KLE TECH

KLE Technological University

Creating Value
Leveraging Knowledge

School of
Electronics and Communication Engineering

Minor Project Report

on

Mapping User Facial Emotions to an Avatar in Metaverse

By:

1. **Vasundhara V.Baligar** 01FE20BEC304
2. **Sahana R.Hiremath** 01FE20BEC034
3. **M.Harshitha** 01FE20BEE057
4. **Sanjana Miskin** 01FE20BEC077

Semester: VI, 2022-2023

Under the Guidance of

Dr. Uma Mudenagudi

Ramesh Ashok Tabib

K.L.E SOCIETY'S
KLE Technological University,
HUBBALLI-580031
2022-2023

SCHOOL OF ELECTRONICS AND COMMUNICATION ENGINEERING

CERTIFICATE

This is to certify that project entitled " **Mapping User Facial Emotions to an Avatar in Metaverse** " is a bonafide work carried out by the student team of " **Vasundhara V. Baligar (01FE20BEC304), Sahana R. Hiremath (01FE20BEC034), M.Harshitha (01FE20BEE057), Sanjana Miskin (01FE20BEC077)**". The project report has been approved as it satisfies the requirements with respect to the mini project work prescribed by the university curriculum for BE (V Semester) in School of Electronics and Communication Engineering of KLE Technological University for the academic year 2022-2023.

Dr. Uma Mudenagudi **Ramesh Ashok Tabib** **Guides**	**Dr. Nalini C.Iyer** **Head of School**	**Dr. B S Anami** **Registrar**

External Viva:

Name of Examiners **Signature with date**

1.

2.

ACKNOWLEDGMENT

We want to convey our heartfelt gratitude to Ramesh Ashok Tabib, our mentor, for his invaluable advice and assistance in completing our project. He was there to assist us every step of the way, and his motivation is what enabled us to accomplish our tasks effectively.We take this opportunity to than Dr. Ashok Shettar, Vice-chancellor and to Dr. B.S. Anami Registrar, KLE Technological University, Hubli. We also take this opportunity to thank Dr. Nalini C. Iyer, Professor and Head of Department, Department of Electronics and Comminucation Engineering for having provided us academic environment to nurture our practical skills and contributed to the sucess of the project.We sincerely thank our project co-ordinators for thier support, inspiration and wholehearted cooperation during the cousr of completion.

-The project associates
Vasundhara V.Baligar
Sahana R.Hiremath
M.Harshitha
Sanjana Miskin

ABSTRACT

In our project, as a nonverbal mode of communication facial expressions of emotion are highly helpful when we are unable to convey our sentiments verbally. We have used open pose artificial intelligence face motion capture to dynamically detect and track emotions. Given a real-time input of a human face through webcam and detect the facial key points using haar cascade algorithm. For avatar creation, we used a blender, we used already rigged avatar and imported into the blender. In online meeting platforms (Microsoft, Zoom, etc) if the user doesn't have a presentable face or doesn't have a good background, an avatar or the deep fake(if a user likes to) comes to the rescue. This covers the user's face with an avatar that they feel similar to and the selected avatar mimics the face.

Contents

List of Figures

Chapter 1

Introduction

In this chapter, we discuss the introduction and the research papers that we shortlisted. Realistic face animation is a key element in 3D virtual environments that show the presence of characters, both human and non-human. Some of their uses include the 3D gaming industry, interactive software, and 3D animated movies. However, making excellent facial animation requires a lot of work for an experienced animator due to the complexity of human facial expressions, which are an essential part of the genuine movement. Even now, research is being done to develop facial animation, especially in two key areas: the technique for facial rigging and the displacement of expression between two virtual 3D characters, or between real people and 3D characters.

Finding people and identifying their key points are both necessary for keypoint detection. The same thing is a key point and an interest point. Particular spatial placements or areas within an image are what piques our interest or stand out. Image rotation, reduction, translation, distortion, etc. have no effect on them. Re-targeting facial expressions from real people to virtual characters is a useful technique in computer graphics and animation. Traditionally, markers or blend shapes were used to make a mapping between the human and avatar faces. nonetheless, these techniques require a laborious 3D modeling process, and performance is based on the modelers' expertise. As a nonverbal mode of communication, facial expressions of emotion are highly helpful when we are unable to convey our sentiments verbally.

1.1 Motivation

- When people are preoccupied with various tasks being presentably ready for meetings and classes seems a little problematic.
- Online classes taken by teachers can seem monotonous and boring to students.
- Gaming in VR with our own avatar seems so much cooler than using existing animated characters in video games

Figure 1.1: Human Face

Figure 1.2: Avatar

1.2 Problem statement

To map i.e retarget user facial emotions example: happy, sad, surprised to an avatar in the metaverse.

1.3 Objectives

- Detecting key points on a human face.
- Creating an avatar similar to the user's face.
- Detecting key points on the avatar created.
- Retargeting facial key points to avatar key points

1.4 Literature survey

1.4.1 A Real-time Facial Emotion Recognition Using Depth Sensor and Interfacing with Second Life Based Virtual 3D Avatar [2]

- Procedure identified for implementation

Face recognition with the Kinect sensor using the Action Units (AU) of facial action coding systems to identify emotions (FACS). Making an avatar's feelings in the second life 3D virtual world and connecting a 3D Second Life avatar with the Kinect depth sensor's tracked emotions

- Objectives
  Over the tracked face, construct a facial mesh and to identify the regions that are crucial for emotion recognition. To assess and identify the feeling also adding feelings to the second life avatar to integrate second life with the emotions recognised by the Kinect sensor.

