Optimizing Facial Expressions of an Android Robot Effectively: a Bayesian Optimization Approach

Dongsheng Yang[1,2], Wataru Sato[2,3], Qianying Liu[1], Takashi Minato[5], Shushi Namba[2], Shin'ya Nishida[1,4]

***Abstract*— Expressing various facial emotions is an important social ability for efficient communication between humans. A key challenge in human-robot interaction research is providing androids with the ability to make various human-like facial expressions for efficient communication with humans. The android Nikola, we have developed, is equipped with many actuators for facial muscle control. While this enables Nikola to simulate various human expressions, it also complicates identification of the optimal parameters for producing desired expressions. Here, we propose a novel method that automatically optimizes the facial expressions of our android. We use a machine vision algorithm to evaluate the magnitudes of seven basic emotions, and employ the Bayesian Optimization algorithm to identify the parameters that produce the most convincing facial expressions. Evaluations by naïve human participants demonstrate that our method improves the rated strength of the android's facial expressions of anger, disgust, sadness, and surprise compared with the previous method that relied on Ekman's theory and parameter adjustments by a human expert.**

I. Introduction

Robot facial expressions (RFEs) have attracted attention in the field of human-robot interactions because facial expressions are essential for human emotional communication in everyday life. Leite et al. [1] suggested that robot expressions can substantially influence human-robot interactions because they attract attention from humans and generate long-term memories.

Giving many action parameters comparable to human facial movements is a reasonable hardware strategy for enabling androids to express rich human-like emotions. A significant number of androids have been developed to reproduce human-like facial expressions (see [2]–[6] and Table I). The android Nikola (Fig. 1), we have developed, is one of the world's leading androids in this respect. Through a series of human experiments, Sato et al. [7] verified that Nikola can produce several basic human-like RFEs.

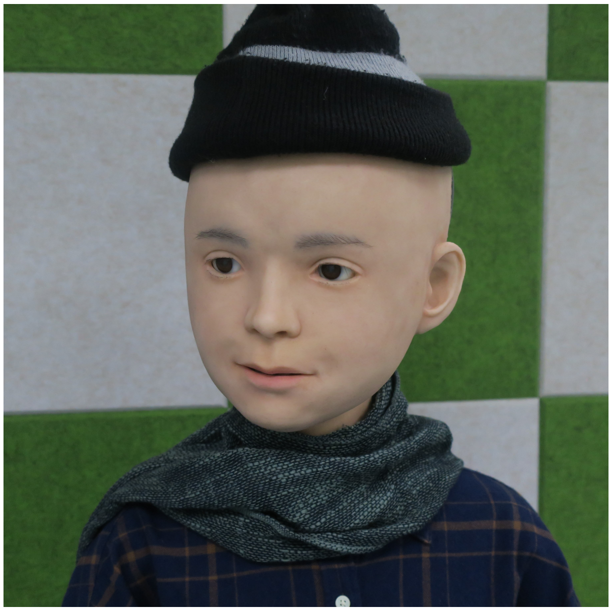

Fig. 1. Nikola, an android developed by RIKEN.

Nikola is an android with the face of a human child; it was designed for use in studies of interactions between humans and robots, with a focus on emotional communications. Soft and deformable silicone skin is attached to the front of Nikola's skull. Beneath the skin, there are 35 pneumatic actuators with pressure control valves: 29 for facial muscle movement control, 3 for head movement control (roll, pitch, and yaw rotation), and 3 for eyeball control (i.e., panning of the individual eyeballs and tilting of both eyeballs) [7].

However, when an android has many controllable parameters, it becomes difficult to identify the optimal parameter set for a particular goal. Recent studies have enabled robots to generate facial expressions, either through knowledge-based human coding or in an automated manner [8]. Sato et al. [7] used the knowledge-based human coding approach. They identified basic axes for each emotion based on Ekman's theory [9], and then asked a human expert to adjust the parameters of each axis. This approach generated convincing positive expressions but did not yield robust negative expressions.

Here, we attempted to improve expressions (i.e., generate emotional expressions that are more convincing to human observers) by controlling more parameters; however, this approach is both challenging and time-consuming for human operators. Therefore, instead of using a human coding approach, we used an automated approach.

Automatic RFE optimization is challenging in two respects. First, RFE performance must be evaluated both reliably and automatically. Because Nikola's face is reasonably human-like, we used Py-Feat [17], an open-source human emotion recognition model, for automatic evaluation

[1]Graduate School of Informatics, Kyoto University, Kyoto, Japan yang.dongsheng.46w@st.kyoto-u.ac.jp, [2]Psychological Process Research Team, Guardian Robot Project, RIKEN, Kyoto, Japan, [3]Field Science Education and Research Center, Kyoto University, Kyoto, Japan, [4]NTT Communication Science Laboratories, Nippon Telegraph and Telephone Corporation, Atsugi, Japan and [5]Interactive Robot Research Team, Guardian Robot Project, RIKEN, Kyoto, Japan

This work is supported by JSP.

TABLE I

COMPARISON OF NIKOLA WITH OTHER FACIALLY EXPRESSIVE HUMANOID ROBOTS.

