VOCALISTMIRROR: A SINGER SUPPORT INTERFACE FOR AVOIDING UNDESIRABLE FACIAL EXPRESSIONS

Kin Wah Edward Lin, Tomoyasu Nakano, Masataka Goto
National Institute of Advanced Industrial Science and Technology (AIST)
Central 2, 1-1-1, Umezone, Tsukuba, Ibaraki 305-8568, Japan
{edward.lin, t.nakano, m.goto}@aist.go.jp

ABSTRACT

We present *VocalistMirror*, an interactive user interface that enables a singer to avoid their undesirable facial expressions in singing video recordings. Since singers usually focus on singing expressions and do not care about facial expressions, when watching singing videos they recorded, they sometimes notice that some of their facial expressions are undesirable. VocalistMirror allows a singer to first specify their undesirable facial expressions in a recorded video, and then sing again while seeing a real-time warning that is shown when the facial expression of the singer becomes similar to one of the specified undesirable expressions. It also displays Karaoke-style lyrics with piano-roll melody and visualizes acoustic features of singing voices. iOS ARKit framework is used to quantify the facial expression as a 52-dimensional vector, which is then used to compute the distance from undesirable expressions. Our experimental results showed the potential of the proposed interface.

1. INTRODUCTION

Although many tools to enable singers to achieve desirable singing expressions of their singing voices have been developed [1–4], to the best of our knowledge, no tools have been developed to enable singers to avoid undesirable facial expressions while singing. For example, Lin et al. [5] developed a singing pitch training interface that visualizes singing pitch (fundamental frequency of singing voice) in real time and gives feedback on the correctness of the pitch while singing. Tsuzuki et al. [6] developed an interface to create derivative choruses by mixing (mashing up) various singing voices sung by different singers for the same song. Such a variety of singing voices are typically available as vocal covers on video sharing services. Ojima et al. [7] developed a real-time vocal-part arrangement system that enables a user to manipulate the vocal part of existing music audio signals. Fragments of singing voices can be manipulated and played back by using a MIDI keyboard. These interfaces and systems focused on singing voices and did not deal with facial expressions in singing.

The importance of facial expressions has already been investigated in the literature. Quinto et al. [8] stated that facial expressions of a singer are known to influence the perception of singing expressiveness, tension, timbre, dissonance, note duration, interval size, phrase structure, and emotion. They also reported the relationship between the singing emotion and the pre-production, production, and post-production of facial expressions. Lyons and Tetsutani [9] and Koh and Yadegari [10] demonstrated how facial expressions can be used to manipulate audio signals. Goto et al. [11] showed how singing and facial expressions of a singer can be imitated by a humanoid robot. However, there are no studies on applications helping singers have desirable facial expressions while singing.

The popularity of recording short singing video clips and uploading them to social media has increased. This can be considered a new form of music interaction and this trend is the most observable among young people. Its popularity is evident in the large market size of singing apps (i.e., smartphone/tablet applications for singing) that enable users to create singing video recordings. For example, TikTok [1], Smule [2], and Yokee [3] are popular and provide functions to add digital makeup or decorate faces and backgrounds. Those makeup functions can be manually used by singers at the post-production stage, but singing apps are not able to detect undesirable facial expressions of singers.

We therefore propose a singer support interface called *VocalistMirror* that enables a user to record a short singing video clip with desirable facial expressions, which could be uploaded to social media by the singer. With the VocalistMirror interface, the user can sing an excerpt of a song while listening to its karaoke track and seeing automatically-scrolling Karaoke-style lyrics with a piano-roll melody line. Acoustic features of the user's singing voice such as the sung pitch (fundamental frequency), and timing are analyzed and visualized in real time. This visualized feedback helps the user be aware of the accuracy of the sung pitch and timing, and thus helps users improve their singing expressions. Furthermore, the user's singing voice and facial expressions are automatically recorded as a video clip and played back after singing. During the

