Research Article

AUTOMATED FACIAL ANIMATION USING MARKER POINT FOR MOTION EXTRACTION

Mehran Sayed [1], Zeeshan Bhatti[1], Azar Akbar Memon [1], Zia Ahmed Shaikh[1], Ahmed Muhammad Sheikh[2], Nisar Ahmed Memon [1]

[1]A.H.S Bukhari Postgraduate Institute of Information and Communication Technology, University of Sindh, Pakistan, [2] Cleveland State University, United States of America

Correspondence:
Zia Ahmed Shaikh
A.H.S Bukhari Postgraduate Institute of Information and Communication Technology, University of Sindh, Pakistan
Email: ziaahmedshaikh@gmail.com
DOI: 10.38106/LMRJ.2024.6.4-07
Received: 23.09. 2024
Accepted: 21.12.2024
Published: 31.12.2024

ABSTRACT

In this research work, an automated 3D face expression generation technique is presented, which is extracted from real life video of face motion. The face expression is extracted from real human face using Huff-transform algorithm to gate the value of x coordinate and y coordinate, Covariance Matrix for detecting face marker points and Mahalanobis Distance to calculate the distance of each marker points within frames. The technique of tracking points on face uses markers placed on key positions of face muscles, then by getting its position from all frames of pre recoded face video using the distance algorithm the movement of each face muscle is detected and measured. The face muscles are marked with particular tracking markers that are detected and tracked by the system. This tracking occurs by using color segmentation, where we detect color of points and track the location and distance of each tracker points. The original and translated position values of each marker points are obtained and recorded in text file in vector values. The tracked values will be transferred in a 3D Animation software like MAYA and applied on a pre-Rigged 3D model of Human face. The 3D face will be rigged using joints to emulate the face muscle behavior.

Key words: Tracker, Mahalanobis Distance, Huff-transform, Covariance, Algorithm

INTRODUCTION:

Animation is an art, in which we create motion of virtual objects and also create imagination world and capture the essence of real world and represent it in virtual environments using animation. Now a days, computer graphics and animation are largely used in multimedia industries where the animator is keenly able to create movies, cartoons, robots and many other activities. In animation user can draw the character from real human to animated face in which he/she can capture the motion and expressions of face through the motion capture technology. In character animation the animator measures the time and dynamics of an object and also the position of character. There are many techniques used in animation and are categorized under 6 main types which are as follows:

- Key frame or video frames animation
- Motion Tracking Animation
- Adjectival / Procedural Animation
- Data Controlled OR Data Driven Animation
- Physics Based Animation
- Traditional Animation or Hand Draw / 2D Animation

In field of animation, the animator uses various different tools and techniques to create the 3D face animation representing the expressions of a real face. The face animation can also be calculated using the geometrical equation and also including the motion capture of human face. It is a long and difficult process to generate the facial expressions from real face that represent a 3D face animation. Every latest movie now requires that the face expressions of the virtual avatar are animated and resembles just like the expressions of a real face, which is very difficult to match.

In human face, many face expressions are generated when they communicate with each other and produce different types of expressions of their face like different expressions of mouth, eyes, checks and nose expressions. In natural languages there are two types of languages to communicate each other one is verbal and other is nonverbal. We use mostly verbal language, but in research point of view the nonverbal communication is largely used to communicate and understand easily, as compare to verbal communication.

This paper focuses on developing an automated technique for generating 3D facial expressions from real-life video footage. The key steps involved in our approach include:

1. Face Expression Extraction: Real human face motion is captured from video using the Huff-transform algorithm to gate the x and y coordinates. This helps in isolating and extracting facial expressions.
2. Marker Points and Covariance Matrix: Marker points are placed on key positions of the face muscles, and a covariance matrix is utilized to detect these markers. The covariance matrix helps in identifying and tracking specific points on the face.
3. Mahalanobis Distance: The Mahalanobis Distance is employed to calculate the distance of each marker point within frames. This distance measurement aids in understanding the movement and changes in each face muscle over time.
4. Tracking and Color Segmentation: The tracking of marker points on the face occurs by using color segmentation. The color of the points is detected, and their location and distance are tracked accordingly.
5. Data Recording: The original and translated position values of each marker point are obtained and recorded in a text file in vector form. This data serves as input for further analysis and animation.
6. 3D Animation Software: The tracked values are then transferred to a 3D animation software, such as MAYA. A prerigged 3D model of a human face is used, with joints set up to emulate the behavior of face muscles.

I. ***Face expressions and animations***

The human face possesses a remarkable ability to convey a wide range of emotions, thoughts, and intentions, serving as a means of communication and expression. Facial expressions play a vital role in conveying messages and communicating nonverbally. These expressions encompass various modes such as smiles, anger, sadness, laughter, and more. While verbal communication relies on natural or learned languages, nonverbal communication utilizes facial expressions to convey thoughts and emotions without spoken words. This form of communication has garnered significant attention in the field of computer graphics, particularly in the creation of virtual face animations for 3D characters. Animation, as an art form, allows designers to bring imaginary worlds to life through the creation of animated movies, games, and cartoons, leveraging the power of facial expressions to enhance storytelling and evoke emotional responses.

II. Facial Muscles and corresponding marker points

In this section, we discuss the muscle-based technique employed in this research, which focuses on identifying and measuring individual face muscles to capture facial actions for extraction and animation purposes. The human face comprises a complex interplay of 43 muscles, each contributing to various facial expressions. By understanding the behavior of these face muscles, animators can accurately replicate and transpose reallife facial actions onto 3D models. In this study, we have categorized the 43 face muscles into different regions of the face to facilitate analysis and animation. For instance, during speech, a subset of 17 muscles exhibits discernible movement, primarily involving the lips, cheeks, and jaw. The intricate coordination of these muscles allows for the formation of various phonetic sounds and mouth shapes. Meanwhile, the eyebrows play a significant role in conveying emotions and expressions, with 15 distinct muscles dedicated to their manipulation. Additionally, the nose, although comprising a smaller set of four muscles, contributes to subtle yet important facial nuances.

To quantify and analyze the changes in facial muscles during different actions, we adopt the widely-used Action Units (AU) system [20]. The AU system provides a standardized framework for identifying and describing specific facial muscle activations associated with different expressions. By mapping the movements of marker points placed on the face to the corresponding AU system, we can precisely capture the intricate interplay of facial muscles during various actions.

Figure 1, provides a visual representation of the face muscles and their corresponding marker points. This figure serves as a reference guide for animators and researchers alike, facilitating the accurate identification and tracking of specific facial muscles during the motion extraction process.

In the following sections, we will explore the methodology employed to track and extract facial motion using marker points, utilizing the information provided by the identified facial muscles and their corresponding Action Units.

Figure 1: Face muscle and Marker Points

III. Related work

The term computer graphics consists of two words, one is 'computer' and other is 'graphics'. The word graphics is from Latin word graphicus which mean to draw or paint (1). Actually the word graphics is the merge of two words Graph and pics, to draw any structure in digital computing we use graph (2). When these graphs combine to gather in proper model and give the colors to model it is known as picture (3). Now what is computer? Basically the computer is an electronic machine to perform and multitask at a time quick and acceptable results of any calculation, methods and classification of programs. Computer is combination of hardware and software there are two types of software system software and application software. There are many application software's we use in our daily life for calculations, drawing, publications, animation, graphics, multimedia, social media, digital image processing, artificial intelligence, neural network, computer vision etc.