Android*	Head DOF**	Head Size	Emotional expression capability	Validation
Nikola (this study)	35	Child	7 emotions	Emotion recognition
Face robot [2]	24	Adult	6 basic emotions	Emotion recognition (no statistical test)
ROMAN [3]	21	Adult	6 basic emotions	Emotion recognition
Saya [4]	24	Adult	6 basic emotions	Emotion recognition (no statistical test)
Albert HUBO [5]	31	Adult	Full range (not specified)	-
Face Robot [6]	39	Adult	6 basic emotions	-
Geminoid F [10]	12	Adult	5 basic emotions	Emotion recognition
FACE [11]	32	Adult	6 basic emotions	Emotion recognition (no statistical test)
Face robot [12]	7	Child	6 basic emotions	Emotion recognition (no statistical test)
EveR [13]	16	Adult	6 basic emotions	-
XIN-REN [14]	7	Adult	-	-
EVA [15]	25	Adult	6 basic emotions	-
EVA 2.0 [16]	25	Adult	-	-

* List only includes androids with human-like appearance

** DOF = degrees of freedom

of RFEs. Second, the complexity of parameter optimization increases exponentially with increasing degrees of freedom (DOF). Nikola has 35 actuators on its face. Thus, Nikola has a greater number of DOF that require optimization than most other androids (Table I).

In a related study concerning the automatic generation of RFEs, Park [18] proposed using a dynamic emotion model (based on the linear affect-expression space model) to generate RFEs. However, because this method does not consider the nonlinear feature of most physical robots, it is difficult to use them. WE-4R [19] used emotion vectors for facial expression generation. Ren et al. [14] proposed a kinematics-based learning method for robot XIN-REN to imitate expression of a human subject. Hyung et al. [13] used a genetic algorithm to identify optimal facial expressions in the search space. Similarly, Habib [20] proposed a learning-based method with a genetic algorithm capable of inverse nonlinear mapping from human faces to robot actuators. However, a recent study suggested that the increasing complexity of robots (i.e., increased number of DOF) is leading to difficulty in the application of genetic algorithm methods [16]. Rawal [21] proposed a deep generative method for generating emotions, in which grid search and gradient descent are combined to automatically identify the parameters of desired RFEs. Importantly, this learning-based method is potentially data-intensive.

We consider this problem to be equivalent to an expensive black-box optimization issue [22] for an entire system. Therefore, we propose the use of Bayesian Optimization [23] to identify a set of optimal parameters that enable Nikola to generate higher-quality facial expressions. We call this method Bayesian optimization-based robot facial expression optimization (**BORFEO**). In summary, our contributions are threefold:

1) We develop a method, BORFEO, which automatically identifies a set of optimal facial parameters that allow Nikola to generate prototypical facial expressions without initialization.
2) BORFEO converges within a relatively small number of trials, which suggests that BORFEO is an efficient RFE optimization procedure.
3) BORFEO can generate RFEs that receive higher human ratings for most basic emotions compared with previous work involving the same robot [7]; this finding indicates that BORFEO is a high-performance method.

The remainder of this paper is structured as follows. Section II describes the preliminary setup. Section III describes the BORFEO methodology. Section IV describes the experiments conducted to generate Nikola's RFEs and evaluate their performance. Section V presents the experimental results and discussion. Section VI presents the conclusions and offers suggestions for future work.

II. PRELIMINARY SETUP

A. Prototypical RFEs in Nikola

Nikola's actuator arrangement design is based on the facial action coding system (FACS) proposed by Ekman [9]. A certified FACS coder conducted the action unit mapping from actuators to action units, as defined by Ekman [9]; this mapping is also the basis of human-coded prototypical RFEs. The detailed correspondence of FACS coding is shown in Table II. Further details regarding the differences between Nikola and other androids are described in our previous work [7].

B. Emotion recognition

Py-Feat [17] is a Python-based open-source facial expression analysis toolbox developed by Cheong and colleagues at Dartmouth College; it includes state-of-the-art facial expression models. Py-Feat can be used for tasks such as face detection, facial landmark detection and alignment, face or head pose estimation, action unit detection, and emotion detection.

For emotion detection, Py-Feat provides four models: a random forest model trained on the histogram of oriented gradients extracted from ExpW, CK+, and JAFFE datasets; a support vector machine model trained on the histogram of oriented gradients extracted from the same dataset; a deep convolution network; and ResMaskNet (a residual masking network for recognition of facial expressions) [24]. We chose the ResMaskNet model because its performance was superior to the other three models in the emotion recognition task using the FER2013 dataset.

C. Experimental setup

Figure 2 shows the experimental setup used for data collection and assessment of our proposed algorithms. We used a Logicool C922 Pro stream web camera (HD1080P) to capture RGB images. For robot control and camera manipulation, we employed a notebook computer (Intel Core i7-10870H CPU, 2.20 GHz, NVIDIA GeForce RTX 3060 laptop GPU, GDDR6 6GB) running the Ubuntu 20.04 system. Nikola uses ROS (Robot Operating System) Noetic Ninjemys as its base system; this software was released on May 23, 2020 and is compatible with Ubuntu 20.04. We connected a web camera to the laptop through a USB cable and connected Nikola to the laptop via Wi-Fi. The position of the web camera was fixed throughout the experiment.

