sequence; but, single face cannot be selected and tracked.

3. DESIGN AND IMPLEMENTATION

The proposed algorithm deals with wider range of features to extract the facial region for tracking, throughout the frames of the video. The implementation of the proposed algorithm for selected face tracking in the input video sequence is divided into 3 parts as follows:

- Pre-processing
- Feature extraction
- Face tracking

3.1 PRE-PROCESSING

The detailed procedure of face selection, RGB to greyscale conversion, greyscale to binary conversion and region borders extraction is described in the following sub-sections.

3.1.1 Face Selection:

The face selection begins by browsing the video that consist of the face to be tracked. After reading the video file, extract the first frame from it; and convert it into greyscale.

After conversion, allow user to select the face in the frame that has to be tracked further. Once the user selects the required face region, it becomes our region of interest (ROI).

The ROI being extracted is used for feature extraction. The user selection of face region in the frame is as shown in Fig.1.

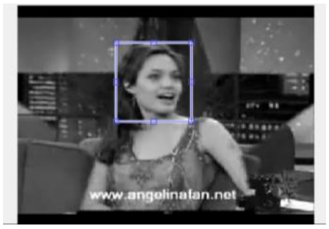

Fig.1. User selected face that needs to be tracked

3.1.2 Converting RGB Frame to Greyscale:

The frame in which we are selecting the face region is converted from RGB color format to greyscale format before it is selected for feature extraction.

Convert the extracted frame which can be in RGB, binary or YC_bC_r color formats to greyscale color scheme. The grey image can be obtained from RGB by using Eq.(1).

$$C_{Grey} = 0.21C_{Red} + 0.72C_{Green} + 0.07C_{Blue} \qquad (1)$$

where, C_{Red} is R, C_{Green} is G and C_{Blue} is B in RGB color format and C_{Grey} is the resulting greyscale color scheme. The resulting frame will be as shown in Fig.2.

Fig.2. Greyscale frame

3.1.3 Converting Greyscale Frame to Binary:

After the conversion from RGB to greyscale, the frame undergoes binary conversion (C_{Binary}) using a threshold of 0.2, as in Eq.(2); resulting in the output as shown in Fig.3.

$$C_{Green} = C_{Grey} > 0.2 \qquad (2)$$

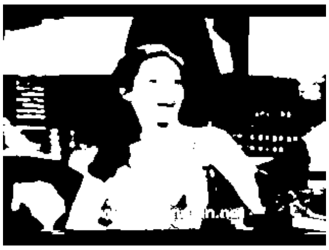

Fig.3. Binary frame

3.1.4 Extraction of Region Borders:

First, greyscale frame is converted to binary as explained in section 3.1.3. From the binary frame, extract the region borders that are needed during feature extraction. Region borders are useful to overcome the problems posed by illumination.

To extract region borders, dilation and erosion is applied onto the binary frame over the ROI area. In dilation process, pixels outside the image limit are given the lowest value supported by the data type; and in erosion process, pixels outside the image limit are given the highest value supported by the data type. Let C be Euclidian space computed as in Eq.(3),

$$C = \sum_{j=1}^{m} x_j \cdot y_j = x_1 \cdot y_1 + x_2 \cdot y_2 + \cdots + x_m \cdot y_m \qquad (3)$$

P be binary image in C_{Binary} and Q is a structuring element, then the dilation "$\oplus$" value of binary images is computed using Eq.(4).

$$P \oplus Q = \{z \in C | (Q^s)_z \cap P \neq \varnothing\} \qquad (4)$$

where, (Q^s) is symmetric of Q, as in Eq.(5).

$$Q^s = \{x \in C | -x \in C\} \qquad (5)$$

Similarly, erosion "$\ominus$" is calculated for binary images using Eq.(6).

$$P \ominus \mathrm{Q} = \{z \in C | Q_z \subseteq P\} \qquad (6)$$

where, Q_z is a translation of Q by the vector z as shown in Eq.(7).

$$Q_z = \{q+z | q \in Q, \forall_z \in C\} \qquad (7)$$

The output data of dilation and erosion on ROI of Fig.1 is shown in Fig.4 below,

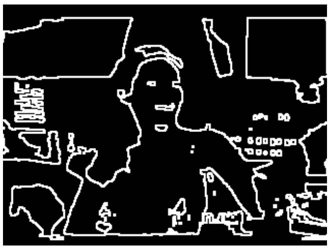

Fig.4. Frame snippet after dilation and erosion resulting in region borders

3.2 FEATURE EXTRACTION

We extract 3 features in proposed system: HOG, Eigen and FAST feature points. All these feature points are computed in and around ROI of the input frame.