Now our main objective is to use computer animation and image processing to generate automated facial animation from real video frames and then apply the face transform values on a 3D face (4). Actually the animation is the sub branch of computer graphics there are a lot of software on animations to create the digital animation from real human face to animated face (5), there is a lot of research work going in area of animation world (6), to generating the animation effective and modeling of character animation (7). When we go back in 90s there is a little work done in this area the how to detect the face expressions and facial motion to describe the facial expression in animation and text based software (8), the pseudo-muscular technique to define tissues and facial expressions (9). Pasquarielloet. al, blend shape animation method for facial expression (10) Pushkar Joshi et al. Similar, the facial expression was detected using trackers (11). and were classified by Lien et. al (12), whereas the entire face head was detected and expressions were recognized from it by Bartlett et. al (13) and Matre et. Al (14) and by Chew et. Al (15). The face expressions again were extracted using sensors in realtime and applied on digital avatar by Weise et. al (16). Here we studied a lot of research work and highlighted related work about face detection and facial expession, Rowely et al (17), present the neural network based filters to detect the in many scales and merge. Osadchy et al (18), explain the joints of face detection and estimation the significance of pos and performance of face detection. As similar work Vaillant et al (19), it describe the two stages of detection in dirt stage it locate the face region and the second define more surmise the location these techniques are define the front face detection.

This research focuses on the analysis of real human facial expressions by employing marker points to track facial motion across different frames in video footage. The motion data, represented by the x and y coordinates of the marker points, is then utilized to generate a 3D animated face using the software Maya. Initially, the study involves male candidates aged between 25 to 36, recognizing that age can influence the manifestation of distinct facial expressions. The research primarily covers four fundamental facial expressions: Worried, Angry, Happy, and Sad. Additionally, in a secondary level of analysis, the position and motion of the mouth, jaw, and lip muscles during opening and closing actions are examined.

To conduct the research, the first step involves recording videos of real human faces. These faces are marked with tracker points strategically placed on specific facial muscles. Matlab code is employed to read and process the video data, including tracking the movement of the marker points, segmenting the video into individual frames, and calculating the distances between the marker points within each frame. This distance calculation is performed using the Mahalanobis distance algorithm, which aids in assessing the variations in facial muscle movement.

The resulting x and y coordinates of the marker points are saved into a CSV file, forming vector sequences that capture the temporal dynamics of facial motion. These generated coordinate values are then imported into Maya, a powerful 3D modeling and animation software. By utilizing the collected data, a 3D model of the face is constructed, incorporating the tracked facial movements and expressions. Figure 2 illustrates the design process involved in generating the 3D model.

Through this research design, a comprehensive understanding of human facial expressions is sought by examining the intricate interplay of facial muscles captured via marker points. The combination of video analysis, distance calculations, and 3D animation techniques provides valuable insights into the dynamics of facial expression, contributing to the advancement of realistic virtual character animations and the exploration of emotional communication in computer graphics.

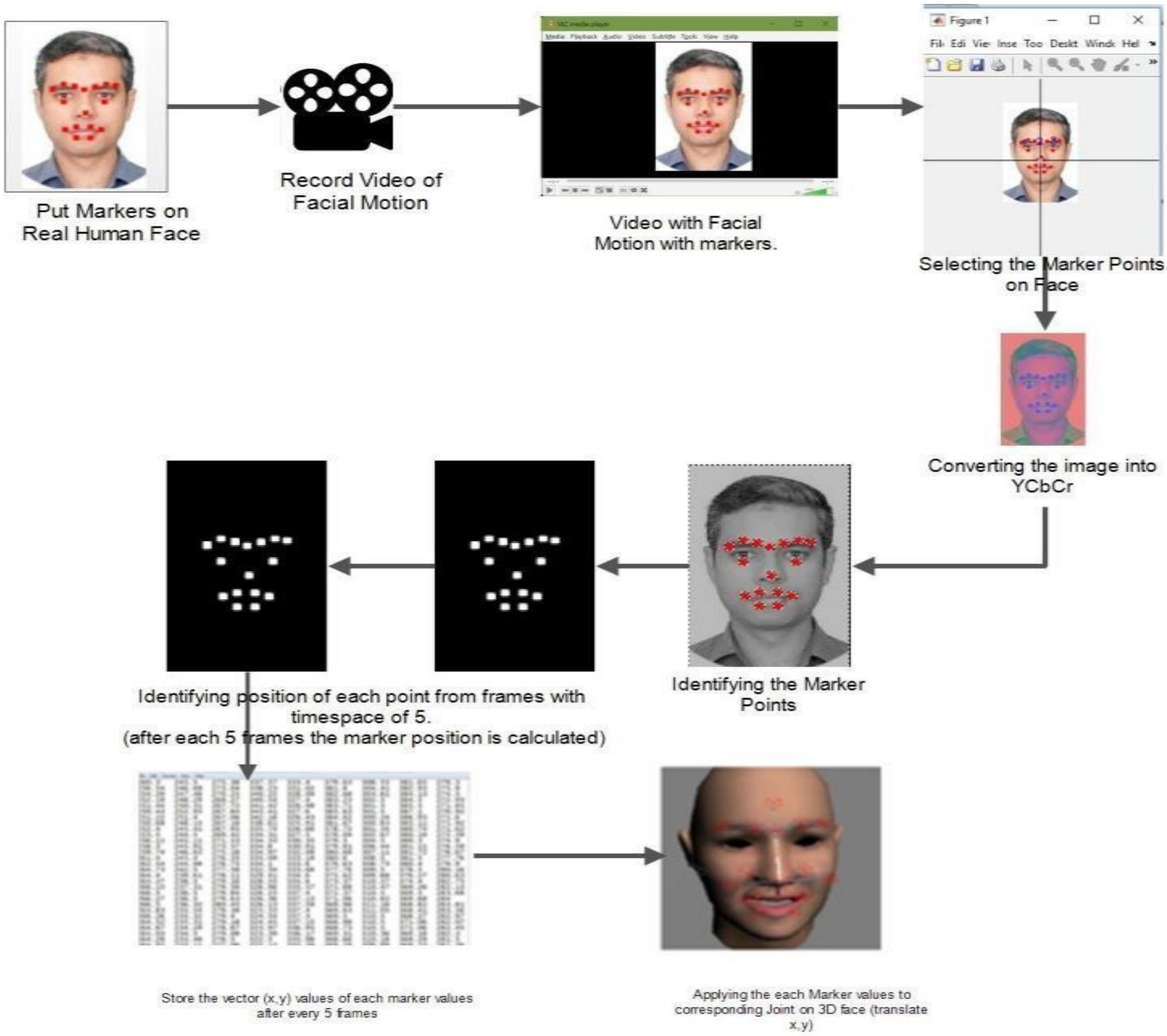

Figure 2: Research design

A. Facial Expressions

The human face possesses an incredible capacity to generate an extensive range of facial expressions, each capable of conveying nuanced emotions and messages, as exemplified in Figure 3. These expressions are a result of the intricate interplay and coordination of numerous facial muscles. Referred to as facial muscle motion, these expressions manifest as a dynamic display of movements across the face. Notably, facial expressions play a crucial role in non-verbal communication, enabling individuals to convey thoughts, emotions, and intentions without relying on spoken words alone.

