2026 IEEE/SICE International Symposium on System Integration (SII)
January 11-14, 2026. Cancún, México

Facial Synchronization System for an Android Avatar with an Immersive Interface to Reproduce the Operator's Intended Expressions

Kaoruko Shinkawa[1], Mizuki Nakajima[2], and Yoshihiro Nakata[1]

Abstract— **To enable natural non-verbal communication during immersive avatar control, it is essential to accurately reproduce an operator's intended facial expressions on an android avatar. This study proposes a facial synchronization system that converts facial parameters captured from head-mounted displays (HMDs), into the corresponding expressions on an android avatar. In conventional systems, actuator commands controlling facial movements are typically mapped linearly to facial parameters. However, the correspondence between android facial actions and facial parameters cannot be clearly defined. We developed a motor command mapping model (the "developer model") that generates the developer's intended avatar expressions based on facial parameter data and actuator commands collected through repeated facial mimicry. We then proposed and evaluated a "personalized model," which adapts the developer model to individual operators to better replicate their intended expressions. In a participant experiment comparing three models (conventional, developer, and personalized), no significant differences were observed. However, the developer model—despite lacking personalization—received evaluations comparable to the conventional model, supporting the effectiveness of the proposed approach. In contrast, the personalized model received mixed evaluations, with many participants rating it lower than the other two models, indicating a need to improve the personalization process. This study offers insights for developing more effective facial synchronization systems for immersive android avatars.**

I. Introduction

Robot avatars, which can be operated as extensions of the human self, are expected to support social engagement beyond the constraints of time and physical presence. Ongoing efforts aim to promote their active use in real-world applications [1]–[3]. In recent years, numerous androids with human-like appearances have been developed, some of which can replicate not only body movements but also a wide range of facial expressions [4]–[6]. Using such androids as avatars is expected to enable remote interactions that feel face-to-face, making them promising tools for communication.

An immersive control system is essential for achieving natural remote communication through an android avatar. Head-mounted displays (HMDs) have been widely adopted as control interfaces in recent robotic avatar systems [7]. In our developed android avatar [4], cameras are embedded in each eyeball, and the captured images are presented stereoscopically to the operator through the HMD. This setup enables the operator to view the remote environment from the avatar's perspective with a strong sense of immersion, as if physically present in the same space [8]. Furthermore, using an HMD with facial motion tracking allows the operator's expressions to be reproduced on the avatar. This enables the avatar to convey the operator's emotions through facial expressions, supporting natural and expressive communication during remote interactions.

Synchronizing the avatar's facial expressions with the operator's is essential for facial-expression-based control. However, since androids generally possess fewer degrees of freedom and exhibit different skin deformation than humans, the operator's facial motions to the avatar is not straightforward. To address this challenge, various methods for replicating human facial expressions on androids have been proposed in recent years. Hu et al. [9] used a human-like robot with multiple facial degrees of freedom and developed a model that generates motor commands to align the robot's facial landmarks with those of the operator, detected through a camera. Wu et al. [10] proposed a more general approach that quantifies human facial expressions captured by a camera using blendshape weights and generates robot expressions by learning the correspondence between the operator's and the android's blendshape weights.

However, prior studies typically assume a setup in which a camera is positioned directly in front of the operator. Facial synchronization systems tailored for immersive control using HMDs remain underexplored. Moreover, although existing methods aim to generate avatar facial expressions that visually resemble those of the operator, the avatar's face often differs structurally from the operator's. As a result, even if facial landmarks or blendshape weights are aligned, the expressions cannot be accurately reproduced and might fail to reflect the operator's intent. Thus, a new facial synchronization system is needed for immersive operation—one in which the avatar's expressions are perceived by the operator as faithfully reflecting the operator's intent.

In this study, we developed a facial synchronization system for android avatars by constructing a mapping between the operator's intended expressions and the motor commands required to reproduce them, using data collected as the operator mimicked the avatar's facial expressions. The contributions of this study are as follows:

- Proposal and development of a motor command map-

*This work was supported by JST Moonshot R&D Grant Number JPMJMS2011. One of the authors is supported by JST SPRING, Grant Number JPMJSP2131 (Scholarship award).

[1]Kaoruko Shinkawa and Yoshihiro Nakata are with the Department of Mechanical and Intelligent Systems Engineering, Graduate School of Informatics and Engineering, The University of Electro-Communications, Chofu, Tokyo, 182-8585, Japan kshinkawa@uec.ac.jp, ynakata@uec.ac.jp

[2]Mizuki Nakajima is with Department of Robotics and Mechatronics, School of Science and Technology for Future Life, Tokyo Denki University, Adachi-ku, 120-8551, Tokyo, Japan. mizuki.nakajima@mail.dendai.ac.jp

ping model to reproduce the operator's intended facial expressions during immersive avatar operation using an HMD
- Accuracy evaluation of the proposed model through comparison with a conventional motor command mapping model based on linear mapping
- Proposal and implementation of a personalized method enabling operator-specific control using a motor command mapping model trained on the developer
- Participant-based evaluation of the proposed system through android avatar operation using the developed facial synchronization system

A preliminary version of this study, focusing on the development and evaluation of the developer model, was previously reported [11].