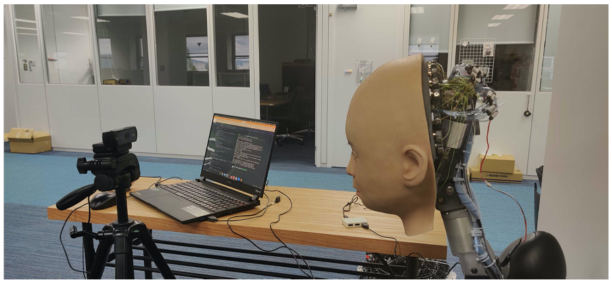

Fig. 2. Video recording setup.

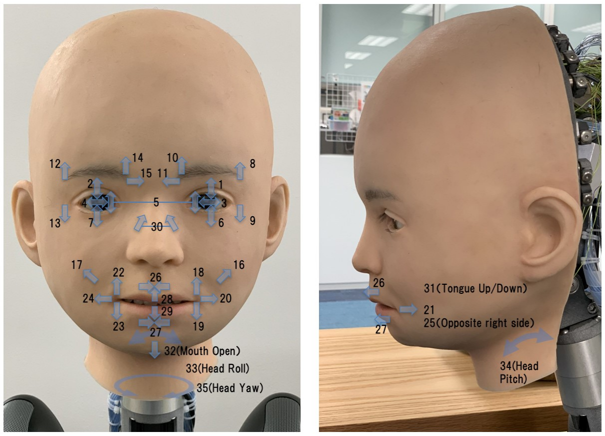

Fig. 3. Actuator arrangement in Nikola.

III. Methodology

Nikola's action unit is controlled by a pneumatic actuator with control inputs that range from 0 to 255 (Fig. 3). To optimize Nikola's facial expressions, we transformed the task into a black-box function optimization problem and proposed the BORFEO method. Because Py-Feat can evaluate a robot's facial expression and generate probability scores for seven basic emotions (anger, disgust, fear, happiness, sadness, surprise, and neutral), we regarded the parameters of the control axes as the input of the black-box function; the Py-Feat probability scores were the black-box output. We sought to identify a set of values for the control axes parameters allowing Nikola's facial expressions to achieve high confidence scores on Py-Feat.

TABLE II
Actuator details.

Actuator	Description	Side
1	Upper lid raiser	L
2	Upper lid raiser	R
6	Cheek raiser/Lid tightener	L
7	Cheek raiser/Lid tightener	R
8	Outer brow raiser	L
9	Cheek raiser	L
10	Inner brow raiser	L
11	Brow lowerer	L
12	Outer brow raiser	R
13	Cheek raiser	R
14	Inner brow raiser	R
15	Brow lowerer	R
16	Cheek puller	L
17	Cheek puller	R
18	Lip corner puller	L
19	Lip corner depressor	L
20	Lip stretcher	L
22	Lip corner puller	R
23	Lip corner depressor	R
24	Lip stretcher	R
28	Lip funneler	T
29	Lip funneler	B
30	Nose wrinkler	-
32	Jaw dropper	-

* L denotes left side, R denotes right side, T denotes top, B denotes bottom, and - denotes not applicable.

There are various methods to address the black-box optimization problem. Grid search and random sampling are used for small-scale systems but are not appropriate for this task because of the very large parameter space. Compared with other algorithmic-based methods such as the genetic algorithm, Bayesian Optimization achieves rapid convergence, and its performance has proven superior to other methods for tasks such as the 2020 Black-Box Optimization Challenge [25], hyperparameter optimization [26], and nutrition problems [27]. Thus, we use Bayesian Optimization to solve the black-box function optimization problem.

In this section, we introduce the task formulation and then describe the use of Bayesian optimization to identify facial expressions.

A. Task formulation

Formally, given $n = 35$ parameters of control axes of the robot $x = (a_0, ..., a_n)$, which satisfies $0 \leq a_i \leq 255, a_i \in \mathbb{Z}$, we define the parameter space as $\mathcal{X}$. Each set of parameters x can generate a corresponding facial expression, which is assigned a score y by Py-Feat. We define the process of obtaining the Py-Feat score y from a set of control axes parameters x as $y = f(x)$. The goal is to identify an optimal set of x that satisfies the following condition:

$$x^* = argmax_{x \in \mathbb{X}} f(x) \tag{1}$$

$f(x)$ is an expensive black-box function, for which no analytical description or gradient information exists concerning the objective function.

B. Bayesian Optimization

The main objective of the algorithm is to update a Bayesian statistical model of the black-box function and use an acquisition function to determine the next search point. We first generate an initial set of candidate solutions and then find the next possible optimal data point based on these solutions points, add that point to the candidate solutions set, and repeat this step until the end of the iteration. Finally, the optimal data point in the set of candidate solutions is taken as the solution to the problem.