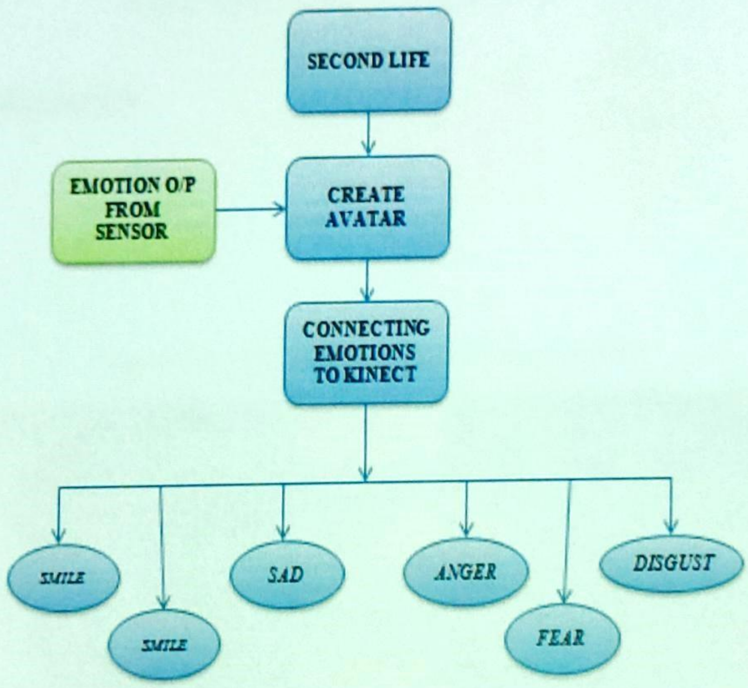

Figure 1.3: Flowchart for expressing emotions in a virtual 3D avatar based on second life

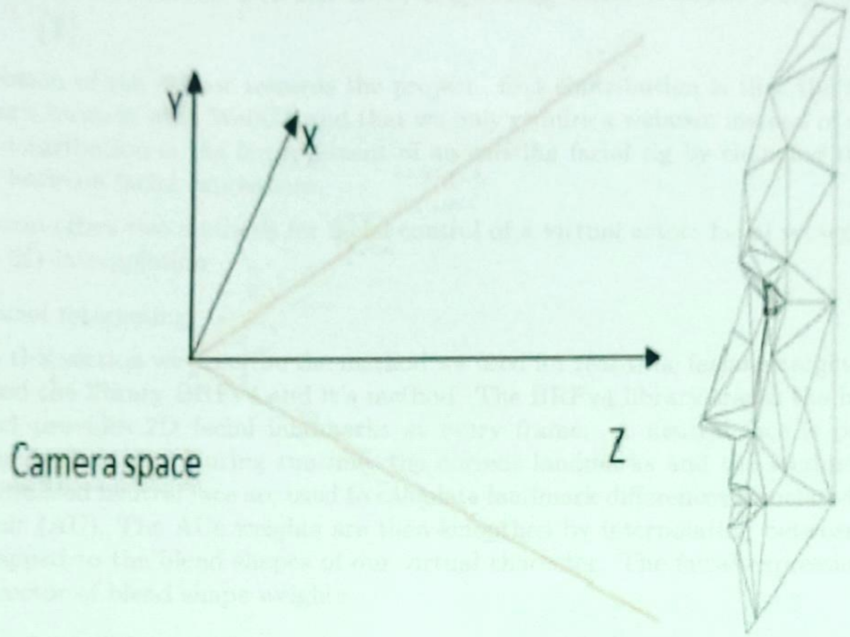

Figure 1.4: Face tracking 3D coordinate system

Figure 1.5: Retargeting

- Conclusion

  The suggested real-time emotion identification system's primary flaw is that it excludes the disgust emotion. When a person is one meter distant from the Kinect depth sensor, the emotions can be seen. The next phase of the experiment will concentrate on classifying disgust and contempt as emotions.

1.4.2 Real-time Facial Retargeting and Facial Rig on the Web [1]

Contribution of the author towards the project. first contribution is that the system works in the client's browser with WebGL and that we only require a webcam instead of a depth camera. Second contribution is the improvement of an existing facial rig by changing the interpolation formula between facial expressions.

The system offers two methods for facial control of a virtual actor: facial retargeting and facial rig with 2D interpolation

- Facial retargeting

  In this section we describe the method we used for real-time facial retargeting system. We used the library BRFv4 and it's method. The BRFv4 library tracks the head orientation and provides 2D facial landmarks at every frame. A neutral face is pre-initialized in our application. During runtime, the current landmarks and the landmarks of the pre-initialized neutral face are used to calculate landmark differences associated to each Action Unit (AU). The AUs weights are then smoothed by interpolating between frames. Then mapped to the blend shapes of our virtual character. The facial expression is defined by a vector of blend shape weights.

- Facial Rig With 2D Interpolation

  In this section we briefly explain the 2D valence-arousal emotional space to control the character's facial expression and introduce the different interpolation methods for the facial rig. In the valence-arousal space we predefine several facial expressions for different positions. The user can select any point of the 2D space and a facial expression is generated automatically according to the surrounding predefined facial expressions. In order to interpolate between the predefined facial expressions and to generate a facial expression at any given point of the 2D space, we implemented natural neighbour interpolation (NNI). Given a point in a 2D space, this method finds the nearest neighbouring points and assigns a normalized weight to each neighbour according to their respective distance.

- Results and discussion of the paper

  We observed that facial retargeting and the facial rig provide a fast method to create facial animations. Although the BRFv4 library is the best implementation in the web we have found, we noticed some noise in the mouth and eyelids landmarks. When blinking, the eyelids did not open and close symmetrically, which created a bizarre effect. Currently, we opted for automatically generated blinks in our applications.