[1] TikTok by Bytedance https://itunes.apple.com/us/app/id835599320 accessed on: 15 Feb 2019.
[2] Smule - The #1 Singing App by Smule https://itunes.apple.com/us/app/id509993510 accessed on: 15 Feb 2019.
[3] Karaoke - Sing Unlimited Songs by Yokee Music https://itunes.apple.com/us/app/id547109049 accessed on: 15 Feb 2019.

playback of this singing video clip, VocalistMirror allows the user to select several frames of *undesirable* facial expressions that the user sometimes wears. Since the user typically does not notice those undesirable facial expressions while singing, the user wants to avoid them. VocalistMirror solves this problem by letting the user sing again while automatically analyzing the user's facial expressions in real time and displaying a warning when one of the selected undesirable facial expressions is detected. VocalistMirror thus acts as a *mirror* enabling the user to notice when the user wears an undesirable facial expression and avoid it in a recorded singing video clip.

Our main contributions are three-fold: (1) we opened up a new way of assisting singers from the viewpoint of facial expressions, (2) we designed and implemented an interface helping singers specify and avoid their undesirable facial expressions in a simple intuitive way, and (3) we evaluated the effectiveness and potential of the interface by conducting a user study.

2. SYSTEM DESIGN AND IMPLEMENTATION

We want our VocalistMirror to be a tool that is capable of creating a short singing video clip with a duration similar to that of the 15 to 30 seconds video clips posted on social media such as TikTok, so that it could encourage novice-level singers to record their own singing video clips. We also want VocalistMirror to be easily accessible and easy-to-use. In the following three subsections, we discuss what platform our VocalistMirror should be deployed on, how facial expressions are quantified, and what acoustic features should be visualized.

2.1 Deployment Platform

Among available popular platforms such as Windows, macOS, Android, and iOS, we decided to develop and deploy our interface on the iOS platform. It is portable and accessible given the size of mobile devices such as iPhone and iPad. It also provides strong software and hardware support for analyzing facial expressions and acoustic features. This choice means we can use Apple's TrueDepth camera system [12, 13] that is available on the iOS platform and expect the audio quality to be high enough [4].

2.2 Quantification of Facial Expressions

The iOS platform has a software framework called ARKit [5] that can quantify facial expressions. It uses a front-facing TrueDepth camera on the iOS device to provide real-time analysis of singer's facial expressions. ARKit quantifies each facial expression as a facial wireframe with 1,220 vertices. It can further analyze those vertices to provide 52 distinct facial shapes [6], which are classified into 5 categories: (1) Left Eye, (2) Right Eye, (3) Mouth and Jaw, (4) Eyebrows, Cheeks, and Nose, and (5) Tongue. Since each of the 52 facial shapes has a value ranging from 0 to 1, we can regard a set of values of the 52 facial shapes as a 52-dimensional *facial vector* that represents the facial expression in a video frame. We quantify how similar the facial expressions in two different video frames are by calculating the distance between the two vectors of those frames.

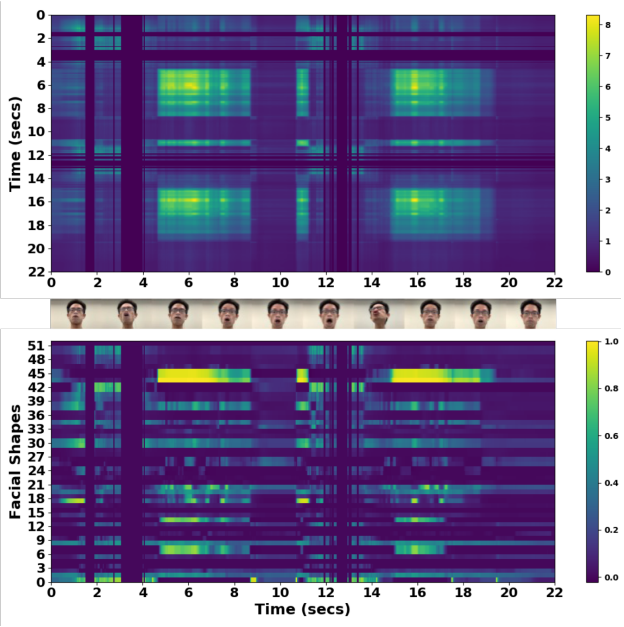

Figure 1. Feasibility study using ARKit framework to quantify facial expressions.