II. Concept of the Facial Synchronization System

We propose a facial synchronization system for immersive avatar operation using an HMD.

A. Problem

Reproducing the operator's intended facial expressions on an avatar requires a model that generates motor commands based on the operator's expressions. HMDs capable of measuring facial movements typically provide only blendshape weights, which represent facial actions. In our immersive android avatar control system, we previously generated facial expressions by linearly mapping each blendshape weight—assumed to correspond to a specific facial action—to the motor's target angles [4]. However, as noted earlier, the mapping between facial parameters and android motor control is not trivial, and camera-based methods proposed in recent studies [9], [10] cannot be directly applied to HMD-based systems. This is because the facial expression data obtained from HMD-integrated sensors and those extracted from camera images differ in both acquisition methods and parameter definitions, making unified representation difficult. As a result, it is challenging to express the operator's facial expressions captured via HMD and the android's expressions using the same parameter set. Therefore, reproducing visually similar expressions requires complex procedures, such as mapping between distinct parameter spaces.

Moreover, since the avatar's face is structurally different from the operator's, it is inherently impossible to reproduce identical expressions. Therefore, a facial synchronization system is needed that allows the operator to feel that avatar's expression genuinely reflects their own.

B. Proposed Method

We propose a facial synchronization system that reproduces the operator's intended expressions by mapping them to the motor commands required to realize them, using data collected as the operator mimics the android's expressions. An overview of the proposed method is shown in Fig. 1.

First, random motor commands are input into the avatar to generate arbitrary facial expressions. The operator then observes the avatar's face reflected in a mirror and mimics the expression. During this process, pairs of the operator's facial parameters and the corresponding motor commands are collected to map the operator's intended expressions to the avatar's resulting expressions. This mapping is used to construct a motor command generation model capable of reproducing the operator's intended facial expressions.

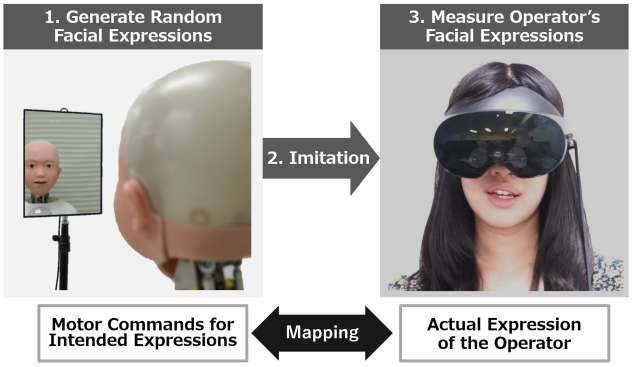


Fig. 1: Proposed method for facial synchronization.

III. Android Avatar

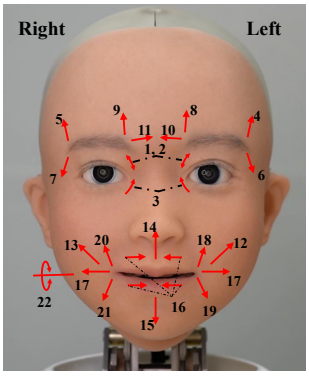
We used the android avatar Yui, developed by our research group [4]. Yui's deformable facial skin enables the execution of 22 facial actions, as shown in Fig. 2, controlled by 15 motors embedded in the head. The end-to-end delay from command transmission to facial actuation was approximately 0.13 s on average across all facial motors.

Around the eyes, Yui is equipped with one motor for yaw rotation of each eyeball and a shared motor for pitch rotation of both eyeballs. This configuration enables independent horizontal movement of each eye and synchronized vertical movement of both eyes. Because the eye motors can be synchronized in real time by directly inputting the angles obtained from the HMD as motor target angles, they were excluded from the model construction in this study. Each of Yui's eyes is equipped with a wide-angle camera, allowing stereoscopic visualization of the avatar's viewpoint to be presented to the operator through the HMD. Although Yui also has three degrees of freedom in the neck, these were fixed in this study to keep the head facing forward.

We used an HMD (Meta Quest Pro, Meta), which outputs facial weights from 0 to 1 for 63 blendshapes [12].

