

Real Time Simple Face-Tracking Algorithm based on Minimum Facial Features

Shruti Asmita

BTech(CSE) KEC,Ghaziabad
MTech Scholar(Computer Sc.)
Banasthali University

Sugandha Agarwal

BTech(CSE) KEC,Ghaziabad
MBA Scholar
IMT Nagpur

Pramod Kumar Sethy

KEC, Ghaziabad
MTech ,Indian Schools of
Mines, Dhanbad

ABSTRACT

This paper presents an algorithm to detect the human face from a video file using the knowledge-based approach and track the motion of the individual by detecting face in each frame. The algorithm detect human face by the geometric correlations between location of face and hairs in each frame of a video file. Range of skin color are used to figure out possible face regions so as to initially localize the face, furthermore, the probable hair blocks in an image frame are determined by means of hair color spectrums. Combined skin and hair blocks decide candidate face areas in light of the geometric relation. The phase correlation motion estimation algorithm is used to analyzing successive frames in a video sequence to identify faces that are in motion and track the human faces from the video file.. The accuracy of single-face tracking is higher than 90% with the frame rate of 10fps. Several algorithms have already been proposed and developed for various applications and employed successfully. But, those algorithms are quite complicated and hard to meet the real-time requirements of specific frame-rate. Therefore, the proposed is able to be expectedly transplanted to an embedded system, like the developing pet robot so as to perform dynamic face detection and tracking. The algorithm can be used for surveillance. The algorithm can be used for developing secure PC camera and web camera. The algorithm is being used for providing laptop security.

General Terms

Image Processing, algorithms.

Keywords

Skin Quantization, Hair Quantization, Motion Vector ,Motion Estimation, Block Matching

1. INTRODUCTION

Human face detection and tracking have been importantly developed recently—surveillance systems, digital monitoring, intelligent robots, notebook, PC cameras, digital cameras, 3G cell phones, and the like. These applications accordingly play an important role in our day to day life. However , the algorithms of the applications are quite difficult to analyze and hard to meet real-time requirements of specific frame-rate.

Over the last few decade much research has been done for improving the performance of human face detection and tracking. Before you begin tracking a face, you need to first detect it. Detection method are categorized into many types (1)Cascade Object Detector to detect the location of a face in a video frame. The cascade object detector uses the Viola-Jones[15] detection algorithm and a trained classification model for detection. By default, the detector is configured to

detect faces, but it can be configured for other object types. (2) Knowledge-based method: This method is directed at finding constant features of a face within a complex environment, thereby localizing the position of the face. Relationships among the features helpfully determine whether a human face appears in an image or not [1]. (3)Feature invariant approaches: Invariant features, unresponsive to different positions, brightness, and viewpoints, are utilized in this approach to detect human faces. A statistical model is usually built up for describing the relations among face features and the presence of the detected faces. Such face features, for instance, are Facial Features [2], Texture [3], and Skin Color [4]. (4)Template matching method: A template cohering with human face features is used to perform a pattern-matching operation based on the template Model [6] are common examples of this method. (5)Appearance-based method: This method, such as Eigen face [7], Neural Network [8], and Hidden Markov Model [9], employs a series of face images to train and establish a face model for the face detection. In general, method (1), (3),(4) and (5) are more complex than method (2); yet the more features are used in method (1), the more complicated it is.

This paper presents a complexity reduce algorithm which is designed for detecting human faces in real-time as proposed. This algorithm, on the basis of Knowledge-based method with minimum features, follows the geometric characteristics of skin and hair color to detect human faces and for tracking Motion estimation algorithm is used. Also this algorithm is able to be expectedly transferred to developing to perform dynamic face detection and tracking. Different Motion estimation algorithms are there like (a)Exhaustive Search Block matching algorithm (b)Fast Algorithm (c)Hierarchical Block Matching Algorithm and (d) Phase Correlation Method. In this paper the Phase correlation motion estimation algorithm is used for tracking.

2. PROBLEM STATEMENT

Face tracking is a critical prior step that localizes the region of the face in video frames, from which a relevant feature set can be extracted and subsequently served as input to the face recognizer. As such, the accuracy of tracking directly impacts the ability to recognize subjects in video.

Visual tracking of objects of interest, such as faces, has received significant attention in the vision community. Accurate tracking is made difficult by the changing appearance of targets due to their non rigid structure, 3D motion, interaction with other objects and changes in the environment, such as illumination.