However, $f(x)$ is a black-box function, thus the key issue here is how to determine the next search point based on the already searched points. As we show in Algorithm 1, the Bayesian optimization algorithm uses Gaussian Process (GP) $\mathcal{M}$ to model the black-box function. For each $x \in \mathbb{X}$ we can obtain the mean μ_x and variance δ_x. Given (μ, δ) pairs, we can evaluate each point with the *Acquisition Function* $\mathcal{S}$ to obtain the next search point.

Algorithm 1: Bayesian Optimization.

```
Input: $f, \mathbb{X}, \mathcal{M}, \mathcal{S}$
Output: $x^*$
initialize $(f, \mathbb{X}) \mapsto \mathcal{D}$;
for $\mathcal{D} \mapsto i$ to $T$ do
    FITMODEL($\mathcal{M}, \mathcal{D}$) $\mapsto p(y|x, \mathcal{D})$;
    $argmax_{x \in \mathbb{X} S(x, p(y|x, \mathcal{D}))}$;
    $f(x_i) \mapsto y_i$;
    $\mathcal{D} \cup (x_i, y_i) \mapsto \mathcal{D}$;
end
```

Where $\mathcal{D}$ is defined as a dataset that consists of several pairs of data, each represented as (x_i, y_i); x_i is a set of hyperparameters, and y_i represents the result corresponding to the set of hyperparameters.

1) Gaussian Process: Consider a sequence of continuous random variables $\{x_i\}$, if the vector formed by any subsequence $[x_{t_0}, ..., x_{t_k}]^\intercal$ obeys a multidimensional normal distribution, this sequence of random variables is defined as a GP.

Notably, our variables x are discrete, while GP requires continuous random variables. We ignore this discontinuity here because the parameters a_i of x are continuous integers that can be regarded as approximately continuous.

Specifically, consider k random variables $x_1, ..., x_k$ that follow a k-dimensional Gaussian distribution:

$$N(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$$

where the mean vector satisfies $\boldsymbol{\mu}_k \in \mathbb{R}^k$ and the covariance matrix satisfies $\boldsymbol{\Sigma}_k \in \mathbb{R}^{k \times k}$.

Because the integral of the normal distribution yields an analytic solution, the marginal and conditional probabilities can easily be obtained. Thus, given a set of control axis parameter values $\boldsymbol{x}_i, i = 1, ..., l$ and the corresponding set of scores y, we can model the black-box function $y = f(x)$ from these samples. Given an input x, we can use the function to predict the label value y and posterior probability $p(y|\boldsymbol{x})$. This process is referred to as *Gaussian Process Regression* (GPR).

GPR can use a set of sampled solution values to fit a Bayesian model and produce the probability distribution of the solution values. Given the black-box function $f(x)$ that satisfies the following mapping:

$$\mathbb{R}^n \to \mathbb{R}$$

GPR will update the model based on $x_i, i = 1, ..., t$ and the corresponding solution $f(x_i)$ to fit the black-box function. Indeed, the model describes the probability distribution of $f(x)$.

2) Acquisition Function: The acquisition function is an inexpensive function that can be evaluated at a given point that guides how the parameter space should be explored for the optimization problem. The acquisition function can then be optimized to select the data points for the next observation. This approach replaces the original optimization problem with another function $a(x)$, which can be optimized more easily.

In our study, we use the Upper Confidence Bound (UCB) acquisition function, which is defined as:

$$a_{UCB}(x; \beta) = \mu(x) - \beta\delta(x) \tag{2}$$

The terms $\mu(x)$ and $\delta(x)$ can be interpreted as explicitly encoding a trade-off between exploitation (evaluating at points with low mean) and exploration (evaluating at points with high uncertainty). $\beta > 0$ is a trade-off hyperparameter.

IV. Experiment

A. Actuator Selection

With some prior knowledge, we restricted the movements of actuators in optimization.

1) **Safety**. Because of Nikola's mechanical limitations, several pairs of fragile points required careful adjustment to random values. For example, actuators s_{18} and s_{19} constitute a pair of fragile points because they will pull the same part in opposite directions.
2) **Psychological knowledge**. Although some psychological evidence suggests that gaze/head direction is involved in emotional expression [28], the gaze/head variant is not implemented in most prototypical facial expression databases or Py-Feat. Therefore, we discarded this factor.
3) **Symmetric**. Nikola's actuator arrangement is symmetrical on the left and right. Considering that the seven basic facial expressions are symmetrical [29], we

