Tracking and Blurring the Face in a Video File

Farah Saad Al-Mukhtar
Computer Science Department, College of Science, Al-Nahrain University, Baghdad-Iraq.

Abstract

This paper shows how to detect, track and blur a face in video frame using the Viola-Jones detection algorithm for detection and KLT algorithm to tracks a set of feature points across the video frames. First, the detection locates the face, and then identifies feature points that can be reliably tracked.

A blurring filter is applied after detection and tracking the face and very good results are obtained that can be useful to blurring the face; for example, when showing a painful view that includes distorted faces in wars reports or crimes. [DOI: 10.22401/ANJS.00.1.27]

Keywords: Viola-Jones, KLT algorithm, Blurring, Face detection, Face tracking.

1. Introduction

Human face detection 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, [1].

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 face tracking 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, [2].

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].

Viola Jones algorithm helps us to 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 first who introduced this algorithm. They made this algorithm mainly by the issue of face detection, [4].

After detecting the face we track it, object tracking is defined as keeping a trace on a particular kind of object, [5].

In this paper Kanade Lucas Tomasi algorithm (KLT) is used for feature tracking. It is the 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.

Then after detecting and tracking the face, blurring filter is applied using a circular averaging filter.

2. Related Work

Ritesh Boda, 2016 [6], in this paper an approach is presented 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.

J. Savitha, 2014 [7], the objective of this paper was to investigate the problem of hand tracking in high-speed videos. Specifically, the reason for this struggle is the physical structure of the hand, which allows significant appearance changes. To solve the problem, two main approaches were found S-PCA method and KLT method. Out of the approaches, KLT aim to track and recover the face which is the ultimate solution to the problem. Nevertheless, practical reasons have been making simpler and faster 2D tracking a more useful alternative in many applications.

Shruti Asmita, 2013 [8], this paper presents an algorithm to detect the human face from video file using the knowledge-based approach and track the motion of the individual by detection face in each frame. The algorithm can be used for developing secure PC camera and web camera.

3. The Viola-Jones Face Detector Algorithm

The viola-Jones is a widely used method for real- time object detection, it's training is slow, but the detection is very fast.

There are three components working in to enable a fast and exact detection:

1. The integral image for feature computation.
2. Ada Boost learning algorithm which is used for feature selection.
3. Method for combining classifiers in a "cascade" for efficient computational resource allocation, [9].

Features

The Viola-Jones algorithm uses Haar basis functions:

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, [9].

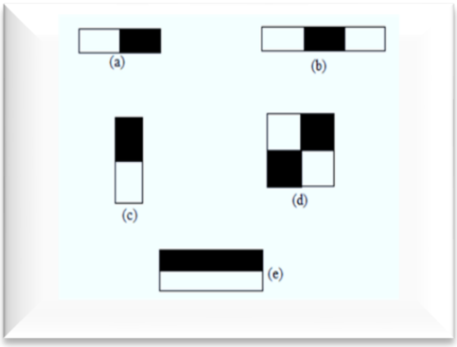


Figure (1) Rectangle Regions, [9]

These haar features are applied to determine the facial features. For example in the Figure (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 (d) is called a 4 rectangle feature, [10]. 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, [10].

3.1 Integral image:

The integral image at location x, y contains the sum of the pixels above and to the left of *x*, *y*, inclusive:

$$g(x,y) = \sum_{x'=0}^{x-1}\sum_{y'=0}^{y-1} f(x',y') \qquad \ldots(1)$$

Using the following pair of recurrences:

$$s(x,y) = s(x,y-1) + f(x,y)$$
$$g(x,y) = g(x-1,y) + s(x,y) \qquad \ldots(2)$$

where *s*(*x*, *y*) is the cumulative row sum, the integral image can be computed in one pass over the original image, [11].