At first, extract Eigen, HOG and FAST feature points on the dilated frame; and store each feature point values in a separate array. Then, combine these three array of points into a single array of combined points say C_p; and keep a copy of this combined points (C_p) as C_p·old, which is useful in case if the value of C_p gets modified during further computation.

3.2.1 *Eigen Feature Points:*

Minimum Eigen values are calculated by sum of the squared difference matrix *D*, as follows,

$$D = \begin{bmatrix} X & Z \\ Z & Y \end{bmatrix} \tag{8}$$

where, *X*, *Y* and *Z* are as shown in the Eq.(9), Eq.(10) and Eq.(11).

$$X = (F_x)^2 \otimes w \tag{9}$$

$$Y = (F_y)^2 \otimes w \tag{10}$$

$$Z = (F_x \cdot F_y)^2 \otimes \mathrm{w} \tag{11}$$

The values F_x and F_y are gradients of input frame, *F* in x-axis and y-axis respectively. The mark "⊗" indicates convolution process. The block multiplies the coefficients vector by its transpose and creates a filter coefficients matrix called "*w*". The resulting frame appears as shown in Fig.5.

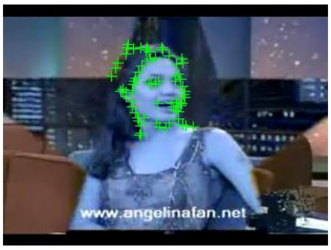

Fig.5. Eigen feature points

3.2.2 *FAST Feature Points:*

Choose a pixel *P* in ROI that has to be classified as either a feature point or not. Allow its intensity to be I_P, choose a threshold value *T*. Now, examine 16-pixels around *P* for test. The pixel *P* is corner, if there reside *N* steady pixels among that 16-pixels which are all brighter than I_P+*T* or darker than I_P-*T*; where, N is chosen to equal to I_P. The output of this process is shown in Fig.6.

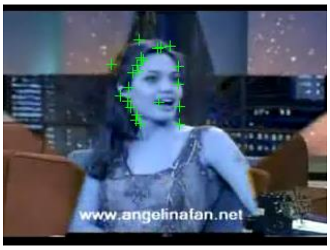

Fig.6. FAST feature points

3.2.3 *HOG Feature Points:*

First, ROI will be filtered using the filter kernels in Eq.(12).

$$[-1\ 0\ 1] \text{ and } [-1\ 0\ 1]^T \tag{12}$$

Then divide the image into blocks and blocks are further divided into cells. Blocks may overlap and same cell can be in several blocks. For each pixel in cell, a horizontal and vertical gradient are obtained using the following Eq.(13) and Eq.(14).

$$G_x(y,x) = Y(y,x+1) - Y(y,x-1) \tag{13}$$

$$G_y(y,x) = Y(y+1,x) - Y(y-1,x) \tag{14}$$

where, $Y(y,x)$ is the pixel intensity at coordinates *x* and *y*; $G_x(y,x)$ and $G_y(y,x)$ are the horizontal and vertical gradients respectively. The magnitude and phase of gradient are determined by using Eq.(15) and Eq.(16). The resulting HOG feature points are shown in Fig.7.

$$G(y,x) = \sqrt{G_x(y,x)^2 + G_y(y,x)^2} \tag{15}$$

$$\theta(y,x) = \arctan\left(\frac{G_y(y,x)}{G_x(y,x)}\right) \tag{16}$$

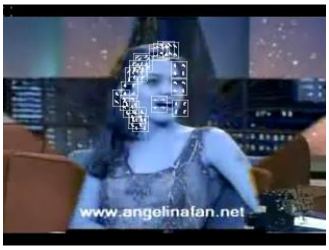

Fig.7. HOG feature points

3.2.4 *Combining Extracted Feature Points:*

In the previous steps, we have extracted the points in and around the selected face region. This gives us various set of points. Once detected, all the points are combined together to establish a common collection of points. After combining the points, the resultant frame representation will be as shown in Fig.8.