Recent tracking methods, such as the Incremental Visual Tracker, attempt to solve these problems using adaptive target appearance models. They represent the target in a low-dimensional subspace which is updated adaptively using the images tracked in the previous frames. Compared to the approaches equipped with a fixed target model such, IVT is more robust to changes in appearance (e.g., pose, illumination). However, the main drawback of the adaptive approaches is their susceptibility to drift: they can gradually adapt to non-targets as the target model is built solely from the previous tracked images accepted by the tracker. Methods such as IVT typically lack mechanisms for detecting

3. APPROACH

The propose algorithm consists of main modules which are as follows (1) Face Detection in each frame (2) Tracking the human face from video file.

Detailed descriptions of each module are as follows:

3.1 Face Detection

Face Detection is achieved by using Algorithm Based on Minimum Facial Features. The Steps of the Algorithm are as follows: Taking color information we have to extract features of skin color and hair from a complex background. The flowchart of the proposed algorithm is shown as Figure 3: Face detection modules again separated in to five modules 1(a)Skin Detection: Using color information to detect possible skin color in an image ,1(b) Hair Detection: Utilizing brightness to out where hair probably is1(c) Skin Quantization: Quantizing skin color pixels and identifying blocks of the skin 1(d)Hair Quantization: Quantizing hair color pixels and identifying blocks of the hair 1(e) Get the Fusion of Features: Determining whether the detected skin area is a part of a human face according to the relative positions between skin and hair regions.

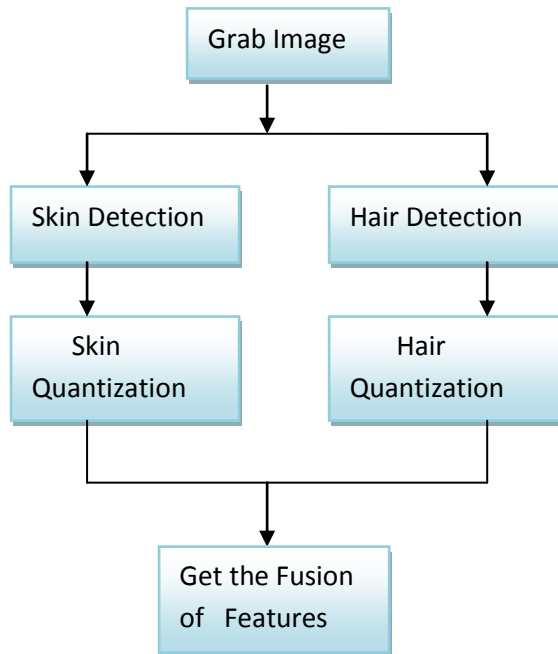


Fig 1: An overview flowchart of Face Detection Algorithm

3.2 Skin Detection

There are many studies have been involved in describing the range of skin colour in an model without normalization was sensitive to variations of light, so that The RGB model was

therefore transformed to the Normalized RGB model. The formulas for the transformation are listed as equation 1 and 2:

$$r = \frac{R}{R + G + B} \quad (1)$$

$$g = \frac{G}{R + G + B} \quad (2)$$

Using these two equation the RGB model transferred to normalized RGB model .Equation (1) represents the normalization of a red pixel while equation (2) stands for that of a green pixel.

These two equations declared above accordingly specify the upper limit $F_1(r)$ and lower limit $F_2(r)$ of the skin color (through analyses of experiments) [11]:

$$F_1(r) = -1.376r^2 + 1.0743r + 0.2 \quad (3)$$

$$F_2(r) = -0.776r^2 + 0.5601r + 0.18 \quad (4)$$

White colour ($r = 0.33$ and $g = 0.33$) is also in the defined range, so the following condition is added for excluding the white colour:

$$w = (r - 0.33)^2 + (g - 0.33)^2 > 0.001 \quad (5)$$

Together equation (3),(4) and (5),the skin color range specified

$$Skin = \begin{cases} 1 & \text{if } (g < F_1(r) \cap g > F_2(r) \cap w > 0.001) \\ 0 & \text{otherwise} \end{cases} \quad (6)$$

Here is another hues element of the robust HSI colour model[12][13] is taken as a new measure for approximating skin colour pixels so as to improve the above insufficiency. The following exemplify the relation between the RGB colour model and the HSI colour model.

$$\theta = \cos^{-1} \left\{ \frac{0.5[(R - G) + (R - B)]}{\sqrt{(R - G)^2 + (R - B)(G - B)}} \right\}$$

$$\begin{cases} H = \theta & \text{if } B \leq G \\ H = 360^\circ - \theta & \text{if } B > G \end{cases} \quad (7)$$

Modified skin detection condition after experiment equation (8) shows

$$Skin = \begin{cases} 1 & \text{if } \left(g < F_1(r) \cap g > F_2(r) \cap w > 0.001 \cap \right. \\ & \left. (H > 240 \cup H \leq 20) \right) \\ 0 & \text{otherwise} \end{cases} \quad (8)$$

Figure 2 is the result of skin detection of the image dealt with by the equation 8.