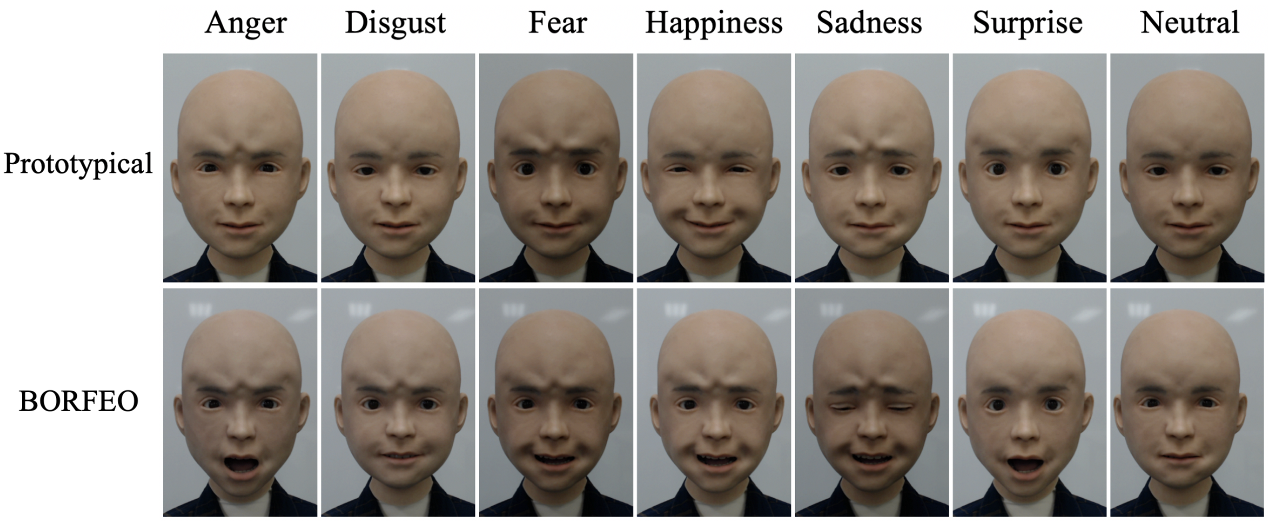

Fig. 4. Illustrations of prototypical and BORFEO-generated RFEs produced by the android Nikola.

simplify the problem and assign the same value to all symmetrical pairs of actuators. $a_i = a_j$, which satisfies $0 \leq i, j \leq 35$, where i and j are center-symmetrical actuators.

Considering these points, we selected 24 actuators with a total dimension equal to 14, as shown in Table II.

B. RFE generation

The BORFEO system was written in Python 3.7.8. We used FFmpeg to control the web camera and Bayesian Optimization library [30] for implementing BORFEO. ROS serves as the bridge system to send instructions to Nikola [31]. To evaluate our method, we conducted a 100-round BORFEO and generated 100 RFEs for each of the seven basic emotions. The number of rounds was determined by the preliminary experiment. Since we didn't optimize the processing time, each round of BORFEO took about 20 seconds including 10 seconds for image capturing by a webcam and 5 seconds for Py-Feat processing. Since we carried out experiments online, each session took about 35 minutes. 4 shows the top-scoring BORFEO-generated expressions, together with the human-coded (mouth closed) prototypical expressions [7].

Figure 5 shows the machine rating results. Clear improvement in the Py-Feat rating was observed for all basic emotions (except neutral), which implies that BORFEO can identify a set of RFEs that receive higher Py-Feat ratings compared with prototypical RFEs.

C. Human evaluation

To evaluate differences in human observers' ratings between human-coded prototype RFEs and optimized RFEs, we recruited 40 naïve Japanese participants (20 men, 20 women; mean ± standard deviation age = 33.7 ± 5.1 years) in an online survey experiment.

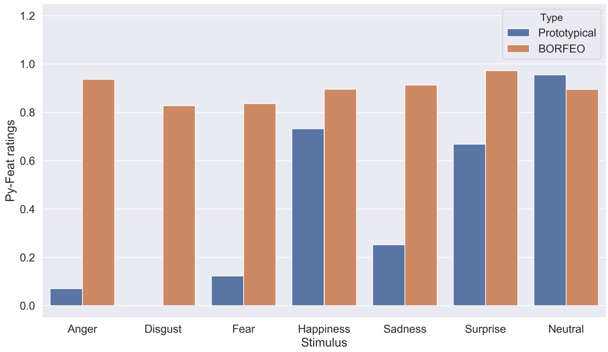

Fig. 5. Py-Feat rating results.

To facilitate correspondence with Py-Feat ratings, we designed a more comprehensive survey compared with most previous studies (see [2]–[4], [10]–[12]). Participants were given stimulus photos of prototypical and BORFEO-generated expressions for seven basic emotions; they were asked to rate the extent to which each photo depicted one of the seven basic emotions using a 7-point scale ranging from 1 (does not depict that emotion) to 7 (strongly depicts that emotion). The prototypical expressions were generated using the method described in [7], based on Ekman's theory and parameter adjustments by a human expert. To increase stimulus variability, we added mouth-open conditions to the original mouth-closed conditions. We selected the five best (but dissimilar) BORFEO-generated expressions for each emotion. The orders of stimulus photos and rated emotions were randomized. The human evaluation experiments were conducted via Qualtrics (an online questionnaire system) using a procedure approved by the ethical committee of RIKEN.

V. Results and Discussion

A. Target emotion recognition

Figure 6 shows the mean target emotion ratings with 95% confidence intervals for the seven basic emotions and two

TABLE III

RESULT OF TWO-WAY REPEATED MEASURES ANALYSIS OF VARIANCE.