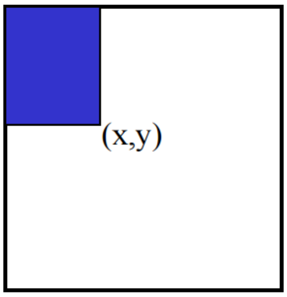


Figure (2) Integral Image, [11]

To compute sum within a rectangle:

- Let A, B, C, D be the values of the integral image at the corners of a rectangle
- Then the sum of original image values within the rectangle can be computed:

sum = A – B – C + D …(3)

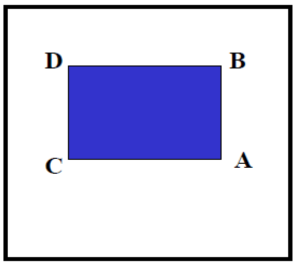


Figure (3) Integral image with A, B, C, D corners, [11]

- Only 3 additions are required for any size of rectangle!

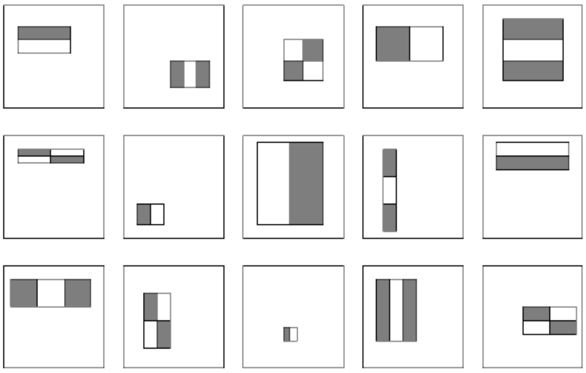


Figure (4) The possible rectangle features, [9]

3.2 Feature selection:

For a 24×24 detection region, the number of possible rectangle features is ~180,000, [12]. A very small number of these features can be combined to form an effective classifier. The main challenge is to find these features.

A variant of **AdaBoost** is used both to select the features and to train the classifier. The first two features selected by AdaBoost for the task of face detection are easily interpreted:

- First feature: the region of the eyes is often darker than the region of the nose and cheeks
- Second feature: the eyes are darker than the bridge of the nose.

So, the systems is essentially based on correlation, [9].

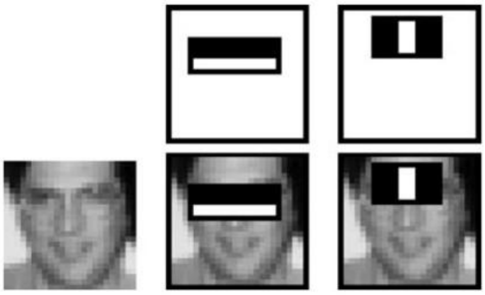


Figure (5) AdaBoost Features, [9]

3.3 Cascaded classifier:

We start with simple classifiers which reject many of the negative sub-windows while detecting almost all positive sub-windows

- Positive results from the first classifier triggers the evaluation of a second (more complex) classifier, and so on
- A negative outcome at any point leads to the immediate rejection of the sub-window, [12].

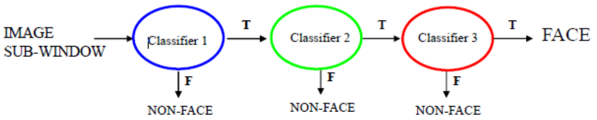


Figure (6) The cascaded classifier for face detection, [12]

The cascaded classifier for face detection has 38 layers and 6060 features:

- first classifier: two features, rejects about 50% of non-faces (while correctly detecting close to 100% of faces).
- second classifier: ten features, rejects 80% of non-faces.
- 3rd and 4th: 25 features.

4. Kanade Lucas Tomasi Algorithm

Kanade-Lucas-Tomasi (KLT) algorithm tracking approach calculate the movements of object in consecutive video frames when the image brightness constancy constraint is fulfilled and image movement is small, [13].

The 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, [6].

Face tracking method is categorized into two parts:

