Real Time Face Detection using VJCMS

C.D. Parmar, Nishanshi Shukla 121030, Rahul Patel-121040, Rajan Mehta-121043

***Abstract*—For real time detection of human face, various algorithms were studied. Approaches like Local Binary Patterns (LBP), Constrained Local Models (CLM) regularized by Landmark Mean Shift and Viola-Jones along with CAMSHIFT and KLT were implemented. Local Binary Patterns is an algorithm which considers various binary features across the human face and then detects them instead of the face as a whole. Constrained Local Models registers a parametrized shape model and then Landmark Mean Shift then tracks the consecutive locations of the face. Viola Jones algorithm uses Haar features to detect the face and CAMSHIFT and KLT are used for tracking the detected face. The results of these algorithms were compared and considering the strengths and weaknesses of each of them a combined approach using Viola-Jones and CAMSHIFT was to developed for real time face detection on video having high frame rate (29 fps) and a resolution of 360x638.**

***Index Terms*—Face detection, Viola-Jones, Constrained Local Model, Local Binary Pattern, CAMSHIFT.**

I. Introduction

The problem at hand was to detect human face in real time from a video of dimensions 360x638. In order to get equipped with the basic understanding of this domain, we began by surveying various research papers and algorithms like skin detection, motion detection, model based detection, Viola-Jones algorithm, CAMSHIFT, Local Binary Patterns, Constrained Local Models and various other state of the art algorithms. The face detection problem in a video becomes complex due to variability in human faces, the requirement of large training set and its dependencies on luminosity of surrounding, camera quality, color scale used, position, posture and the expressions of the face. Also the challenges of high data rate still remains. Different algorithms deal with a few of the above mentioned limitations. After coming across the limitations of all, it was clear that no single algorithm is fully equipped to optimally solve the problem in real time. So we clubbed a few detection and tracking algorithms and implemented them. The three algorithms that we implemented are Local Binary Patterns, Constrained Local Models regularized using Landmark Mean Shift and Viola-Jones along with CAMSHIFT. From these three, the final algorithm that we propose is Viola-Jones face detection along with CAMSHIFT tracking, as at this stage it satisfies the objective of dealing with high data rate and maximum detection rate simultaneously on the given video.

II. Various techniques implemented for face detection

A. Viola-Jones [1][2]

Viola-Jones Face Detection Algorithm is a Supervised form of learning where in which we give training by providing images of faces and non-faces. The basic principle of the Viola-Jones algorithm is to scan a sub-window capable of detecting faces across a given input image. The standard image processing approach would be to rescale the input image to different sizes and then run the fixed size detector through these images. This approach turns out to be rather time consuming due to the calculation of the different size images. Contrary to the standard approach Viola-Jones rescale the detector instead of the input image and run the detector many times through the image each time with a different size. At first one might suspect both approaches to be equally time consuming, but Viola-Jones have devised a scale invariant detector that requires the same number of calculations whatever the size. The algorithm combines the following four main concepts:

1) Haar Features
2) Integral Image
3) Adaboost
4) Cascading

B. Face Detection by Local Binary Pattern [3][4]

In LBP or local binary pattern, the value of an image is stored in binary form using a specific calculation. A NxN window, here 3x3, of surrounding pixels is taken, in order to calculate the value of center pixel. If the value of the surrounding pixel is more than or equal to the center pixel, it is replaced by one, and zero elsewhere, i.e. the value is calculated by using threshold. This binary value is now considered as a pattern or LBP code.

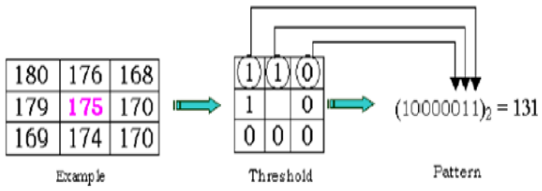

Fig. 1. Example of LBP calculation

After completion of all calculations, histogram is taken and each bin is considered as a micro-texton. Different types of patterns are codified by these bins.