To illustrate the feasibility of using these 52-dimensional facial vectors to represent and detect similar facial expressions, we first asked a singer to sing a 22-second song that repeats two similar musical phrases. We also asked the singer to try to express the same facial expression for each of the musical phrases. We used the ARKit framework to capture a set of facial vectors during this singing performance. Then we calculated a self-similarity matrix of this set of vectors. Fig. 1 shows the self-similarity matrix of these facial vectors on the top, snapshots of singer's facial expressions in the middle, and the corresponding facial vectors at the bottom. This figure clearly shows that similar facial expressions of similar phrases were repeated twice. We can therefore use the 52-dimensional facial vectors obtained from the ARKit framework to detect similar facial expressions.

The similarity between facial expressions is quantified by calculating the exponential moving average of the L1-norm distance. The smaller the L1-norm distance is, the more similar they are. The real-time warning for undesirable facial expressions is displayed only when the L1-norm distance is below a threshold. We leave other distance calculations and the corresponding threshold setting for future work.

[4] Mobile Audio Quality Index from JUCE `https://juce.com/maq` accessed on: 15 Feb 2019.

[5] ARKit `https://developer.apple.com/documentation/arkit` accessed on: 15 Feb 2019.

[6] ARKit Face Blendshape by Apple `https://developer.apple.com/documentation/arkit/arfaceanchor/blendshapelocation` accessed on: 15 Feb 2019.

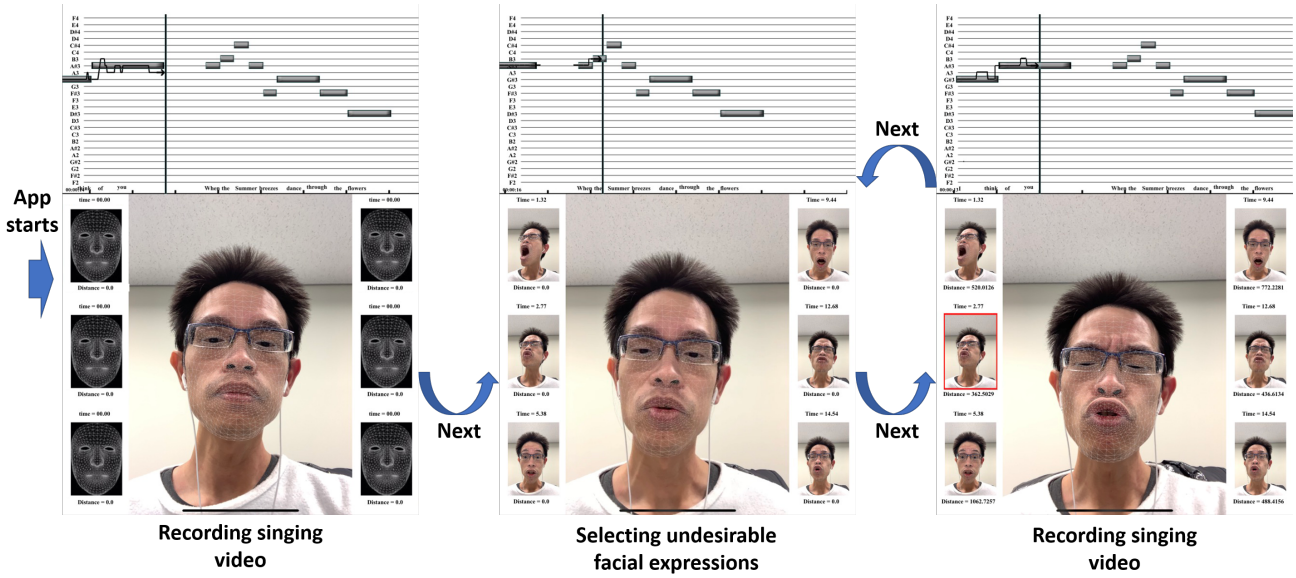

Figure 2. Interface and workflow of VocalistMirror.

