



FACE DETECTION AND TRACKING USING KLT AND VIOLA JONES

Ritesh Boda and M. Jasmine Pemeena Priyadarsini

School of Electronics and Communication Engineering, VIT University, India

E-Mail: riteshboda9@gmail.com

ABSTRACT

In this paper we use two approaches for detecting a face and track it continuously. Basically video sequences provide more information than a still image. It is always a challenging task to track a target object in a live video. We undergo challenges like illumination; pose variation and occlusion in pre-processing stages. But this is can be overcome by detection of the target object continuously in each and every frame. Face tracking by Kanade Lucas Tomasi algorithm that is used to track face based on trained features. Whereas the Viola Jones algorithm is used detect the face based on the haar features.

Keywords: face tracking, kanade lucas tomasi (KLT), viola jones.

INTRODUCTION

Object detection has been a great challenge from the past few years. In computer vision human face detection is an important research topic. It is needed for many computer applications like HCI, surveillance, human-robot interaction, etc. In this field facial tracking is finding more growth of use in security and safety applications to detect various situations¹. This tracking domain can be used to control or communicate with robots.

Detecting human faces in a video is a great challenging problem. These configurations may be like angle of view, background intensity, and various illuminations. This is due to high variety of configurations that may occur. The complexness of the face results in a particular degree of issue for fast detection and tracking [2].

For face detection and tracking in a given video sequence different algorithms have been introduced over the past few years. Each algorithm has got its own advantages and disadvantages. But any facetracking algorithm will have some errors which will cause deviation from the required object. The tracker can be accurate if and only if it is able to minimize this deviation. The technique used in this paper is one of the effective approaches. It is quicker and simpler. Where haar features of the Viola Jones algorithm gives a way for detecting faces and harris detector provides a route for tracking [3].

RELATED WORK

Face tracking

Object tracking is defined as keeping a trace on a particular kind of object. In this paper as we are mainly concentrating on face, we track human faces based on the given input features. Continuous tracking makes us leave the problems like illumination, variation in pose etc. aside [4]. Here tracking of human faces in a video sequence is done and also live video tracking using a webcam is done.

Kanade LucasTomasi(KLT) algorithm

Kanade Lucas Tomasi algorithm is used for feature tracking. It is a most popular one. KLT algorithm

was introduced by Lucas and Kanade and their work was later extended by Tomasi and Kanade. This algorithm is used for detecting scattered feature points which have enough texture for tracking the required points in a good standard [5].

Kanade-Lucas-Tomasi (KLT) algorithm is used here for tracking human faces continuously in a video frame. This method is accomplished by them finding the parameters that allow the reduction in dissimilarity measurements between feature points that are related to original translational model.

Firstly in this algorithm we calculate the displacement of the tracked points from one frame to another frame. From this displacement calculation it is easy to compute the movement of the head. The feature points of a human face are tracked by using optical flow tracker [6]. KLT tracking algorithm tracks the face in two simple steps, firstly it finds the traceable feature points in the first frame and then tracks the detected features in the succeeding frames by using the calculated displacement.

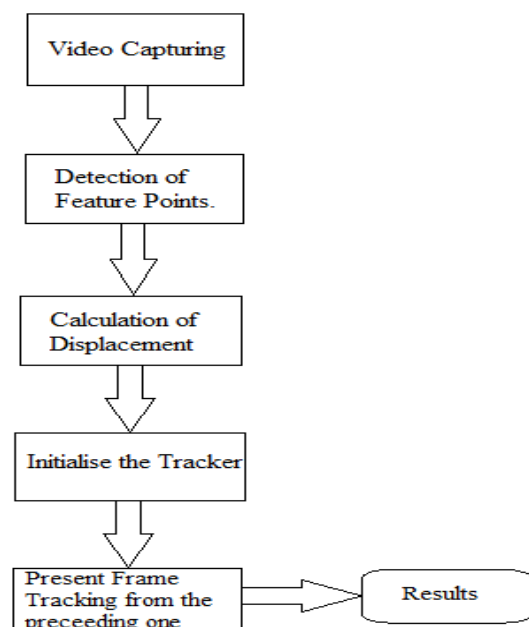


Figure-1. KLT algorithm flow chart.



In this algorithm firstly it detects harris corners in the first frame. And then using optical flow it continues to detect the points by computing the motion of the pixels of an image. For each translation motion of the image optical flow is computed. The harris corners are detected by linking the motion vectors in successive frames to get a track for each harris point. Just not to lose the track of the video sequence we apply harris detector at every 10 to 15 frames. This is nothing but making sure by checking the frames periodically. In this way new and old harris points are tracked. Here in this paper we consider only 2-D motion i.e. translation movement.