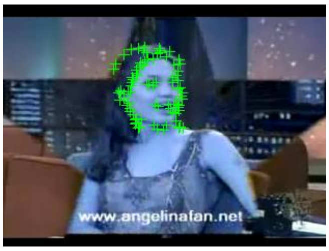

Fig.8. Combined points

3.3 FACE TRACKING

Tracking part needs 3 important elements: input video sequence, C_p and a point tracker. A point tracker initializes the tracker to track the input points. C_p is as in Eq.(17).

$$C_p = \text{concatenate}(D,N,G). \tag{17}$$

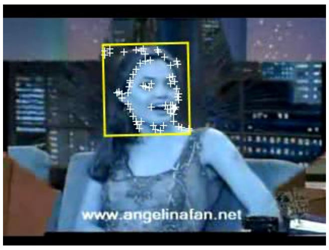

Fig.9. Output frame

3.3.1 Tracking the Combined Feature Points (C_p):

The tracking process in the proposed system goes frame by frame rather than whole video at a time. So, system tracks the video until last frame is processed. Tracking begins by extracting next frame and finding points present in C_p array. It is not always feasible that all points get matched accurately; so, we estimate new points and take C_p as old points. Next, estimate the geometric transformation between old and new points; and estimate outliers. Finally, insert the bounding box around the estimated transformed region. This method continues until the ROI in last frame is tracked. Display the annotated frame using a video player object as shown in Fig.9.

3.3.2 Workflow Architecture:

The Fig.10 shows that, how the workflow architecture of the proposed algorithm for selected face tracking is implemented using the design steps discussed. The steps are combined together to form a complete system; and each step in the system adds its own importance for the algorithm.

4. EXPERIMENTAL EVALUATION

The discussion in this section is about the outcome of proposed work. This section comprises datasets, performance metrics, results and analysis and time analysis as sub-sections.

4.1 DATASETS

We have taken eight technically challenging videos of (three) different background categories for testing. One video is from VidTIMIT dataset [22] and seven videos belong to YouTube Celebrities dataset [23]. Out of eight, six videos come under the category of static sensor and moving face(s); one each under moving sensor and static face(s) and moving sensor and moving face(s) categories. The videos pose many challenges like low resolution, abrupt motions, occlusion (e.g. helmet, spectacle and beard), variations in illumination, pose and expression etc.

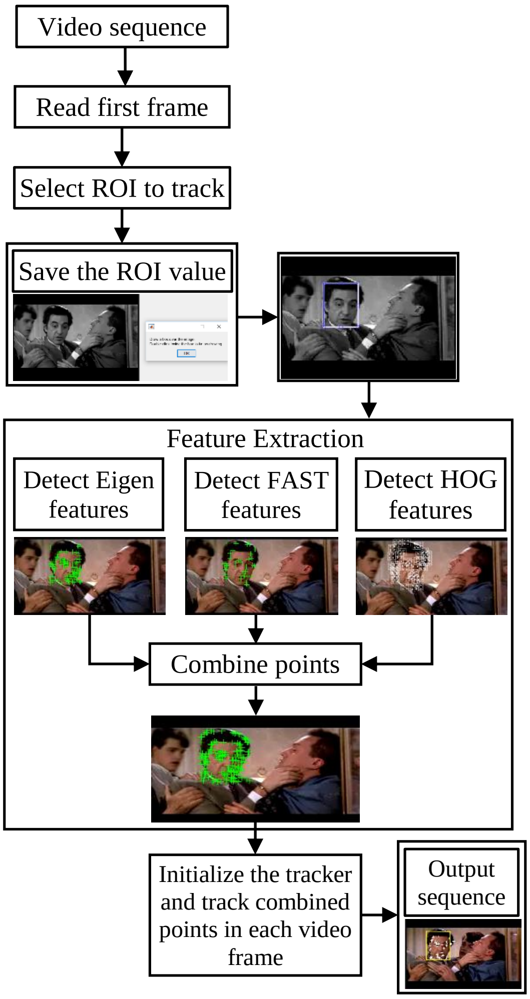

Fig.10. Workflow diagram

4.2 PERFORMANCE METRICS

Performance metrics [24] are calculated based on the following four values.