2.3 Visualization of Acoustic Features

To sing better, singers have to intentionally control several audio features of their singing voices, including but not limited to, the singing pitch (fundamental frequency), timing, vibrato, tremolo, and breathing. Among these acoustic features, we chose the singing pitch, timing, and vibrato for the visualization, and we leave the others for future work. To visualize the singing pitch and vibrato in the right timing in real time, we need a fast and accurate fundamental frequency estimator. A third-party iOS AudioKit framework [7] not only provides such an estimator [14], but also seamlessly cooperates with the iOS platform that we use for the ARKit framework. Since AudioKit also provides various audio effects and analysis tools, VocalistMirror could use them to extend its functions in the future. For example, we could use the facial expressions to call the AudioKit API to create the Wah-Wah audio effect on the singing voice. We could also use AudioKit analysis tools to visualize the singing formants so that the phonetic quality of the singing voice could be visually examined.

3. INTERFACE DESIGN

With the above design principles in mind, we implemented and deployed the VocalistMirror application on an 11-inch iPad Pro 2018 with iOS 12.1.4. Since Lin [15] showed that many singers prefer a larger screen for singing apps, we chose iPad, rather than iPhone for our implementation. In this section, we first describe the workflow of VocalistMirror and then discuss its interface details.

3.1 Workflow of VocalistMirror

Fig. 2 shows an overview of our proposed interface. We call the lower part of the interface a *facial-expression interface* and the upper part of the interface an *acoustic-feature interface*.

When the VocalistMirror app starts, it is at the stage of singing video recording, which is shown in the left of Fig. 2. A singer is expected to sing a short excerpt of a specified song while looking at the acoustic-feature interface as a singing guide. While the singer is singing and listening to the corresponding karaoke accompaniment, the acoustic-feature interface uses the AudioKit framework to provide real-time visual feedback on the singing pitch (fundamental frequency), timing, and vibrato. The singing voice and its corresponding facial expressions are recorded separately into two different files, one for the audio track and the other for the video track. After the singer finishes recording, VocalistMirror is moved to the next stage where the singer can play back and watch the recorded video clip.

At the second stage of selecting undesirable facial expressions, which is shown in the center of Fig. 2, the acoustic-feature interface uses the AudioKit framework to trace the fundamental frequency of the recorded singing voice in the audio track, so that the singer is able to see the pitch, timing and vibrato of the recorded singing voice. Moreover, while the facial-expression interface plays back the recorded video track, the singer is expected to select the singer's undesirable facial expressions by tapping the iPad screen. Due to the limitation of the screen size, the singer can select only six or fewer undesirable facial expressions. The snapshot of each selected facial expression is displayed along with its occurrence time in the video clip. Once the singer finishes selecting undesirable facial expressions, VocalistMirror is moved back to the stage of singing video recording, which is shown in the right of Fig. 2.

The singer at this time is expected to record a new and more satisfying singing video clip by avoiding the selected undesirable facial expressions. The facial-expression interface helps the singer do so by displays a real-time warning when the singer's current facial expression in singing

[7] AudioKit `https://audiokit.io/` accessed on: 15 Feb 2019.

becomes very similar to one of the selected undesirable expressions. This warning is represented by adding a red rectangular border to the snapshot of the most similar expression.

These two stages keep alternating until the singer selects another song, quits the app, or most likely, is satisfied with a recorded short singing video clip without undesirable facial expressions

3.2 Facial-Expression Interface Design

The facial-expression interface is used to record facial expressions of the singing performance and display the similarity between the singer's current facial expression and each of the selected undesirable facial expressions. To make sure that the facial expression is well captured by the ARVideoKit framework [8], a thin facial wireframe is overlaid on the singer's face whenever the singer faces the TrueDepth camera. Therefore, when the thin facial wireframe disappears, the disappearance alerts the singer to face the TrueDepth camera.

