VRmoji : Natural Avatar Movement based on Real-time Facial Expression Recognition System

Changhyeon Park*
KAIST

Youjin Sung†
KAIST

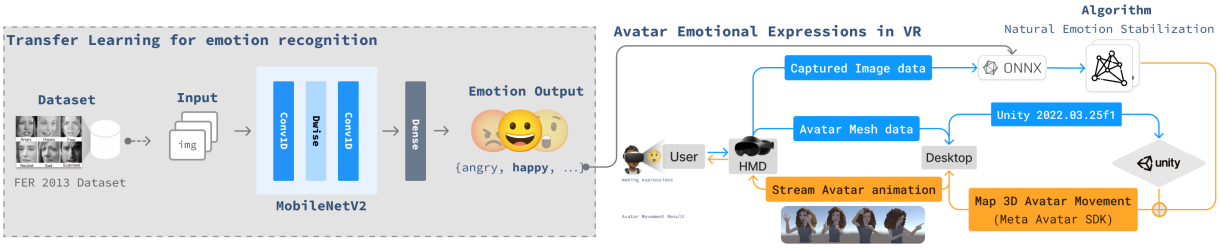

Figure 1: System Flow of the VRmoji. Emotion Recognition Model(upper), Avatar Animation Generation(lower).

Abstract

With the development and popularization of virtual reality communication content, the need to recognize user emotions for realism and immersion in interaction is emerging. Due to the instability of existing emotion recognition machine learning models, there are limitations in using them for real-time recognition in VR. In this paper, we propose an algorithm for stable and continuous emotion recognition which calculates emotion intensity and generates appropriate 3D Avatar Emotion Animations. By supporting VR specialized natural avatar movement in real-time, it enhances the user experience while minimizing the uncanny valley effect of avatars in VR communication situation.

Index Terms: VR Avatar, Emotion Recognition

1 Introduction

While we are having a conversation, we communicate with various verbal and non-verbal expressions. However, in Virtual Reality (VR), it is difficult to express it because the face is half-covered while wearing the head-mounted displays (HMDs). In order to address this issue, there is a growing interest in the developing natural avatar movement based Facial Expression Recognition (FER).

Emotional expressions in 3D avatars are not a recent phenomenon, encompassing familiar elements like text-based emojis and customizable Memojis. However, it is the ongoing challenge of rendering natural emotional expressions in VR avatars since VR employs distinct sensing technologies compared to traditional mobile environments.

In this paper, we used machine learning (ML) to infer emotion from user facial expression images and then applied stabilization and emotion intensity calculation algorithms to generate natural avatar emotion motion.

2 Related Works

In VR, there are primarily two methods for expressing emotions through avatars; (1) directly mapping the user's facial expressions onto the avatar, and (2) activating preset avatar animations. Reconstructing facial expressions onto a 3D avatar is actively researched. However, this method leads to a uncanny valley effect, as the avatar's movements may not fully capture the subtle expressions of a real person. To address this, preset animation are utilized. For instance, users in VRChat[1] manually activate preset animations to express emotions. Additionally, non-emotional facial expressions are also used to classify user emotions[8]. For instance, specific movements in facial part can be analyzed as emotion by using Facial Action Coding Systme (FACS)[2]. While this enables natural animation, it still leaves room for improvement in terms of the manual control. Consequently, there is a trend towards using machine learning models on existing datasets for more sophisticated analysis of user emotions[10, 7, 6, 3, 4]. However, as the primary focus of this kind of research is on classification, there remains significant scope for improvement in implementing theses techniques in real-time avatar motion. Thus, we focused on natural real-time avatar emotion recognition and expression process.

3 Emotion Recognition Algorithm

3.1 Facial Expressions Recognition Model

In order to recognize the user's facial expression in real time, computation speed is one of the key requirements while inferring or generating appropriate emotional expressions. To minimize the latency caused by computation process of emotion inference, we utilized MobileNetV2 which is pre-trained on image net data and requires less computations[9]. We used MobileNetV2 as a feature extractor and added three dense layers at the end to serve as the classifier. The designed model used an facial image of size $(224 \times 224 \times 3)$ as input data and provided the probability of 4 emotions $(neutral, happy, angry, surprised)$ as a final output(See **Section 4**).