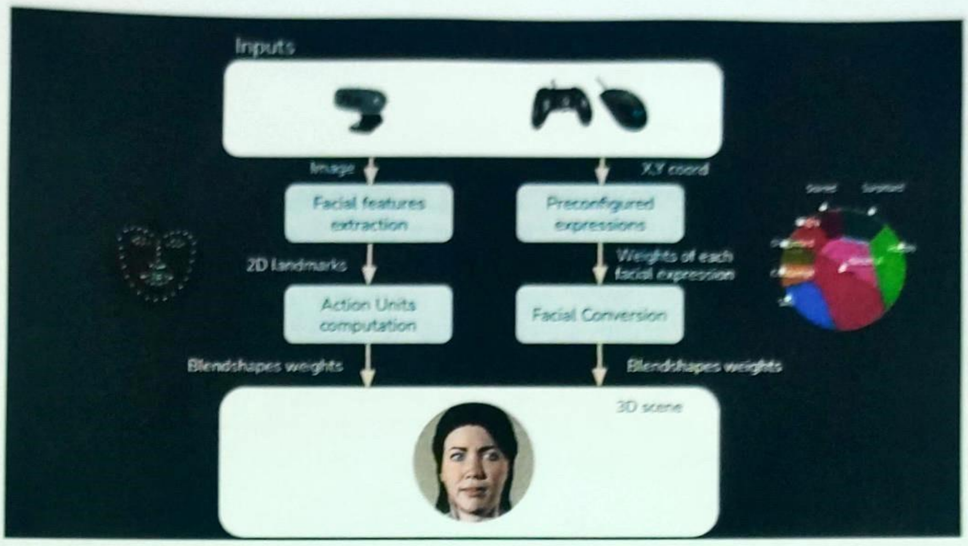

Figure 1.6: Pipeline of facial emotions mapping to avatar

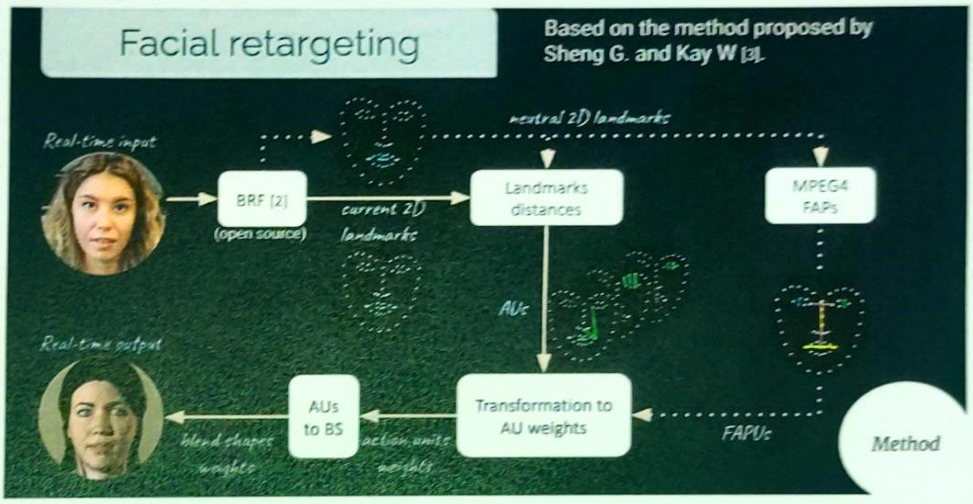

Figure 1.7: Real-time facial retargeting and a facial rig on the web

1.4.3 Individual's Social Perception of Virtual Avatars Embodied with their Habitual Facial Expressions and Facial Appearance [3]

- Procedure identified for implementation
  - **Participants** - To guarantee that individuals could reliably recognize visual cues, we only included participants with a corrected vision of at least 0.7. Each participant completed a consent form after being informed of the experiment's goals and steps.
  - **Video Stimulus** -In the present study, participants' facial reactions to video stimuli were elicited in order to generate data for an individualized avatar. We chose video materials that were known to arouse emotions; this was empirically confirmed by an experiment done at Stanford University and made available by them (n = 411, [37]). We chose two candidate stimuli from Stanford's materials for each emotional state (positive and negative) [37]. All candidate materials underwent a manipulation check. Participants

thought the two video stimuli were positive in terms of the positive stimuli. There was no discernible difference between the outcomes and those of the Stanford research.

- **Video Analysis** - We utilized Open Face, which is free software that uses deep neural networks to recognize faces [38]. The Facial Action Code System (FACScore)'s unit of analysis was the AU (Action Unit) [39]. Figure 1 illustrates the procedure. The facial region from the participant films was first adjusted. The 200 x 200-pixel images that make up the video were placed in order. We extracted the 68 face landmarks and the degree of AU movement from this image series (see Figure 2). Key facial areas' coordinates (x,y) are extracted by landmarks (such as the chin, lips, nose, brows, and eyes). In HOG (Histograms of Oriented Gradients), the AU vector data were used to determine the movement and intensity of AU [40] virtual Avatar- To represent the participant's expressive habits and facial features, we created two basic avatars, one for men and one for women (see Figure 3). We altered an open-source, publicly accessible model for the female model.

- **Procedure-** A week passed between the first and second sessions of the experiment, which was conducted twice. The attendees in the first experiment were given an overview of the study's methods and intent. Then, as the two emotional cues appeared on the display, participants watched them calmly. Instead of pushing any certain attitude, they were instructed to show the natural expression they felt from the audience. The webcam on the display recorded a video of the participant's facial expressions for 90 seconds.

- Conclusion of the paper

  - This study is the first to show that people can identify with a virtual person who mimics their distinctive facial movements, which has important implications for the creation of virtual agents.