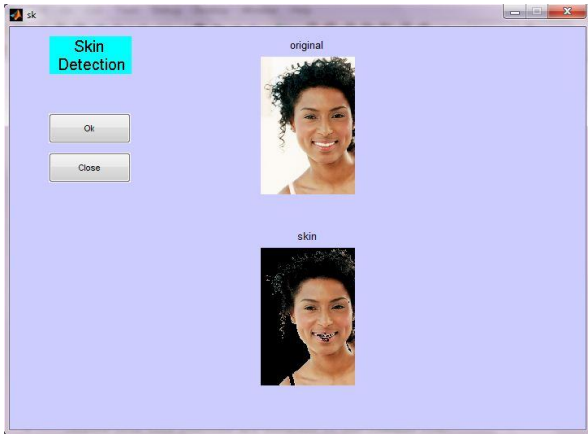


Fig 2: Result after skin detection

3.3 Hair Detection

The intensity element of HSI colour model is used for detecting the hair colour. The relation between the intensity element and RGB elements is as follows:

This intensity element is available for measuring the brightness of an image; moreover, the hair colour is specified

$$I = \frac{1}{3} (R+G+B) \quad (9)$$

to be in the range of dark. Equation 10 specifies the intensity range of the hair colour based on experiments.

$$Hair = \begin{cases} 1 & \text{if } \left\{ \begin{array}{l} (I < 80 \cap (B - G < 15 \cup B - R < 15)) \\ \cup \\ (20 < H \leq 40) \end{array} \right. \\ 0 & \text{otherwise} \end{cases} \quad (10)$$

The condition $(B - G < 15 \cup B - R < 15)$ closes out the pixels which are easily categorized as deep blue while hair colour pixels commonly and statistically match the condition $(B - G < 15 \cup B - R < 15)$. The condition $(I < 80)$ includes the pixels which are dark. The condition $(20 < H \leq 40)$ includes the pixels which are brown. Figure 6 shows that many non-hair pixels are misjudged, yet the Hair Quantization module performs the follow-up operations of the whole algorithm so as to minimize the impact from the misjudging.

Figure 3 shows image processed by the equation 10. Non-hair color pixels are replaced by white pixel.

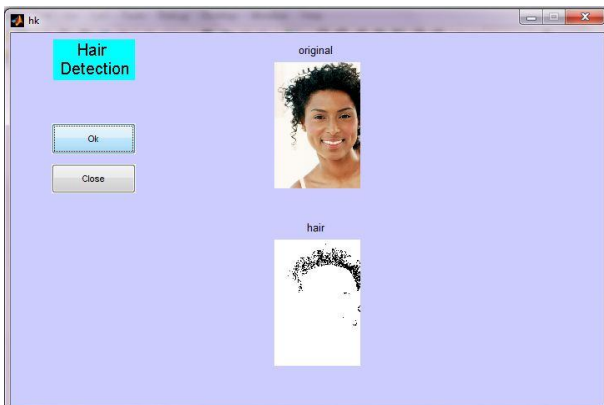


Fig 3: Result after Hair detection

3.4 Skin Quantization

This module quantizes skin color pixels and uses a 3x3 pixel square to express whether or not a region included pixels of skin color. The quantization lowers the image resolution; nonetheless, the following calculation of the geometric locations of pixels is speeded up. The number of black pixels within 9 pixels is counted and these 9 pixels regarded as non-skin color blocks if the number is beyond a threshold value 4. The rest are viewed as skin color blocks. demonstrates the image of after the Skin Quantization. Figure 4 ,Skin quantization demonstrates the image of Figure 2.