Machine learning models including FER images as input have its limitation on being sensitive to environmental factors and quality of the images. Therefore, when you decide to use the raw output of the model without post-processing, you should be aware of the fluctuating changes in the avatar's emotional expression, especially in the real-time based system.

To overcome system instability, we designed an algorithm that takes temporal factors into account. We provided animation effects for each emotion intensity to maximize the effectiveness of

*e-mail: sac7160@kaist.ac.kr
†e-mail: 672@kaist.ac.kr

[1] https://hello.vrchat.com/

avatar emotion expression. For continuous expression intensity estimation, data labeling for facial expression intensity is difficult and time-consuming, so regression techniques have been the dominant method for continuous expression intensity estimation [11]. However, to minimize inference delay and simplify the model within the Unity VR scene, we used the output of the emotion classification model for emotion intensity estimation. To summarize, to calculate emotion intensity, we utilized the emotion probability of the emotion classification machine learning model output[1] as well as comparing the similarity of images to implement reliable detection of emotion intensity.

3.2 Algorithm 1 : Natural Emotion Stabilization

Algorithm 1 `Natural Emotion Stabilization`

Input : Inferred Emotion, Inferred Probability
Output : Final Emotion, Probability Average of the Emotion
threshold = Emotion Probability to use

1: **while** *Inference progresses* **do**
2: **if** $EmotionProbability > threshold$ **then**
$queue \leftarrow \{EmotionIndex, EmotionProbability\}$ from **ML Model** per inference
3: **else**
Discard inferred values
4: **end if**
Emotion = Find the largest number of emotions in the queue
EmotionProbability = The average of the probability of the emotion
5: **end while**
6: **return** $Emotion, EmotionProbability$

To prevent sudden fluctuations in avatar emotion motion due to incorrect inference of the emotion inference model, we observe the emotion inference results over a period of time and apply the mode value to the avatar emotion motion, rather than applying the results of each model inference to the avatar emotion motion. We used a *queue* as a buffer as shown in *Algorithm* 1 (**3.2**). Also we set a threshold to exclude uncertain inference results and utilized inference results only when they are larger than the threshold. The threshold was arbitrarily set to 0.7 for stable accuracy.

In VR environments, humans typically perceive delays of 30-50*ms* or more for visual cues. When we deploy our trained model in Unity and use it to make inferences, it takes about 3*ms* at a rate of about 300Hz. To take this into account and maximize the size of the *queue*, we set 50 / 3 $\approx$ 17(elements) to be half the size of the *queue*, resulting in a *queue* size of 34(elements).

3.3 Algorithm 2 : Calculating Emotion Intensity

We utilitzed two factors to calculate intensity of the emotion. When calculating the final emotion intensity, we utilized the two factors in equal proportions and matched the scales of each factor.

- Emotion Probability from *Algorithm* 1(**3.2**)
- Facial image similarity to neutral facial image (**Figure 2**)

We utilized the similarity of facial expression images as well as the emotion probability of the output of the machine learning model, as in previous research.[1] In general, the higher the intensity of an emotion, the greater the change in facial expression, so image similarity can be used to calculate the intensity of an emotion, which is very intuitive. The user registers a neutral facial expression image to the system before starting the application and compares the similarity with this image. For image similarity comparison, a histogram of each image is generated and correlation is

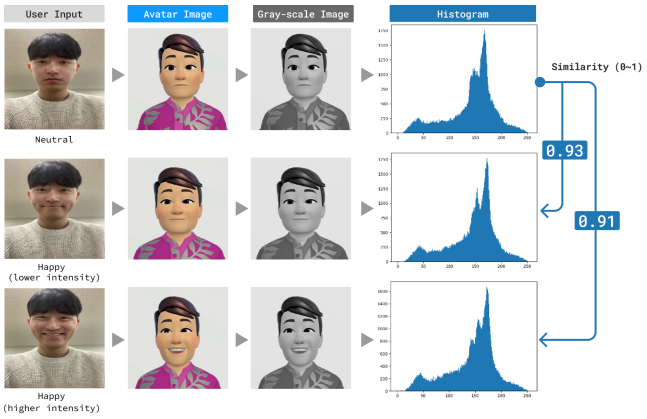

Figure 2: We converted facial expression images to gray scale, generated an image histogram, and measured the similarity between images to calculate emotion intensity.

analyzed to measure the correlation between the two histograms to measure the similarity between images.