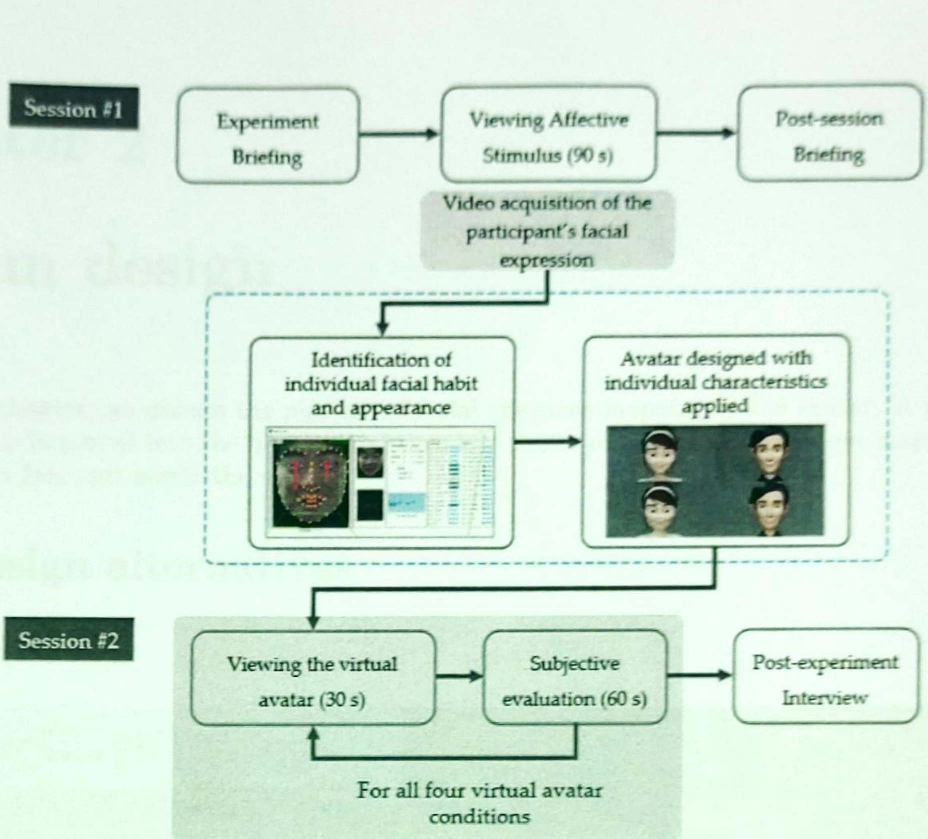

Figure 5. The experiment consists of two sessions, with one week in between for each participant.

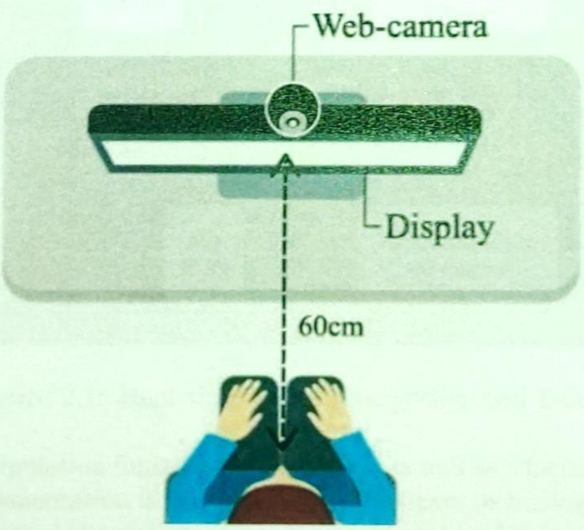

Figure 1.8: The experiment environment

Chapter 2

System design

In this chapter, we discuss the pipeline of facial emotions mapping to the avatar. A pre-rigged model is imported into the blender and then key points of the rigged model get mapped onto the user's face and mimic the same.

2.1 Design alternatives

2.1.1 Facial rig

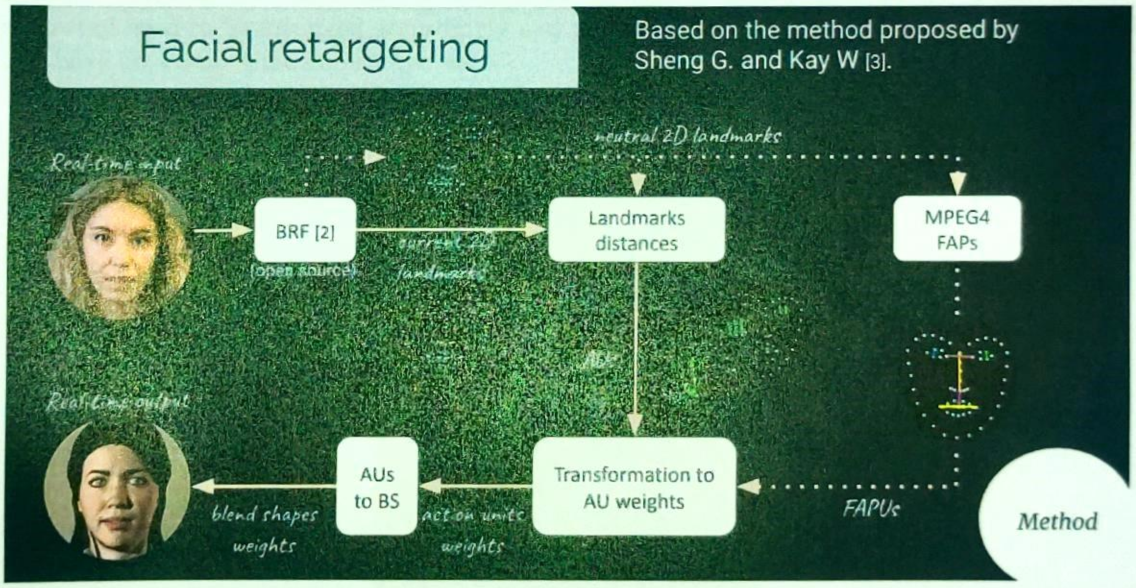

Figure 2.1: Real-time facial retargeting and a facial rig on the web

A new interpolation function for a facial rig as well as a facial retargeting system. This shows that such implementation is feasible and uses webcam technology in the browser and these tools with various virtual characters. It needs to undertake additional user and animator reviews. In addition to supporting these technologies, this application is built on an open-source platform, allowing others to access and alter the tools. We hope that the techniques described here will expand the potential for facial animation on the web and enhance the animation pipelines, enabling more effective interactive virtual agents that may be used online.

2.1.2 Using mediapipe to extract facial keypoints

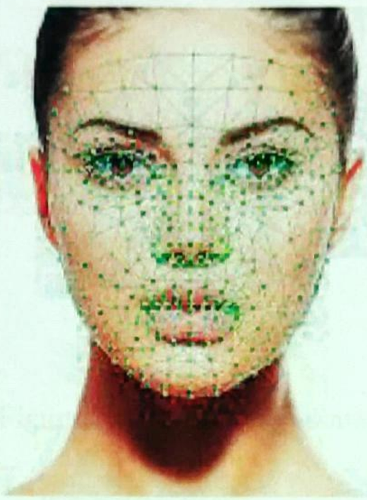

Figure 2.2: Extracting keypoints using mediapipe

We extract the key points using Mediapipe and store this in mathematical form(data) and further, we do the same with the avatar that we created using Unity 3D(other platforms like Blender, real engine). Once we get these data we need to map it i.e., the key points that we extracted need to be arranged accordingly and put onto the avatar that we created. Finally, we will have an avatar that mimics the user.