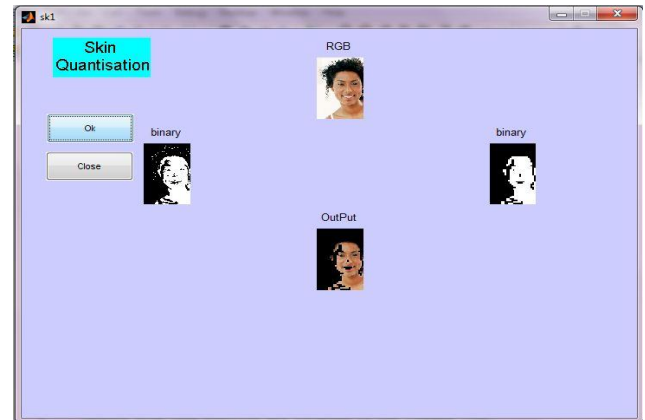


Fig 4: Skin Quantization

3.5 Hair Quantization

This module, similar to the Skin Quantization, quantizes hair color pixels and uses a 3x3 pixel square to express whether or not a region included pixels of hair color. The number of white pixels within 9 pixels is counted and these 9 pixels considered as non-hair color blocks if the number is beyond a threshold value 4. The rest are treated as hair color blocks.

Figure 5 displays the image of Figure 3 processed by the Hair Quantization.



Fig 5: Result of Hair Quantization

3.6 Get the Fusion of Features:

This module, mentioning to the results from Skin Quantization, and the result from Hair Quantization and determines the existence of a human face as well as its relation. Figure 6 illustrates the flowchart of the human face detection

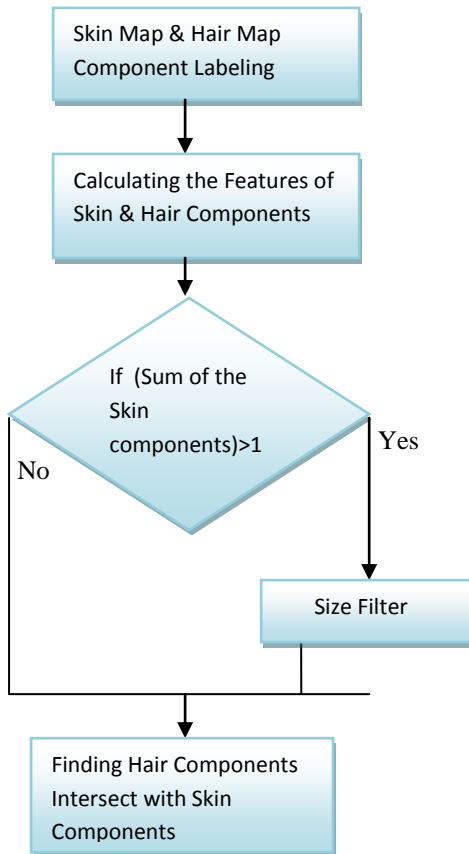


Fig 6: Flowchart of single human face detection

In addition the result from Skin Quantization and Hair Quantization is calculated using Component Labeling[14]. Adjacent pixels are defined as one component; two components have no adjoining pixels on the boundary. Component Labeling, applying 8 connectivity method, is able to search out all components in an image and get each of them labeled.. Calculating the Features of Skin & Hair Components:

The detection algorithm in Figure 6 calculates the features for all presently labeled components. For a component, features to be calculated are area, center of gravity (COG), and the coordinates of the extreme left, top, right, and bottom pixels, respectively.

Size Filter[14] is used to filter out noises and to conserve main skin features. The filtering conditions can be changed according to necessities such as the area of a component,. Only one component is required to detect a human face. Among all the components only the largest component is preserved while the other small components are excreted.

Finding Hair Components Intersect with Skin Component :
The step defined here to find hair components that intersect with the skin component. The presence of human face requires that hair components and skin component intersect. Thus, when there is no hair component intersecting with skin component, there is no human face present in the picture. From practical observations, the most common twelve hair styles that we would like to detect are shown in Figure 7.