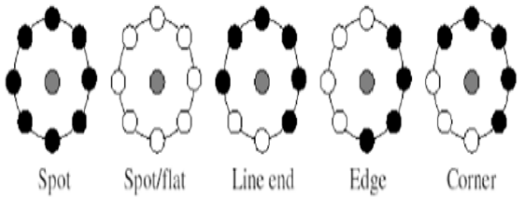

Fig. 2. Example of texture primitives

Some bins contain more information than the others, i.e. the bins with more number of bit transitions (0 to 1 or 1 to 0) in the pattern are more informative than the ones with no transitions at all. Patterns which have more than two transitions in a single bin are called LBP descriptors.

Each face is considered as a collection of micro-patterns which are to be detected by the LBP detector. So the face is divided into M non-overlapping regions and then histograms of all those are concatenated. This histogram describes shape of a face.

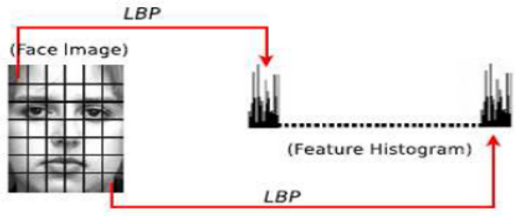

Fig. 3. LBP based facial representation

Gentle Adaboost is used to train the classifiers. The weak classifiers are selected such that they best differentiate the positive and the negative class. All of them are combined to form a strong classifier. Various stages in the form of cascade classifiers are used to enhance the performance. The later stages are stricter than the previous ones. A positive result from previous stage is evaluated again by the next stage which is set to have higher detection rate, while the negative ones are at ones rejected.

This is a detection scheme based on appearance and large training set is required. LBP features are simple and are able to detect face and non-face patterns. But the speed is not real time and the false positive ration is higher here.

C. Tracking human face using Constrained Local Models regularized by Landmark Mean Shift [5]

1) Constrained Local Model(CLM): Finding faces in an image is not easy, even difficult is finding position and shape of eyes, nose and mouth, etc. The job of CLM is to find these feature points, given a face image.

2) CLM Intuition:

- We already know what eyes/nose generally look like, otherwise we cannot find them even if they are in the patch of image. Or speak technically, we have a model of the look of eye/nose, and we use this model to search for them in a given patch of image.
- We also know the arrangement of eyes/nose on a face, eyes on top, nose in the center, etc. We can think of this as a constraint. This constraint is good for us, because when asked to search for nose in a face image, we do not need to go the length of searching the whole image, only the center or a close neighborhood of center (local region).

3) CLM Conceptual: We use CLM after Viola-Jones face detector. The output of V-J face detector is a rectangle containing a face. The task of CLM search is to find in this rectangle

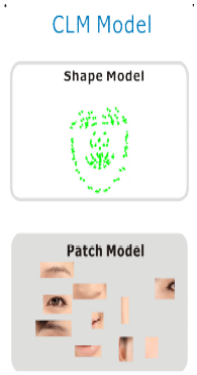

Fig. 4. Models in CLM

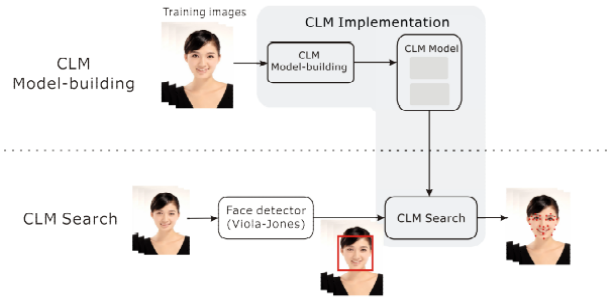

Fig. 5. Conceptual diagram of CLM Model and Search

the position of each feature point. We won't go into details of building model and CLM search method.