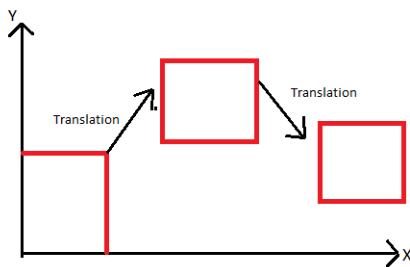


Figure-2. Translation movement of frames.

Let us assume that the initially one of the corner point is (x, y) . Then in the next frame, if it is displaced by some variable vector (b_1, b_2, \dots, b_n) , the displaced corner point of the frame will be the sum of the initial point and displaced vector.

The coordinates of the new point will be $x' = x + b_1$ and $y' = y + b_2$.

So, the displacement now should be calculated with respect to each coordinate. For this we use warp function which is a function with coordinates and a parameter. It is denoted as $W(x; p) = (x + b_1; y + b_2)$. The warp function is used to estimate the formation.

Alignment estimation

In the first frame the initially detected points is taken as a template image. In the later stages difference between displacement and the preceding point is taken to get the next tracking points. The alignment is calculated by

$$\sum_x [I(W(x; p)) - T(x)]^2 \quad (1)$$

Where p is the displacement parameter

Assume initial estimate of p as a known parameter and find Δp

$$\sum_x [I(W(x; p + \Delta p)) - T(x)]^2 \quad (2)$$

The displacement Δp is calculated by finding Taylor series and then differentiating it w.r.t. Δp

$$\Delta p = H^{-1} \sum_x \left[\nabla I \frac{\partial W}{\partial p} \right]^T \cdot [T(x) - I(W(x; p))] \quad (3)$$

Where H is called as hessian matrix. This is how we estimate the displacement Δp and find the next traceable point [7].

Face detection

Detecting a face is a computer technology which let us know the locations and sizes of human faces. This helps in getting the facial features and avoiding other objects and things. In the present situation human face perception is a biggest research area. It is basically about detecting a human face through some trained features [10]. Here face detection is preliminary step for many other applications such as face recognition, video surveillance etc.

Viola Jones algorithm

This algorithm helps us detect features of a face in a particular frame of a video sequence. This is the first object detection framework which gives a competition to real time detection rates. Paul Viola and Michael Jones are the ones who introduced this algorithm. They made this algorithm mainly by the issue of face detection. There are four steps which have to be followed to detect a face. Firstly, we train the system with the haar features [12-14]. Haar features are a kind of rectangular boxes which are black and white.

Haar features are simple rectangular feature which is the difference of the sum of pixels of areas inside the rectangle. This rectangle can be at any position of the frame and can scale the image. This modified feature set is called 2-rectangle feature. Each feature type can indicate the existence or the absence of certain characteristics in the frame, such as edges or changes in texture.

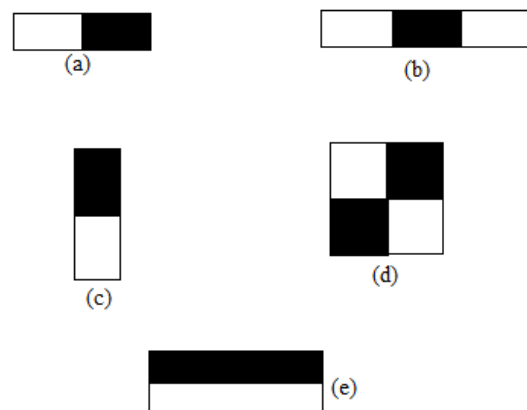


Figure-3. Haar features.

These haar features are applied to determine the facial features. For example in the Figure-3 (b), this part is used to detect nose feature of a human face as the black colored part defines the presence of a nose which is located at the center of the face. And the Figure-3 (e) is called a 4 rectangle feature. Where the black part is denoted as +1 and the white part is denoted as -1.

The result is calculated by subtracting the sum of pixels under the white rectangle from the sum of pixels



under black rectangle. Initially some threshold is taken for particular features. Average sum of each black and white is calculated. Then difference is checked with threshold. If the value is above or matches with the threshold then it is detected as relevant feature [11].

Integral image

The integral image part is used to sum all the pixels of a particular box to its left and above ones. The four corner values of the area are to be calculated. This makes avoid summing of each pixel in the region. This integral image conversion process is introduced just to speed up the process in calculating pixels [9].

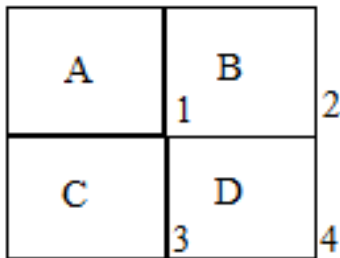


Figure-4. Integral image calculation.