To find out whether the skin and hair components are extracted from a human face, an estimation of the joint

geometric relationship. Figure 7 shows the geometric relation between a skin component and a hair component

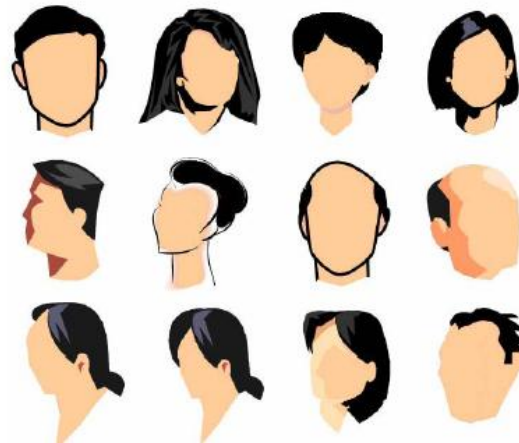


Fig 7 Twelve hair styles like to detect

The twelve complex geometric relations are simplified into some simple mathematical operations, The estimate is done by the joint geometric relationship using the “minimum bounding box” approach. The minimum bounding box is calculated for each skin component as well as each hair component. If the minimum bounding box of a skin component intersects with the minimum bounding box of a hair component, they are “related”. Observe that the coordinates of each minimum bounding box has already been obtained in the step “calculating the features of Skin and Hair component”. The intersection of skin bounding box and hair bounding box is shown in Figure 8.

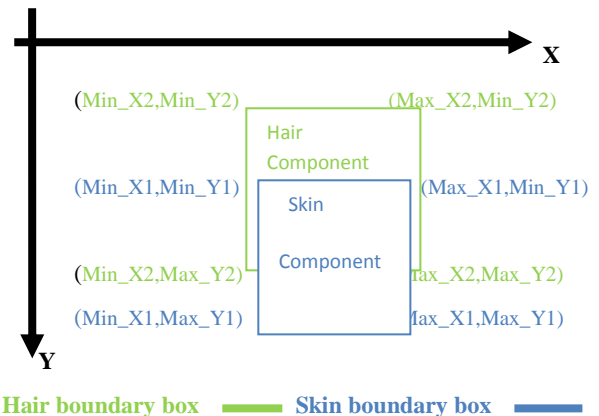


Figure 9 summarizes the various intersection relations between skin and hair bounding boxes of Figure 7.

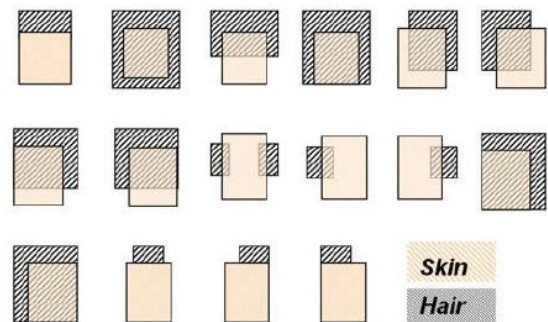


Fig 9: Intersection relations between hair and skin bounding boxes.

Observe that not all intersection relationships are possible from a human face. Restrict the intersections to be the ones shown in Figure 9. That is, a hair component is associated with a skin component only if the hair bounding box intersects with skin bounding box according to one of the relations shown in Figure 9. If one hair component is associated with a skin component, the operation for finding hair component is thus finished. Figure 10 shows the result of the face detection

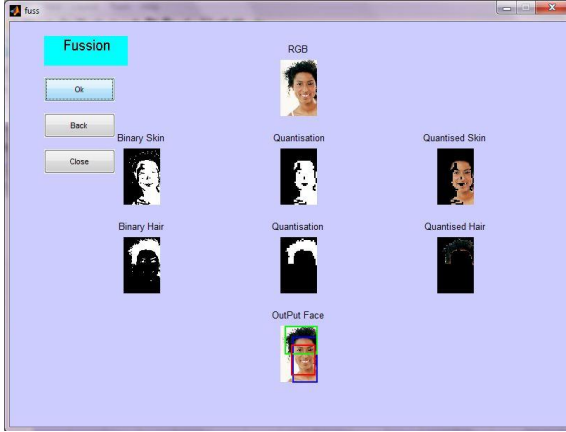


Fig. 10: Result of face detection

3.2 FACE TRACKING

Once human face is being detected using above algorithm, then track the human face from the video file using Phase Correlation Method motion estimation algorithm. Before this algorithm we have described the general methodologies in motion estimation criteria

3.2.1 General Methodologies in Motion Estimation Criteria

Motion estimation is the process of studying successive frames in a video sequence to identify faces that are in motion. The motion of a face is usually described by a two-dimensional (2D) motion vector (MV), which encodes the length and direction of motion.