4) Tracking face in video: Model fitting is the problem of registering a parametrized shape model to an image such that its landmarks correspond to consistent locations on the object of interest. It is a difficult problem as it involves an optimization in high dimensions, where appearance can vary greatly between instances of the object due to lighting conditions, image noise, resolution and intrinsic sources of variability.

Most deformable model fitting methods employ a linear approximation to how the shape of a non-rigid object deforms, coined the point distribution model (PDM) by Cootes and Taylor (1992). It models non-rigid shape variations linearly and composes it with a global rigid transformation, placing the shape in the image frame:

$$x_i = sR(\bar{x}_i + \Phi_i q) + t$$

where x_i denotes the 2D-location of the PDMs i^{th} landmark and p = (s,R, t,q) denotes the PDM parameters, which consist of a global scaling s, a rotation R, a translation t and a set of non-rigid parameters q. Here, $\bar{x}_i$ denotes the mean location of the i^{th} PDM landmark in the reference frame (i.e. $\bar{x}_i = [\bar{x}_i; \bar{y}_i]$ for a 2D model) and Φ_i denotes the sub matrix of the basis of variations, Φ, pertaining to that landmark.

CLM fitting is generally posed as the search for the PDM parameters, p, that jointly minimizes the misalignment error over all landmarks, regularized appropriately:

$$Q(p) = R(p) + \sum_{i=1}^{m} D_i(x_i, l)$$

where R penalizes complex deformations (i.e. the regularization term) and D_i denotes the measure of misalignment for the i^{th} landmark at x_i in the image I (i.e. the data term). Example : R can be Gaussian Mixture Model and D_i is often chosen as the least squares difference between the template and the image. Since a landmarks misalignment error depends only on its spatial coordinates, an independent exhaustive local search for the location of each landmark can be performed efficiently.

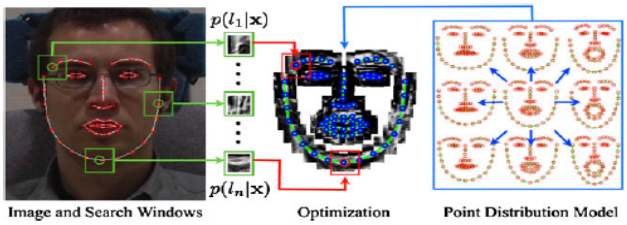

Fig. 6. Illustration of CLM fitting and its two components: (i) an exhaustive local search for feature locations to get the response maps and (ii) an optimization strategy to maximize the responses of the PDM constrained landmarks.

5) Regularized Landmark Mean-Shift Algorithm : The CLM objective in can be interpreted as maximizing the likelihood of the model parameters such that all of its landmarks are aligned with their corresponding locations on the object in an image.(Assuming conditional independence between detections for each landmark)

Algorithm : Require I and p

1) Compute Responses
2) while not_converged(p) do
3) Linearize shape model
4) Compute mean-shift vectors
5) Compute PDM parameter update
6) Update parameters: $p \leftarrow p + \Delta p$
7) end while
8) return p

6) KLT [6][7]: KLT algorithm is used for tracking an object. The process to realize tracking is the process to find out the position of the same object in image sequence. Tracking can be classified based on model base, feature base, contour base and area base. KLT tracking is feature based tracking algorithm which is invented by Teacher Kanade and his students lucas and Tomasi. KLT is based on intensity observation called optical flow based tracking. The motion of object can be described by the motion field in three dimensional space. In KLT first three dimensional object is converted to two dimensional image plane and motion of an object is segmented by distribution of grey scale in an image. Intensity of successive images is considered for movement tracking of an object. Steps for simple KLT algorithm

1) Detect Harris corners in the first frame
2) For each Harris corner compute motion (translation or affine) between consecutive frames
3) Link motion vectors in successive frames to get a track for each Harris point.
4) Introduce new Harris Points by applying Harris detector at every m(10 or 15) frames.
5) Track new and old Harris points using steps 1-3.

7) CAMSHIFT [6]: CAMSHIFT is built upon mean shift with the addition capabilities of working on dynamic distribution by readjusting the search window size for the next frame based on the zeroth moment of the current frame distribution. Its essential to know about mean shift and back projection to understand the phenomenon of window shifting in CAMSHIFT.