- True Positive (*TP*) – Positive pixels rightly classified as positive.
- False Negative (*FN*) – Positive pixels wrongly classified as negative.
- True Negative (*TN*) – Negative pixels wrongly classified as negative.
- False Positive (*FP*) – Negative pixels wrongly classified as positive.

The most important metrics are recall (also called sensitivity), precision, specificity, accuracy and F1 score. The combination of sensitivity and specificity elevates the accuracy measurement; and F1 score combines recall with precision to determine the test accuracy. Let $TP = \alpha$, $TN = \beta$, $FP = \gamma$ and $FN = \delta$. Using these, the five performance metrics are defined in the following sub-sections.

4.2.1 Recall (R):

Recall is defined as the amount of correct occurrences fetched from the total amount of correct occurrences. Hence, *R* can be defined as follows:

$$R = \alpha/(\alpha + \delta) \tag{18}$$

4.2.2 *Precision (P):*

Precision is defined as the amount of correct occurrences among the fetched occurrences. Hence, *P* can be defined as follows:

$$P = \alpha/(\alpha + \gamma) \tag{19}$$

4.2.3 *Specificity (S):*

Specificity is defined as the amount of incorrect occurrences fetched from the total amount of incorrect occurrences. Hence, *S* can be defined as follows:

$$S = \beta/(\beta+\gamma) \tag{20}$$

4.2.4 *Accuracy (A):*

Accuracy is defined as the ratio of correctly guessed occurrence to the total occurrences. Hence, *A* can be defined as follows:

$$A = (\alpha+\beta)/(\alpha+\beta+\gamma+\delta) \tag{21}$$

4.2.5 *F1 Score (F1):*

*F*1 Score is defined as the weighted mean of recall and precision. Hence, *F*1 can be defined as follows:

$$F1 = 2\cdot(Precision \times Recall) / (Precision + Recall) \tag{22}$$

Equations defined above i.e. Eq.(18), Eq.(19), Eq.(20), Eq.(21) and Eq.(22) compute *R* (recall), *P* (precision), *S* (specificity), *A* (accuracy) and *F*1 (*F*1 score) respectively. The outcomes of these equations are probabilistic values, with 1 being the best and 0 being the worst result value.

4.3 RESULTS AND ANALYSIS

This section discusses about the results of the proposed algorithm for the performance metrics discussed before. The results are then compared with the similar well-known algorithms such as KLT and CAMSHIFT; and a recent time algorithm which works based on points, let us call it as Algorithm* [16]. We have used both indoor and outdoor videos from the datasets discussed above. The few experimental results of eight videos are given in following figures.

The Fig.11 shows the frames of a low resolution video which comes under the category of static sensor and moving face(s); and belong to YouTube Celebrities dataset. It contains two human faces, out of which one with pose and expression variations is selected for tracking as shown in Fig.11.

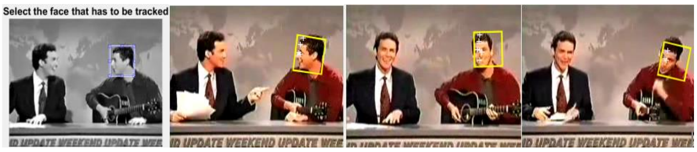

Fig.11. Selected face tracking result of video sequence 0031_01_031_adam_sandler.avi (337 frames) with frames #01, #04, #44 and #243 displayed

The Fig.12 houses multiple faces, some of them are merged and some are occluded. As the motion in video is due to human movement, it comes under the category of static sensor and moving face(s); and belong to YouTube Celebrities dataset. A single human face is selected and tracked as shown in figure.