The 2D motion estimation techniques described in this paper address the problem of estimating the motion between two frames: $f(x, y, t_1)$, which will be called the anchor frame, and $f(x, y, t_2)$, which will be referred to as the target frame. The anchor frame may be either before or after the target frame in time. It will call forward motion estimation the case where $t_2 > t_1$ and backward motion estimation the case where $t_1 > t_2$. A motion vector $d(X) = [d_x d_y]^T$ will represent the displacement of a particular point $X = (x_i, y_i)$ between time t_1 and t_2 . Figure 14 illustrates these concepts and associated notation.

A Feature-based Motion Estimation approach, Establish a correspondence between pairs of selected feature points in the two frames and attempt to fit them into a motion model using methods such as least-squares fitting. Feature-based methods are more often used in applications such as face tracking and 3D reconstruction from 2D video.

Once a model and region of support have been chosen, the problem shifts to the estimation of the model parameters. One of the most popular motion estimation criteria and the only one described in this paper is based on the displaced frame difference (DFD). It consists in computing the differences in intensity between every point in the anchor frame and the corresponding point in the target frame. Mathematically, the objective function can be written as

$$E_{DFD}(d) = \sum_{x \in R} (f_2(X + d) - f_1(X))^p \quad (11)$$

where $f_1(x)$ is the pixel value at x in time t_1 , $f_2(x + d)$ is the pixel value at $(x + d)$ in time t_2 , d is the estimated displacement, and R is the region of support.

When $p = 1$, the error $EDFD(d)$ is called the mean absolute difference (MAD), and when $p = 2$, it is called the mean squared error (MSE).

The error functions described by equation can be minimized using multi resolution search. This method searches the motion parameter starting at a coarse resolution and propagating the partial solutions to finer resolutions, which can then be successively refined. It is a good trade-off between speed and does not suffer from the problem of being trapped into a local minimum.

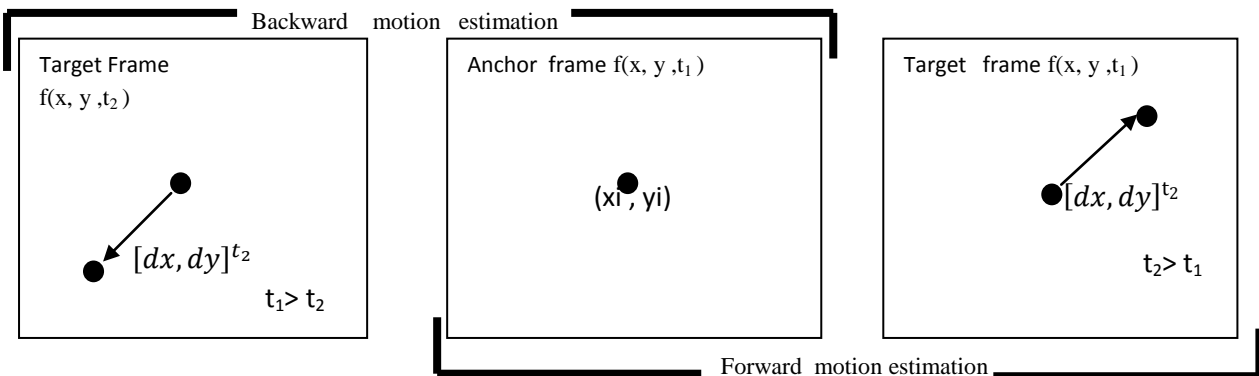


Fig 11: Anchor and target frames in forward and backward estimation

3.2.2 Phase correlation Motion Estimation Algorithm

The phase correlation method is a frequency-domain motion estimation technique that uses a normalized cross-correlation function computed in the frequency domain to estimate the relative shift between two blocks (one in the anchor frame, another in the target frame). The basic idea consists in performing spectral analysis on two successive fields and computing the phase differences. These differences undergo a reverse transform, which reveals peaks whose positions correspond to motions between fields.

1. Compute the discrete Fourier transform (DFT) of each block in the target and anchor frames (which we will denote frames k and $k+1$)

2. Calculate the normalized cross-power spectrum between the resulting DFTs according to equation (12):

$$C_{k,k+1} = \frac{F_k + 1F_k^*}{|F_k + 1F_k^*|} \quad (12)$$

where F_k and F_{k+1} are the DFT of two blocks at the same spatial location in frames k and $k+1$ and $*$ denotes the complex conjugate.

3. Compute the phase correlation function, PCF(x), defined as the inverse DFT of $C_{k,k+1}$, as follows:

$$PCF(X) = F^{-1}(C_{k,k+1}) = \delta(X + d) \quad (13)$$

where d is the displacement vector.