Source[a]	SS[b]	NDF[c]	DDF[d]	Mean squares	F-value	p-unc[e]	η_p^2 [f]
RFE	0.763050	1	39	0.763	12.52	<0.0001	0.2430
Emotion	40.695060	6	234	6.78	41.35	<0.0001	0.5146
RFE * Emotion	3.147407	6	234	0.524	8.18	<0.0001	0.1734

[a] "Source" column shows groups of RFEs (3 RFEs per group).

[b] SS = sums of squares

[c] NDF = degrees of freedom (numerator)

[d] DDF = degrees of freedom (denominator)

[e] p-unc: uncorrected p-value

[f] η_p^2: partial eta-square effect size

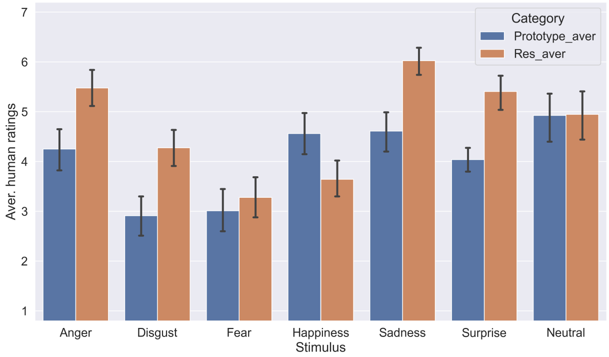

Fig. 6. Main results of the human evaluation experiment.

types of RFEs. The target emotion ratings were analyzed using two-way repeated measures analysis of variance with RFE (prototypical and BORFEO) and emotion (anger, disgust, fear, happiness, sadness, surprise, and neutral) as independent variables (Table III). The results showed significant main effects of RFE and emotion, as well as a significant interaction between these factors.

Follow-up analyses for the simple main effects of RFE revealed that, compared with prototypical expressions, BORFEO-generated expressions had significantly higher target emotion recognition scores for angry, disgusted, sad, and surprised expressions ($p < 0.001$). In contrast, prototypical happy expressions had significantly higher target emotion ratings than BORFEO-generated happy expressions ($p < 0.001$).

The results in Figure 6 do not exclude the possibility that the BORFEO method increases human ratings for both target (e.g., anger) and non-target expressions (e.g., disgust). If such an effect is present, we may be unable to determine whether the BORFEO method produces a more convincing target expression. To address this concern, we also analyzed the normalized target emotion rating, i.e., the target emotion rating divided by the sum of all emotion ratings for the same face image (Fig. 7). We found that the effects of BORFEO were very similar when the analysis focused on raw (Fig. 6) and normalized ratings (Fig. 7), thus eliminating the above concern.

Our results indicate that, compared with prototypical RFEs, the BORFEO method improved emotion rating for the RFEs of anger, disgust, sadness, and surprise. No significant effect of BORFEO was observed for fearful and neutral expressions. Regarding happiness, emotion rating was higher for prototypical than BORFEO-generated expressions.

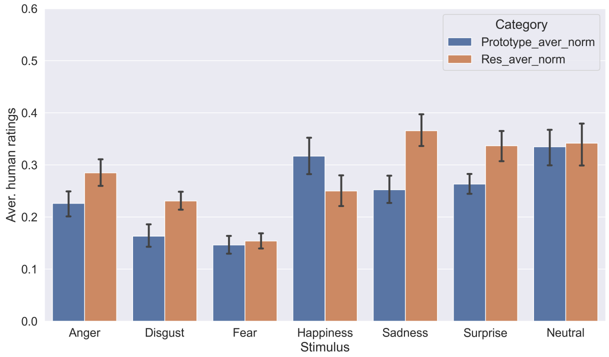

Fig. 7. Normalized ratings in the human evaluation experiment.

B. Correlation between Py-Feat ratings and human ratings

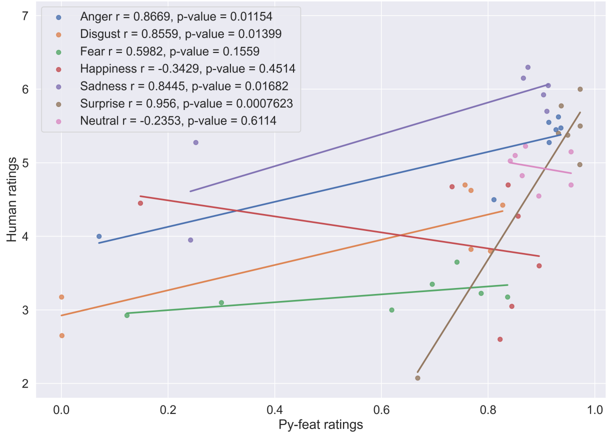

Fig. 8. Scatter plots and regression lines for different types of stimuli.