1. Detect a face.
2. Identify facial features to track.

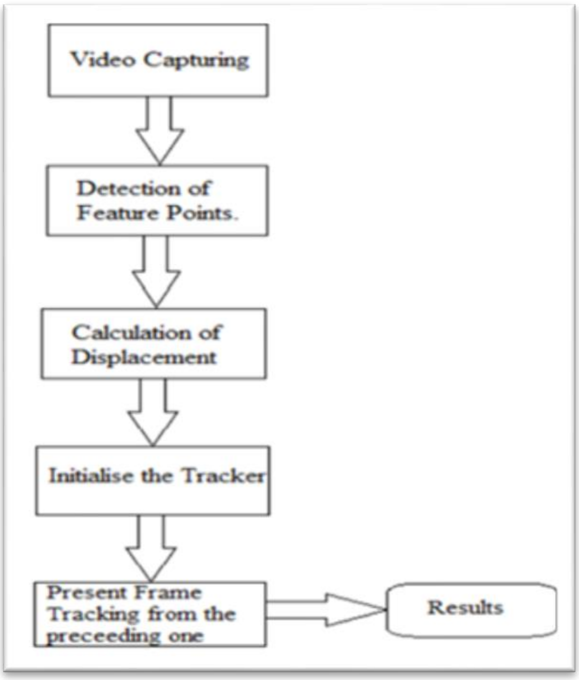

Video Capturing
Detection of Feature Points.
Calculation of Displacement
Initialise the Tracker
Present Frame Tracking from the preceeding one
Results

Figure (7) KLT Algorithm Flow Chart, [6].

4.1 Detect a face:

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, [6]. Here face detection is preliminary step for many other applications such as face recognition, video surveillance etc.

4.2 Identify facial features to track:

The KLT algorithm detects a set of object points across the video frames. Once the detection trace the face, the next step detects feature points that can be constantly tracked. The basic information has now been established to solve the displacement d of a feature from one window to the next, [12]. For simplicity, we redefine the second window as $B(x) = I(x,y,t+)$ and the first window as $A(x–d) = I(\mathrm{x}–d) = I(x–\Delta x,y–\Delta y,t)$ where $x = (\mathrm{x,y})$. The relationship between the two images is given by:

$$B(x) = A(x - d) + (x) \qquad \ldots(4)$$

where (x) is a noise function caused by interference. An error function which has to be minimized in order to minimize the effect of noise:

$$\text{€} = [A(x\text{-}d) - B(x)]\,[A(x - d) - B(x)]\, wdx \qquad \ldots(5)$$

As we see € is a quadratic equation of *d*. To minimize €, we need to differentiate € with *d* and set the result to 0.

$$d\text{€}/dd = [A(x) - B(x) - d]\, gwdA = 0 \qquad \ldots(6)$$

where *A* refers to area of window *W*. If the terms of 4 are rearranged and if we use the fact that $(d)g = (g)d$ it follows that:

$$(g\ wdA)d = [A(x) - B(x)]\, gwdA \qquad \ldots(7)$$

Using a similar derivation as for the KLT, Shi and Tomasi showed that the search can be performed using the formula $T_{\mathrm{Z}} = a$, where T_a matrix of gradients is, *Z,* is a vector of affine coefficients and *a*, is a vector of affine coefficients and $\nabla d = e$

$$I(x; y; t + \text{¢}t) = I(x + \text{¢}x; y + \text{¢}y; t) \qquad \ldots(8)$$

Let:

$$x = (x; y)T \ \text{ and } \ v = (\text{¢}x; \text{¢}y) \qquad \ldots(9)$$

In the presence of image noise

$$rI(x; t + \text{¢}t) = I(x + d; t) + r \qquad \ldots(10)$$

KLT will compute the displacement vector d that minimizes the following error [14]

$$r = XW(I(x + d; t)\ \text{¡}\ I(x; t + \text{¢}t)) \qquad \ldots(11)$$

5. Blurring

Blur is an optical feature of image in which can makes something to be unclear to observer.

It can be bothersome to human, as it provides difficulty for human to focus on the features of an image.

There are some methods that can be used to blur an image. Image have spatial domain and frequency domain where blurring can be performed on. Spatial domain is the domain where the space reveals the real image that can be seen, whereas the frequency domain displays the frequency of each pixel values that are present in the image.

In this research, a circular averaging filter is used to make blur on the face after detecting and tracking it.

h = fspecial (‘disk’, radius) returns a circular averaging filter (pillbox) within the square matrix of size 2×radius + 1.

6. The Methodology

In this research the tracking problem is divided into three parts:

1. Detect a face.
2. Identify facial features to track.
3. Track the face.

Then blurring the face after detecting and tracking it.