IV. Motor Command Mapping Model

The motor command generation process was defined as follows. First, based on the 63 facial parameters obtained from the HMD, a weight between 0 and 1 was assigned to each of the 22 avatar facial actions defined in Fig. 2. Let the set of assigned facial action weights be denoted as the facial weight vector $f_i, (i = 1, \cdots, 22)$, and the target angles for the avatar's 15 facial motors be denoted as $w_i, (i = 1, \cdots, 15)$. The motor target angles were then calculated as



#	Motor No.	Facial Action	#	Motor No.	Facial Action
1	1+	Upper eyelid open	12	7	Left cheek pull
2	1-	Upper eyelid close	13	8	Right cheek pull
3	2-	Lower eyelid close	14	9	Upper lip up
4	3+	Left outer eyebrow up	15	10	Lower lip down
5	4+	Right outer eyebrow up	16	11+ & 12+	Mouth pull
6	3-	Left outer eyebrow down	17	11- & 12-	Mouth pucker
7	4-	Right outer eyebrow down	18	13+	Left mouth corner up
8	5+	Left inner eyebrow up	19	13-	Left mouth corner down
9	6+	Right inner eyebrow up	20	14+	Right mouth corner up
10	5-	Left inner eyebrow frown	21	14-	Right mouth corner down
11	6-	Right inner eyebrow frown	22	15+	Jaw open

Fig. 2: Facial actions and motors of Yui, where $+$ and $-$ indicate motor forward and reverse rotation.

follows:

$$\begin{bmatrix} w_1 \\ \vdots \\ w_{15} \end{bmatrix} = \boldsymbol{F}_o + \boldsymbol{T} \begin{bmatrix} f_1 \\ \vdots \\ f_{22} \end{bmatrix}. \tag{1}$$

Here, $\boldsymbol{F}_o \in \mathbb{R}^{15\times 1}$ represents a constant baseline vector of motor target angles when all facial weights are zero. $\boldsymbol{T} \in \mathbb{R}^{15\times 22}$ is a weight matrix that maps facial weights to motor commands (-1 to 1). The rows correspond to motors, and the columns correspond to facial actions. Each element is set to 1 if the motor controls the corresponding facial action (or -1 if in the reverse direction), and 0 otherwise. Note that certain combinations of facial actions, such as Upper eyelid close with Eyebrow up, Cheek pull with Mouth pucker, and Cheek pull with Mouth corner down, posed a risk of tearing the avatar's skin. To prevent this, when the target values of Upper eyelid close or Cheek pull exceeded half of their motion range, the interfering actions were disabled.

In Yui's conventional facial synchronization system, the most relevant facial parameters among the 63 obtained from the HMD were directly selected as f_i $(i = 1, \cdots, 22)$ to represent each avatar's facial action. For facial actions linked to multiple parameters, the facial weight was taken as either the maximum or the average, depending on the expression's characteristics. In contrast, this study introduces a machine learning approach to estimate the facial weights f_i $(i = 1, \cdots, 22)$ that best reproduce the operator's intended expressions.

A. Training Data Collection

To collect training data, we asked the operator to mimic the avatar's facial expressions generated in response to randomly selected motor commands. To avoid commanding infeasible facial actions due to hardware constraints, we first operated the avatar using the conventional system across a wide range of expressions and extensively recorded the corresponding facial weights used in motor command generation. We then randomly selected and normalized these recorded facial weights, converted them back into motor commands, and used them to generate facial expressions on the avatar for imitation. To ensure that each motor command was unique, duplicate entries were removed prior to data collection.

The model was specifically designed for the case in which the operator and developer were the same person, namely, the author of this paper. Data collection was performed while the developer wore an HMD. To ensure clear visualization of the avatar's facial expressions during data collection, a camera was installed in front of the avatar, and a mirrored (left–right flipped) feed from this camera was displayed in the HMD. When a random motor command was applied to the avatar, the developer mimicked the resulting facial expression and pressed a handheld button to indicate the completion of the imitation. At the moment of button press, the system recorded both the operator's facial parameters and the facial weights that had been used to generate the avatar's expression. Following each recording, the next motor command was applied to the avatar, and the operator repeated this process until 1000 data pairs were collected.

B. Model Training

The goal was to develop a facial weight generation model (63 inputs, 22 outputs) that converts the operator's facial parameters—obtained via the HMD—into facial weights. When passed through (1), these weights yield motor commands that reproduce the operator's intended facial expressions. We selected a Multilayer Perceptron (MLP), a type of neural network, as the machine learning algorithm for this task. MLPs offer a relatively simple architecture, support fast computation, and can model complex relationships between input features. We implemented the MLP model using the MLPRegressor class from scikit-learn, a Python-based machine learning library. Of the 1000 collected samples, 800 were used for training and 200 randomly selected samples were set aside as the test set. To determine the optimal hyperparameters, we conducted a grid search with five-fold cross-validation across 36 hyperparameter combinations.

- Hidden Layers : 1, 2, 3, 4
- Nodes : 32, 64, 128
- L2 regularization coefficient : 0.01, 0.001, 0.0001

All other hyperparameters were set to the default values of MLPRegressor. Additionally, the model outputs were clipped to stay within the range of 0 to 1.