Mean shift: The algorithm gets initialized by a particular window size at a given position. The centroid of the data points lying inside the window is calculated and matched with the center of the window. If the centers do not match, the center of the window is moved to the centroid and mean is calculated again. This process is repeated until the center of the window and the centroid of the points lying inside the window coincides. Thus we ultimately lend up in that region of space having maximum density of the points w.r.t to the intial location.

Back Projection: For a given window on an image, it generates a matrix (2 D) having the same dimension as that of the image on which it is applied where each pixel in the matrix denotes the probability of that pixel lying inside the window. Hence, its an essential tool for image segmentation.

First of all the histogram of the pixels lying inside the window is calculated. This histogram of the window of the last frame is passed as an input to the back projection applied on the current frame. Thus we have a matrix where each pixel denotes the probability of falling inside the window of the previous frame in the current frame. The window is initialized from the location of the window in the past frame. To converge to a location having the maximum points with maximum probability of belonging inside the window we use mean shift. One point to note here is that that if more than one location is probable-having high probability of matching with the window in the previous frame- then the algorithm will choose the location which is closer from the window in the past frame. In other words, the window will converge to the local maxima. CAMSHIFT does all this along with dynamically changing the window size based on the zeroth moment.

III. Proposed Algorithm

As mentioned earlier there have been various approaches studied and implemented to counter the daunting task at hand. Out of these experiments two things clearly emerged out. First, the results of Viola Jones algorithm for face detection were best, yielding high detection rate and low false positive rate. The major setback using it was its speed; it was extremely slow to be implemented alone to detect faces in real time. Second, tracking of objects/faces is extremely fast as compared to detection algorithm. Looking at the speed of the tracking algorithm anyone will be tempted to use these algorithms. But the major drawback with them was it needed human intervention to point out which object to track. Also, once the object leaves the frame it starts generating erroneous results i.e. it may start tracking random objects in the frame.