7. The Results

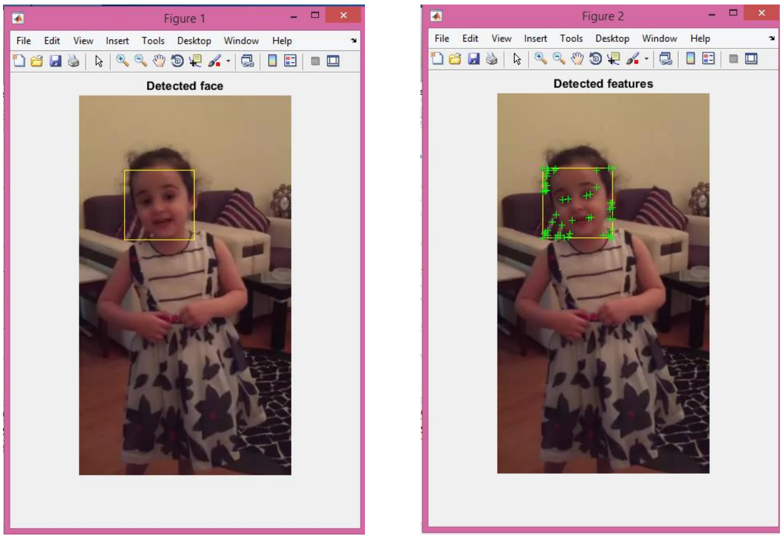
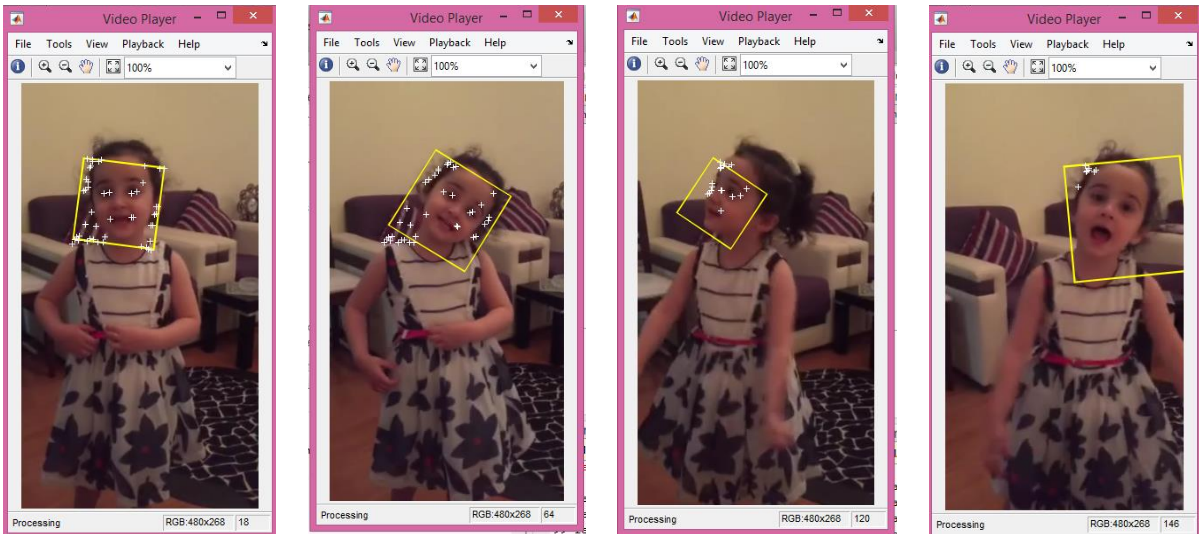


Figure 1
File Edit View Insert Tools Desktop Window Help
Detected face
Figure 2
File Edit View Insert Tools Desktop Window Help
Detected features