Figure 8 shows the relationships (i.e., scatterplots and regression lines) between Py-Feat and human ratings for the stimulus photos of each target emotion. The human ratings indicated that Py-Feat successfully predicts human ratings for angry, sad, disgusted, and surprised expressions. However, Py-Feat did not successfully predict fearful, happy, or neutral expressions. Visual inspection of Figure 8 suggests a possible reason for the problems with these three expressions: positive correlations between Py-Feat and human ratings were observed for angry, disgusted, sad, and surprised expressions, while negative correlations were observed for fearful, happy, and neutral expressions.

Based on these results, we speculate that the poor performance of BORFEO concerning fearful, happy, and neutral expressions is related to disagreement between Py-Feat and human recognition of Nikola's expressions. Py-Feat (ResMaskNet) is a machine learning algorithm trained on human ratings of human, rather than android, expressions. In addition, the results may have been influenced by cultural differences in emotion recognition [10] between the original raters and our participants. Prediction of how our participants would recognize Nikola's facial emotions is an out-of-domain task for Py-Feat. We may be able to improve BORFEO performance by adjusting Py-Feat via domain adaptation [32] or other transfer learning strategies.

Humans often exhibit asymmetrical facial expressions [33]. Although Nikola can produce asymmetrical expressions, we only considered symmetrical RFEs to simplify our analysis. Nikola may be able to produce more convincing expressions for human observers if we expand the parameter search range to include complex asymmetrical expressions.

VI. Conclusion

In this work, we proposed and implemented the BORFEO method for an automatic RFE optimization task. The BORFEO method can be easily applied to any facial robot because it only requires a web camera, laptop, and means of transmitting robot control commands from the laptop; it does not require human experts with appropriate knowledge of facial expressions. We obtained a set of optimal control parameters for Nikola's RFEs. The results of our human evaluation experiment showed that the BORFEO method improves Nikola's RFEs for anger, disgust, sadness, and surprise compared with the RFEs generated by our previous method based on human coding [7]. We also explored why the BORFEO method could not improve some expressions and offered suggestions to overcome this problem in future work.

VII. Acknowledgement

This work was supported in part by JST, the establishment of university fellowships towards the creation of science technology innovation, Grant Number JPMJFS2123, and the JSPS Grants-in-Aid for Scientific Research (KAKENHI), Grant Number JP20H05957.

References

[1] I. Leite, C. Martinho, and A. Paiva, "Social robots for long-term interaction: a survey," *International Journal of Social Robotics*, vol. 5, no. 2, pp. 291–308, 2013. pages

[2] H. Kobayashi, F. Hara, G. Uchida, and M. Ohno, "Study on face robot for active human interface—mechanisms of face robot and facial expressions of 6 basic emotions—," *Journal of the robotics society of Japan*, vol. 12, no. 1, pp. 155–163, 1994. pages

[3] K. Berns and J. Hirth, "Control of facial expressions of the humanoid robot head roman," in *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2006, pp. 3119–3124. pages

[4] T. Hashimoto, S. Hitramatsu, T. Tsuji, and H. Kobayashi, "Development of the face robot saya for rich facial expressions," in *2006 SICE-ICASE International Joint Conference*. IEEE, 2006, pp. 5423–5428. pages

[5] J.-H. Oh, D. Hanson, W.-S. Kim, Y. Han, J.-Y. Kim, and I.-W. Park, "Design of android type humanoid robot albert hubo," in *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2006, pp. 1428–1433. pages

[6] T. Hashimoto, S. Hiramatsu, and H. Kobayashi, "Dynamic display of facial expressions on the face robot made by using a life mask," in *Humanoids 2008-8th IEEE-RAS International Conference on Humanoid Robots*. IEEE, 2008, pp. 521–526. pages

[7] W. Sato, S. Namba, D. Yang, S. Nishida, C. Ishi, and T. Minato, "An android for emotional interaction: Spatiotemporal validation of its facial expressions," *Frontiers in Psychology*, vol. 12, 2022. [Online]. Available: https://www.frontiersin.org/article/10.3389/fpsyg.2021.800657 pages

[8] N. Rawal and R. M. Stock-Homburg, "Facial emotion expressions in human–robot interaction: a survey," *International Journal of Social Robotics*, pp. 1–22, 2022. pages

[9] J. C. H. Paul Ekman, Wallace V Friesen, *Facial action coding system*. Salt Lake City, UT : Research Nexus eBook, 2002. pages

[10] C. Becker-Asano and H. Ishiguro, "Intercultural differences in decoding facial expressions of the android robot geminoid f," *Journal of Artificial Intelligence and Soft Computing Research*, vol. 1, no. 3, pp. 215–231, 2011. pages

[11] D. Mazzei, N. Lazzeri, D. Hanson, and D. De Rossi, "Hefes: An hybrid engine for facial expressions synthesis to control human-like androids and avatars," in *2012 4th IEEE RAS & EMBS International Conference on biomedical robotics and biomechatronics (BioRob)*. IEEE, 2012, pp. 195–200. pages

[12] C.-Y. Lin, C.-C. Huang, and L.-C. Cheng, "An expressional simplified mechanism in anthropomorphic face robot design," *Robotica*, vol. 34, no. 3, pp. 652–670, 2016. pages