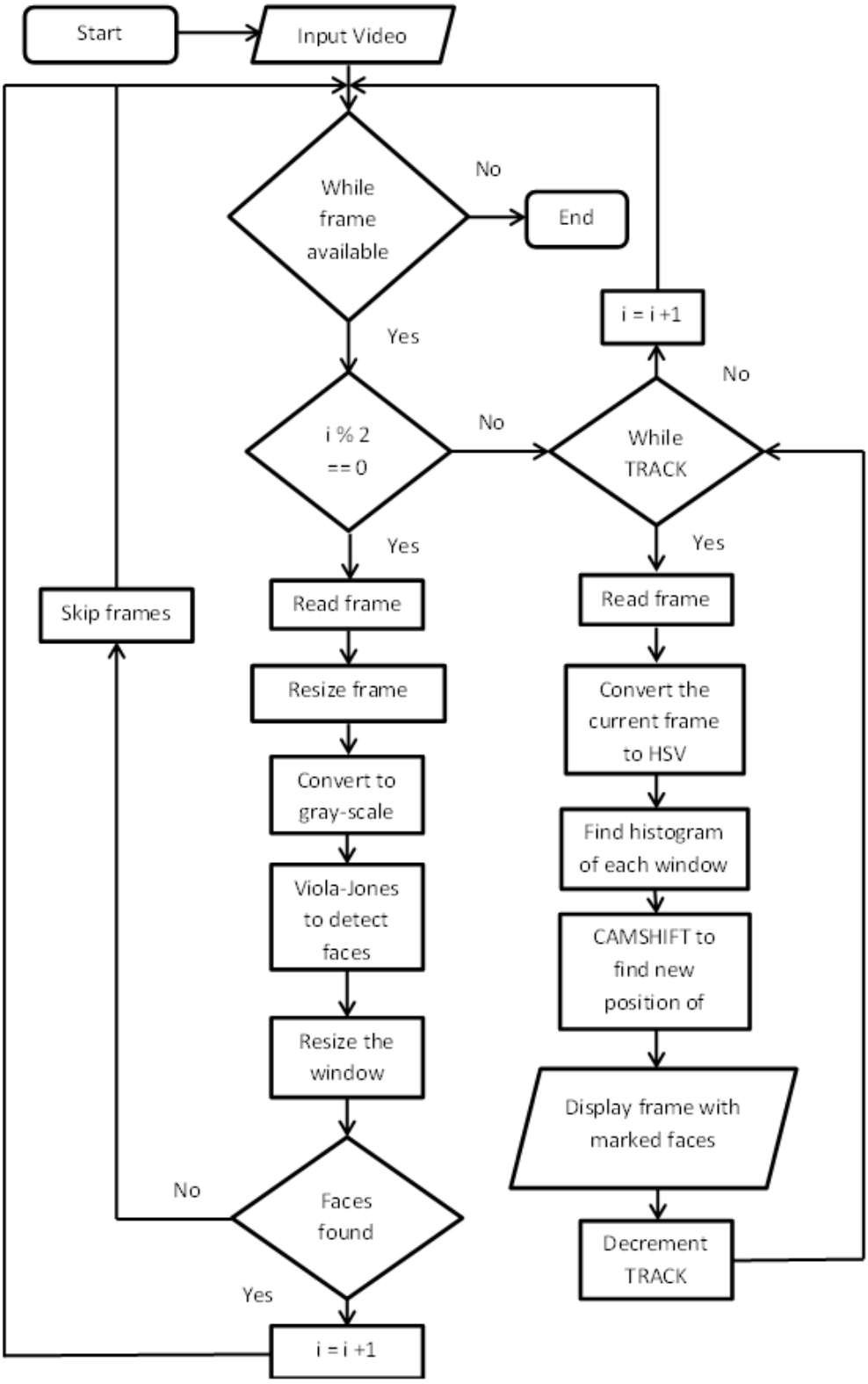

Fig. 7. This image shows the flowchart of the proposed algorithms combining Viola-Jones and CAMSHIFT

To remind you of the task at hand, we need to device an algorithm which can do real time face detection on videos (360x638 and 720x1280) with high frame rates (29 fps). Thus the two important parameters will be speed along with high detection rates and low false positive rate. The algorithms mentioned above have one of the intended features each for the required algorithm. Hence we chose Viola Jones face detection along with one of the tracking algorithms. The tracking algorithm we chose was CAMSHIFT. In our algorithm Viola-Jones face detection and CAMSHIFT tracking works in tandem and it forms the core of algorithm. There is certain pre-processing done before giving the frame to Viola-Jones for face detection. The frame is resized/compressed and converted to gray-scale. We then try to find faces using Viola-Jones face detection. It gives back the coordinates of the faces found in frame but these coordinates are with respect to the resized/compressed frame. Hence we need to map them to the original frame. The coordinates of the faces found acts as an input to the CAMSHIFT. For a certain number of upcoming frames we try to track those faces using the CAMSHIFT. If no faces are found then we skip certain number of frames by just displaying them as it is at the output and then again check for faces. A pass of execution is given to CAMSHIFT only when we detect faces.

Fig. 8. Unexpected increase in the window size observed for the subsequent frames whose window is initialized by Viola-Jones. Later on discovered that this increase in window size is subject to deviation of mean from the expected due to background pixels inside the window.

There is slight trick involved after mapping coordinates of faces found to the original frame and passing them to CAMSHIFT as input. One more manipulation needs to be done before passing these coordinates to CAMSHIFT. This manipulation is crucial for the success of this algorithm. The bounding box which we obtain from the coordinates found also contains a small amount of background pixel. These pixels disturb the mean of the actual object which we want to track, leading to abnormal window sizes enclosing the faces. Hence we need to further decrease the size of the enclosing window in such a manner that it discards as many unwanted pixels as possible and contains only the pixels of the object which we want to track. This help in better estimating the probability density function of the color of object which plays a crucial role in tracking the object.