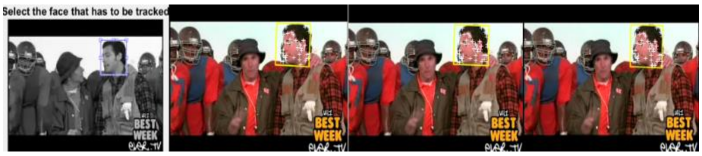

Fig.12. Selected face tracking result of outdoor video sequence 0047_03_004_adam_sandler.avi (28 frames) with frames #01, #05, #12 and #21 displayed

The Fig.13 contains the frames of an actress who is giving the interview. It comes under the category of static sensor and moving face(s); and belong to YouTube Celebrities dataset. We can observe that proposed algorithm is able to track the face, facing the challenges of change in illumination and abrupt motion from frame to frame.

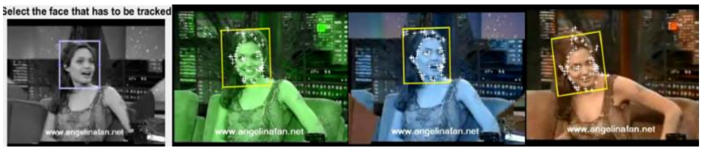

Fig.13. Tracking result of single face video sequence 0286_01_016_angelina_ jolie.avi (186 frames) with frames #01, #48, #170 and #110 displayed

The Fig.14 shows video frames with multiple faces. Due to camera motion and movement of face(s), it comes under the category of moving sensor and moving face(s); and belong to YouTube Celebrities dataset. For tracking, we have selected a human face which moves along with the camera.

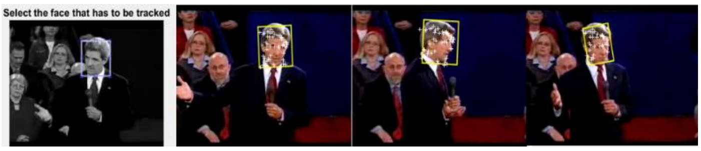

Fig.14. Selected face tracking result of indoor video sequence 1167_01_011_ john_kerry.avi (121 frames) with frames #01, #04, #55 and #111 displayed

The Fig.15 shows low resolution video frames where two celebrities are giving an interview. This video comes under the category of static sensor and moving face(s); and belong to YouTube Celebrities dataset. The video contains background poster and there are chances of tracking non-faces; but, proposed algorithm continues to track only the selected face.

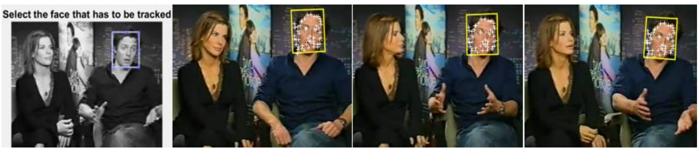

Fig.15. Selected face tracking result of video sequence 0818_02_013_hugh_grant.avi (196 frames) with frames #01, #190, #112 and #10 displayed

The Fig.16 shows the result of tracking extreme poses of a single face. This video comes under the category of static sensor and moving face(s); and belong to VidTIMIT dataset.

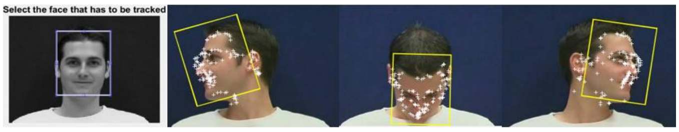

Fig.16. Tracking result of video sequence mcem0_head.mpg (363 frames) with frames #01, #38, #196 and #123 displayed

The Fig.17 shows video frames where people are quarrelling. This video comes under the category of static sensor and moving face(s); and belong to YouTube Celebrities dataset. Here, a partially visible face is selected for tracking.

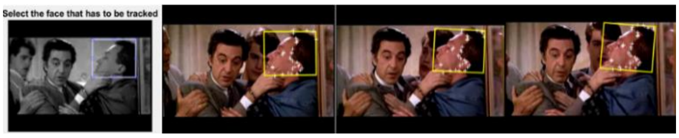

Fig.17. Selected face tracking result of indoor video sequence 0159_02_023_al_pacino.avi (144 frames) with frames #01, #38, #140 and #93 displayed

The Fig.18 shows the frames of a video containing two persons. Here, a person is in the position of rest and speaking to other person; but, camera is moving towards him. So, this video comes under the category of moving sensor and static face(s); and belong to YouTube Celebrities dataset.