Since the degree of facial expression change varies considerably between emotions, we adjusted the intensity level before playing the avatar animation or each emotion, . For example, unlike 'happy', which has significant facial expression change, 'angry' has less facial expression change. Furthermore, the main feature of 'angry', the furrowed brow, is not represented in the avatar, making the avatar's facial expression change even less.

4 3D Avatar Animation

For the initial implementation of natural emotional expressions in avatars, we selected 4 expressions from basic human emotions: *neutral*, *happy*, *angry*, and *surprised*. These were chosen based on their distinctiveness and recognizability within a VR. For instance, *happy* is often challenging to distinguish accurately in VR while *angry* and *surprised* were relatively easily recognized[5]. A total 9 different preset is designed (3 emotions $\times$ 3 levels) from Avatar Recording App from prefrontal cortex[2]. To distinguish the level of the emotion, we applied logic for each action. For instance, we set single thumbs up in level 1, double thumbs in in level 2, and double thumbs up with clapping action in level 3 for happy.

In our system architecture(Figure1)for VR environments, the integration of user interaction and 3D avatar synchronization is enabled using the Quest Pro. Real-time facial expressions and movements captured by the HMD cameras are transmitted via Air Link to a desktop system. On the desktop, the Unity software platform (version 2022.03.25f1) utilizes these data to animate a 3D avatar that accurately reflects the movements (Meta Avatar SDK).

5 User Study Design

We generated 3D avatar animation (Figure 3) mapped them to the user's facial expressions in real-time for each emotion and its intensity level (level 1 to level 3). As we reduced the latency, the whole process can be done without disturbing user experience. To measure the performance, we conducted pilot study with 2 people (1 female, median 25 years old) with two conditions.(See Figure 4). As a result, participants reported that our system supports natural communication in terms of the emotion expression for all emotions (angry, happy, and surprised). In the following interview, P2 mentioned that *"The system allowed me to express richer expressions beyond the limits of conventional facial expression recognition. Be-*

[2]https://prefrontalcortex.de/en/

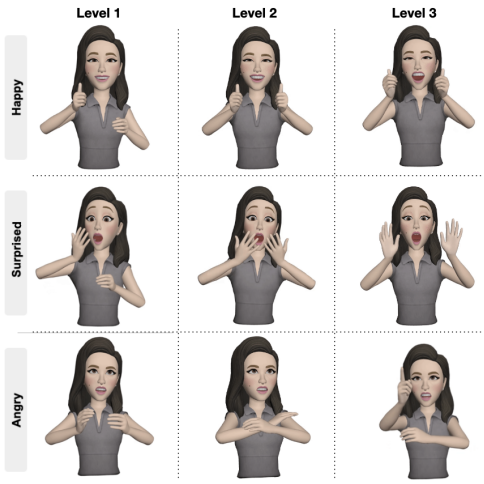

Figure 3: For each emotion (happy, angry, surprise), we created a different avatar emotional animation based on the intensity of the emotion.

ing able to adjust the thresholds myself was very convenient for freely expressing my emotions.".

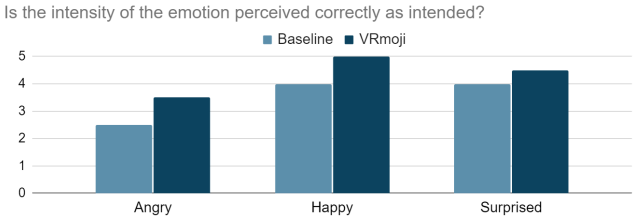

Figure 4: Questionnaire on emotion recognition accuracy and intensity accuracy between avatar only (baseline) and our system (VRmoji) (1-5 Likert Scale).