Among the myriad facial expressions, the eyes hold a special significance. They serve as a captivating focal point and are instrumental in communicating a wealth of emotions. The eyes express a wealth of information through subtle cues, including blinking patterns that provide insights into the speaker's state of mind and emotional expressions.

The variations in facial expressions stem from the activation and modulation of specific facial muscles. Each expression involves the orchestrated movement of multiple muscles working in concert. These muscles can contract, relax, or assume specific configurations to convey a particular emotion or convey a message. By analyzing and understanding the intricate patterns of these muscle movements, researchers gain valuable insights into the mechanisms underlying facial expressions.

In Figure 3, we observe a selection of six distinct facial expressions portrayed by women. Each expression represents a unique combination of facial muscle movements, capturing a specific emotional state or communicative intent. By studying and categorizing these expressions, researchers can delve deeper into the intricacies of human communication, exploring the subtle nuances and universal characteristics that define facial expressions across cultures.

In the subsequent sections, we will delve further into the analysis and extraction of facial expressions, examining the underlying muscle movements and their relevance in computer graphics and animation. Through a comprehensive understanding of facial expressions, we can unlock new avenues for creating more realistic virtual characters and enhancing the emotional impact of digital media.

Figure 3: Face Expressions generated by human face [internet]

B. 3D Face Construction and Rigging

To enable facial animation, a 3D model of the human face is essential. In this research, Autodesk Maya was utilized for constructing the 3D model. The process involved creating a polygonal structure to represent the face using established face modeling techniques. Figure 4 depicts the wireframe surface of the face muscles, which serves as the foundation for constructing the 3D polygonal head. Subsequently, textures and hair were applied to the head, while facial bones were introduced to control the movements of the muscles.

The rigging phase of the virtual 3D human face was a critical step in the animation process. This involved establishing a hierarchical joint structure within Maya that corresponds to the key face muscles observed in a real face. Each joint chain was strategically positioned around the relevant facial muscle areas, enabling control over the movement and deformation of the polygon vertices using Maya's built-in skinning tool. Consequently, when the face is rigged, it becomes capable of manipulating and rotating the key facial muscle points, which are obtained through the face expression feature extraction algorithms discussed in the subsequent chapter. Figure 5 showcases the final 3D face model, complete with the joint structure and corresponding muscle control.

By leveraging the capabilities of Maya and employing a carefully designed joint hierarchy, the 3D face model becomes a dynamic and expressive entity. The joint structure, synchronized with the extracted facial muscle data, empowers the virtual face to mirror and emulate the intricate movements and expressions observed in a real face. This fusion of 3D modeling and rigging techniques forms the foundation for realistic and engaging facial animation.

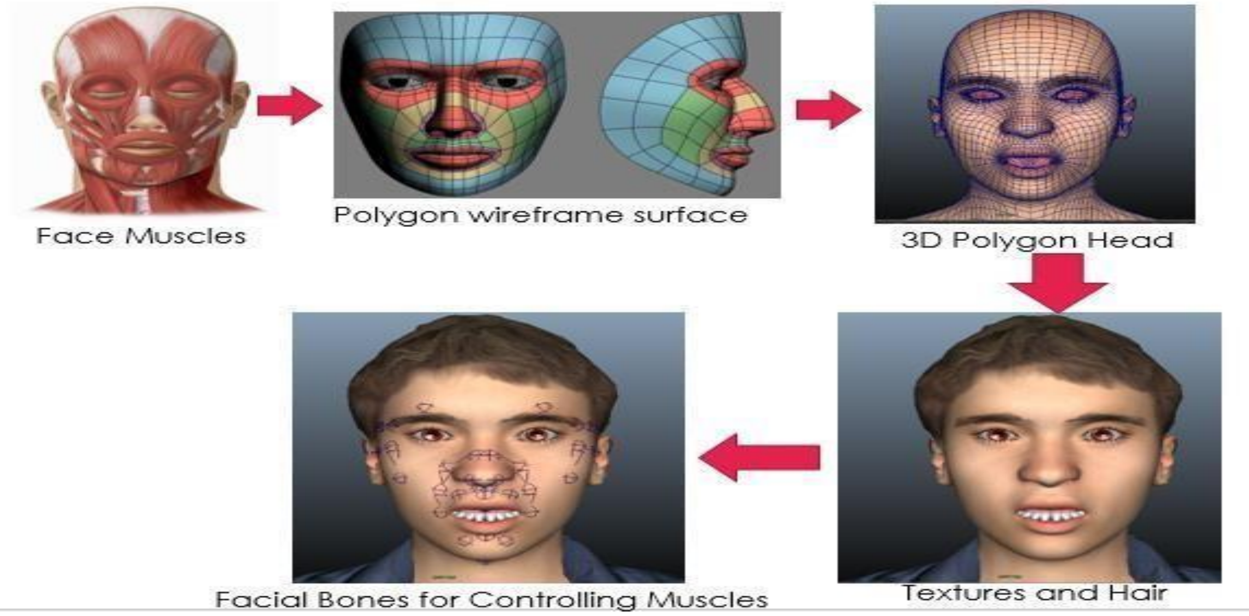

Figure 4. Face construction in 3D with joint

In the upcoming sections, we will delve into the details of the face expression feature extraction algorithms, which play a pivotal role in capturing the nuances of facial movements and driving the animation process. The seamless integration of these algorithms with the 3D face model allows for the creation of compelling and lifelike virtual characters capable of conveying a wide range of emotions and expressions.

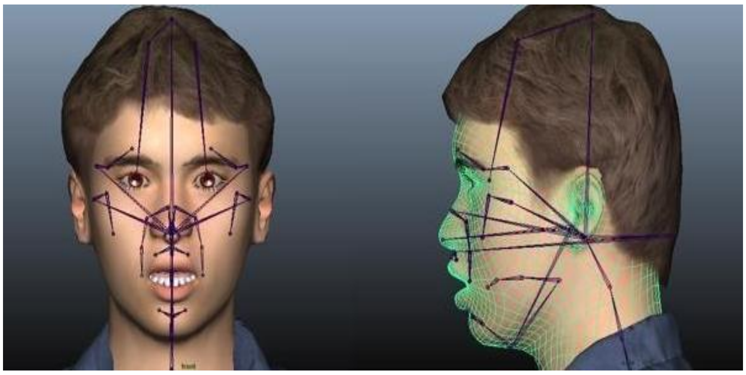

Figure 5: Polygon face model with joint rig to control the key face muscles.

IV. Face expression feature extraction

The process of extracting face expression features in this system begins with the tracking of points on the face, enabling the capture of different frames showcasing varying facial positions. These tracked points are then utilized for color segmentation, aiding in the detection of facial tracking points across the frames. The positional data of these tracking points in each frame is subsequently stored in a text file for further analysis.

In the realm of computer vision, segmentation refers to the partitioning of a digital image into multiple distinct segments or subsets of pixels. The primary objective of segmentation is to define a more meaningful representation of the image that is amenable to analysis. Essentially, image segmentation simplifies the representation of image pixels and assigns marker points to describe the characteristics of each pixel. Consequently, the image is divided into different segmented regions, facilitating further analysis and processing (20).