IV. Results

A. Video8

The algorithm produces high detection rates and low false positive rates along with speed. Hence we can say that the primary objective is satisfied. One of the prime reasons for such high detection rates is, the orientations of faces are mostly frontal which helps Viola-Jones face detection algorithm to produce best output. Secondly, separation between the foreground and background color in the color space is also high which helps CAMSHIFT to better track the object.

Fig. 9. Comparison of results generated by algorithm for compression ration of 2:1 and 3:1 respectively. It can noted that for lesser compression higher detection is observed.

Also the detection of faces is depend upon the compression ratio. For compression ratio of 3:1 and 2:1 there was a significant increment in the detected faces along with degradation in the processing speed of the video. This fact will apply to all videos. The detection rate is inversely proportion to the processing

B. Benchmark

The algorithm produces extremely low detection rates and low false positive rates along with speed. Hence we can say that the primary objective is not satisfied. One of the prime reasons for such low detection rates are the orientations of faces which are mostly non-frontal. Secondly, the face is subjected to various objects like caps, glares, helmet, etc. and also to various emotions which makes it difficult for the algorithm to detect it as a face as the whole model of face is disturbed. Thirdly, separation between the foreground and background color in the color space is low in few cases which makes CAMSHIFT to erroneously track other object other than the desired one. Also, the movement of faces is high, hence it might be possible that the frames in which we are applying Viola-Jones, might not have a frontal face and hence the algorithm might skip certain frames, not able to detect faces.

V. Conclusion and Future Work

After going through various algorithms for face detection available today, we can say that there is a lot of scope of research in this domain as no algorithm is up to the mark. Each algorithm has its own set of benefits and drawbacks. Hence to optimally utilize the capabilities of them, we need be fully aware about the requirements of our system and the conditions or the environment in which the algorithms will be subjected to to better exploit them in our favor. Also, the performance depends upon the computational complexity of the algorithm, the environment(operating system), processing power and RAM of the computer.

Viola-Jones algorithm works great for frontal faces but is very slow and it also fails when subjected to non-frontal faces.CAMSHIFT tracking works great when the foreground and background color in the color space are not too close.

CLM built upon Viola-Jones takes into account various emotions and is also having good speed but it fails when subjected to sudden changes in the orientation of face. Also, currently it is restricted to track only one face. Hence, in the future we would like to put our efforts in CLM, making it more robust to sudden orientation changes and also increasing its ability to detect and track multiple faces. Probably we would like to devise an algorithm which tracking based on model rather then color.

REFERENCES

[1] Viola, Paul, and Michael Jones. "Rapid object detection using a boosted cascade of simple features." Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on. Vol. 1. IEEE, 2001.

[2] Viola, Paul, and Michael J. Jones. "Robust real-time face detection." International journal of computer vision 57.2 (2004): 137-154.

[3] Jo Chang-yeon. "Face Detection using LBP features."

[4] T. Ojala and M. Pietikainen. Multiresolution Gray-Scale and Rotation Invariant Texture Classification with Local Binary Patterns, IEEE Trans on Pattern Analysis and Machine Intelligence, Vol. 24. No.7, July, 2002.

[5] Saragih, Jason M., Simon Lucey, and Jeffrey F. Cohn. "Deformable model fitting by regularized landmark mean-shift." International Journal of Computer Vision 91.2 (2011): 200-215.

[6] Suhr, Jae Kyu. "Kanade-Lucas-Tomasi (KLT) Feature Tracker." Computer Vision (EEE6503) (2009): 9-18.

[7] Detection and Tracking of point features ,Technical Report, CMU-CS-91-132 Karlo Tomasi,Tokyo Kanade, April 1991

[8] Bradski, G.R., Real time face and object tracking as a component of a perceptual user interface, Applications of Computer Vision, 1998. WACV 98. Proceedings., Fourth IEEE Workshop on , vol., no., pp.214,219, 19-21 Oct 1998