6 Application

For the possible use case scenario, we suggest general communication situation in VR. When the user wants to convey more beyond text or voice, such as nonverbal communication, VRmoji can support user to effectively express own emotion as intended. In **virtual meetings** and **conferences**, participants can interpret their colleagues' emotions more effectively through smiles and nods, which improves communication during discussions and presentations. Meanwhile, **multiplayer gaming** and **role-playing** experiences are enriched with natural facial emotes, giving characters greater depth and realism, resulting in more immersive and dynamic interactions between players and NPCs. We believe this is essential in an increasingly advanced and ubiquitous digital world, including VR.

7 Conclusion

This paper suggested natural avatar emotion using facial expression recognition in a VR environment. This study is significant in that it goes beyond emotion recognition using facial expression images and applies it to real-time VR applications, but further research is needed for more stable and accurate emotion recognition and expression. First of all, we need to find the optimal value of the threshold set arbitrarily in the *Algorithm* 1 (**3.2**) and the utilization ratio of the two factors (*probability*, and *image similarity*) used in **3.3**. In the future, we will consider this and find the optimal value through user study by experimenting with various number of cases. Next, we need to develop more advanced 3D avatar animation that can be applied to richer and more diverse use case scenarios for universal use of avatar emotion motion. To this end, we will improve the emotion recognition model and further segment the animation according to emotion intensity to enable richer emotional expressions.

References

[1] A. A. Alharbi, M. Dhopeshwarkar, and S. Savant. Detection of emotion intensity using face recognition. In K. C. Santosh and B. Gawali, eds., *Recent Trends in Image Processing and Pattern Recognition*, pp. 207–213. Springer Singapore, Singapore, 2021. 2

[2] P. Ekman and W. V. Friesen. Measuring facial movement. *Environmental psychology and nonverbal behavior*, 1:56–75, 1976. 1

[3] M.-I. Georgescu, G.-E. Duţă, and R. T. Ionescu. Teacher–student training and triplet loss to reduce the effect of drastic face occlusion: Application to emotion recognition, gender identification and age estimation. *Machine Vision and Applications*, 33(1):12, 2022. 1

[4] M.-I. Georgescu and R. T. Ionescu. Recognizing facial expressions of occluded faces using convolutional neural networks. In *Neural Information Processing: 26th International Conference, ICONIP 2019, Sydney, NSW, Australia, December 12–15, 2019, Proceedings, Part IV 26*, pp. 645–653. Springer, 2019. 1

[5] C. N. Geraets, S. K. Tuente, B. P. Lestestuiver, M. van Beilen, S. A. Nijman, J.-B. C. Marsman, and W. Veling. Virtual reality facial emotion recognition in social environments: An eye-tracking study. *Internet interventions*, 25:100432, 2021. 2

[6] T. Gotsman, N. Polydorou, and A. Edalat. Valence/arousal estimation of occluded faces from vr headsets. In *2021 IEEE Third International Conference on Cognitive Machine Intelligence (CogMI)*, pp. 96–105. IEEE, 2021. 1

[7] B. Houshmand and N. M. Khan. Facial expression recognition under partial occlusion from virtual reality headsets based on transfer learning. In *2020 IEEE Sixth International Conference on Multimedia Big Data (BigMM)*, pp. 70–75. IEEE, 2020. 1

[8] T. Ortmann, Q. Wang, and L. Putzar. Facial emotion recognition in immersive virtual reality: A systematic literature review. In *Proceedings of the 16th International Conference on PErvasive Technologies Related to Assistive Environments*, pp. 77–82, 2023. 1

[9] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L. Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4510–4520. IEEE Computer Society, Los Alamitos, CA, USA, jun 2018. doi: 10.1109/CVPR.2018.00474 1

[10] H. Yong, J. Lee, and J. Choi. Emotion recognition in gamers wearing head-mounted display. In *2019 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*, pp. 1251–1252. IEEE, 2019. 1

[11] R. Zhao, Q. Gan, S. Wang, and Q. Ji. Facial expression intensity estimation using ordinal information. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3466–3474, 2016. doi: 10.1109/CVPR.2016.377 2