Color segmentation, a subset of segmentation techniques, categorizes pixels based on their colors, employing methods such as covariance matrix and Hough-transform algorithms to detect marker point circles with predefined radii. By applying these algorithms, the system identifies and segments the different colored regions in the image, allowing for the localization of specific facial features and markers. To quantify the movement and changes in facial expressions, the Mahalanobis algorithm is utilized. This algorithm calculates the distance between a marker point in one frame and the corresponding marker point in another frame. By leveraging different frames from a recorded video clip, the system computes the values of each marker point across the frames, providing insight into the temporal dynamics of facial expressions. Figure 6 provides a visual representation of the process, illustrating the tracking and extraction of marker points across different frames.

Through the face expression feature extraction process, the system captures and quantifies the subtle variations in facial expressions, enabling a deeper understanding of the dynamic nature of facial movements. By leveraging color segmentation, covariance matrix, Hough-transform algorithms, and the Mahalanobis distance calculation, the system gains valuable insights into the spatial and temporal characteristics of facial expressions. These extracted features serve as crucial input for subsequent stages of the research, facilitating the generation of realistic 3D facial animations that accurately reflect the intricacies of human expression.

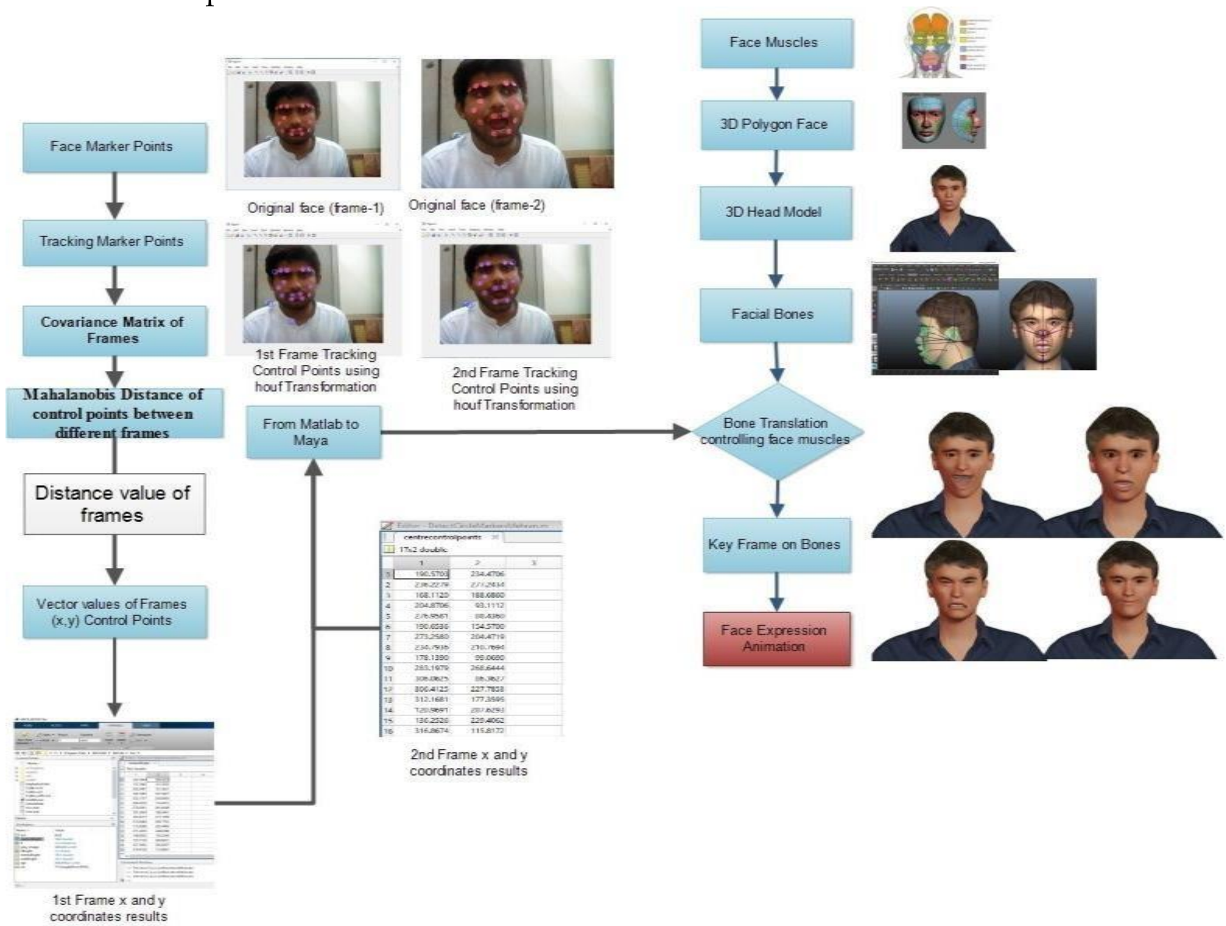

Figure 6: Face Expression feature Extraction model

A. *Algorithm Steps.*

1) First read video and show first frame then allow user to detect points in first frame.

2) Circular Hough Transform finds circles with approximately the specified RADIUS, in pixels, in the input image A. A can be a gray scale, RGB or binary image.

3) And then convert image into YCRCB.

4) Calculate the inverse of covariance of matrix.

5) Calculate the distance of marker points in different frames and using color segmentation on marker points.

6) After detecting tracker points color make some morphological operation on result, compute centroid through region properties for each point in each frame.

7) Finally write tracker values result in text file

B. *Co-Variance Matrix*

The Co-variance have a two kinds of concepts one is matrix covariance and matrix these are inherent of covariance matrix. Now let`s start from matrix early we have learn about the matrix that was in schools or you are enrolled in school that you have to know about the basics of linear algebra and solve the equations of matrices. I highly suggest you to learn it seriously it is very complicated problems to solve many mathematical methods.

Let to know about if you are at the research level I suggest you that you don't have to go deeply because this is very complicated method in mathematics you have need to learn some methods and solve the problem and implement it in your application because it is a very long method in matrices many notations it take a time and I suggest you to learn the method that how to solve the problem and you can do and follow the steps and solve the problem. In co-variance matrix we have to 1st solve the mean of object then we solve the variance of numbers and calculate mean and variance both the final result you take that is co-variance matrix.

Let

$$X = \begin{bmatrix} X_1 \\ \vdots \\ X_n \end{bmatrix}$$

are random variables, each with finite variance, then the covariance matrix Σ is the matrix whose (i, j) entry is

$$\Sigma_{ij} = \mathrm{cov}(X_i, X_j) = \mathrm{E}\left[(X_i - \mu_i)(X_j - \mu_j)\right]$$

Where

$$\mu_i = \mathrm{E}(X_i)$$

$$\Sigma = \begin{bmatrix} \mathrm{E}[(X_1-\mu_1)(X_1-\mu_1)] & \mathrm{E}[(X_1-\mu_1)(X_2-\mu_2)] & \cdots & \mathrm{E}[(X_1-\mu_1)(X_n-\mu_n)] \\ \mathrm{E}[(X_2-\mu_2)(X_1-\mu_1)] & \mathrm{E}[(X_2-\mu_2)(X_2-\mu_2)] & \cdots & \mathrm{E}[(X_2-\mu_2)(X_n-\mu_n)] \\ \vdots & \vdots & \ddots & \vdots \\ \mathrm{E}[(X_n-\mu_n)(X_1-\mu_1)] & \mathrm{E}[(X_n-\mu_n)(X_2-\mu_2)] & \cdots & \mathrm{E}[(X_n-\mu_n)(X_n-\mu_n)] \end{bmatrix}.$$