3.3 Acoustic-Feature Interface Design

Since the acoustic-feature interface is mainly used as the singing guide, the singer is likely to concentrate on this interface while singing. And since the TrueDepth camera is located at the middle top of the iPad screen, we position this interface at the upper part of the screen so that the singer is more likely to face the camera. The display of the piano-roll melody line as well as the karaoke-style lyrics is implemented by using the Songle web service [16] for the melody line and the TextAlive web service [17] for the lyrics timing. We use SpriteKit [9] to implement the graph drawing. The singing pitch, timing, and vibrato are visualized using the arrow on the timeline cursor shown in Fig. 2.

4. EXPERIMENT

In this section, we first describe the setup of our subjective experiment. We then report and discuss experimental results. This preliminary experiment illustrates the effectiveness and potential of the proposed interface.

4.1 Experiment Setup

Each participant used VocalistMirror alone in a separated and quiet room to create their recording of an excerpt (the first two musical phrases) from *RWC-MDB-P-2001 No.87* in the RWC Music Database (Popular Music) [18]. The duration of the excerpt is 22 seconds. Since each participant was also told to use earphones to listen to the karaoke accompaniment of this song, accompaniment sounds were not recorded by the microphone the participant used and we could record solo singing without any accompaniment. The experiment followed the steps described below and on average took half an hour for each participant. After the experiment, it was optional for the participant to provide further feedback and suggestions.

4.1.1 Pre-experiment questionnaires:

Each participant first filled out an online form that asked about their age (like 20's, 30's, or 40's), gender, musical background, and experience creating their own singing video clips.

4.1.2 Introduction and demonstration:

We then used about 10 minutes to explain and demonstrate the intended usage of VocalistMirror. During this tutorial, we also clarified whatever questions they raised so that they would be comfortable using VocalistMirror alone in a room to create a singing video clip.

4.1.3 Singing video recording:

Since we asked the participants to use VocalistMirror until they had finished a satisfying recording, each participant knew that there was no time limit on recording a video clip.

4.1.4 Post-experiment questionnaires:

Once they finished their recording, they filled out another online form alone in the room. The form was used to evaluate VocalistMirror on a 7-point scale of degree of appreciation, with 7 being the most and 1 being the least. They were told that they should answer the following questions with their first impressions.

- How much do you like the exterior design of the app? (i.e., The graphics and the user interface.)
- How much do you like the features of avoiding undesirable facial expressions?
- How much do you like the features of visualizing acoustic features of your singing voice?
- How much do you think the app has helped you improve your facial expressions after multiple uses?
- How much do you like the app as a whole?

4.2 Experimental Results

Eight participants (four male and four female) participated in this experiment. Their ages ranged from the 20's to the 40's. Five of them had basic western music education and had several years of musical instrument (e.g., piano) practices during their teenage period. These participants are considered to have had no serious music background. The other three had more advanced music education and had serious musical instrument practice (including singing) for at least six years. These participants are considered to have had a serious music background. Only one participant, who was the youngest, has used Instagram [10] to create a short singing video clip before. We conclude that these participants were suitable for evaluating our proposed interface because their genders were evenly distributed, they

[8] ARVideoKit `https://github.com/AFathi/ARVideoKit` accessed on: 15 Feb 2019.

[9] SpriteKit `https://developer.apple.com/documentation/spritekit` accessed on: 15 Feb 2019.

[10] Instagram `https://itunes.apple.com/us/app/id389801252` accessed on: 15 Feb 2019.

How much do you like	Least 1	2	3	4	5	6	Most 7	Median	Mode
Exterior Design		2	1	2	3			4	5
Features of Avoiding		1	1	1	2	2	1	5	5,6
Features of Visualizing				4		3	1	5	4
Improving Expressions		2		2	1	2	1	4.5	2,4,6
Overall Impression			1	1	5		1	5	5

Table 1. Evaluation of the proposed interface by all eight participants. Each number represents the number of participants who selected the corresponding item.

How much do you like	Least 1	2	3	4	5	6	Most 7	Median	Mode
Exterior Design		2	1					2	2
Features of Avoiding		1	1		1			3	2,3,5
Features of Visualizing				3				4	4
Improving Expressions		2		1				2	2
Overall Impression			1	1	1			4	3,4,5

Table 2. Evaluation of the proposed interface by three participants who had a serious music background. Each number represents the number of participants who selected the corresponding item.