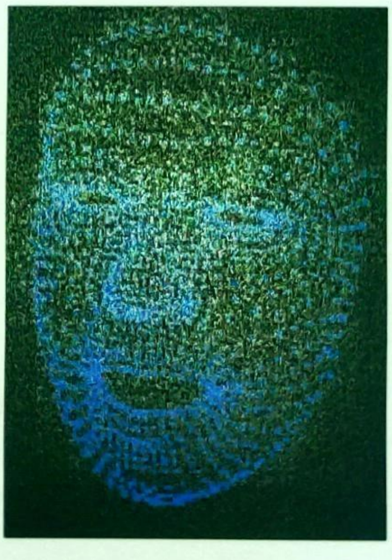

Figure 2.3: Human keypoint

2.1.3 3D registration

This is a comprehensive workflow for building fully rigged, unique 3D facial avatars from hand-held footage. By adjusting a blendshape template to an image series of captured expressions using an optimisation that incorporates feature tracking, optical flow, and shape from shading,

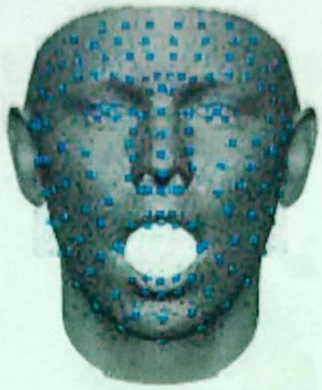

Figure 2.4: Avatar keypoints

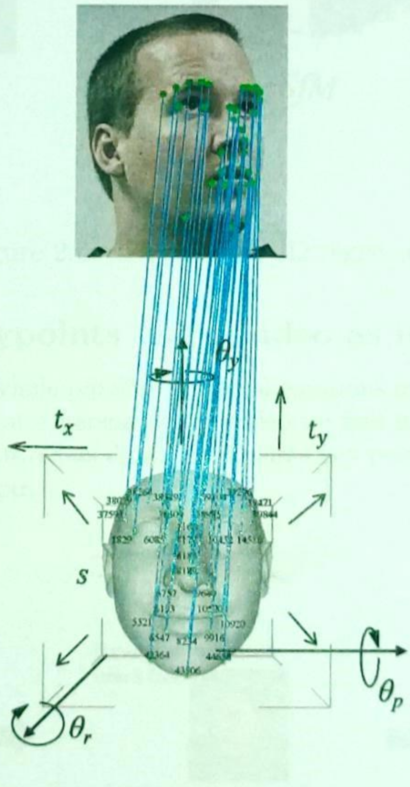

Figure 2.5: Retargeting the keypoints

this method accurately recreates the user's facial expression dynamics. Normal maps and ambient occlusion maps each separately capture fine-scale characteristics like wrinkles. Develop a regressor for on-the-fly detail synthesis during animation from this user- and expression-specific data in order to increase the perceptual realism of the avatars. This approach illustrates that even with a basic acquisition setup and minimal user assistance, the deployment of suitable reconstruction priors results in compelling face rigs.

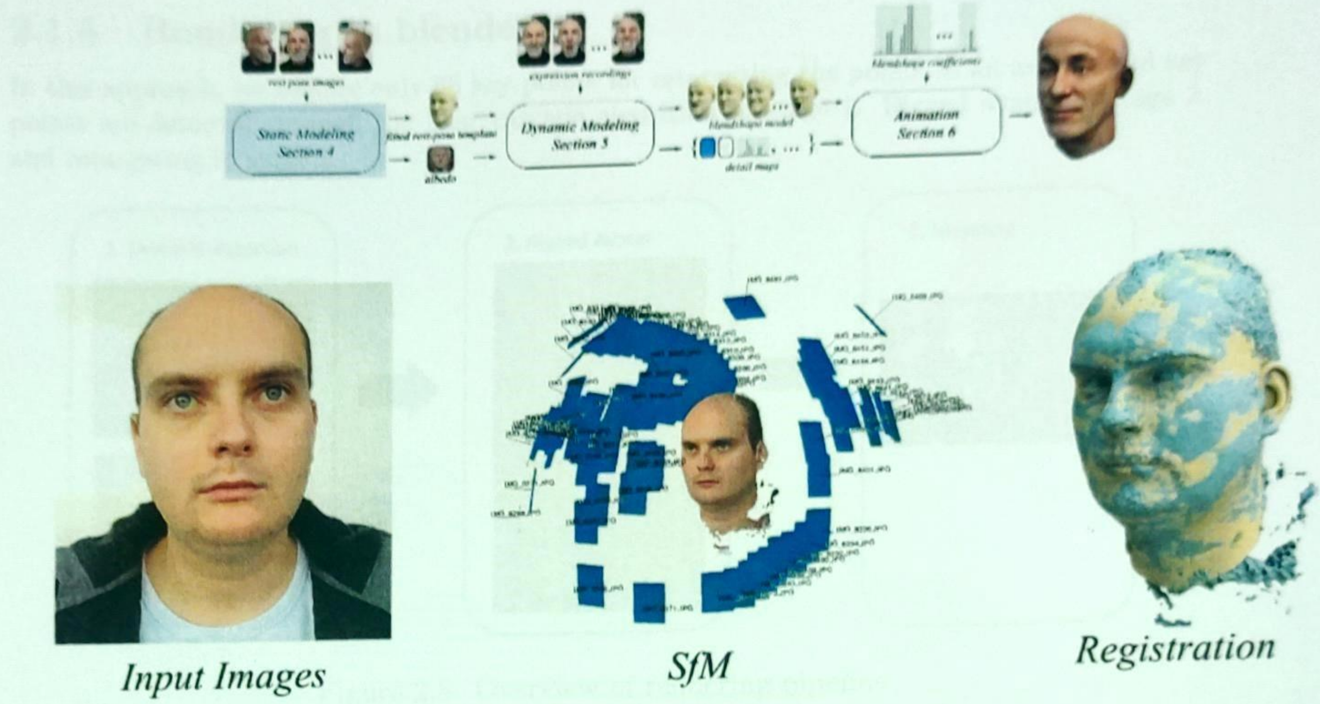

Figure 2.6: Overview of 3D registration

2.1.4 Detecting keypoints using video as input

In Figure 2.1, we discuss the whole pipeline for facial emotions mapping to Avatar. The camera captures the real-time image of a person whose video we had given as the input earlier. Now key points get extracted and stored as data. Next, these key points are mapped onto the avatar and we have a mimicking avatar.