The calculation of the sum of pixels of part D in the fig.4 is $(1+4)-(2+3)$ i.e. $[A+(A+B+C+D)] - [(A+B+A+C)]$ which gives D.

Adaboost

It is a process used to find out relevant and irrelevant features. It uses the weak classifiers and weights to form a strong classifier. It finds the single rectangular feature and threshold which is the best to separate the faces and non-faces in training examples in terms of weighted error. It firstly starts with uniform weights while training. Next it evaluates the weighted error for each feature and picks the best [13].

We reevaluate the examples where incorrect classifiers will have more weight and correct classifiers will have less weight. Finally the classifier will contain the combination of correct classifiers which are having less weight. To reduce the computational time non-faces are discarded.

Cascading

This step is introduced to speed up the process and give an accurate result. This step consists of several stages where each stage consists of a strong classifier. All features are grouped into several stages. It detects faces in the frame by sliding a window over a frame [8].

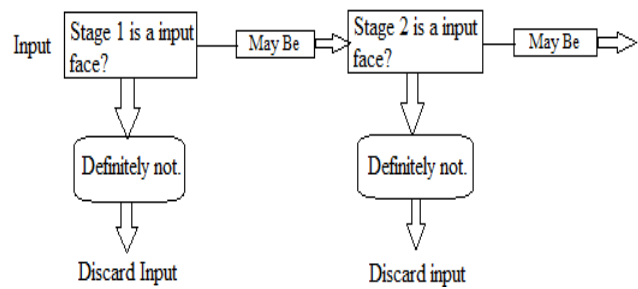


Figure-5. Cascading process.

Just to understand the Figure-5, when an input is given it checks for certain classifier in the first stage and then so on. But it is passed to the successive stage if and only if it satisfies the preceding stage classifier.

SIMULATION RESULTS

This process is implemented using Matlab software. Here input is given as a video format or else it can take a live video with the help of a webcam. The video input should be given only in avi format.

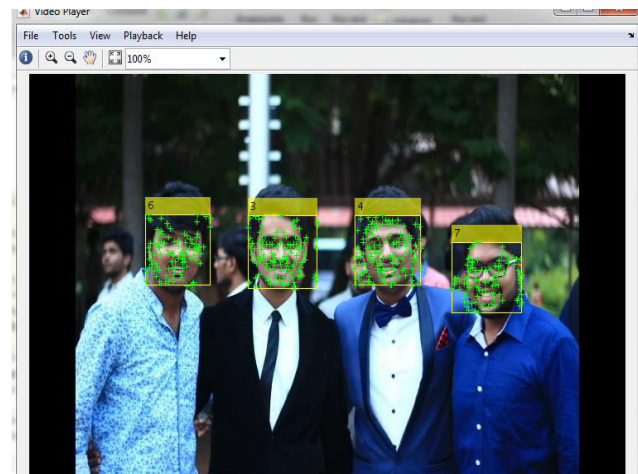


Figure-6. Snapshot 1 of a video from VIT database.

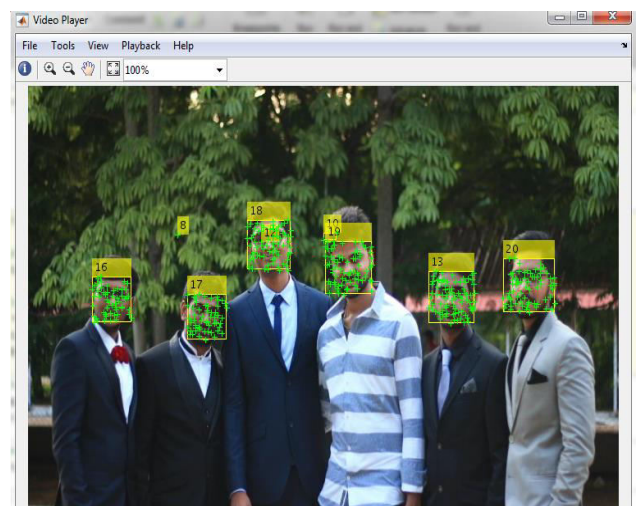


Figure-7. Snapshot 2 of a video from VIT database.



These two snapshots (snapshot 1 and snapshot 2) of a video sequence are the ones captured from a playing video in Matlab. The detection accuracy is shown in later.

Now let us see the results from a live webcam. Where it starts tracking as soon as the webcam gets detected and switched on by Matlab.