4. Locate the peaks of the PCF.

3.2.3 Tracking and Matching

At the “tracking and matching” stage, the face detected in the current frame are compared with the face currently being tracked from previous frames in order to answer the following questions: Is any face new to the scene?, has any face left the scene? and where did the faces currently being tracked go?

When a new face is discovered (i.e., one for which no current face match), this face is placed in a “three-frame buffer.” The buffer is necessary to dismiss false alarms caused, for example, by noise artefacts introduced in the early stages of segmentation. If the same face is successfully tracked over three frames, it is then removed from the buffer and entered in the “active tracking list.” Any face that is currently in this list is considered as an object officially being tracked, and any tracking information collected here will be reported in the final output. Figure 12 illustrates the process.

To determine the new position of an existing face A (which is contained in the active tracking list), we consider any face in the current frame whose centroid falls within the bounding box of face A as a possible candidate. If only one face

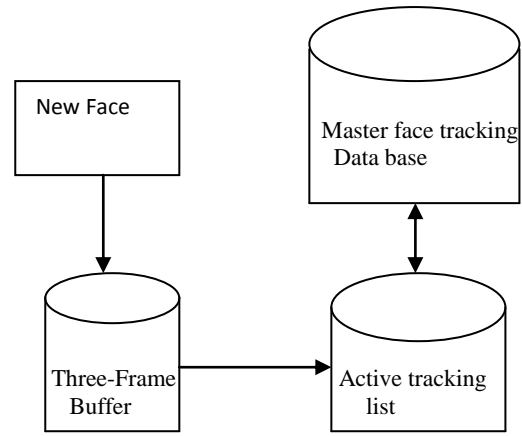


Fig 12: keeping track of existing and candidate faces.

meets this criteria, then consider it a match, run an additional test to ensure that face cannot be considered a possible candidate for any other faces (since we now know what that face is), and update the tracking data for face A . If there are more than one possible candidate for the new position of face A , we then compare face sizes, and that with the closest size to face A is considered a match. Finally, if at any given frame a face cannot be matched up with a face in the subsequent frame, that face is removed from the “active tracking list,” although its history remains in the “master face tracking database”

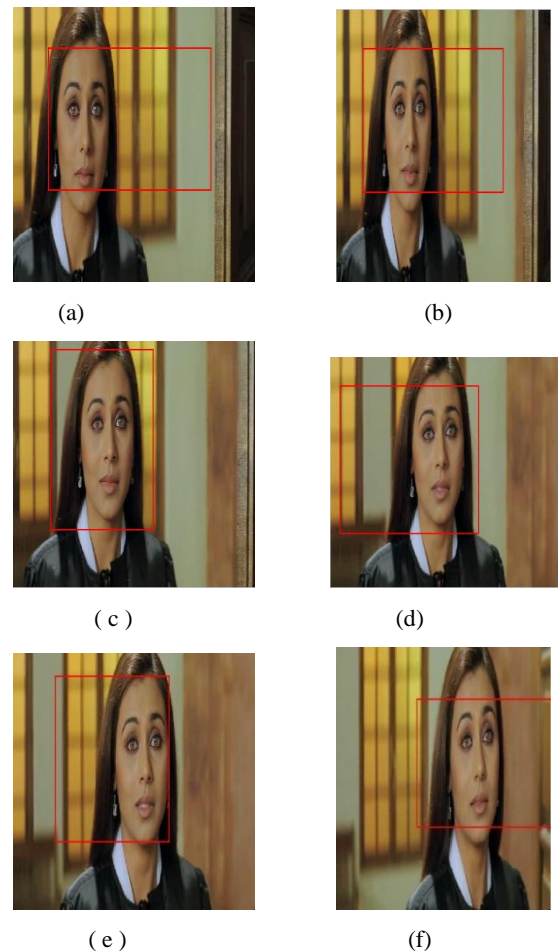


Fig 13:(a),(b),(c),(d),(e) and (f) are Face Tracking from the video file

In this process, the proposed algorithm detects the face of the person present in the video and then tracks the movement of the same person. Figure 13 shows, the detection and tracking of a person whose movement is being made in the direction from left to right (the movement of the red-square (detected face) is visible to us).