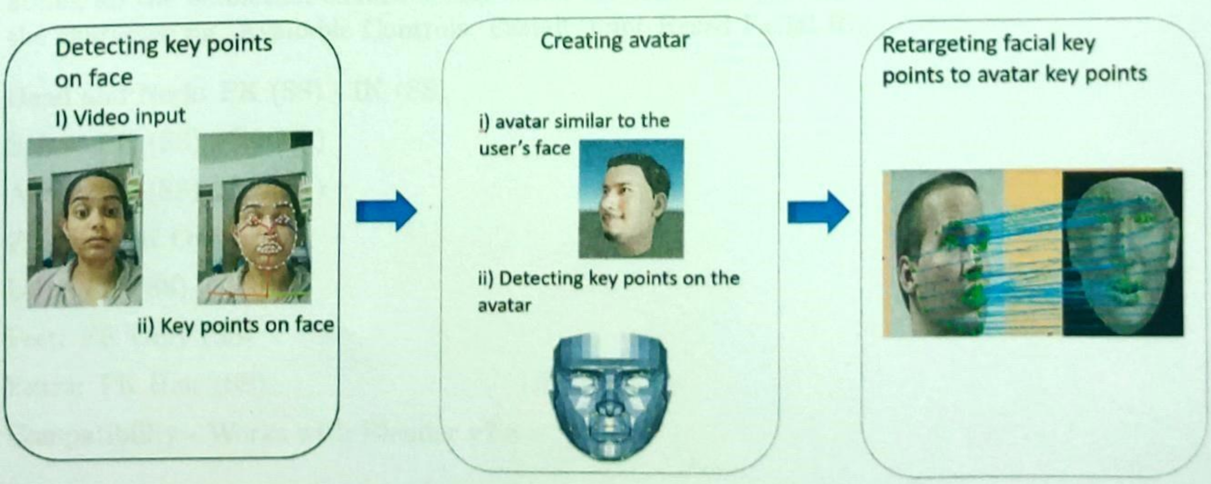

Figure 2.7: Overview of rendering pipeline

2.1.5 Rendering in blender

In this approach, we require only 68 key points for retargeting the points to an avatar, and key points are detected through the haar cascade algorithm in stage 1. Rigged avatar in stage 2, and retargeting in stage 3.

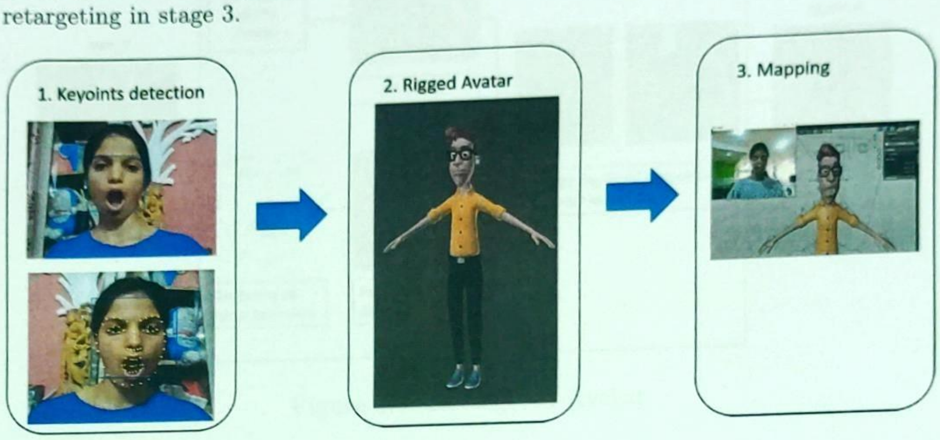

Figure 2.8: Overview of rendering pipeline

2.2 Choosen design

The Vincent rig comes with a learning curve, it's an excellent introduction to Blender's comprehensive rigging controls. The rig works with squash and stretch but doesn't include the bendy bones so often seen in stylized characters. Skeletal System Despite having complex manipulation options, the skeletal system for the Vincent rig is simplistic and rather sparse. An interesting side note is that the pole vectors for the IK system are attached directly to the knee joints.

Controls:

The "Picker System" that has been put in place is the main emphasis of this control rig. Enabling all the animation elements they think are essential, provides the user total control over the character rig. Available Controls: Facial: Joint-Based Facial Rig

- Head and Neck: FK (SS) - IK (SS)
- Spine: FK (SS) - IK (SS)
- Arms: FK (SS) - IK (SS)
- Fingers: FK Only (SS)
- Legs: FK (SS) - IK (SS)
- Feet: FK Only (SS)
- Extra: FK Hair (SS)
- Compatibility - Works with Blender v2.8

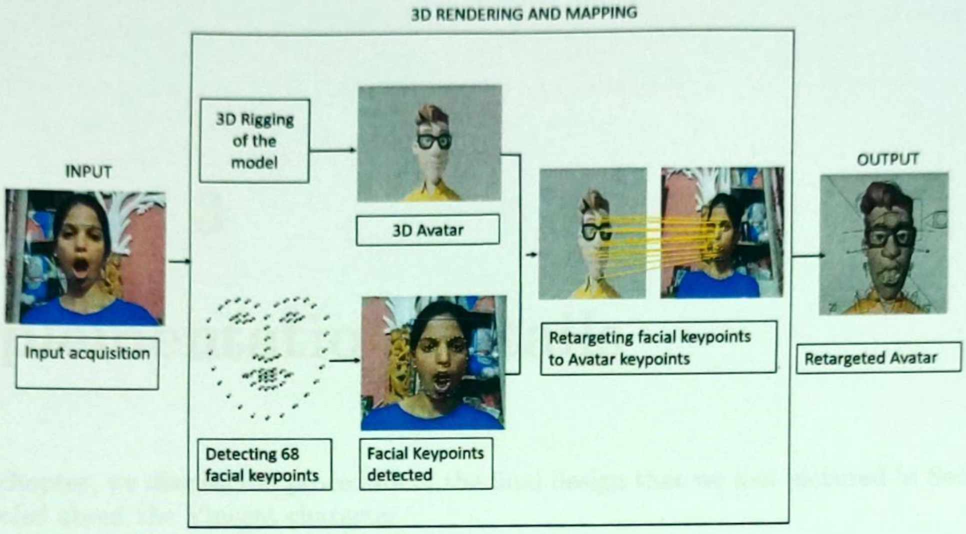

Figure 2.9: Retargeted Avatar

In this block diagram, we take input through the camera, detecting facial key points (68) and we have an character which is been rigged earlier. Retargeting facial key points of the user to the avatar key points and hence the retargeting makes the avatar to mimic same as the user.