The 63 facial parameters used as input represent the states of various facial regions. However, the relationship

TABLE I: Hyper-parameters

Configuration	Hidden Layers	Nodes	L2 regularization coefficient
Full-facial	4	128	0.001
Eye-region	4	128	0.01
Mouth-region	4	128	0.01

between these parameters and facial actions is less direct for regions around the eyes and mouth. It was therefore hypothesized that including such redundant inputs might reduce the model's prediction accuracy. To investigate this, the dataset was divided into two subsets: one containing parameters related to eye-region movements (brows and eyelids), and the other corresponding to mouth-region movements (cheeks, lips, mouth corners, and jaw). Two separate configurations were trained:

- Eye-region configuration: (20 inputs, 11 outputs) predicts facial weights associated with eye-related expressions (e.g., brow and eyelid movements).
- Mouth-region configuration: (43 inputs, 11 outputs) predicts facial weights associated with mouth-related expressions (e.g., cheeks, lips, mouth corners, and jaw movements).

These configurations were compared with the full-facial configuration, which used all 63 facial parameters to predict all 22 facial weights. The hyperparameters for the eye-region and mouth-region configurations were selected using the same five-fold cross-validated grid search across 36 hyperparameter combinations, consistent with the full-facial configuration. The specific hyperparameter settings used for each configuration are summarized in Table I.

C. Accuracy Evaluation

The accuracy of the model was evaluated using the Mean Absolute Error (MAE) between the predicted facial weights and the corresponding values in the test dataset. Fig. 3(a)–(c) presents the MAE for the conventional model and the proposed configurations, evaluated across all facial actions, eye-region actions, and mouth-region actions. For the prediction of all facial actions (Fig. 3(a)), the full-facial configuration showed improved accuracy compared to the conventional model. Similarly, for eye-region action prediction (Fig. 3(b)), the full-facial configuration outperformed both the conventional model and the eye-region configuration. In contrast, for mouth-region action prediction (Fig. 3(c)), the mouth-region configuration achieved the highest accuracy.

Fig. 4 compares the prediction errors for individual facial actions between the full-facial and conventional models. For all 11 eye-region facial actions, the full-facial configuration achieved higher prediction accuracy than the conventional model. Similarly, for the 11 mouth-region facial actions, the full-facial configuration outperformed the conventional model in all cases except for Right mouth corner down.

Furthermore, Fig. 5(a)–(d) illustrates an example of the operator's facial expression, the corresponding target expression, and the expressions generated by both the full-facial configuration and the conventional model. The expression generated by the full-facial configuration (Fig. 5(c)) more closely resembles the target expression originally mimicked by the operator (Fig. 5(b)) compared to the conventional model (Fig. 5(d)), indicating better reproduction of the intended facial expression.

D. Discussion

The accuracy evaluation results confirmed that the proposed model achieved higher accuracy in predicting static facial expressions than the conventional model. A comparison of prediction errors for individual facial actions indicated that the improvement was not limited to specific expressions but resulted from enhanced prediction performance across nearly all facial actions.

For eye-region actions prediction, the full-facial configuration outperformed the eye-region configuration, which was trained exclusively on eye-related parameters. This suggests that incorporating correlations among facial actions across the entire face can improve the model's ability to reproduce the operator's intended expressions. In contrast, for mouth-region actions, the mouth-region configuration slightly outperformed the full-facial configuration, implying that expressions involving the mouth may rely more on localized features. These findings indicate that while full-facial input enhances overall synchronization, region-specific models may better capture subtle localized expressions. Nevertheless, the full-facial configuration consistently achieved high prediction accuracy across all facial actions. Compared to region-specific configurations, it likely captures latent correlations among various facial movements more effectively, enabling more robust and generalizable facial synchronization. Therefore, considering both system simplicity and ease of implementation, the full-facial configuration represents a practical and effective choice for facial synchronization.

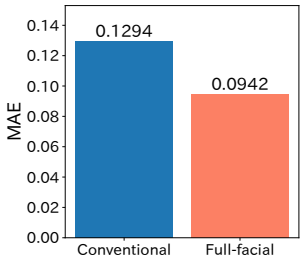
Furthermore, a visual comparison of the generated facial expressions confirmed that the full-facial configuration produced expressions that closely resembled the target expressions. In addition, when comparing the operator's expression (Fig. 5(a)) with the target expression (Fig. 5(b)), specifically in the mouth region, a noticeable difference in mouth shape was observed. This finding suggests a possible discrepancy between the facial expression an individual believes they are making and the one they actually display.

V. Personalization for Each Operator

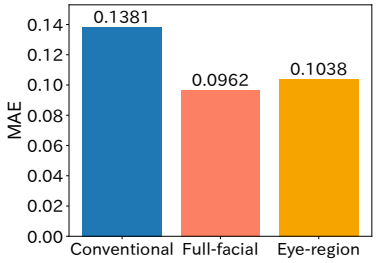
Next, we developed a personalization system that adapts the motor command mapping model, originally trained on developer data, for different operators.

A. Proposed Method

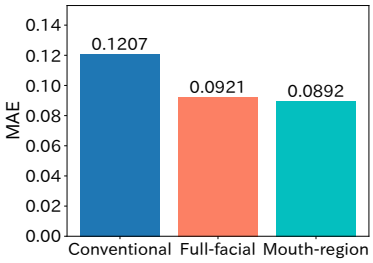
In this study, we propose a personalization approach that applies a transformation matrix to the output of the developer-trained model, converting it into facial weights that reflect the operator's intended avatar expression based on their own facial parameters. First, the operator wears the HMD and mimics a series of diverse avatar expressions, during which their facial parameters are recorded as calibration



(a) Full-facial actions.