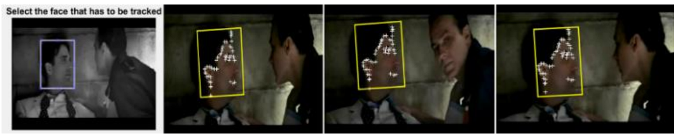

Fig.18. Selected face tracking result of video sequence 0265_03_003_andy_garcia.avi (143 frames) with frames #01, #138, #105 and #65 displayed

For simplicity and clear representation of tabular results, we shall represent 0031_01_031_adam_sandler.avi, 0047_03_004_adam_sandler.avi, 0159_02_023_al_pacino.avi, 0286_01_016_angelina_jolie.avi, 0818_02_013_hugh_grant.avi, 1167_01_011_john_kerry.avi, mcem0_head.mpg and 0265_03_003_andy_garcia.avi with numbers from 1-8 respectively. Using the metrics discussed, the results of four algorithms are tabulated in Table 1, Table 2, Table 3 and Table 4. The "-" in the tables denote failure or no result possible due to lack of available data which contradicts that there is a failure in tracking of that particular video by the given algorithm.

Table.1. Results obtained using proposed algorithm

Video	TP	TN	FP	FN	R	P	S	A	F1
1	337	0	0	0	1	1	0	1	1
2	28	0	0	0	1	1	0	1	1
3	144	0	0	0	1	1	0	1	1
4	186	0	0	0	1	1	0	1	1
5	196	0	0	0	1	1	0	1	1
6	121	0	0	0	1	1	0	1	1
7	363	0	0	0	1	1	0	1	1
8	143	0	0	0	1	1	0	1	1

Table.2. Results obtained using Algorithm*

Video	TP	TN	FP	FN	R	P	S	A	F1
1	-	-	-	-	-	-	-	-	-
2	-	-	-	-	-	-	-	-	-
3	-	-	-	-	-	-	-	-	-
4	186	0	0	0	1	1	0	1	1
5	-	-	-	-	-	-	-	-	-
6	-	-	-	-	-	-	-	-	-
7	363	0	0	0	1	1	0	1	1
8	-	-	-	-	-	-	-	-	-

Table.3. Results obtained using CAMSHIFT

Video	TP	TN	FP	FN	R	P	S	A	F1
1	-	-	-	-	-	-	-	-	-
2	-	-	-	-	-	-	-	-	-
3	-	-	-	-	-	-	-	-	-
4	-	-	-	-	-	-	-	-	-
5	-	-	-	-	-	-	-	-	-
6	-	-	-	-	-	-	-	-	-
7	363	0	0	0	1	1	0	1	1
8	-	-	-	-	-	-	-	-	-

Table.4. Results obtained using KLT

Video	TP	TN	FP	FN	R	P	S	A	F1
1	-	-	-	-	-	-	-	-	-
2	-	-	-	-	-	-	-	-	-
3	-	-	-	-	-	-	-	-	-
4	186	0	0	0	1	1	0	1	1
5	-	-	-	-	-	-	-	-	-
6	-	-	-	-	-	-	-	-	-
7	363	0	0	0	1	1	0	1	1
8	-	-	-	-	-	-	-	-	-

In the results tabulated, we can observe that the proposed algorithm produces the best results for all undertaken videos. The Fig.19 illustrates the accuracy of different approaches graphically.

4.4 TIME ANALYSIS

Time is one of the major factor for evaluation of performance and speed up achieved by any algorithm. Speed of an algorithm is dependent on the system in which it is running. So, to test for time taken for processing, it is advisable to use the same system for execution of all algorithms. The time taken has been tabulated in Table.5.