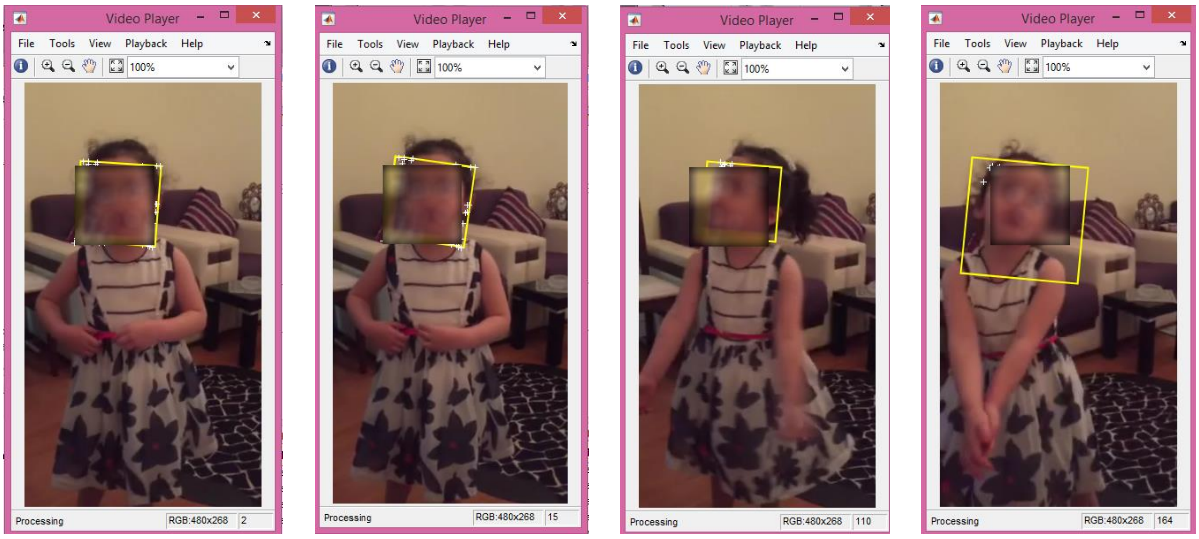
(a) (b)

Figure (8) (a) Detect the face; (b) Identify facial features to track.



(a) (b) (c) (d)

Figure (9) a,b,c,d Track the face.



(a) (b) (c) (d)

Figure (10) a,b,c,d Blurring.

8. Conclusion

In this paper, a simple face tracking system that automatically detects and tracks a single face is created, by using Viola-Jones detection algorithm for detection, KLT algorithm for tracking and then blurring the face.

By applying this method a very good results are obtained that can be useful to blurring the face for example, when showing a painful view that's include distorted faces in wars reports or crimes or even in coverage the witnesses faces.

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.

[2] 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.

[3] Dorin Comaniciu and Visvanathan Ramesh. 2011. “Robust Detection and Tracking of Human Faces with an Active Camera”. IEEE Visual Surveillance.

[4] L. Stan and Z. Zhang. 2004. “FloatBoost learning and statistical face detection”. IEEE Trans. On Pattern Analysis and Machine Intelligence. 26 (9).

[5] MAMATA S. KALAS. 2014. “Real Time Face Detection and Tracking Using OPENCV”. International Journal of Soft Computing and Artificial Intelligence.

[6] Ritesh Boda and M. Jasmine Pemeena Priyadarsini, 2016, “Face Detection And Tracking Using KLT And Viola Jones”, ARPN Journal of Engineering and Applied Sciences, VOL. 11, No. 23.

[7] J. Savitha, Dr. A.V. Senthil Kumar, 2014, “Face Tracking and Detection using S-PCA & KLT Method”, International Journal of Advance Research in Computer Science and Management Studies, Volume 2, Issue 2, ISSN: 2321-7782

[8] Shruit Asmita, Sugandha Agarwal, 2013, “Real Time Simple Face-Tracking Algorithm based on Minimum Facial Features”, International Journal of Computer Applications, Volume 79, No. 12.

[9] Richard Szeliski, 2010, “Computer Vision: Algorithms and Applications”, page 44, ©2010 Springer.

[10] Peter Gejguš and Martin Šperka. 2003. “Face tracking for expressions simulations”. International Conference on Computer Systems and Technologies.

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

[12] Nandita Sethi and Alankrita Aggarwal. 2011. “Robust Face Detection and Tracking Using Pyramidal Lucas Kanade Tracker Algorithm”. IJCTA, Vol. 2.

[13] Nandita Sethi and Alankrita Aggarwal, 2011, “Robust Face Detection and Tracking Using Pyramidal Lucas Kanade Tracker Algorithm”, IJCTA, vol. 2.

[14] Stan Birchfield, 2018, “Image Processing and Analysis”, ©2018 Cengage Learning, USA, ISNB: 978-1-285-17952-0.