How much do you like	Least 1	2	3	4	5	6	Most 7	Median	Mode
Exterior Design				2	3			5	5
Features of Avoiding				1	1	2	1	6	6
Features of Visualizing				1		3	1	6	6
Improving Expressions				1	1	2	1	6	6
Overall Impression					4		1	5	5

Table 3. Evaluation of the proposed interface by five participants who had no serious music background. Each number represents the number of participants who selected the corresponding item.

were from different generations, and their music backgrounds were almost evenly distributed.

Table 1 shows their responses toward the post-experiment questionnaires. Based on the median values, participants had positive impressions (4 and above) towards VocalistMirror. The lowest median value (4) in the exterior design suggests that we should invite a professional graphic designer to improve our interface in terms of aesthetics. By identifying the participants who mostly gave points above or below 4, we realize there could be two distinct user groups.

Table 2 shows the responses of the three participants who had a serious music background. Table 3 shows the responses of the other five participants, who had no serious music background. By comparing the responses of these two groups of people, we realize that each group of people has distinct and contradictory opinions of VocalistMirror. Examples are shown below.

- Participants with no serious music background would appreciate more on the features of avoiding undesirable facial expressions and would more agree that their facial expression is improved after multiple uses. They mentioned that the alternating stage design makes VocalistMirror easy to use and, most importantly, encourages them to be more aware of their undesirable facial expressions. They felt that a process of simply selecting some undesirable facial expressions is sometimes helpful enough for them to avoid undesirable facial expressions. However, the other group of participants with a serious music background less agreed as they would demand more sophisticated features for selecting their undesirable facial expressions in the first place. For example, they may only dislike the mouth shape or the head tilt. Hence, they demand a precise navigation function so they can navigate the recorded video clip precisely to find that particular set of frames.

- Participants with serious music background would demand the interface design to match their music understanding. For example, once the experienced singers know the key of the song after hearing the first few notes of the song, they just need the pitch information which is related to the key of the song (i.e., the solfeggio - do, re, me fa, so, la, si, do) in order to sing in tune. Hence, the experienced singers demand the interface to show the relative pitch information. However, novice-level singers would prefer the interface to display the absolute pitch information (e.g., C4, D4, and so on) so that they could feel the pitch control is just like pressing the note on the piano keyboard.

These opinions suggest that we will need two different versions of VocalistMirror in the future if we want to target both of the user groups. It also suggests that the current version of VocalistMirror is more appreciated by the participants with no serious music background.

5. CONCLUSION

We described *VocalistMirror*, a real-time user interface helping a singer avoid the singer's undesirable facial expressions. Although facial expression is an essential feature in singing, to the best of our knowledge, this is the first study that focuses on undesirable facial expressions of singers. VocalistMirror analyzes the singing voice taken from a microphone input in real time and visualizes its pitch trajectory as well as a piano-roll melody line with Karaoke-style lyrics scrolling automatically. Moreover, it analyzes the singer's facial expression taken from a camera in real time and displays a warning if it becomes similar to one of undesirable facial expressions specified by the singer. In our current implementation, VocalistMirror uses iOS ARKit framework to quantify the facial expression as a 52-dimensional facial vector with each dimension ranging from 0 to 1. We can tell how similar two facial

expressions are by calculating the exponential moving average of the L1-norm distance between the corresponding two facial vectors. Experimental results showed the effectiveness and potential of the proposed interface and, most importantly, they provide a direction for further improving our VocalistMirror interface.

Acknowledgments

This work was supported in part by JST ACCEL Grant Number JPMJAC1602, Japan. In this work, we used the RWC Music Database (RWC-MDB-P-2001) [18].

References

[1] D. Hoppe, M. Sadakata, and P. Desain, "Development of real-time visual feedback assistance in singing training: A review," *Journal of Computer Assisted Learning*, vol. 22, pp. 308–316, 2006.

[2] F. Moschos, A. Georgaki, and G. Kouroupetroglou, "FONASKEIN: An interactive application software for the practice of the singing voice," in *Proc. Sound and Music Computing Conference (SMC 2016)*, 2016, pp. 326–331.