Chapter 3

Implementation details

In this chapter, we discuss the procedure of the final design that we had pictured in Section 2.3 and briefed about the Vincent character

3.1 Procedure

Adding controls to a character mesh is the process of 3D rigging. Your character is powerless without a rig. In reality, if an object has to interact with a character in a scenario, they are given basic rigs. When animating characters using motion capture or keyframes, a character rig is used.

Inverse-Kinematics (IK) and Forward-Kinematics (FK) are the two basic categories of rig controls. FK rigs enable more mechanical movement because you have complete control over the rotation and location of each joint. IK rigs are more sophisticated since they can move a whole chain of joints. The fundamental idea behind IK rigs is that if someone grabbed your wrist and pulled it to the left, your elbow joint would move and your shoulder joint would even spin.

The majority of pre-rigged characters feature a "Control Rig" A control rig is a group of predetermined controls that have a direct impact on how the skeletal system of the character moves. All arrows, icons, or wire shapes (often in bright green) are a part of the control rig that you'll use to move the character, even though there's a lot going on in the background of a rig. Any rig's skeletal system will replicate the subject's bone structure, such as the human body or skeleton. This skeleton system is skinned to a character mesh. The character mesh's vertices will then move as the rig moves going forward. That's what gives the appearance of motion.

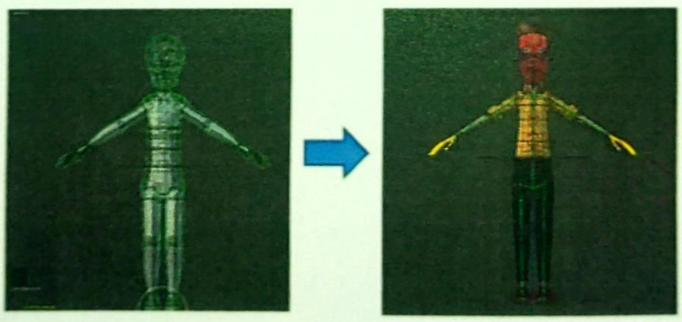

Figure 3.1: Rigging of the rendered character

Once we understand the above topics, it's time for the actual implementation!

- Finding interesting points in a picture of a human face is a method known as facial landmark detection. Because it has so many useful applications, the computer vision community has recently witnessed rapid growth. For instance, we have demonstrated the capability to recognize emotion through facial motions, eye tracking, face swapping, adding visuals to the face, and manipulating virtual characters.
- Here we will be using haar cascade xml file A feature-based object detection system called Haar Cascade is used to identify things in photos. For detection, a cascade function is trained on a large number of both positive and negative images. The approach can run in real-time and doesn't call for a lot of computing.
- Now, these key points are stored in the form of data(array).
- Keeps a moving average of a given length.
- Detect faces and detect only the biggest face, and keep it.
- Detect the landmarks.
- 3D points are extracted and stored in the form of data.
- Determine head rotation.
- Set bone rotation/positions.
- Display camera pop-up on Windows.
- starts mapping the extracted points onto the rendered character (avatar).
- Avatar starts to mimic the user.

3.2 Algorithm

Haar Cascade is a feature-based object detection algorithm to detect objects from images. For detection, a cascade function is trained on a large number of both positive and negative images. The approach can run in real time and doesn't call for a lot of computing. Haar cascade is 1-2 percent better than LBP cascade in accurately detecting the location of a face. A Haar-like feature is a straightforward rectangular feature that can be used to detect image edges and lines. Face detection Haar-like features are often constructed of rectangular patches with varying gray-level intensities. A rectangle with dark pixels at the top and light pixels at the bottom, for example, could be one Haar-like feature. The classifier can detect whether the feature is present in the image by analyzing the intensity values of the pixels within each rectangle.

Algorithm 1 An algorithm for our proposed approach

Ensure: .

Acquire a new image from camera.
Convert color to gray scale.
Detect face using Cascade algorithm.
if Face detected **then**
    Detect eyes in face
    if Two eyes detected **then**
        Normalize face image size and orientation
        Apply Contrast and lightning enhancement.
        Facial recognition using PCA classifier.
    else.
        Detect face
    end if
else
    Detect face
end if

3.3 Flowchart

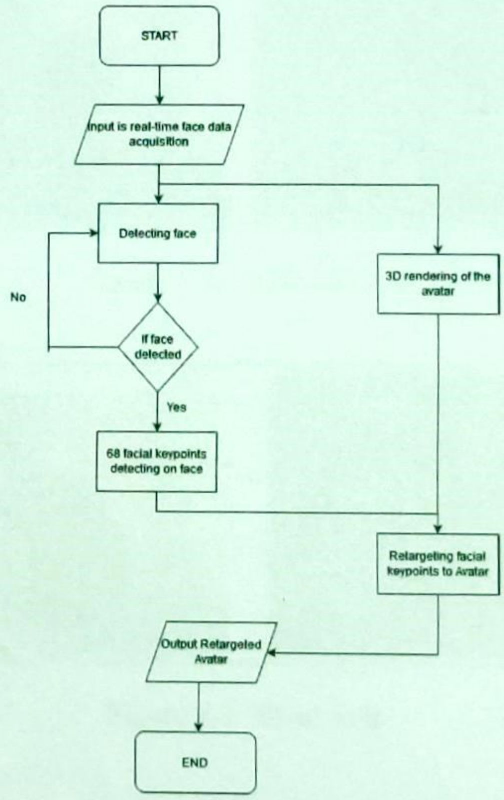

Figure 3.2: Flowchart

Chapter 4

Results and discussions

In this chapter, we discuss the key points detected on the face using haar cascade algorithm, and the avatar is rendered in blender.