(b) Eye-region actions.



(c) Mouth-region actions.

Fig. 3: Mean absolute error comparison by configurations.

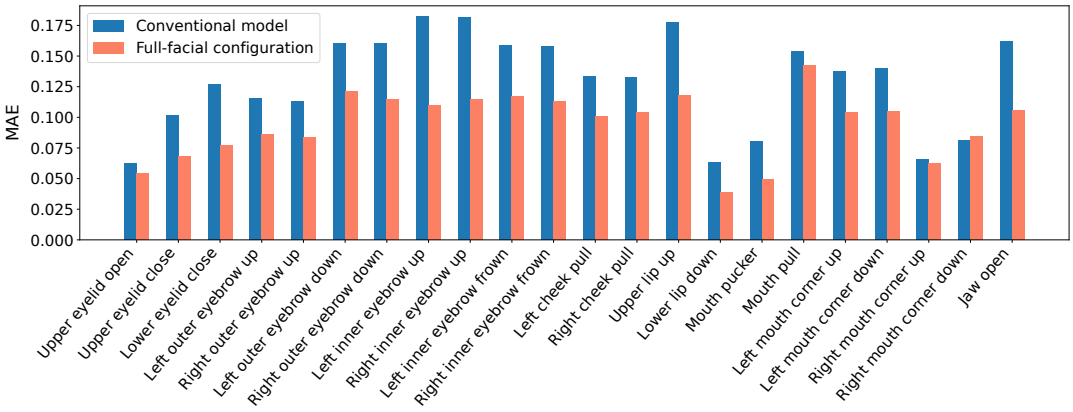


Fig. 4: Mean absolute error comparison by avatar facial actions.

(a) Operator. (b) Ground truth. (c) Full-facial. (d) Conventional.

Fig. 5: Actual avatar expressions generated by each model.

data. Next, the recorded facial parameters are input into the developer's motor command mapping model to obtain the corresponding facial weight outputs. A transformation matrix is then calculated to convert these output facial weights into the target facial weights corresponding to the avatar expressions that the operator had previously mimicked. Applying this transformation matrix to the model's output in real time allows the system to generate motor commands that more accurately reproduce the operator's intended expressions.

The transformation matrix used to adjust the facial weights is computed as follows. Let the facial weights used to generate k distinct avatar expressions be represented as

$$\boldsymbol{a}_1, \boldsymbol{a}_2, \boldsymbol{a}_3, \ldots, \boldsymbol{a}_k \quad \left(\boldsymbol{a}_i \in \mathbb{R}^{22\times 1}\right),$$

and let the corresponding output vectors—obtained by inputting the operator's facial parameters into the developer-trained model—be

$$\boldsymbol{b}_1, \boldsymbol{b}_2, \boldsymbol{b}_3, \ldots, \boldsymbol{b}_k \quad \left(\boldsymbol{b}_i \in \mathbb{R}^{22\times 1}\right).$$

We then define the matrices $\boldsymbol{A}$ and $\boldsymbol{B}$ as follows:

$$\boldsymbol{A} = \begin{bmatrix} \boldsymbol{a}_1^{\mathrm{T}} \\ \vdots \\ \boldsymbol{a}_k^{\mathrm{T}} \end{bmatrix} \left(\boldsymbol{A} \in \mathbb{R}^{k\times 22}\right), \tag{2}$$

$$\boldsymbol{B} = \begin{bmatrix} \boldsymbol{b}_1^{\mathrm{T}} \\ \vdots \\ \boldsymbol{b}_k^{\mathrm{T}} \end{bmatrix} \left(\boldsymbol{B} \in \mathbb{R}^{k\times 22}\right). \tag{3}$$

Let the transformation matrix for converting the facial weights be denoted as $\boldsymbol{M}$. Then, the following relation holds:

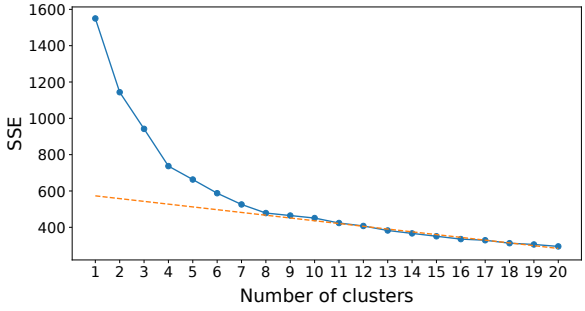


Fig. 6: SSE by number of clusters. The dotted line represents the approximate linear trend for the number of clusters ranging from 10 to 20.

$$\boldsymbol{A} = \boldsymbol{B} \cdot \boldsymbol{M} \quad \left(\boldsymbol{M} \in \mathbb{R}^{22\times 22}\right), \tag{4}$$

$$\boldsymbol{M} = \boldsymbol{B}^{\dagger} \cdot \boldsymbol{A} \quad \left(\boldsymbol{B}^{\dagger}\text{: pseudoinverse of } \boldsymbol{B}\right). \tag{5}$$

From this, the transformation matrix $\boldsymbol{M}$ can be derived. The matrix $\boldsymbol{M}$, as defined in (4), serves to convert the 'avatar expression intended by the developer' into the 'avatar expression intended by the operator' for a given set of facial parameters.