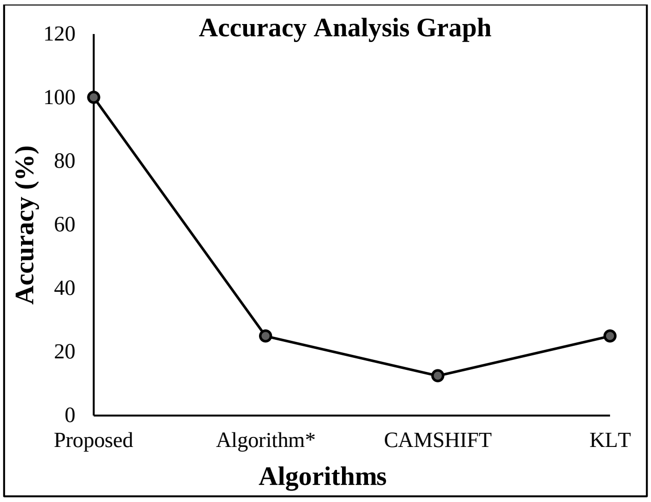

Fig.19. Accuracy analysis of different algorithms

Table.5. Time taken to process videos (in seconds)

Video	Proposed	Algorithm*	CAMSHIFT	KLT
1	2.8	-	-	-
2	1.3	-	-	-
3	1.7	-	-	-
4	1.9	4.6	-	11.7
5	2.9	-	-	-
6	1.4	-	-	-
7	4.9	7.6	5.8	26
8	2.3	-	-	-

When we plot the results of Table.5, we observe the graph shown in Fig.20 below. The lines on and below X-axis are the result of "-" (False/Failure). We can observe that time required for proposed algorithm is constant and very less compared to other algorithms.

5. CONCLUSION AND FUTURE WORK

The proposed algorithm is capable of tracking human face appearing in the frames of different background low resolution video sequences. It tackles the problems of change in illumination, expression, pose and occlusion effectively. The results obtained shows that, performance of the proposed algorithm to track face is better compared to other algorithms.

Proposed algorithm is able to track any selected single face appearing only in the first frame of the video sequence. Hence, future work involves selected single face tracking in videos which contain face appearing in any frame of the video sequence. Further, the tracker can be improved to track multiple selected faces.

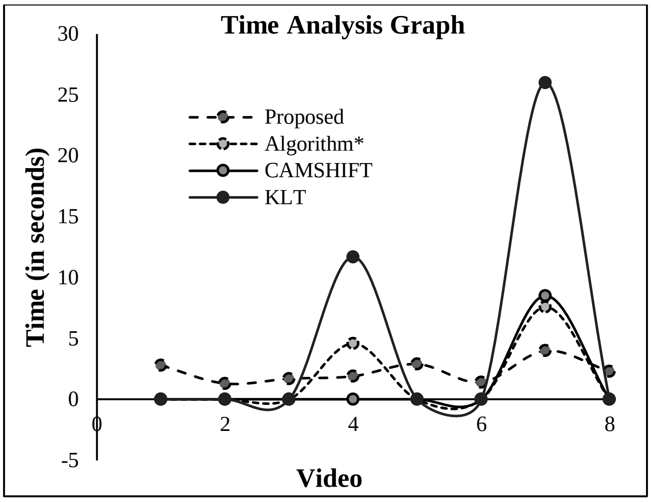

Fig.20. Time analysis of different algorithms

REFERENCES

[1] S. Ranganatha and Y.P. Gowramma, "Face Recognition Techniques: A Survey", *International Journal for Research in Applied Science and Engineering Technology*, Vol. 3, No. 4, pp. 630-635, 2015.

[2] S. Ranganatha and Y.P. Gowramma, "A Comprehensive Survey of Algorithms for Face Tracking in Different Background Video Sequence", *International Journal of Computer Applications*, 2018.

[3] P. Viola and M. Jones, "Rapid Object Detection using a Boosted Cascade of Simple Features", *Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition*, pp. 511-518, 2001.

[4] P. Viola and M. Jones, "Robust Real-Time Face Detection", *International Journal of Computer Vision*, Vol. 57, No. 2, pp. 137-154, 2004.

[5] Jianbo Shi and Carlo Tomasi, "Good Features to Track", *Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition*, pp. 593-600, 1994.