Covariance matrix of group is computed using centered data matrix

$$\mathbf{C}_i = \frac{1}{n_i}\hat{\mathbf{X}}^T\hat{\mathbf{X}}$$

It is common to group all these numbers into a square table called the Covariance Matrix of the distribution according to the following layout:

$$\Sigma = \begin{pmatrix} \sigma_{11} & \cdots & \sigma_{1n} \\ \sigma_{21} & \sigma_{22} & \vdots \\ \vdots & & \ddots \\ \sigma_{n1} & \cdots & \sigma_{pp} \end{pmatrix}$$

In this image we show the marker points and capture image from the frames of video and set the numbers on marker points. [figure.7]

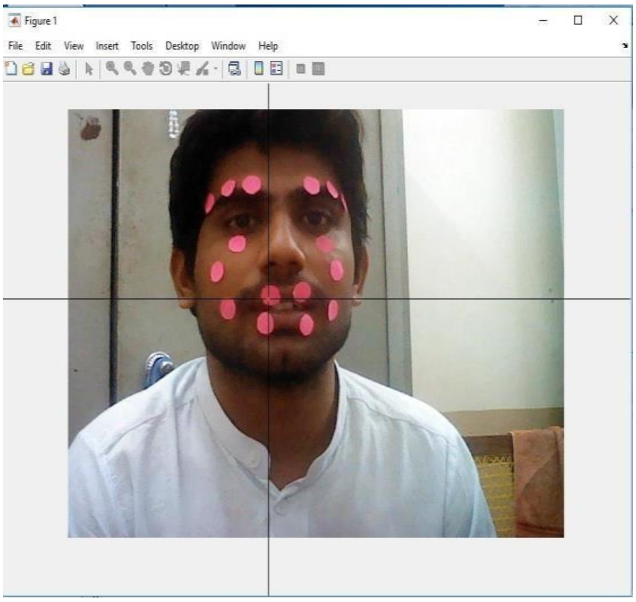

Figure 7: Face Marker Points from initial video frame

In figure 8 we select the 10 marker points of face and set the numbers on marker points for tracking these markers points.

Figure 8: Numbered Face Marker points

In figure 9 we convert the face into YBCR and show the accurate face marker point which are they selected.

Figure 9: YBCR image of the video frame with numbered marker points

C. Circular Hough Transform Algorithm

The Hough transform algorithm is basically used in image processing technique where the object detect the circular points in image. We are using the digital image when we find the circular calculations of the object from face (21).

In CHT (Circular Hough Transform Algorithm) to detect the circular objects it has a feature extraction technique to detect the objects which a in circular form. Actually we use this technique to detect the proper circle which we place on face and to find their x-coordinates and y-coordinates of the circular object in other words we called it accumulator matrix. The Hough transform algorithm that is define the Sensitivity of the circular label and radius of the object (22). The equation of the circular Hough transform algorithm that is describe in two dimensional.

$$(x-a)^2 + (y-b)^2 = r^2 \quad (1)$$

According the equation a and b, where (a,b) is the center of the circle, and r is the radius. If a 2D point (x,y) is fixed, then the parameters can be found according to (1). The parameter space would be three dimensional, (a, b, r). And all the parameters that satisfy (x, y) would lie on the surface of an inverted right-angled cone whose apex is at (x, y, 0). In the 3D space, the circle parameters can be identified by the intersection of many conic surfaces that are defined by points on the 2D circle. This process can be divided into two stages. The first stage is fixing radius then find the optimal center of circles in a 2D parameter space. The second stage is to find the optimal radius in a one dimensional parameter space (23).

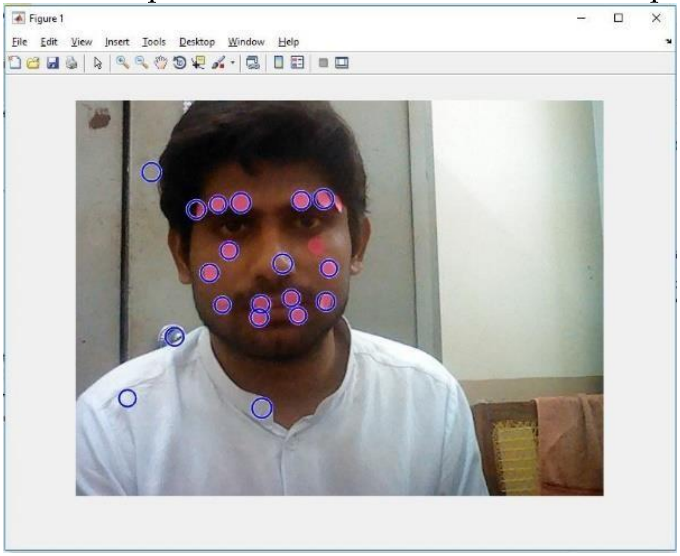

Figure 10: face marker points being detected with circle over them using Huf transform

Changing the values of radius, sensitivity and threshold values within the huff- transform gave various different results and certain noise detected in recognition of the marker points as shown in figure 11 and 12.

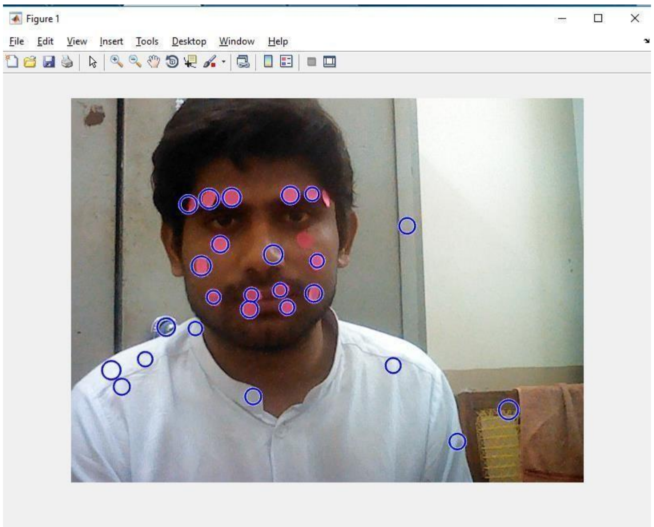

Figure 11: Face marker detection with Radius=8,12

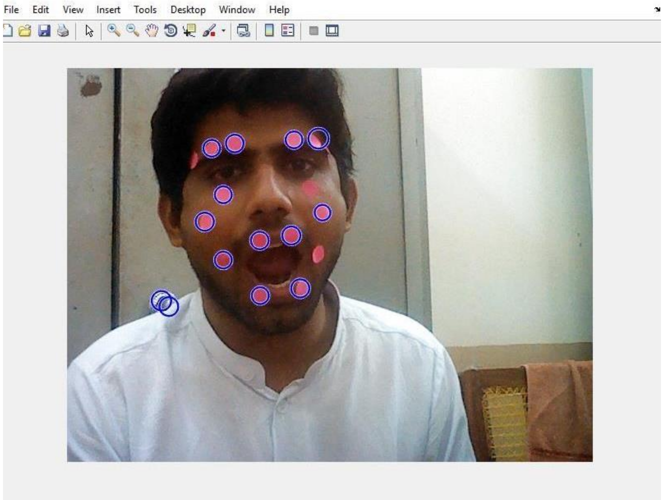

Figure 12: Face marker detection with Radius=9, Threshold=0.85& Sensitivity=0.95

V. ***Face detection***

There are many techniques to use face detection (21-24). The automated face detection Muhammad Yeasin using Robust technique (20). Segmentation face regions Rowley (21). Boosting technique to generate the framework and detect face Marian Stewart Bartlett (28). To draw geometry of face by code on face edges and show in graph Y Tong (24). Iren Kotsia using rule based algorithm to detect face and its classification with neural network.