B. Selection of Target Expressions for Imitation

To construct a transformation matrix that can generalize to a wide range of operators using only a small amount of imitation data, it is important to select distinctive expressions from those that the avatar is capable of producing. To this end, we performed clustering on the 1000 facial weight vectors used as imitation targets during the model training process described in Section IV-A. The centers of the resulting clusters were then input into the model, and the corresponding outputs were used as the avatar expressions to be imitated by the operator.

We used k-means clustering for this purpose. The number of clusters was determined using the elbow method. We computed the sum of squared errors (SSE) for cluster numbers ranging from 1 to 20; the results are shown in Fig. 6. Since the SSE curve began to flatten at around 10 clusters, we fitted a linear approximation to the SSE values between 10 and 20, shown as a dotted line in the figure. Based on the shape of the curve, we determined the number of clusters to be 8. We applied k-means clustering (k = 8) and visualized the result in 3D using t-SNE [13], as shown in Fig. 7.

The eight avatar expressions generated from the facial weights at each cluster center are shown in Fig. 8. The operator then imitated each of these expressions, and the corresponding facial parameters were collected as calibration data.

VI. Experiment

To examine whether the developed system can accurately reproduce the operator's intended facial expressions, we conducted a subjective evaluation in which the operator assessed the expressions generated by the system.

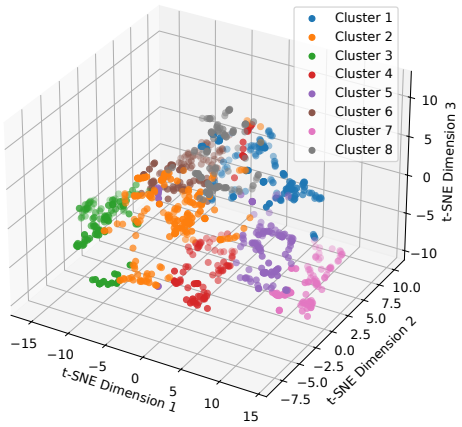


Fig. 7: t-SNE visualization of clustering results.

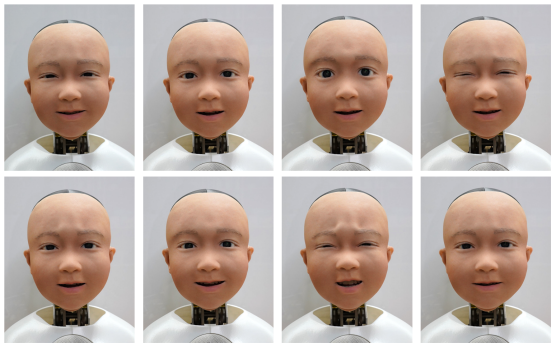


Fig. 8: Avatar expressions generated by the facial weights of each cluster center.

A. Method

The following sections describe the participants, the materials used, and the experimental procedure.

1) Participants: A total of 18 students (10 males and 8 females; aged 18–28 years) from the University of Electro-Communications participated in the experiment. Each participant received a cash voucher worth 1000 JPY as a token of appreciation.

2) Materials: The experimental setup is shown in Fig. 9. To prevent participants from directly viewing the avatar, it was positioned in a separate location behind partitions. A web camera was positioned in front of the avatar and connected to the operator's PC. During the experiment, participants sat while wearing an HMD and headphones and pressed a button placed on the table at the instructed timing.

As a subjective measure of how well the avatar's facial expression reflected the operator's intent, a 7-point Likert scale questionnaire was used. Participants were asked the following question: "Did the android avatar reproduce the facial expression you intended?" Responses were recorded on a scale from 1 (Not at all) to 7 (Very much so).

3) Procedure: The experimental design, task content, and procedure are described below.

a) Design: A within-subjects design was used, in which each participant experienced all three model conditions.

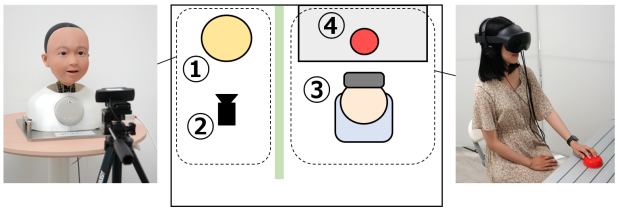


Fig. 9: Layout and photos of the experimental environment: 1) android avatar Yui; 2) web camera; 3) operator; 4) button. The right photo shows a reproduced participant scene, demonstrated by the author.