[3] R. Gong, Y. Yang, and X. Serra, "Pitch contour segmentation for computer-aided jingju singing training," in *Proc. Sound and Music Computing Conference (SMC 2016)*, 2016, pp. 172–178.

[4] M. Pérez-Gil, J. Tejada, R. Morant, D. Martos, and A. Pérez-González, "Cantus: Construction and evaluation of a software solution for real-time vocal music training and musical intonation assessment," *Journal of Music, Technology & Education*, vol. 9, no. 2, pp. 125–144, 2016.

[5] K. W. E. Lin, H. Anderson, M. Hamzeen, and S. Lui, "Implementation and evaluation of real-time interactive user interface design in self-learning singing pitch training apps," in *Proc. International Computer Music Conference and Sound and Music Computing Conference (Joint ICMC SMC 2014 Conference)*, 2014, pp. 1693–1697.

[6] K. Tsuzuki, T. Nakano, M. Goto, T. Yamada, and S. Makino, "Unisoner: An interactive interface for derivative chorus creation from various singing voices on the web," in *Proc. International Computer Music Conference and Sound and Music Computing Conference (Joint ICMC SMC 2014 Conference)*, 2014, pp. 790–797.

[7] Y. Ojima, T. Nakano, S. Fukayama, J. Kato, M. Goto, K. Itoyama, and K. Yoshii, "A singing instrument for real-time vocal-part arrangement of music audio signals," in *Proc. Sound and Music Computing Conference (SMC 2017)*, 2017, pp. 443–449.

[8] L. R. Quinto, W. F. Thompson, C. Kroos, and C. Palmer, "Singing emotionally: a study of pre-production, production, and post-production facial expressions," *Frontiers in Psychology*, vol. 5, p. 262, 2014.

[9] M. J. Lyons and N. Tetsutani, "Facing the music: A facial action controlled musical interface," in *CHI '01 Extended Abstracts on Human Factors in Computing Systems*, 2001, pp. 309–310.

[10] E. S. Koh and S. Yadegari, "Mugeetion: Musical interface using facial gesture and emotion," in *Proc. International Computer Music Conference (ICMC 2018)*, 2018.

[11] M. Goto, T. Nakano, S. Kajita, Y. Matsusaka, S. Nakaoka, and K. Yokoi, "VocaListener and VocaWatcher: Imitating a human singer by using signal processing," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (IEEE ICASSP 2012)*, 2012, pp. 5393–5396.

[12] M. Abbaszadegan, S. Yaghoubi, and I. S. MacKenzie, "Trackmaze: A comparison of head-tracking, eye-tracking, and tilt as input methods for mobile games," in *Proc. HCI International 2018*, 2018, pp. 393–405.

[13] A. Scicali, I. Nwogu, and J. Geigel, "Mobile facial emotion recognition engine," in *ACM Symposium on Applied Perception*, 2018.

[14] J. C. Brown and M. S. Puckette, "A high resolution fundamental frequency determination based on phase changes of the Fourier transform," *The Journal of the Acoustical Society of America (JASA)*, vol. 94, no. 2, pp. 662–667, 1993.

[15] K. W. E. Lin, "Singing voice analysis in popular music using machine learning approaches," Ph.D. dissertation, Singapore University of Technology and Design, 2018.

[16] M. Goto, K. Yoshii, H. Fujihara, M. Mauch, and T. Nakano, "Songle: A web service for active music listening improved by user contributions," in *Proc. International Society for Music Information Retrieval Conference (ISMIR 2011)*, 2011, pp. 287–288.

[17] J. Kato, T. Nakano, and M. Goto, "TextAlive: Integrated design environment for kinetic typography," in *Proc. ACM SIGCHI Conference on Human Factors in Computing Systems (ACM CHI 2015)*, 2015, pp. 3403–3412.

[18] M. Goto, T. Nishimura, H. Hashiguchi, and R. Oka, "RWC Music Database: Popular, classical, and jazz music databases," in *Proc. International Conference on Music Information Retrieval (ISMIR 2002)*, 2002, pp. 287–288.