4.1 Result analysis

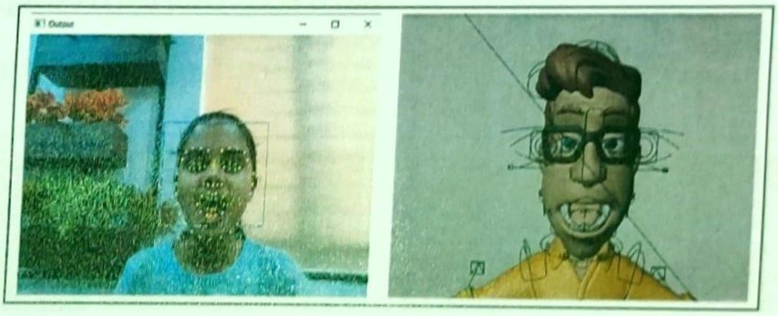

Figure 4.1: Wide Mouth

Figure 4.2: Head Left

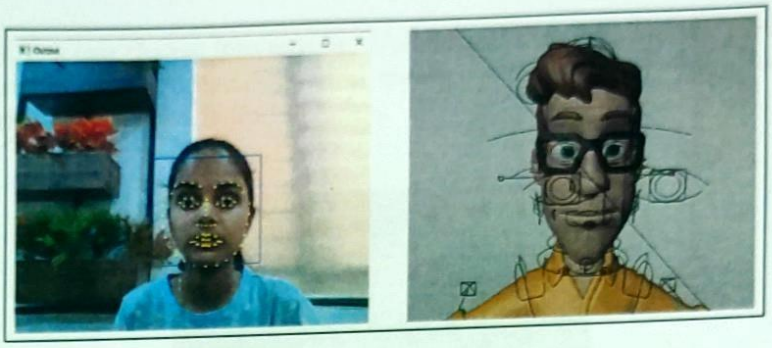

Figure 4.3: Wide Eyes

In this project, we are detecting key points using haar cascade algorithm in xml file format and we set webcam resolution to width 640 and height 480 and formed an array of coordinates with the values of the nose tip, chin, etc. By finding the faces and landmarks and determining head rotation set bone rotation and position, mouth position, eyebrows, eyelids, and drawing face markers, drawing a detected face.

4.2 Experimental results

4.2.1 Unity face capture

Figure 4.4: Unity face capture

As mentioned we intend to map the facial key points to the key points of the avatar which we created. Here is a snippet of what our output should look like, we have used an application called Unity face capture by IOS to demonstrate the same. This APK is available only in IOS currently we intend to make this available on Android and other devices too.

4.2.2 Avatarify

An animated image file can be created using the multimedia software program Avatarify. An advanced neural network engine is used by the app to add animated emotions and expressions to the image. For photographs containing a human subject, such as a selfie or a family portrait, Avatarify is appropriate.

Using Avatarify, you can change into anyone. Upload a picture of a famous person or your boss and make a short video. An advanced neural network will add all of your emotions and facial expressions to the shot to spice it up.

Figure 4.5: Monalisa mimicking facial emotions in Avatarify

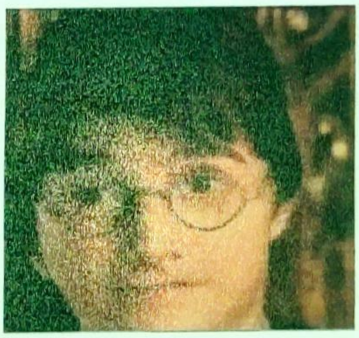

Figure 4.6: Harry mimicking facial emotions in Avatarify

Figure 4.7: Obama mimicking facial emotions in Avatarify

Chapter 5

Conclusions and future scope

In this chapter, we discuss the conclusions and future scope of facial emotions mapping to the avatar

5.1 Conclusions

We mapped the user's facial emotions to an avatar, this is done using Blender where in we give them real-time facial expressions as input, and these expressions are mapped onto the avatar which is a rigged character. When the input facial expression changes, the avatar's expression is also changed successfully.

5.2 Future scope

Metaverse is a platform that will replace the internet in the coming days.

- VR chat (platform for an online virtual world) – Avatar represents the user and this creates a world that has no limits and can be explored more.
- Online meeting platforms (Microsoft, Zoom, etc) – If the user doesn't have a presentable face or doesn't have a good background, an avatar or the deep fake(if a user likes to) comes to the rescue. This covers the user's face with an avatar that they feel similar to and the selected avatar mimics the face and feels as if the avatar is speaking.
- The above mentioned point will be achievable by creating a cloud server and exporting the rendered character(avatar) to it and make it available to other user who wants to get in communication with the user.
- Use of CSI(Camera serial interface) of the raspberry pi which will be replaced as web camera, with this we can alter the resolution as needed and the next procedure remains the same.

Furthermore, we can also include options such as converting the text to speech if the background has more noise,this can be done using deep learning and many more add on's. This can be a complete change.

Bibliography

[1] (ICGI 2019)" "By Eva Valls Garolera et al. "real-time facial retargeting and a facial rig on the web".

[2] (ICRAIE 2014)" "By Monika Kakarla et al. "a real-time facial emotion recognition using depth sensor and interfacing with second life based virtual 3d avatar".

[3] (SENSORS 2021)" "By Sung Park et al. "individual's social perception of virtual avatars embodied with their habitual facial expressions and facial appearance".

cav

ORIGINALITY REPORT

12%	8%	8%	%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	media.rokoko.com Internet Source	3%
2	www.mdpi.com Internet Source	2%
3	Alexandru Eugen Ichim, Sofien Bouaziz, Mark Pauly. "Dynamic 3D avatar creation from hand-held video input", ACM Transactions on Graphics, 2015 Publication	2%
4	Sung Park, Si Pyoung Kim, Mincheol Whang. "Individual's Social Perception of Virtual Avatars Embodied with Their Habitual Facial Expressions and Facial Appearance", Sensors, 2021 Publication	1%
5	ijarcce.com Internet Source	1%
6	www.irjmets.com Internet Source	1%
7	www.tib.eu	

Internet Source

1 %

8	Mounika Kakarla, G.Ram Mohana Reddy. "A real time facial emotion recognition using depth sensor and interfacing with Second Life based Virtual 3D avatar", International Conference on Recent Advances and Innovations in Engineering (ICRAIE-2014), 2014 Publication	<1 %

Exclude quotes On

Exclude bibliography On

Exclude matches < 5 words