- **Conventional Model**: A baseline model that directly uses selected facial parameters as facial weights without any transformation.
- **Developer Model**: A motor command mapping model trained using the developer's data, corresponding to the full-facial configuration described in Section IV.
- **Personalized Model**: A model that applies a transformation matrix to the developer model's output, computed from each operator's calibration data.

b) Task: In this experiment, one of the models was used to synchronize the avatar's facial expressions with the participant's in real time, while a mirrored image of the avatar was displayed in the HMD. When the participant pressed the button, the avatar began synchronization using one of the three models. Participants were instructed to freely vary their facial expressions for 30 seconds while observing the avatar's mirrored face. To prevent auditory distractions, white noise was played continuously through headphones during the task. After 30 seconds, the avatar stopped synchronizing, and participants were instructed to remove the HMD and headphones. They were then asked to complete a questionnaire. Each participant repeated this procedure three times, using a different model each time, thereby completing the task under all model conditions. There were six possible orders in which participants could perform the task across the three conditions, and participants were evenly distributed across these orders to ensure counterbalancing.

c) Experimental Steps: First, calibration data for the personalized model were collected. Participants were instructed to wear the HMD and headphones, mimic the avatar's expressions displayed on the screen, and press the button repeatedly. After completing this task, a transformation matrix for personalization was generated.

Next, the main task was conducted. Before beginning the task, participants were informed that they would operate the avatar by freely changing their facial expressions for 30 seconds, observe the avatar during this time, and then complete a questionnaire assessing how well the avatar's expressions matched their intended expressions. Participants completed the questionnaire after operating the avatar with each model. During calibration and task phases, facial parameters, personalization matrices, facial weights, motor commands, and avatar video were collected.

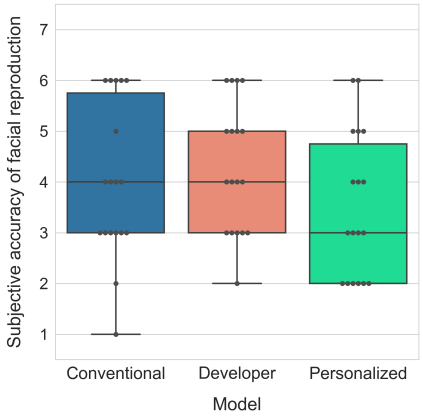


Fig. 10: Subjective evaluation of facial reproduction accuracy across models.

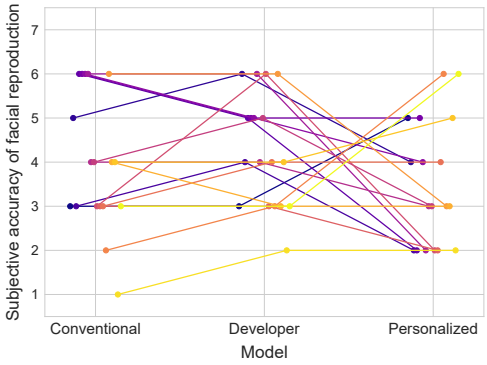


Fig. 11: Individual ratings for each model connected by participants.

4) Ethical Considerations: The research protocol used in this study was approved by the Ethics Committee of the University of Electro-Communications (No. H25003). All participants provided written informed consent prior to the start of the experiment.

B. Results

The results of participants' subjective evaluations of expression synchrony are shown in Fig. 10. The median score was the same for the Conventional Model ($Mdn = 4$) and the Developer Model ($Mdn = 4$), while the Personalized Model yielded the lowest median score ($Mdn = 3$). A Friedman test revealed no significant differences among the three conditions ($\chi^2(2) = 4.33$, $p = 0.147$). Fig. 11 shows the distribution of expression reproducibility ratings across participants for each model.

VII. Discussion

The Developer Model received ratings comparable to the Conventional Model, despite lacking personalization. This suggests that the proposed approach, which constructs correspondences through facial imitation, may help generate expressions aligned with the operator's intent.

Although the difference was not statistically significant, the Personalized Model received the lowest overall ratings.

Nine of the 18 participants rated the Personalized Model noticeably lower than the other two models. Participant comments included statements such as, "The avatar's facial movements were exaggerated and did not reflect the expressions I was making," and "Even subtle movements, like slight mouth openings, were overly amplified." These remarks indicate a disconnect between the participant's intended expressions and those generated by the Personalized Model. One possible explanation is that the amount of calibration data used for personalization was relatively small compared to the training data used for the Developer Model. This imbalance may have caused overfitting in the transformation matrix, resulting in exaggerated or unintended facial expressions.

To examine whether calibration imitation accuracy affected evaluations, we analyzed all participants' imitation performance. We focused on eyelid and mouth movements, which were often mentioned in participants' comments. Specifically, we analyzed two of the 22 facial actions: Upper eyelid close and Jaw open.