[13] H.-J. Hyung, H. U. Yoon, D. Choi, D.-Y. Lee, and D.-W. Lee, "Optimizing android facial expressions using genetic algorithms," *Applied Sciences*, vol. 9, no. 16, p. 3379, 2019. pages

[14] F. Ren and Z. Huang, "Automatic facial expression learning method based on humanoid robot xin-ren," *IEEE Transactions on Human-Machine Systems*, vol. 46, no. 6, pp. 810–821, 2016. pages

[15] Z. Faraj, M. Selamet, C. Morales, P. Torres, M. Hossain, B. Chen, and H. Lipson, "Facially expressive humanoid robotic face," *HardwareX*, vol. 9, p. e00117, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2468067220300262 pages

[16] B. Chen, Y. Hu, L. Li, S. Cummings, and H. Lipson, "Smile like you mean it: Driving animatronic robotic face with learned models," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 2739–2746. pages

[17] J. H. Cheong, T. Xie, S. Byrne, and L. J. Chang, "Py-feat: Python facial expression analysis toolbox," *arXiv preprint arXiv:2104.03509*, 2021. pages

[18] J. W. Park, H. S. Lee, and M. J. Chung, "Generation of realistic robot facial expressions for human robot interaction," *Journal of Intelligent & Robotic Systems*, vol. 78, no. 3, pp. 443–462, 2015. pages

[19] K. Itoh, H. Miwa, M. Zecca, H. Takanobu, S. Roccella, M. C. Carrozza, P. Dario, and A. Takanishi, "Mechanical design of emotion expression humanoid robot we-4rii," in *Romansy 16*. Springer, 2006, pp. 255–262. pages

[20] A. Habib, S. K. Das, I.-C. Bogdan, D. Hanson, and D. O. Popa, "Learning human-like facial expressions for android phillip k. dick," in *2014 IEEE International Conference on Automation Science and Engineering (CASE)*. IEEE, 2014, pp. 1159–1165. pages

[21] N. Rawal, D. Koert, C. Turan, K. Kersting, J. Peters, and R. Stock-Homburg, "Exgennet: Learning to generate robotic facial expression using facial expression recognition," *Frontiers in Robotics and AI*, vol. 8, 2022. [Online]. Available: https://www.frontiersin.org/article/10.3389/frobt.2021.730317 pages

[22] D. R. Jones, M. Schonlau, and W. J. Welch, "Efficient global optimization of expensive black-box functions," *Journal of Global optimization*, vol. 13, no. 4, pp. 455–492, 1998. pages

[23] M. Pelikan, D. E. Goldberg, E. Cantú-Paz, *et al.*, "Boa: The bayesian optimization algorithm," in *Proceedings of the genetic and evolution-*

ary computation conference GECCO-99, vol. 1. Citeseer, 1999, pp. 525–532. pages

[24] L. Pham, T. H. Vu, and T. A. Tran, “Facial expression recognition using residual masking network,” in *2020 25th International Conference on Pattern Recognition (ICPR)*. IEEE, 2021, pp. 4513–4519. pages

[25] R. Turner, D. Eriksson, M. McCourt, J. Kiili, E. Laaksonen, Z. Xu, and I. Guyon, “Bayesian optimization is superior to random search for machine learning hyperparameter tuning: Analysis of the black-box optimization challenge 2020,” in *NeurIPS 2020 Competition and Demonstration Track*. PMLR, 2021, pp. 3–26. pages

[26] H. Alibrahim and S. A. Ludwig, “Hyperparameter optimization: comparing genetic algorithm against grid search and bayesian optimization,” in *2021 IEEE Congress on Evolutionary Computation (CEC)*. IEEE, 2021, pp. 1551–1559. pages

[27] S. Gumustekin, T. Senel, and M. A. Cengiz, “A comparative study on bayesian optimization algorithm for nutrition problem,” *Journal of Food and Nutrition Research*, vol. 2, no. 12, pp. 952–958, 2014. pages

[28] O. Semyonov, A. Ziv-El, E. G. Krumhuber, S. Karasik, and H. Aviezer, “Beyond shared signals: The role of downward gaze in the stereotypical representation of sad facial expressions.” *Emotion*, vol. 21, no. 2, p. 247, 2021. pages

[29] P. Ekman and W. V. Friesen, “Facial action coding system,” *Environmental Psychology & Nonverbal Behavior*, 1978. pages

[30] F. Nogueira, “Bayesian Optimization: Open source constrained global optimization tool for Python,” 2014–. [Online]. Available: https://github.com/fmfn/BayesianOptimization pages

[31] S. A. I. L. et al., “Robotic operating system (ros melodic morenia),” 2018. [Online]. Available: https://www.ros.org pages

[32] H. Daumé III, “Frustratingly easy domain adaptation,” in *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*. Prague, Czech Republic: Association for Computational Linguistics, June 2007, pp. 256–263. [Online]. Available: https://aclanthology.org/P07-1033 pages

[33] A. Lindell, “Lateralization of the expression of facial emotion in humans,” *Progress in brain research*, vol. 238, pp. 249–270, 2018. pages