In face detection or an object detection there are many algorithm and techniques were used to detect the real human face or any object. When to detect the different part of face or different segments of face using bounding box technique to detect the defined face parts like right eye, left eye, eye brows, nose and mouth these parts shown of detect with bounding box. In which the DAG (Dense Adversary Generation algorithm) used to detect the face or object in box in other words we call it Bounding box face detection technique.

VI. Mahalanobis distance

The Mahalanobis distance, initially introduced by P.C. Mahalanobis in 1936, is a statistical measure utilized to assess the dissimilarity or similarity between data points. It provides a means to analyze patterns of correlation between variables and aids in identifying relationships within a dataset. Unlike the Euclidean distance, which considers only the spatial separation between points, the Mahalanobis distance incorporates the correlation structure of the dataset, making it scale-invariant.

In our research work, we employ a two-dimensional plane to implement the Mahalanobis distance equation. The equation calculates the distance, denoted as d(x, y), between two points, x and y, in the two-dimensional space. It is computed as the integral of the squared differences between the corresponding coordinates of the points. Specifically, for each dimension, the difference between the x-coordinate of one point and the y-coordinate of the other point is squared, and the resulting values are summed. This process is repeated for both dimensions, and the squared differences are then summed to obtain the final distance value.

$$d(x,y)=\int(x1-y1)^2+(x1-y1)^2$$

By utilizing the Mahalanobis distance equation in our research, we can effectively measure the dissimilarity between marker points in different frames of the recorded video. This allows us to analyze the variations in facial muscle movements and quantify the differences between facial expressions. The Mahalanobis distance, incorporating the correlation structure of the data, provides a robust and comprehensive measure for assessing the distances between marker points, aiding in our understanding of facial dynamics and supporting the animation process.

VII. ***Implementation and results***

The implementation of the proposed methodology and the presentation of results were carried out using Matlab and Maya software. The algorithmic components of image detection and face expression extraction were implemented in Matlab, employing the methodologies discussed in the previous chapter. Subsequently, the 3D face animation process was executed using Autodesk Maya.

The implementation process began by reading the video footage and displaying the first frame to the user. The user was then provided with the capability to manually detect the key points on the face in the first frame. Following this, the color space of the frame was converted from RGB to YCRCB, enabling further analysis. The inverse of the covariance matrix was calculated to facilitate subsequent computations on the image.

In the color segmentation stage, the square of the Mahalanobis distance was computed and compared to a predefined threshold. This facilitated the identification and detection of color-marked points on the face. Morphological operations were subsequently applied to the segmentation result, followed by the computation of connected components. Through region properties analysis, centroids were determined for each point in every frame. The centroid computation process was repeated to ensure accuracy. Finally, the obtained results were written into a text file for further analysis and visualization.

By implementing the described steps, the research methodology was executed, allowing for the extraction of facial expression features and the subsequent generation of 3D facial animations. The utilization of Matlab and Maya software enabled seamless integration and processing of the data, facilitating the analysis and interpretation of the results.

The results obtained from this implementation provided valuable insights into the dynamics of facial expressions and their translation into realistic 3D animations. The accuracy and effectiveness of the proposed approach were evaluated through visual observation and comparison with ground truth data, allowing for an assessment of the fidelity and authenticity of the generated animations.

The implementation and results demonstrate the feasibility and effectiveness of the proposed methodology in automated facial animation using marker points for motion extraction. The combination of Matlab and Maya software provided a robust platform for executing the algorithms and generating expressive 3D facial animations, paving the way for advancements in virtual character animation and computer graphics.

Step 1. Read Video (figure 13)

```
A = aviread('.mp4');
Implay (A);
```

Step 2- Allow user to select point in first frame

```
for i=1:10
%get ten point from user
[x,y]=ginput(1);
user_input=[x,y ];
h=text(x,y,num2str(i));
set(h,'Color',[0 0 1], 'LineWidth',4);
end
```

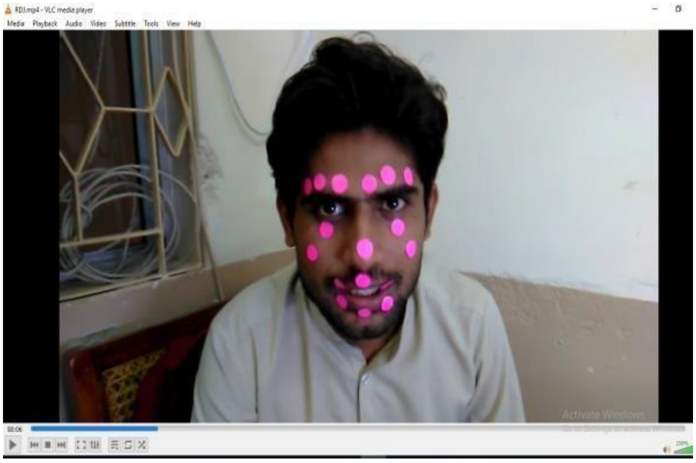

Figure 13: Read video

Step 3: Get the 10 points from user (figure 14)

```
[x,y]=ginput(1);
user_input=[x,y ];
h=text(x,y,num2str(i));
set(h,'Color',[0 0 1], 'LineWidth',4);
```

Figure 14: select 10 marker points of real face

Step 4: Convert frame into YCRCB (figure 15)

```
YCBCR = rgb2ycbcr(i(1).cdata);
```

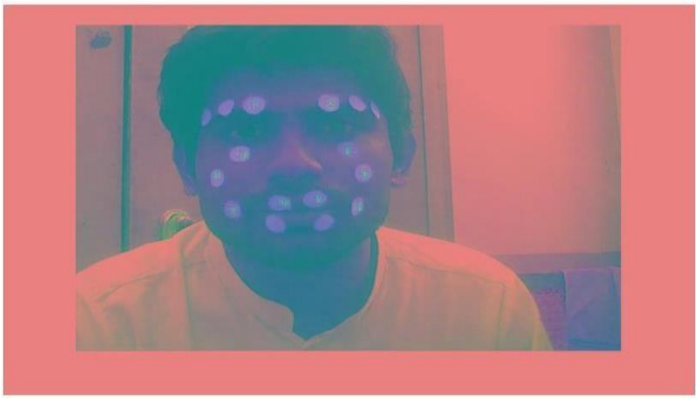

Figure 15: convert image into YCBCR

Step 5. Finds circular objects (figure 16)

```
[centersBright, radiiBright,
metricBright]=imfindcircles(rgb,[9 13 ],
('ObjectPolarity','bright','Sensitivity',0.945,'EdgeThreshold',0. 08)
```