For each participant, we compared the facial weights for these two actions in the target avatar expressions with those generated by the Conventional Model using their imitation facial parameters. We then assessed whether the absolute difference between the target and predicted values exceeded 0.5, which corresponds to the midpoint of the facial weight range (0 to 1). The analysis showed that all participants, except one, had at least one instance in which the absolute difference exceeded the threshold. No clear relationship was found between the number of large deviations and participants' ratings for any specific model. These findings suggest that even facial actions with visibly distinct movements were not always accurately imitated. This implies that full-face imitation may have been cognitively or physically demanding for participants. A more refined calibration process—such as staging imitation by facial region (e.g., mouth and eyes)—may help improve precision and reduce participant burden.

On the other hand, four out of 18 participants rated the personalized model noticeably higher than the other two models. In their free-form comments, some participants mentioned that "the facial movements felt smoother, which made the expression seem more accurately reproduced," and "the expression was best reproduced, although the mouth movement appeared slightly exaggerated." Regardless of whether the feedback was positive or negative, many comments on the personalized model focused on the extent of facial movement. These findings suggest that participants' evaluation of whether the expression matched their intention was influenced not only by the static resemblance of facial shapes but also by dynamic aspects such as the magnitude and fluidity of facial changes. This implies that the perceived alignment between an avatar's expression and the operator's intent may vary according to individual subjective values and interpretation.

Based on these results, the proposed method—enabling operator-specific adaptation through a linear transformation of facial weights using limited imitation data—can be considered effective, although certain limitations remain. In particular, refining the calibration process and introducing evaluation metrics that capture the dynamic characteristics of expression reproduction are expected to further improve the system's accuracy.

VIII. Conclusion

In this study, we developed a facial synchronization system for an android avatar that reflects the operator's intended expressions during immersive operation using an HMD. By having a developer imitate randomly generated avatar expressions, we established a mapping between facial parameters and motor commands, enabling high-accuracy facial synchronization. We also proposed a personalization method that uses a small amount of imitation data to apply a linear transformation to the model output, allowing the system to adapt to individual operators. Although the personalized model presented certain challenges, experimental results suggest that it offers a promising direction for generating operator-specific expressions. Future work will aim to improve the calibration system, with the goal of reproducing the operator's intended expressions more accurately using a limited amount of data.

References

[1] K. Takeuchi, Y. Yamazaki, and K. Yoshifuji, "Avatar work: Telework for disabled people unable to go outside by using avatar robots," in *Companion of the 2020 ACM/IEEE International Conference on Human-Robot Interaction*, 2020, pp. 53–60.

[2] L. Neumann, S. Skoubo, and F. Wamsler, "(Tele) present in the classroom: exploring the international use of telepresence robots for inclusive learning environments," *Learning Environments Research*, pp. 1–22, 2025.

[3] B. Kang, I. Hwang, J. Lee, S. Lee, T. Lee, Y. Chang, and M. K. Lee, "My being to your place, your being to my place: Co-present robotic avatars create illusion of living together," in *Proceedings of the 16th Annual International Conference on Mobile Systems, Applications, and Services*, 2018, pp. 54–67.

[4] M. Nakajima, K. Shinkawa, and Y. Nakata, "Development of the lifelike head unit for a humanoid cybernetic avatar 'Yui' and its operation interface," *IEEE Access*, vol. 12, pp. 23 930–23 942, 2024.

[5] E. Arts, "Ameca," https://www.engineeredarts.co.uk/robot/ameca/, retrieved on July 25, 2025.

[6] D. F. Glas, T. Minato, C. T. Ishi, T. Kawahara, and H. Ishiguro, "Erica: The erato intelligent conversational android," in *2016 25th IEEE International symposium on robot and human interactive communication (RO-MAN)*. IEEE, 2016, pp. 22–29.

[7] L. D. Spano, "Teleoperating humanoids robots using standard VR headsets: A systematic review." *International Conference on Computer-Human Interaction Research and Applications*, 2021.

[8] K. Shinkawa, M. Nakajima, and Y. Nakata, "Vision-sharing system for android avatars to enable remote eye contact," *ROBOMECH Journal*, vol. 11, no. 1, p. 16, 2024.

[9] Y. Hu, B. Chen, J. Lin, Y. Wang, Y. Wang, C. Mehlman, and H. Lipson, "Human-robot facial coexpression," *Science Robotics*, vol. 9, 2024.

[10] B. Wu, C. Liu, C. T. Ishi, T. Minato, and H. Ishiguro, "Retargeting human facial expression to human-like robotic face through neural network surrogate-based optimization," in *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2024, pp. 4724–4730.

[11] K. Shinkawa, M. Nakajima, and Y. Nakata, "Consideration of a facial synchronization system for android avatars," in *Proceedings of the 42nd Annual Conference of the Robotics Society of Japan (RSJ)*, 2024, pp. 1–4, in Japanese.

[12] Meta Horizon OS Developers, "Face Tracking for Movement SDK for Unity," https://developers.meta.com/horizon/documentation/unity/move-face-tracking/, retrieved on July 25, 2025.

[13] L. Van der Maaten and G. Hinton, "Visualizing data using t-SNE." *Journal of machine learning research*, vol. 9, no. 11, 2008.