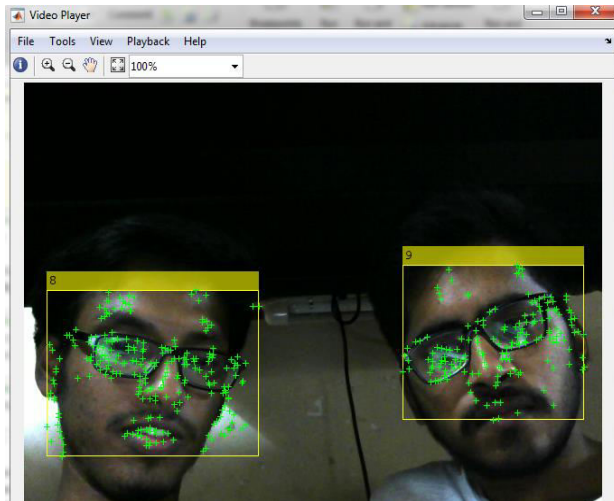


Figure-8. Snapshot 3 of a live video from webcam.

Table-1. Accuracy of the simulation results.

| Snapshot | Number of faces detected | Number of non-faces detected | Number of faces not detected |
|----------|--------------------------|------------------------------|------------------------------|
| 1. | 4 | 0 | 1 |
| 2. | 6 | 1 | 0 |
| 3. | 2 | 0 | 0 |

Table-1 shows the accuracy of the simulation results. As the video keeps playing it is a bit difficult for it to track the faces exactly. But upon checking it practically we can say that it has accuracy above 95%.

CONCLUSION AND FUTURE WORKS

The approach presented here for face detection and tracking decreases the computation time producing results with high accuracy. Tracking of a face in a video sequence is done using KLT algorithm whereas Viola Jones is used for detecting facial features. Not only in video sequences, it has also been tested on live video using a webcam. Using this system many security and surveillance systems can be developed and required object can be traced down easily. In the coming days these algorithms can be used to detect a particular object rather than faces. Future work is to work on the same domain but to track a particular face in a video sequence. That is like avoiding all other faces except the face required.

REFERENCES

- [1] Ragini Choudhury Verma, Cordelia Schmid and Krystian Mikolajczyk. 2013. Face Detection and

Tracking in a Video by Propagating Detection Probabilities. IEEE Transactions on Pattern Analysis and Machine Intelligence. 25(10).

- [2] DorinComaniciu and Visvanathan Ramesh. 2011. Robust Detection and Tracking of Human Faces with an Active Camera. IEEE Visual Surveillance.
- [3] Christian Ku'blbeckand Andreas Ernst. 2009. Face detection and tracking in video sequences using the modified census transformation. Electronic Imaging Department, Fraunhofer Institute for Integrated Circuits, Hsin -Chu, Germany.
- [4] MAMATA S. KALAS. 2014. Real Time Face Detection and Tracking Using OPENCV. International Journal of Soft Computing and Artificial Intelligence.
- [5] Nandita Sethi and Alankrita Aggarwal. 2011. Robust Face Detection and Tracking Using Pyramidal Lucas Kanade Tracker Algorithm. IJCTA, vol. 2.
- [6] Zhu Liu and Yao Wang. Face Detection and Tracking in Video Using Dynamic Programming. Department of Electrical Engineering, Polytechnic University, Brooklyn.
- [7] Etienne Corvee and Francois Bremond. Combining face detection and people tracking in video sequences. Pulsar team, INRIA.
- [8] Ragini Choudhury Verma, Cordelia Schmid and Krystian Mikolajczyk. 2008. Face Detection and Tracking in a Video by Propagating Detection Probabilities. IEEE Transactions on Pattern Analysis And Machine Intelligence. 25(10).
- [9] Dmitry Mikhaylov, Anton Samoylov, Peter Minin and Alexey Egorov. 2014. Face Detection and Tracking from Image and Statistics Gathering. IEEE Conference on Signal-Image Technology and Internet-Based Systems.
- [10] Divya George and Arunkant A. Jose. 2015. Face Detection and Tracking at Different angles in Video using Optical Flow. ARPN Journal of Engineering and Applied Sciences. 10(17): 7678-7683.
- [11] Peter Gejguš and Martin Šperka. 2003. Face tracking for expressions simulations. International Conference on Computer Systems and Technologies.
- [12] Paul Viola and Michael Jones. 2001. Rapid Object Detection using a Boosted Cascade of Simple



Features. Conference on Computer Vision and Pattern Recognition.

- [13] L. Stan and Z. Zhang. 2004. FloatBoost learning and statistical face detection. IEEE Trans. On Pattern Analysis and Machine Intelligence. 26(9).
- [14] http://people.ece.cornell.edu/land/courses/ece5760/FinalProjects/f2010/kaf42_jay29_teg25/teg25_jay29_kaf42/index.html.
- [15] <http://cs.mcgill.ca/~gpetro6/links/report.pdf>.