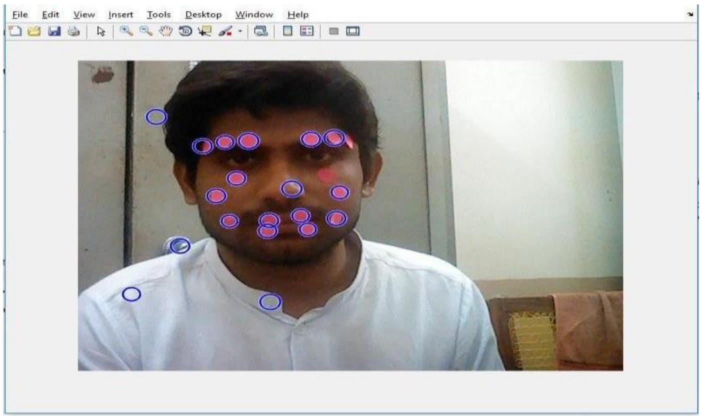

Figure 16: find circular marker points using hough transform algorithm

Step 6: Find the radius of each circle interactively using the distance tool

```
d = imdistline;
```

Step 7: Then computing the square of the Mahalanobis distance and comparing it to a threshold. (Figure 17)

```
%convert to double
image = im2single(YCBCR);
CB = image(:,:,2);
CR=image(:,:,3);
CB_out=single(CB)-0.4684;
CR_out=single(CR)-0.7107;

%convert CB into Vector
reshape_CB= reshape(CB_out,[],1);

%convert CR into Vector
reshape_CR=reshape(CR_out,[],1); horzcat1=horzcat(reshape_CB,reshape_CR);

%get inverse of covariance matrix
inv_1=single(inv(1.0e-003 *[0.0216 -0.0211 0.2555]));
%compute Distance
%horzcat()concatenates arrays horizontally
Distance1 = sqrt((horzcat1*inv_1).*horzcat1);
Distance1 = sum(Distance1);
result = reshape(Distance1,[],640);
%compare result with mahanobis
Distance I= (result<6);
```

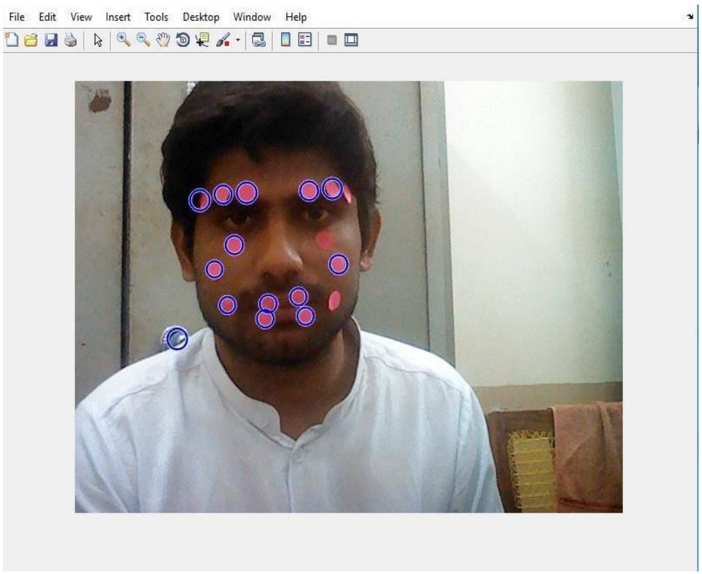

Figure 17: Value of x coordinate and y coordinate marker points

Some more results on the X-Y values in table format

- X-V values of Each marker separately.
- X-Y values of same markers after change of pose
- X-Y values from different frames

Once each marker point is detected, its x,y location values in 2D transform plane are obtained and stored in a text file in CSV format as shown in figure 18.

1	201.145	124.8098	96.47824	165.6201	229.7756	286.6748	181.8263	168.4321	229.7756	286.6748	237.1647
2	270.0098	261.9197	213.5785	247.399	203.2269	121.6684	267.7879	104.6989	203.2269	121.6684	198.6459
3	274.3606	122.5335	155.3255	286.2471	191.6945	188.9893	194.9017	109.0541	191.6945	188.9893	215.4428
4	223.0076	264.5532	153.197	249.9055	173.0364	120.525	265.7906	246.0804	173.0364	120.525	190.2304
5	186.5456	183.2419	197.7332	154.5259	167.4941	217.7341	222.1622	228.3669	167.4941	217.7341	287.7955
6	163.6165	209.8818	127.5038	168.1899	284.6904	116.1776	218.1126	248.8148	284.6904	116.1776	123.9766
7	261.5736	240.4552	97.00011	252.5337	233.139	255.8421	119.8233	288.4187	233.139	255.8421	115.3427
8	308.4822	204.5256	168.6686	161.662	278.9694	282.0384	256.7738	222.4234	278.9694	282.0384	174.4282
9	190.0216	234.8963	120.6191	322.3936	184.032	257.7252	171.5745	113.7636	184.032	257.7252	317.3365
10	277.1454	87.60493	227.4222	162.3456	119.7492	286.7783	298.609	102.7104	119.7492	286.7783	186.4761
11	235.1459	210.7776	129.1931	290.9113	304.1669	115.25	174.3983	236.1446	304.1669	115.25	283.5637
12	190.459	154.6811	183.6898	296.335	313.9628	209.3393	222.7794	375.9547	313.9628	209.3393	242.5206

Figure 18: calculate the different frames of video x coordinate and y coordinate marker point

Once the CSV file is created form Matlab, the file is transferred and read into Maya for creating the face animation based on the x –y coordinate of each marker point. Each frame number is correlated with the time frame value inside Maya for creating key frame of animation.

This image shows that the real human face expression convert into 3D animation expressions Figure 19.

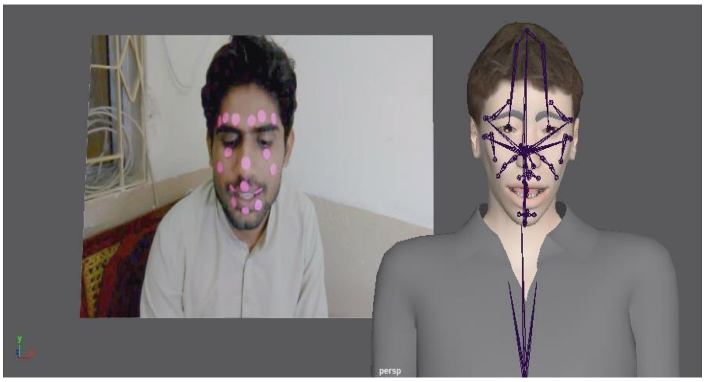

Figure 19: Face Expression with corresponding 3D animated face

The initial face marker points were detected and then saved as CSV file with coordinate location of each marker points with respect to time. These coordinates of each face marker point is then applied to face bones which has been rigged to the polygon vertices representing face muscles. This then drives these joints with respect to time in frames, and key frame is generated by the Maya system at each time interval saving the face pose.

This image shows us the surprise face of real human face and also in 3D animation model face Figure 20.

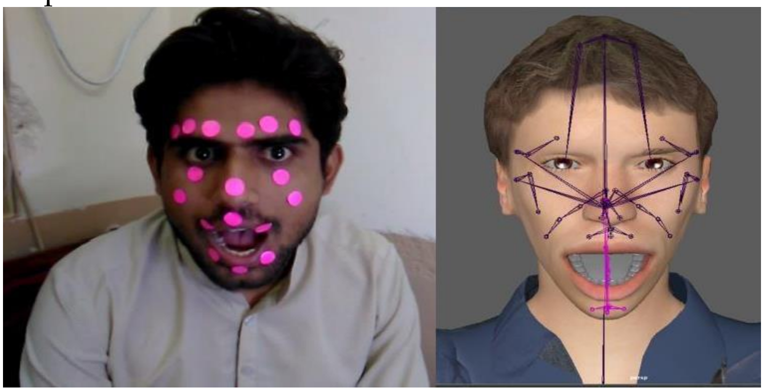

Figure 20: Surprise Face Expression with corresponding 3D animated face

Figure 21: Face Expression obtained from real face

We see multiple type of face expressions in our daily routine life and we cannot recognize all the expressions of real human face. Researcher and scientists are described there are 21 distinct face expressions. In natural there are 6 human face expressions which can be Happy, Sad, angry, Natural, Surprise and Fear. Here we also define these 6 natural human face expressions.