[6] E. Rosten and T. Drummond, "Fusing Points and Lines for High Performance Tracking", *Proceedings of IEEE International Conference on Computer Vision*, pp. 1508-1515, 2005.

[7] N. Dalal and B. Triggs, "Histograms of Oriented Gradients for Human Detection", *Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition*, pp. 886-893, 2005.

[8] Stephen M. Smith and J. Michael Brady, "SUSAN-A New Approach to Low Level Image Processing", *International Journal of Computer Vision*, Vol. 23, No. 1, pp. 45-78, 1997.

[9] C. Harris and M. Stephens, "A Combined Corner and Edge Detector", *Proceedings of 4th Alvey Vision Conference*, pp. 147-151, 1988.

[10] D. Comaniciu and P. Meer, "Mean Shift: A Robust Approach toward Feature Space Analysis", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 24, No. 5, pp. 603-619, 2002.

[11] K. Fukunaga and L.D. Hostetler, "The Estimation of the Gradient of a Density Function, with Applications in Pattern

Recognition", *IEEE Transactions on Information Theory*, Vol. 21, No. 1, pp. 32-40, 1975.

[12] Yizong Cheng, "Mean Shift, Mode Seeking, and Clustering", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 17, No. 8, pp. 790-799, 1995.

[13] G. Bradski, "Computer Vision Face Tracking for Use in a Perceptual User Interface", *Intel Technology Journal*, pp. 12-21, 1998.

[14] Bruce D. Lucas and Takeo Kanade, "An Iterative Image Registration Technique with an Application to Stereo Vision", *Proceedings of International Joint Conference on Artificial Intelligence*, pp. 674-679, 1981.

[15] Carlo Tomasi and Takeo Kanade, "Detection and Tracking of Point Features", Available at: https://cecas.clemson.edu/~stb/klt/tomasi-kanade-techreport-1991.pdf.

[16] S. Ranganatha and Y P Gowramma, "A Novel Fused Algorithm for Human Face Tracking in Video Sequences", *Proceedings of IEEE International Conference on Computation System and Information Technology for Sustainable Solutions*, pp. 1-6, 2016.

[17] Supriya Mangale, Ruchi Tambe and Madhuri Khambete, "Object Detection and Tracking in Thermal Video Using Directed Acyclic Graph (DAG)", *ICTACT Journal on Image and Video Processing*, Vol. 8, No. 1, pp. 1566-1574, 2017.

[18] S. Ranganatha and Y.P. Gowramma, "An Integrated Robust Approach for Fast Face Tracking in Noisy Real-World Videos with Visual Constraints", *Proceedings of IEEE International Conference on Advances in Computing, Communications and Informatics*, pp. 772-776, 2017.

[19] S. Ranganatha and Y.P. Gowramma, "Development of Robust Multiple Face Tracking Algorithm and Novel Performance Evaluation Metrics for Different Background Video Sequences", *International Journal of Intelligent Systems and Applications*, Vol. 10, No. 8, pp. 19-35, 2018.

[20] S. Ranganatha and Y.P. Gowramma, "Image Training, Corner and FAST Features Based Algorithm for Face Tracking in Low Resolution Different Background Challenging Video Sequences", *International Journal of Image, Graphics and Signal Processing*, Vol. 10, No. 8, pp. 39-53, 2018.

[21] S. Ranganatha and Y.P. Gowramma, "Image Training and LBPH Based Algorithm for Face Tracking in Different Background Video Sequence", *International Journal of Computer Sciences and Engineering*, Vol. 6, No. 9, pp. 349-354, 2018.

[22] C. Sanderson and B.C. Lovell, "Multi-Region Probabilistic Histograms for Robust and Scalable Identity Inference", *Proceedings of International Conference on Biometrics*, pp. 199-208, 2009.

[23] M. Kim, S. Kumar, V. Pavlovic and H. Rowley, "Face Tracking and Recognition with Visual Constraints in Real-World Videos", *Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition*, pp. 1-8, 2008.

[24] T. Fawcett, "An Introduction to ROC Analysis", *Pattern Recognition Letters*, Vol. 27, No. 8, pp. 861-874, 2006.