4. CONCLUSION

A more reliable face detection and tracking approach based on the geometric relations between skin of human face and hair. The algorithm also takes the advantage of an exact skin color extraction. The frame rate is up to 10 fps on a digital computer having 1.4GHz processor. For each face, the system keeps track of its size, position, velocity, and color. The size information can be used to separate into application- and view point dependent classes, for example, large, medium, and small. Similarly, brightness and color information can be used to classify objects into categories such as "bright" or "red." Position is reported as the instantaneous pixel position of the centroid of the object within the current frame. Similarly, velocity is reported as an instantaneous change in position between consecutive frames, which can be easily calculated since we know the frame rate of the sequence. The proposed algorithm can be extended to perform multi-face detections and tracking. The difference between single-face and multi-face detections is in the operation of the Size Filter. For single-face detection we only preserve the largest skin component. For multi-face detection, to keep any component larger than 10 pixels. Find the hair bounding box intersecting with for each such skin bounding box, to determine if a corresponding human face exists.

5. ACKNOWLEDGMENTS

We are thankful to Mr. Baijnath Kaushik for his valuable guidelines and suggestions which help us in improving the quality of the paper in a systematic way. We are also thankful to the anonymous reviewers for providing constructive suggestions which led to the overall improvement of the paper.

6. REFERENCES

- [1] Jerome, M. S. Shapiro, 1993. "Embedded Image Coding Using Zero-tress of Wavelet Coefficients", IEEE Transaction on Signal Processing Vol.41No.12.
- [2] T.K.Leung, M.C.Burl, and P.Perona, 1995 "Finding Face in Cluttered Scenes Using Random Labeled Graph Matching", Fifth IEEE Int'l Conf.Computer Vision, 1995, pp637-644..
- [3] Y. Dai and Y.Nakano, 1996 "Face-Texture Model Based on SGLD and Its Application in Face Detection in a Color Scene", Pattern Recognition, vol. 29, no. 6, 1996, pp.1007-1017.
- [4] J. Yang and A.Waibel, 1996 "A Real-Time Face Tracker", Third Workshop Applications of Computer Vision, 1996, pp. 142-147.
- [5] I. Craw, D. Tock, and A. 1992 Bennett, Proc. Second European Conf. Computer Vision, 1992, pp. 92-96. "Finding Face Features,"
- [6] A. Lanitis, C.J. Taylor, and T.F. Cootes, 1995 "An Automatic Face Identification System Using Flexible Appearance Models", Image and Vision Computing, vol. 13, no. 5, 1995, pp. 393-401.
- [7] M. Turk and A. Pentland, 1991 "Eigenface for Recognition", J.Cognitive Neuroscience, vol.3, 1991, pp.71-86.
- [8] H. Rowley, S. Baluja, and T. Kanade, 1998 "Neural Network-Based Face Detection," IEEE Trans. Pattern Analysis and Machine Intelligence, vol.20, no.1, Jan. 1998.
- [9] A. Rajagopalan, K. Kumar, J. Karlekar, R. Manivasakan, M. Patil, U. Desai, P. Poonacha, and S. Chaudhuri, 1998 "Finding Faces in Photographs", Proc. Sixth IEEE Int. Conf. Computer Vision.
- [10] Linda G. Shapiro, George C. Stockman, 2001 "Computer Vision", Prentice Hall, pp.192-193.
- [11] M. Soriano, S. Huovinen, B. Martinkauppi, and M. Laaksonen, 2000 "Using the Skin Locus to Cope with Changing Illumination Conditions in Color-Based Face Tracking," Proc. of IEEE Nordic Signal Processing Symposium, pp. 383-386.
- [12] Rafael C. Gonzalez, Richard E. Woods, 2002 "Digital Image Processing", 2nd Edition, Prentice Hall, pp.299-300.
- [13] William K. Pratt, 2001 "Digital Image Processing", 3rd Edition, John Wiley & Sons, pp.63-87.
- [14] Ramesh Jain, Rangachar Kasturi, Brian G. Schunck, 1995 "Machine Vision", McGraw-Hill, pp.44-48.
- [15] Viola, Paul A. and Jones, Michael J, 2001 "Rapid Object Detection using a Boosted Cascade of Simple Features", IEEE CVPR.
- [16] Yao-Jiunn Chen and Yen-Chun Lin, 2007 "Simple Face-detection Algorithm Based on Minimum Facial Features", IEEE Industrial Electronics Society (IECON) Nov. 5-8.
- [17] Oge Marques, 2011 "Practical Image and Video Processing using MATLAB", A John Wiley & Sons Inc Publication, pp.561-578.