CONCLUSION

In this research work a model is proposed based on covariance matrix and Mahalanobis distance algorithms to detect, process and calculate the tracking points from human face video. The system first captures the face video and send it to Matlab for processing. Through Matlab the Face tracking points are identified and isolated. Then each tracking point distance in motion is calculated using Mahalanobis algorithm. Through this system, automated animation of facial expressions can be achieved very easily and conveniently. the animator would be required to record their video acting the required expressions. And the system will automatically transfer the expressions into a 3D model.

In this research work the aim was to generate realistic human face expressions, by extracting the expression of real human from a video source. We recorded the video and used Circular Hough Transform to detect the control points. These control points were placed on real human face along with key face muscles to allow accurate muscle movement detection. **Covariance Matrix** calculate X-coordinates and Y-coordinates of Radius in frames from these tracker movements the distance was measured of each muscle movement using Mahalanobis Distance Algorithm.

These distance values were transferred to 3D model in Maya, which was rigged with joints replicating human muscles. The final results are very satisfying with accurate detection of tracker points, trackers displacement and distance calculation and finely transfer of information on 3d model of human face.

REFERENCES

1. Bouaziz, S., Wang, Y., & Pauly, M. (2013). Online modeling for realtime facial animation. ACM Transactions on Graphics (TOG), 32(4), 40
2. Pasquariello, S., &Pelachaud, C. (2002). Greta: A simple facial animation engine. In Soft computing and industry (pp. 511- 525). Springer, London.
3. Deng, Z., & Neumann, U. (2008). Data-driven 3D facial animation.
4. Joshi, P., Tien, W. C., Desbrun, M., &Pighin, F. (2006, July). Learning controls for blend shape based realistic facial animation. In ACM Siggraph 2006 Courses (p. 17). ACM.
5. LIN, I.-C., AND OUHYOUNG, M. 2005. Mirror mocap: Automatic and efficient capture of dense 3d facial motion parameters from video. The Visual Computer 21, 355–372
6. DENG, Z., CHIANG, P.-Y., FOX, P., AND NEUMANN, U. 2006. Animating blendshape faces by cross-mapping motion capture data. In I3D, 43–48
7. MA, W.-C., JONES, A., CHIANG, J.-Y., HAWKINS, T., FREDERIKSEN, S., PEERS, P., VUKOVIC, M., OUHYOUNG, M., AND DEBEVEC, P. 2008. Facial performance synthesis using deformation-driven polynomial displacement maps. ACM Trans. Graph. 27, 121:1–121:10\
8. BICKEL, B., LANG, M., BOTSCH, M., OTADUY, M. A., AND GROSS, M. 2008. Pose-space animation and transfer of facial details. In SCA, 57–66.
9. HUANG, H., CHAI, J., TONG, X., AND WU, H. 2011. Leveraging motion capture and 3d scanning for high-fidelity facial performance acquisition. ACM Trans. Graph. 30, 74:1– 74:10. [10] Gustafson, D. E., & Kessel, W. C. (1979, January). Fuzzy clustering with a fuzzy covariance matrix. In Decision and Control including the 17th Symposium on Adaptive Processes, 1978 IEEE Conference on (pp. 761-766). IEEE.
10. De Maesschalck, R., Jouan-Rimbaud, D., &Massart, D. L. (2000). The mahalanobis distance. Chemometrics and intelligent laboratory systems, 50(1), 1-18.
11. Lien, J. J. J., Kanade, T., Cohn, J. F., & Li, C. C. (2000). Detection, tracking, and classification of action units in facial expression. Robotics and Autonomous Systems, 31(3), 131-146

12. Bartlett, M. S., Littlewort, G., Fasel, I., &Movellan, J. R. (2003, June). Real Time Face Detection and Facial Expression Recognition: Development and Applications to Human Computer Interaction. In Computer Vision and Pattern Recognition Workshop, 2003. CVPRW'03. Conference on(Vol. 5, pp. 53-53). IEEE.
13. Matre, G. N., & Shah, S. K. (2013, December). Facial expression detection. In Computational Intelligence and Computing Research (ICCIC), 2013 IEEE International Conference on (pp. 1-3). IEEE.
14. Chew, S. W., Lucey, P. J., Lucey, S., Saragih, J., Cohn, J., & Sridharan, S. (2011). Person-independent facial expression detection using constrained local models. Proceedings of FG 2011 Facial Expression Recognition and Analysis Challenge, 915-920.
15. Weise, T., Bouaziz, S., Li, H., & Pauly, M. (2011, August). Realtime performance-based facial animation. In ACM transactions on graphics (TOG) (Vol. 30, No. 4, p. 77). ACM.
16. M. Frey, A. Jenny, E. Stüssi, S. Handschin, and R. Manktelow, "Development of a new paresis scoring system for pre- and postoperative evaluation of facial paresis," European Archives of Oto-Rhino-Laryngology, pp. S182–S184, 1994.
17. M. Isono, K. Murata, H. Tanaka, M. Kawamoto, and H. Azuma, "An objective evaluation method for facial mimic motion," Otolaryngology—Head and Neck Surgery, vol. 114, no. 1, pp. 27–31, 1996
18. P. C. Johnson, H. Brown, W. M. Kuzon Jr., R. Balliet, J. L. Garrison, and J. Campbell, "Simultaneous quantitation of facial movements: the maximal static response assay of facial nerve function," Annals of Plastic Surgery, vol. 32, no. 2, pp. 171–179, 1994.
19. C. J. Linstrom, "Objective facial motion analysis in patients with facial nerve dysfunction," Laryngoscope, vol. 112, no. 7 I, pp. 1129–1147, 2002.
20. C. J. Linstrom, C. A. Silverman, and D. Colson, "Facial motion analysis with a video and computer system after treatment of acoustic neuroma," Otology and Neurotology, vol. 23, no. 4, pp. 572–579, 2002.
21. E. W. Sargent, O. A. Fadhli, and R. S. Cohen, "Measurement of facial movement with computer software," Archives of Otolaryngology: Head and Neck Surgery, vol. 124, no. 3, pp. 313– 318, 1998.
22. K. L. Schmidt, J. F. Cohn, and Y. Tian, "Signal characteristics of spontaneous facial expressions: automatic movement in solitary and social smiles," Biological Psychology, vol. 65, no. 1, pp. 49–66, 2003.
23. K. L. Schmidt, J. M. Van Swearingen, and R. M. Levenstein, "Speed, amplitude, and asymmetry of lip movement in voluntary puckering and blowing expressions: implications for facial assessment," Motor Control, vol. 9, no. 3, pp. 270–280, 2005
24. M. Frey, A. Jenny, P. Giovanoli, and E. Stüssi, "Development of a new documentation system for facial movements as a basis for the international registry for neuromuscular reconstruction in the face," Plastic and Reconstructive Surgery, vol. 93, no. 7, pp. 1334–1349, 1994.